UNIVERSITÀ DEGLI STUDI DI BARI
"ALDO MORO"

DEPARTMENT OF COMPUTER SCIENCE

PHD PROGRAMME IN COMPUTER SCIENCE AND MATHEMATICS

XXXVII CYCLE

SCIENTIFIC DISCIPLINARY AREA IINF-05/A

BIOMETRICS FOR SAFETY AND HEALTH IN REAL-WORLD CONTEXTS

Coordinator:
Prof. Francesca MAZZIA

FRANCESCA
MAZZIA
26.05.2025 18:01:44
GMT+02:00

Tutor:
Prof. Antonio PICCINNO

ANTONIO PICCINNO
23.05.2025 19:28:07
GMT+02:00

Co-tutor:
Prof. Donato IMPEDOVO

Firmato digitalmente da: Impedovo Donato
Data: 23/05/2025 18:13:35

PhD candidate:
Vincenzo GATTULLI

Vincenzo
Gattulli
23.05.2025
15:45:43
GMT+02:00

FINAL EXAM 2025

FUNDED BY PON RICERCA E INNOVAZIONE 2014-2020 FSE REACT-EU, AZIONE IV.4 "DOTTORATI E CONTRATTI DI RICERCA SU TEMATICHE DELL'INNOVAZIONE (CUP: H99J21010060001)

Index

Abstract

The PhD thesis *"Biometrics for Safety and Health in Real-World Contexts"* explores the application of behavioral biometrics for monitoring and enhancing quality of life, with a particular focus on preventing bullying and cyberbullying among young individuals. The research leverages innovative technologies such as Human Activity Recognition (HAR) and continuous authentication through touch dynamics to construct behavioral models capable of timely identifying anomalies and potential threats to safety and psychological well-being.

This study is part of the *PRIN-2017 "BullyBuster" project*, an interdisciplinary initiative aimed at combating bullying through artificial intelligence and computer vision technologies. During the research, an Android app and a web platform were developed, integrating a questionnaire designed to collect biometric and interaction data during use, allowing for the classification of users based on their responses and behavioral activities.

Through data collection in school and university settings, datasets were created to train and validate machine learning models capable of identifying behaviors associated with bullying and victimization. The results demonstrate how behavioral biometrics can serve as a valuable tool for supporting health and safety in real-world environments, offering a prevention and monitoring system adaptable to contemporary needs.

Sommario

La tesi di dottorato "*Biometrics for Safety and Health in Real-World Contexts*" esplora l'applicazione delle biometrie comportamentali per monitorare e migliorare la qualità della vita, con particolare attenzione alla prevenzione del bullismo e del cyberbullismo tra i giovani. La ricerca sfrutta tecnologie innovative come il Human Activity Recognition (HAR) e l'autenticazione continua basata sulle dinamiche di tocco, al fine di costruire modelli comportamentali capaci di identificare tempestivamente anomalie e potenziali minacce alla sicurezza e al benessere psicologico.

Questo studio si inserisce nel progetto *PRIN-2017 "BullyBuster",* un'iniziativa interdisciplinare volta a contrastare il bullismo attraverso l'intelligenza artificiale e tecnologie di computer vision. Durante la ricerca sono stati sviluppati un'app Android e una piattaforma web, integrando un questionario progettato per raccogliere dati biometrici e di interazione durante l'utilizzo. Questi dati hanno permesso la classificazione degli utenti in base alle loro risposte e alle attività comportamentali.

Attraverso la raccolta di dati in contesti scolastici e universitari, sono stati creati dataset per l'addestramento e la validazione di modelli di machine learning, capaci di identificare comportamenti associati al bullismo e alla vittimizzazione. I risultati dimostrano come le biometrie comportamentali possano rappresentare uno strumento prezioso per supportare la salute e la sicurezza in ambienti reali, offrendo un sistema di prevenzione e monitoraggio adattabile alle esigenze contemporanee.

Figures Index

Table Index

Ringraziamenti

Desidero esprimere il mio più sentito ringraziamento a tutte le persone che hanno reso possibile il completamento di questo percorso di dottorato.

In primo luogo, rivolgo un sincero ringraziamento al mio relatore, **Prof. Antonio Piccinno**, e al co-relatore, **Prof. Donato Impedovo**, per i preziosi consigli e il significativo contributo scientifico che hanno saputo offrirmi.

Un ringraziamento speciale e affettuoso va ai colleghi/**AMICI** del laboratorio, in particolare *Alessia, Davide, Francesco, Gianfranco, Giacomo, Lucia, Luca, Paolo, Stefania e Vincenzo* (elencati in ordine alfabetico) che negli anni hanno saputo alleggerire una delle esperienze più impegnative e difficili della mia vita.

Non posso infine dimenticare la mia **famiglia** e i miei **amici**, il cui sostegno emotivo e continuo incoraggiamento sono stati per me una fonte di forza nei momenti più difficili.

Ringrazio altresì tutte le persone che, in modo diretto o indiretto, hanno contribuito alla realizzazione di questo lavoro.

Senza di voi non sarei stato qui!!

1. Introduction

In the contemporary landscape, a pressing need has arisen to devise innovative solutions that prioritize individual safety and well-being. Within this framework, Behavioral Biometrics have emerged as potent, versatile tools offering continuous, personalized, and user-centered monitoring. These technologies not only safeguard data and access but also foster users' physical and psychological wellness.

This study is conducted as part of the Ph.D. program in Computer Science and Mathematics and has been funded by the PON Research and Innovation 2014-2020 FSE REACT-EU, Action IV.4 *"Ph.D. Programs and Research Contracts on Innovation Topics"* (CUP H99J21010060001). Academic supervision was provided by Prof. Antonio Piccinno, with the support of Co-Tutor Prof. Donato Impedovo.

The PhD thesis entitled *"Biometrics for Safety and Health in Real-World Contexts"* delves deeply into the integration of biometric techniques in real-life settings to monitor and enhance quality of life. It addresses emerging issues such as bullying and cyberbullying, with a particular focus on young individuals. The application of behavioral biometrics techniques has proven instrumental in preventing and analyzing these phenomena in classroom settings, utilizing anonymized data for added privacy. The research area, termed *"Behavioral Biometrics for Safety and Health,"* represents a multidisciplinary approach combining expertise in computer science and psychology to develop technological solutions that meet the real and pressing needs of contemporary society.

The decision to focus on **health** and **safety** stems from the recognition of the importance of continuously monitoring individuals' well-being through innovative and technologically advanced methods. Within the **Health** sector, wellness monitoring through behavioral biometrics has enabled the detection of changes in an individual's behavior that may signal stress, depression, or other physical and mental issues. These systems have not only facilitated the prevention of such phenomena but also have had a particularly significant impact on psychology. Behavioral biometrics can identify signs of emotional distress before more severe symptoms emerge, offering preventive tools to maintain psychological well-being. In parallel, within the **Safety** sector, authentication based on behavioral biometrics has ensured that access to IT systems remains secure by analyzing in real-time the way a user interacts with devices such as smartphones and computers. For instance, analyzing a user's typing behavior on their smartphone can reveal anomalies indicative of unauthorized access attempts, significantly enhancing the protection of sensitive data. Furthermore, these technological interventions have proven highly effective in counteracting issues like cyberbullying, providing tools capable of monitoring and blocking instances of bullying and cyberbullying. Prevention has been a pivotal element in this field, as timely intervention has the potential to avert significant harm to individuals' safety and psychological well-being.

The **motivations** behind this research lie in the integration of behavioral biometric techniques to enhance the ability to detect and prevent bullying and cyberbullying through the use of touch

interaction sensors and data on smartphones. This approach has demonstrated significant multidisciplinary importance, combining insights from various fields to address a complex and widespread issue. Currently, behavioral biometric techniques specifically targeting bullying and cyberbullying are lacking, particularly those focusing on young people, who are often the primary victims of these phenomena. Thus, this research aimed to fill this gap by offering innovative solutions that can be implemented in real-world contexts, such as schools and universities, where the risk of bullying and cyberbullying is particularly high.

The **objective** of the thesis was to implement behavioral biometric models in support of a questionnaire on bullying and cyberbullying. These models relied on analyzing collected data, allowing for the identification of anomalous behaviors that might be linked to specific, high-relevance questions related to the phenomenon. Through continuous analysis of users' interactions with their devices, a unique behavioral profile was developed, enabling the detection of significant deviations from this profile and signaling potential security threats or signs of psychological distress. Complementing this objective was the creation of a behavioral dataset based on the BullyBuster Questionnaire (BBQuest) in real-world contexts, a focal part of the BullyBuster project. The term "*Real-World Contexts*" reflects the data collection settings in schools and universities. This dataset encompassed data from smartphone sensors and users' touch interactions while completing the questionnaire, also including users' responses to facilitate categorization. Additionally, a *Human Activity Recognition (HAR) dataset* was constructed in a laboratory setting to support HAR experimentation, including fall cases detected by smartphone sensors. These datasets were fundamental for training and validating machine learning models, ensuring that the proposed solutions were accurate and reliable across various real-world scenarios.

The **research pipeline** design of this PhD study involved several essential phases, each contributing significantly to achieving the set objectives. Initially, an Android application and a web platform were designed and implemented, integrating a questionnaire validated by the psychologist of the BullyBuster project (the Android application was primarily developed during the corporate phase with the academic spin-off *Digital Innovation Srl*). These data were then processed through statistical calculations to assign each user a personality index, classifying them into five categories: *Bullying-Victim, Bullying-Bully, Cyberbullying-Victim, Cyberbullying-Cyberbully, and External*. These classes represented the various dynamics of social interaction and were fundamental for subsequent analysis through machine learning techniques.

Applying machine learning techniques to the data collected via BBQuest addressed several crucial aspects of the research. First, Human Activity Recognition (HAR) enabled the identification of user activities, providing deeper insight into the context of their behaviors while completing the questionnaire. This step was essential for accurately interpreting the data and developing robust models. Secondly, anomaly detection allowed for the identification of unusual sensor behaviors that might be linked to questions relevant to each target category. Lastly, continuous authentication via touch interactions provided an additional layer of security, constantly monitoring the user to ensure that only the legitimate user could access the device and complete the questionnaire.

The **PhD journey** spanned three years, each contributing substantially to the overall progress of the project:

First Year: The focus was on the state of the art in Human Activity Recognition (HAR), essential for understanding the most advanced technologies and methodologies in the field. This preliminary study enabled the acquisition of a solid theoretical foundation and the identification of the most promising techniques to apply within the research's specific context. Alongside this, an in-depth analysis of the psychological nature of the questionnaire and the assignment of behavioral classes was conducted, ensuring that the questionnaire was structured effectively and had psychological validity. This step was crucial to ensure that the data collected was meaningful and valuable for subsequent analysis.

Second Year: The research concentrated on methodologies and techniques for implementing machine learning models within the HAR and continuous authentication contexts. This included the study of advanced algorithms, data preprocessing techniques, and model validation methods—key elements in developing robust and reliable solutions. Additionally, the Android-specific *"BB Questionnaire"* was developed to gather data and assess the incidence of bullying and cyberbullying, a vital tool for collecting empirical data for further analysis and intervention (conducted during the six-month industry collaboration with the academic spin-off *Digital Innovation srl*). Internal experiments and tests with first-year UNIBA students were conducted to test and validate the models, ensuring their accuracy and effectiveness before broader application.

Third Year: This phase represented a period of consolidation and practical application of the results obtained. The "*BB Questionnaire*" web platform was developed and made accessible on iOS devices, facilitating data collection through questionnaires and making the process accessible and user-friendly to a broader user base. Preprocessing techniques were applied to real data collected from schools in Avellino and Cagliari, including data cleaning, normalization, and preparation for subsequent analysis, ensuring that the datasets were ready for machine learning models. Field experiments were conducted to test the platforms and models' effectiveness in real-life situations, collecting feedback and data for further improvements.

Finally, the **corollary**, throughout all three years, involved enriching current research in the challenging analysis of touch patterns in continuous authentication and HAR patterns, providing a more comprehensive and detailed view of user behavior both related and unrelated to the topics of bullying and cyberbullying.

The PhD thesis is structured into seven main chapters, each delving into specific aspects of integrating behavioral biometrics to enhance safety and health in real-world contexts. The chapters are organized to follow the research progression in the correct sequence:

1. **Introduction**: This chapter introduces the research context, emphasizing the importance of behavioral biometrics in monitoring individual well-being and preventing bullying and cyberbullying among youth. It discusses the motivations behind choosing this topic, highlighting the need for innovative solutions in a rapidly evolving technological era.

- **1.1 Real-World Problem - Bullying and Cyberbullying**: Definitions and explanations of bullying and cyberbullying phenomena are provided, examining their characteristics and consequences.
- **1.2 PRIN-2017 BullyBuster Project**: The BullyBuster research project is presented as a multidisciplinary initiative using advanced technologies to combat bullying, with a focus on university collaboration and employed methodologies.

2. **Human Activity Recognition with Smartphone-Integrated Sensors: A Survey**: This chapter provides an overview of human activity recognition (HAR) through smartphone-integrated sensors, exploring machine learning models used for sensor data analysis.
   - **2.1 Smartphone Sensors**: Describes the various sensors in mobile devices and their roles in activity recognition.
   - **2.2 Architecture of a HAR System**: Analyzes the structure of a HAR system, including data acquisition, feature extraction and selection, and classification processes.
   - **2.3 Activities, Datasets, and Performances of HAR**: Summarizes human activities, datasets used, and performance outcomes in prior studies.
   - **2.4 Discussion**: Provides critical insights on the techniques and technological choices in HAR.
   - **2.5 Conclusions**: Summarizes key conclusions and implications for the future of research in this field.
3. **Cyberbulling from Detection to Evaluation:**
   - **3.1 BB Questionnaire Design**: This subchapter focuses on designing the BullyBuster questionnaire, outlining the necessary requirements and technologies involved in its creation.
     - **3.1.1 Requirements Specification**: Defines specific requirements for the questionnaire, ensuring alignment with research objectives.
     - **3.1.2 Design**: Describes the design of the questionnaire, including details on formulating questions for psychological validity.
   - **3.2 BB Questionnaire Implementation**: This subchapter discusses the practical implementation of the questionnaire on both Android and web platforms.
     - **3.2.1 Android Smartphone Application Implementation**: Details the development process for the Android application, including versions and libraries used.
     - **3.2.2 Questionnaire Web Platform Implementation**: Examines the implementation of the web platform for the questionnaire, highlighting the web technologies utilized.
   - **3.3 Experimental Strategy**: This subchapter presents data collection strategies and analyzes the results of experiments conducted in various locations.
     - **3.3.1 Real-Context Data**: Describes data collection methods, including test settings in different Italian cities for the project.
     - **3.3.2 Experimental Strategies – University Students**: Details testing strategies and related publications that substantiate the experimental approach, specifically connected to the dataset of UNIBA university students.
     - **3.3.3 Experimental Strategies - Comparison of University Student and School Student**: Provides a comparative analysis between university students (UNIBA) and school students (Cagliari-Avellino), presenting the results of this comparison and highlighting the differences and similarities between the two populations.
4. **Human Activity Recognition for Security and Safety**: This chapter explores additional analyses related to HAR and continuous authentication, supporting the scientific research:
     - **4.1 Touch Events and Human Activities**: Examines data on touch patterns and their relationship with human activities.
     - **4.2 Two-Factor Authentication by Combining PIN and Biometrics**: Discusses the use of two-factor authentication in security contexts.
     - **4.3 Human Activity Recognition Using Smartphone Sensors**: Analyzes human activity recognition through smartphone sensors, with particular attention to fall detection.
5. **Contribution:** In this chapter, I will outline the contributions (*papers*) for this PhD thesis.

6. **Conclusion**: The concluding chapter summarizes the main findings and implications of the research, emphasizing the role of behavioral biometrics in enhancing safety and health.

This structure perfectly reflects the three-year path of my PhD, in which each phase has consistently contributed to the evolution of the research work. In the initial phase, I delved into the theoretical study of bullying and cyberbullying phenomena, along with the state of the art in Human Activity Recognition (HAR), thereby establishing a solid conceptual foundation (Chapter 2). Subsequently, I designed and implemented a mobile application featuring a structured questionnaire, developed for the collection of behavioral data. In parallel, I applied Artificial Intelligence methods to analyze the collected data, both through the questionnaire and via smartphone sensors to recognize human activities and implement continuous authentication mechanisms (Chapter 3). During the second and third years, I also pursued a parallel line of research focused on the integration of HAR and mobile device security. This work led to further experimental and theoretical investigations and publications, culminating in the analysis of authentication techniques and user-device interaction patterns (Chapter 4).
This PhD thesis is specifically focused on the **real-world problem of cyberbullying**, addressed within the design context of the **PRIN BullyBuster project**. Indeed, the core of my thesis work is closely tied to **BullyBuster**, a pioneering and **multidisciplinary initiative** aimed at combating bullying and cyberbullying through the use of **advanced artificial intelligence technologies** and **computer vision applied to Human Activity Recognition (HAR)**. Two of the most relevant publications from my PhD journey are directly associated with this project. These two papers clearly reflect the **scientific level achieved** and represent the **main outcomes** of my research activities in the field of HAR and in identifying behaviors associated with bullying and cyberbullying dynamics.

The **first paper**, published in **Expert Systems with Applications** and entitled **"Human activity recognition with smartphone-integrated sensors: A survey"**, provides a detailed overview of the **state of the art** in human activity recognition through mobile devices. In this study, I actively contributed to **data collection**, **comparative analysis** of major recognition methods, and the definition of **open challenges** in the field. The goal of the article is to offer a **solid foundation for the development of future applications** in safety and behavior monitoring. This contribution is thoroughly addressed in **Chapter 2** of the thesis.

The **second cornerstone paper** of my PhD, entitled **"Classification bullying/cyberbullying through smartphone sensor and a questionnaire application"** (published in **Multimedia Tools and Applications**), presents an **innovative system** for the **automatic classification** of bullying and cyberbullying episodes through HAR sensors. The study is based on a **HAR dataset** that I collected using smartphone-integrated sensors within an **experimental protocol** specifically designed for the **BullyBuster project**. To support the analysis, a **BB questionnaire** was integrated, aiming to enhance the **accuracy and contextualization** of bullying and cyberbullying recognition. My contribution included the **design of the experimental protocol**, **data collection, analysis, and interpretation**, as well as the **validation of the proposed HAR classification model**. This second paper is analyzed and discussed in **Chapter 3**, dedicated to the experimentation and design of the PhD thesis.

Both papers were published in **high-impact scientific journals (Q1 according to SCImago)** and represent a **concrete synthesis of my skills** in the design of **smart systems**, in **managing and interpreting sensor data from mobile devices**, and in the **integration between technology, security, and well-being**. I believe these contributions provide a **solid basis for future developments**, aimed at preventing risky behaviors in school and social contexts. The journals published for the project are listed in **Chapter 5 "Contributions"**.

The **survey on Human Activity Recognition (HAR)** was a true starting point for my PhD journey. From this initial work, new experiments emerged that expanded and consolidated my research, steering it towards the **application of HAR in the context of bullying and cyberbullying**. These studies not only enabled the exploration of **innovative approaches for continuous authentication** but also laid the groundwork for **intelligent systems capable of recognizing risky behaviors in real time**.

These contributions fit perfectly within the research path outlined during my PhD, where **mobile device security** was not treated solely as a technical issue, but as a **complex challenge** that also involves **behavioral, social, and psychological aspects**. The integration of **Human Activity Recognition, behavioral biometrics, and continuous authentication systems** now represents a **promising frontier** for developing **adaptive and intelligent solutions** capable of effectively responding to **emerging threats** such as cyberbullying and **silent attacks** targeting the most vulnerable users. This **multidisciplinary approach** guided all phases of my experimentation, further strengthening the connection between **technological innovation** and **active individual protection**. This analysis is mainly covered in **Chapter 4**, as a corollary to the main work presented in **Chapter 3**.

1.1 Real Problem - Bullying and Cyberbullying

This subsection addresses the analysis of bullying, with a particular emphasis on the phenomenon of cyberbullying. Initially, various definitions proposed over time by scholars regarding both bullying and cyberbullying are presented. Subsequently, the current state of research on these phenomena is examined, including methods for prevention and mitigation.

One of the earliest definitions of bullying was proposed by Hinduja and Patchin in 2007 [1], who describes bullying as a prolonged form of mistreatment exerted by an individual with malicious intentions and a perceived power advantage over the victim. In 2008, Rigby introduced another definition, describing bullying as an intentional, malevolent act aimed at harming a vulnerable person, characterizing it as *"a systematic abuse of power in interpersonal relationships."* This definition highlights the significance of power imbalance awareness, exploited to inflict suffering on a victim who feels helpless and incapable of retaliation.

Face-to-Face (FTF) bullying can manifest in physical, verbal, or relational forms [2]. Physical and verbal bullying are direct forms of bullying: the former includes acts such as hitting, pushing, or taking objects, while the latter involves mocking, insults, and ridicule. Relational bullying, on the other hand, is an indirect form, enacted through spreading rumors and false stories [2].

So far, bullying has been discussed in general terms; now, cyberbullying is analyzed further, presenting several definitions formulated by recognized authors in the field:

- Bullying that occurs on communication platforms such as email, chat rooms, cell phones, and websites is defined as cyberbullying [3];
- Cyberbullying is generally defined as the repeated and harmful use of offensive, insulting, or attacking language toward another individual [4];
- Patchin and Hinduja define cyberbullying as deliberate and repeated harm inflicted through electronic text [1], [5];
- Cyberbullying can be described as "the use of the Internet, cell phones, or other devices to send or post text or images that can hurt or embarrass another person";
- In research literature, cyberbullying is defined as an intentional aggressive act carried out by an individual or group using electronic forms of contact, repeatedly and over time, against a victim who cannot easily defend themselves" [6];
- Cyberbullying is generally defined as using electronic media (e.g., social networking sites, email, chat rooms, SMS, MMS, etc.) to harm another person who cannot defend themselves; Cyberbullying involves online attacks based on insults, threats, embarrassment, or the deliberate harassment of people on the Internet.

Various types and forms of cyberbullying also exist, including:

- **Flaming**: Sending violent and vulgar electronic messages to incite conflicts within a network, often in the context of video game challenges or confrontations.
- **Harassment**: Persistent and repeated words, behaviors, or actions directed toward a specific person.
- **Cyberstalking**: Behaviors aimed at harassing victims to the point of severe physical aggression.
- **Denigration**: The distribution of false or derogatory messages about victims through network messages or SMS, often involving images, photos, or videos posted online.
- **Impersonation**: A perpetrator who knows the victim's username and password can modify their credentials or send messages in the victim's name to another person.
- **Tricky Outing**: Deceptively obtaining private and intimate information from the victim, only to disclose it through electronic media.
- **Exclusion**: When a cyberbully intentionally bans or excludes another user from their group, chat, or game.
- **Happy Slapping**: Recording video of a victim enduring various forms of abuse, then sharing the footage online without their knowledge.

These cyberbullying forms share certain characteristics, as listed below:

- *Pervasiveness and Accessibility*: The cyberbully can reach their victim at any time and in any place, enabled by the constant connectivity of cell phones.
- *Wide Audience*: Cyberbullying can reach a vast audience online, contrasting with the typically private interactions of FTF bullying.

- *Message longevity*: Message longevity: Unlike FTF bullying, in which harm occurs within a certain period of time, offensive online posts or videos remain accessible long after the initial posting, and cyberbullies retain limited control over content distribution once online [7] [8];
- *Lack of emotional feedback*: the cyberbully, not seeing his victim's reactions to his behaviors, is never fully aware of the harm he does this makes him more uninhibited and lowers his levels of self-control;
- *Bully anonymity*: anonymity can lead to disinhibition, encouraging behaviors that users might not exhibit in real life [9] [10] [1];
- *Multiplication of Cyberbullies*: Cyberbullying's online nature enables many people to become cyberbullies simply by sharing or promoting harmful content.
- *Adult Underestimation*: Many youths feel that adults fail to grasp the pervasiveness and impact of online bullying.

These bullying and cyberbullying phenomena can have various consequences for victims, depending on the type, frequency, and personality of the individual. The most recognized consequences include:

- Victims report feelings of sadness, anxiety, and fear, as well as difficulty concentrating at school [11];
- Studies indicate that cyberbullying victims often have lower social status among their peers, problematic relationships with parents, and low self-esteem[12];
- Depression, substance abuse, and delinquency are significantly higher among youths who report cyberbullying experiences [13];
- A recent study explored the relationship between adolescents' experiences with cyberbullying and their self-esteem [5];
- Cyberbullying victims exhibited lower self-esteem levels than unaffected individuals;
- Feedback from social networking sites (SNS) can influence an adolescent's self-esteem: positive feedback enhances well-being, while negative feedback diminishes it [14];
- Research into the relationship between technology use and self-esteem shows that children who play video games extensively tend to have lower self-esteem than those who play less [15];
- Williams and Guerra examined relationships among self-esteem, normative beliefs, and school climate[16]. Their study found that in negative school environments, low self-esteem predicted higher bullying perpetration, while positive climates reduced aggressive behavior;
- Other studies link cyberbullying victimization to psychological distress and low self-esteem [17][18], [19].

Many behaviors analyzed in this subsection have been incorporated into the BB Questionnaire. Questions have been designed to actively or passively engage aspects of bullying and cyberbullying, enriching the study's scope.

1.2 PRIN-2017 BullyBuster Project

A significant portion of the PhD research was integrated into the BullyBuster project, a pioneering, multidisciplinary initiative aimed at countering bullying and cyberbullying through advanced artificial intelligence and computer vision technologies.

BullyBuster brought together four prominent professors from renowned universities in Southern Italy: Professor Carlo Sansone from the *University of Naples Federico II, Gian Luca Marcialis from the University of Cagliari, Professor Donato Impedovo from the University of Bari Aldo Moro*, and Professor Donatella Curtotti from the University of Foggia. This collaborative synergy aimed to develop a series of AI-based software tools capable of effectively and reliably identifying and monitoring bullying and cyberbullying behaviors.

The core of BullyBuster centered around the creation of advanced tools utilizing computer vision and AI to detect bullying and cyberbullying actions. The project drew upon the latest research in behavioral biometrics and crowd analysis in computer vision systems, while also integrating criminal behavioral models from psychological and legal disciplines. This interdisciplinary approach ensured a thorough understanding and a more precise detection of bullying dynamics, both in real and virtual contexts.

BullyBuster utilized various data types to ensure comprehensive and multifaceted coverage of the bullying phenomenon:

1. **Video Analysis**: Through segmentation and scene characterization techniques, the system examined videos using temporal and spatial textual descriptors. This enabled the detection of specific bullying actions based on the crowd's movements surrounding the victim and, when possible, on the victim's facial expressions. Analysis of facial expressions and body movements provided crucial indicators for identifying aggressive or intimidating behaviors.
2. **Textual Analysis**: This component focused on identifying words and phrases commonly associated with harassment, oppression, and stalking in cyberbullying contexts. Using text mining techniques and semantic models, the system could detect offensive or threatening content in written communications, such as instant messages, emails, and social media posts (1.2.3 (Paper) Cyber Aggression and Cyberbullying Identification on Social Networks).
3. **Behavioral Analysis**: Detection of typing dynamics and touch analysis on smartphones represented a further frontier for identifying suspicious behaviors. Typing dynamics, including the speed and rhythm with which an individual typed, provided valuable clues for identifying stress or aggression associated with cyberbullying (Other papers from chapters 5).

The statistical and generative models underpinning BullyBuster were inspired by psychological models of criminal behavior. This approach ensured that the algorithms developed were not only technically advanced but also capable of comprehending and interpreting human motivations and behaviors with greater accuracy. Integrating such models guaranteed higher precision in bullying detection, reducing false positives and enhancing the overall reliability of the system.

A fundamental aspect of BullyBuster was its attention to legal and privacy implications associated with using surveillance and personal data analysis technologies. The project addressed these issues by examining the current legislative framework and proposing solutions to overcome potential legal obstacles. Particular attention was paid to ensuring that the developed tools complied with data protection and privacy regulations, avoiding violations and ensuring ethical use of the technologies.

BullyBuster set out to achieve several specific objectives that formed the fundamental pillars of the project:

1. **Interdisciplinary Exploitation**: Combining legal and psychological models to design effective detection algorithms for bullying and cyberbullying. This interdisciplinary approach allowed for the integration of knowledge from various fields, enriching the quality and effectiveness of the developed tools.
2. **Implementation and Testing**: Developing a preliminary set of tools to be tested by interested institutions, such as schools and law enforcement agencies. Field tests allowed for assessing the tools' effectiveness in real-world scenarios, providing valuable feedback for further improvements.
3. **Dataset Creation**: Collecting and publishing examples of bullying and cyberbullying actions to support future research. The creation of a well-structured and tagged dataset enabled the scientific community to use these data for developing and testing new algorithms, promoting ongoing innovation in the field.
4. **Dissemination and Engagement**: Organizing workshops and collaborating with key stakeholders, such as families, schools, and cyberbullying observers, to field-test the developed tools and receive realistic feedback on system effectiveness. This dissemination phase was crucial to ensure that technological solutions were actually used and valued by end users.

The main motivation behind BullyBuster lay in the growing social emergency represented by bullying and cyberbullying, phenomena with devastating impacts on the physical, psychological, and social well-being of young people. In Italy, according to 2014 data from the National Institute of Statistics (ISTAT), approximately 50% of Italian adolescents experienced bullying at least once a month. This phenomenon, widely recognized and prevalent in other European countries and the United States, has been steadily increasing, fueled by the widespread adoption of smartphones and social networks, which facilitate aggressive behaviors in both real and virtual worlds. Traditional prevention and suppression methods, mainly entrusted to schools and law enforcement agencies, have proven insufficient in addressing the complexity and rapid evolution of these behaviors. The lack of advanced technological tools that could detect and proactively monitor bullying and cyberbullying actions has hindered timely and effective intervention.
BullyBuster was created to fill this gap, offering technological solutions that support institutions in their role of prevention and repression, reducing victims' psychological isolation, and enabling prompt intervention.

The BullyBuster project adopted an integrated methodological approach, combining expertise from various disciplines to develop cutting-edge technological solutions.

This approach was structured into several phases:

1. **Behavioral Modeling**: Developing an innovative protocol for modeling bullying and cyberbullying behaviors, defining a set of measurements to be extracted from different data types (video, text, biometric).
2. **Legal Framework Identification**: In-depth analysis of current legislation, particularly Italian law no. 71 of 2017, to ensure that the developed tools complied with legal provisions for cyberbullying prevention and repression.
3. **Data Collection and Analysis**: Defining an experimental plan for data collection, through the emulation of bullying and cyberbullying actions. This included creating videos simulating physical or psychological aggression, images of facial expressions of perpetrators and victims, representative text messages of harassment, and signals of typing dynamics.
4. **Development of BullyBuster Tools**: Designing a suite of tools based on computer vision and AI, capable of analyzing collected data according to the proposed behavioral model. Developed algorithms could detect anomalies in video events, identify offensive textual content, and analyze typing dynamics to identify suspicious behaviors.
5. **Testing and Validation**: Implementing a testing protocol involving interested institutions, such as schools and law enforcement, to evaluate the tools' effectiveness in real-world scenarios. Collected feedback was used to further refine technological solutions.

BullyBuster not only advanced scientific knowledge in computer vision and AI applied to bullying detection but also made a significant social impact. Through collaboration among universities and the integration of interdisciplinary expertise, the project created a dynamic, innovative research environment capable of addressing one of the most severe social emergencies of our time.

Additionally, the creation of a dataset of bullying and cyberbullying examples promoted transparency and collaboration within the scientific community, encouraging further research and technological development in this field. The involvement of law enforcement and educational institutions ensured that the developed solutions were not only theoretically sound but also practical and effective in real-world contexts. Further information is available at https://www.bullybuster.unina.it/.

During the PRIN project, I contributed to the writing of these papers as well:

1. *Leveraging Artificial Intelligence to Fight (Cyber)Bullying for Human Well-being: The BullyBuster Project (Paper)*
2. *Development of technologies for the detection of (cyber)bullying actions: the BullyBuster project (Paper)*
3. *Cyber Aggression and Cyberbullying Identification on Social Networks (Paper)*

1.2.1 Leveraging Artificial Intelligence to Fight (Cyber)Bullying for Human Well-being: The BullyBuster Project

This article focuses on the BullyBuster project, an interdisciplinary initiative promoted by four universities in southern Italy, aimed at combating bullying and cyberbullying through the use of

artificial intelligence and computer vision. Bullying is examined as a significant social issue that negatively impacts the psychophysical well-being of victims, compromising everyday environments such as schools and online platforms. With the increased adoption of digital technologies, cyberbullying has evolved into a more pervasive and complex threat, making innovative solutions essential. The BullyBuster project implements an integrated framework composed of several modules. These include group behavior analysis through video surveillance, textual analysis to detect offensive or aggressive online content, typing dynamics to gauge victims' emotional states, and deepfake detection to prevent the dissemination of manipulated and harmful multimedia content. Each module is designed to address specific aspects of bullying or cyberbullying, employing psychological models to describe the behaviors of victims, aggressors, and bystanders.

A crucial element is the crowd behavior analysis module, which utilizes computer vision algorithms to identify behavioral anomalies, such as signs of physical violence. This module examines the formation and dispersion of groups to detect sudden changes, potentially signaling unusual events, including bullying incidents.

Textual analysis, an essential component of the framework, focuses on identifying verbal aggression by processing comments on platforms like Twitter. Patterns of offensive language are recognized, with special attention to vulgar language, the use of capital letters to emphasize aggression, and the recurrence of negative words. These patterns contribute to the development of a keyword dictionary for automated online content analysis, enabling the timely identification of cyberbullying.

Another innovative area is typing dynamics analysis, which uses users' typing habits to detect their emotional states. This technology allows for monitoring stress or discomfort levels during message composition, facilitating prompt interventions to prevent bullying and mitigate associated emotional harm. The deepfake detection module is also vital, as the growing use of manipulated videos exacerbates cyberbullying, necessitating the development of reliable algorithms to distinguish authentic videos from falsified ones.

Initial results from the project are encouraging, demonstrating that the framework can detect and prevent incidents of bullying and cyberbullying across various contexts. The multidisciplinary approach ensures that technological, psychological, and legal aspects are addressed holistically, with particular attention to privacy and data protection. The project has received international recognition, including its selection among the top 100 artificial intelligence projects by the International Research Center on Artificial Intelligence (IRCAI) under the auspices of UNESCO, underscoring its potential impact on achieving the United Nations' sustainable development goals.

1.2.2 Development of technologies for the detection of (cyber)bullying actions: the Bully-Buster project

The article *"Development of technologies for the detection of (cyber)bullying actions: the Bully-Buster project"* introduces the BullyBuster initiative, a multidisciplinary effort aimed at detecting and preventing bullying and cyberbullying through the integration of artificial intelligence (AI) techniques and psychological models. The project's objective is to automatically identify harmful

content by analyzing behavioral patterns and linguistic models across various data sources, including text, photos, and videos, to prevent potential harm and alert relevant authorities.

The proposed approach is designed to be applicable in real-world settings, particularly in schools, and involves using behavior analysis systems based on AI algorithms. The framework integrates advanced computer vision techniques for analyzing surveillance videos, textual analysis to detect potential verbal attacks on social media, and typing dynamics analysis to assess the emotional state of victims. Additionally, it tackles deepfake detection to identify manipulated multimedia content used to defame or embarrass individuals.

The experimental phase relies on distinct AI modules that analyze group dynamics to detect abnormal behaviors in videos, such as panic episodes or physical aggression. For manipulated video content, the system employs multiple AI models in combination to improve reliability in detecting tampering. In terms of verbal abuse detection, the system uses text analysis of social media comments, employing classification models that accurately identify aggression and vulgarity with a high degree of precision.

An innovative component of the project is the use of typing dynamics to recognize users' emotional stress. This module leverages behavioral dynamics by analyzing keyboard typing rhythms, detecting any emotional anomalies linked to bullying situations. The data collected is anonymized and untraceable, and results indicate a substantial capacity of the system to detect stress conditions associated with bullying episodes.

The project's legal aspects have been carefully considered, with particular attention to privacy protection and GDPR compliance, especially regarding the use of surveillance video and biometric data of minors. Data anonymization and pseudonymization systems have been implemented to ensure the security of collected information.

The framework has been implemented in various prototypes, including a questionnaire used in schools to collect behavioral data on students, as well as tools designed for teachers and students that enable the analysis of chat conversations and surveillance videos. The system generates reports that assess the risk of bullying and cyberbullying episodes, providing teachers with a tool to monitor and intervene promptly.

The project has received international recognition and has been included among the top 100 AI projects globally by the IRCAI under UNESCO’s auspices. With its innovative approach, the BullyBuster project represents a significant step towards creating safer physical and digital environments.

1.2.3 Cyber Aggression and Cyberbullying Identification on Social Networks

The article titled "*Cyber Aggression and Cyberbullying Identification on Social Networks*" outlines the specific contributions of the *University of Bari (UNIBA)* within the project [20].

This paper examines the development of an automated system for identifying cyberbullying and aggression on social networks, with particular focus on Italian-language comments from Twitter. Cyberbullying, which includes online harassment, abuse, and discrimination, is one of the most prevalent issues on social media, where offensive and aggressive content spreads readily. The main goal of this work is to automatically identify such behaviors and detect profiles of users who repeatedly act aggressively.

The proposed system operates in three main phases:
Comments were selected from posts by well-known Italian public figures who are known to attract polarized audience responses. Four prominent figures, including singer Achille Lauro and politicians Matteo Renzi and Giuseppe Conte, were chosen for comment collection. The posts were gathered from a specific period (November-December 2020), during which the Italian government crisis and the COVID-19 pandemic were ongoing, sparking significant online reactions.

This phase is essential for extracting useful information from the comments. Nine different features are analyzed to identify potential aggression:

- **Number of negative words:** Calculated using a lexicon of 540 vulgar words commonly used to offend or humiliate.
- **Use of negations ("no" or "not"):** The presence of negations often indicates a polemical or aggressive tone.
- **Use of capital letters:** Comments written entirely in uppercase are often interpreted as virtual “shouting,” a sign of potential aggression.
- **Positive/negative weight of the comment:** Each word is weighted with a positive or negative score using lexical resources like WordNet and SentiWord-Net.
- **Use of the second person:** Direct attacks often involve second-person verb or pronoun forms (e.g., "I'll kill you").
- **Presence of threats:** Phrases that incite violence or suicide are treated as a specific feature of aggressive language.
- **Typical bullying words:** Identification of insults and terms frequently associated with bullying (e.g., "idiot," "stupid").
- **Comment length:** Many aggressive comments tend to be short and concise.
- **Classification and performance measurement:** Four supervised machine learning algorithms were used to classify comments as aggressive or non-aggressive: Support Vector Machine (SVM), Random Forest (RF), Multi-Layer Perceptron (MLP), and Decision Tree (DT). Each algorithm was evaluated based on metrics like accuracy, precision, recall, and F1-score.

The training dataset, titled *"Aggressive Italian Dataset,"* comprises 3028 comments extracted from Twitter, equally divided between aggressive and non-aggressive comments. A unique aspect of this dataset is its construction and balance: each comment was manually labeled by a group of ten individuals, whose assessments were combined to minimize ambiguity. This approach ensured that comments labeled as aggressive included verbal attacks that might not contain vulgarities but could still harm the victim’s sensitivity.

Two primary experiments were conducted:

1. **First Experiment:** The aim was to identify aggressive comments within the posts. The Random Forest (RF) model achieved the best results, with precision, recall, and F1-score values ranging from 90% to 98% for non-aggressive comments and from 75% to 90% for aggressive ones. The overall accuracy of the system was very high (80%-98%), though performance was lower in specific cases, such as sarcastic comments or insults masked by grammatical errors.
2. **Second Experiment:** This experiment aimed to identify profiles exhibiting cyberbullying behavior. The system tracked the frequency of aggressive comments posted by a user, enabling the profiling of serial cyberbullies. A table was created listing users with the highest number of offensive comments, revealing persistent behavior directed at various social targets, such as singers and politicians.

The results demonstrate that the approach used to detect cyberbullying and online aggression was effective, with the Random Forest model achieving the highest performance among the tested algorithms. The system's overall accuracy is high, though it shows some limitations in detecting sarcastic comments or those with intentional grammatical errors. Additionally, the system was able to identify users posting aggressive comments across multiple posts, suggesting its potential for monitoring and preventing cyberbullying.

The article concludes that the automatic detection system could be further improved by expanding the vocabulary of colloquial expressions and broadening the dataset with comments from ordinary users, not just celebrities. Moreover, the possibility of continuously monitoring aggressive user profiles could offer an effective solution for preventing cyberbullying in more localized settings, such as schools. Another area of development could include analyzing the frequency of attacks and the motivations driving users toward aggressive behaviors, paving the way for a psychological analysis of the phenomenon.

2. Human activity recognition with smartphone-integrated sensors: A survey

Human Activity Recognition (HAR) is an essential area of research related to the ability of smartphones to retrieve information through embedded sensors and recognize the activity that humans are performing. Researchers have recognized people's activities by processing the data received from the sensors with Machine Learning Models. This work is intended to be a hands-on survey with practical tables capable of guiding the reader through the sensors used in modern smartphones and highly cited developed machine learning models that perform human activity recognition. Several papers in literature have been studied, paying attention to the preprocessing, feature extraction, feature selection, and classification techniques of the HAR system. In addition, several summary tables illustrating HAR approaches have been provided: most popular human activities in the literature with paper references, the most popular datasets available for download (*Analyzing their characteristics, such as the number of subjects involved, the activities recorded, and the sensors with online-availability*), *co-occurrences between activities and sensors, and a*

summary table showing the performance obtained by researchers. The work's goal is to recommend, through the discussion phase and thanks to the tables, the current state of the art on this topic.

Human Activity Recognition (HAR) identifies physical activities performed by a physical person by analyzing signals recorded by digital devices. Thanks to the recent spread of the *Internet of Things (IoT)*, it has become increasingly common to adopt HAR to recognize activities of daily living. Some examples of activities could be sitting, standing, climbing stairs, and walking. Detecting such activities has numerous applications ranging from neurological disease progression analysis to gaming and many more. Smartphones have many sensors which could be used to detect activities: accelerometer, gyroscope, magnetometer, and so on. The combination of them could lead to activity recognition through a common smartphone [21] .

Several assumptions and hypotheses shall be considered when designing a system for *Human Activity Recognition (HAR)*: Advances in traditional recognition methods have successfully focused on finding meaningful information from raw data. However, the main challenge is that HAR models demonstrate effectiveness primarily in controlled environments and for specific tasks. As a result, performance in complex HAR tasks depends on the Feature Extraction techniques adopted and the limitations of domain knowledge. In this context, no direct and unambiguous solutions emerge to infer human actions from sensory information in a general way. The scientific literature presents varied choices for experimental settings, data types, tasks, and sensors. Moreover, data from such sensors incorporate intrinsic elements such as the temporal nature of the data, storage, and so on. Conducting a specific survey on human activity recognition using smartphones is imperative. This investigation provides the reader with a direct and practical approach to building a robust HAR system. The main idea is also to provide a detailed study plan, especially for people who are inept in this domain.

The survey focuses strongly on Human Activity Recognition approaches using smartphones that provide the reader with a direct, practical, and easy-to-understand approach to building a robust HAR system. In addition, smartphone sensors were studied in more detail to understand their potential applications, thus finding the optimal configuration. These aspects are fundamental to building a HAR system. This investigation also aims to conduct an in-depth study of HAR architectures using standard machine learning (also known as *Shallow Learning*) and *Deep Learning* techniques. The main innovation, besides the careful structuring of the work, is the composition of several summary tables that can be used to guide the approach and for quick reference by the reader:

1. *Activities found in Literature with the corresponding reference in* Table 4: This table collects scientific papers from the past five years and groups the scientific papers that specifically addressed them by HAR activities. Readers who want to study only one activity could consult the included references. In addition, the histogram below the table has also been added for quick understanding.
2. *Co-occurrences between activities and sensors found in Literature with the corresponding reference in* Table 5: This table collects the scientific papers from the last five years and

groups not only the activities from the previous work but also the sensors better to detail the search for the correct literature reference. In addition, the histogram below the table has also been added for quick understanding. This table is helpful to emphasize our detailed analysis of the sensors. In addition, for quick understanding, the histogram below the table has also been added.

3. *Summary and comparison among the different datasets found in Literature in* Table 6: This table collects the scientific datasets of the last five years inherent to HAR models and makes a careful analysis of them considering several characteristics, namely: *Name of Devices, Number of users, Sensor, Activities, Laboratory/Real, Raw data/pre-processed, Availability.* The table promotes quick readability of the essential features and convenience in finding the suitable dataset and downloading it simultaneously from the link.
4. *Summary of the experimentation settings with performance scores found in the Literature in* Table 7: This table collects scientific papers from the last five years and summarizes their performance. The table favors the choice of the dataset to the model used to favor better performance. In addition, for quick understanding, the histogram below (Figure 4) that analyzes the contrast of using Shallow/Deep models in HAR approaches has also been added.

The selected surveys in Table 1 did not go into all these aspects, especially the survey tables analyzed and viewed above. Finally, a careful discussion of relevant results is provided to advise the reader to use suitable models, datasets, and tips to create an optimal HAR system.

Ref.	Description
[21]	It covers smartphone sensors, sampling rate, location, and orientation. Furthermore, the study focuses on the recognition of the number of activities. Additionally, different classification methods used for the recognition process are evaluated. Finally, the various challenges and facets of these studies are discussed.
[22]	The authors examined state of the art in recognizing human activity based on acceleration components. The online and offline activity recognition systems, traditional machine learning algorithms, and Deep Learning have been defined. Forty-eight studies qualitatively compare activities, devices used, learning patterns, datasets, and recognition accuracy. Finally, the different challenges and problems of these studies are discussed.
[23]	This work includes three popular methods of activity recognition, namely HAR using pose estimation (vision-based), smartphone sensors, and wearable sensors. The pros and cons of the technologies are also discussed by considering a comparison of their accuracy.
[24]	It summarizes recognizing human physical activity from mobile phone sensors. The analysis is based on three parameters: human activity, smartphone position, location, and classification method. The accuracy of the other classification methods for various activities is compared. The limitations of the different classifiers are also discussed. Finally, real applications enabled by activity recognition are introduced.
[25]	The authors compared HAR works between 2010-2020 with smartphone sensors. The comparison charts highlight: the type of sensor used, the activities, the positioning of the sensor, the type of HAR system (offline, online), the processing device, the classifier (variety of algorithms), and the levels of system accuracy.

Table 1 - Surveys on HAR methods

The organization of the work is the following: Sub-Chapter 2.1 Smartphone Sensors, describes sensors present in smartphones and used for HAR. Sub-Chapter 2.2 Architecture of a HAR system, explains various architectures for HAR: It starts by explaining the different pre-processing techniques and feature extraction, feature selection, and classification. Sub-Chapter 2.3 Activities, Datasets, and performances of HAR, illustrates summary tables on human activities, datasets, and performance on various state-of-the-art works. Sub-Chapter 2.4 Discussion, provides Discussions and observations concerning the design and technology choices to address the problem

and this sub-chapter contains the most used applications of a HAR system, and Sub-Chapter 2.5 Conclusions, sketches conclusions and final considerations.

2.1 Smartphone Sensors

Smartphones have a variety of sensors, including *GPS, Accelerometers, Gyroscopes, Magnetometers, Proximity Sensors, Brightness Sensors*, and more, all of which have specific uses:

- *Accelerometer*: Recognizes changes in position. They are often found in electronic gadgets, essential for screen orientation, and not mainly made for very accurate activity recognition;
- Gyroscope: Tracked motions of a smartphone in three dimensions. Flipping the gadget over silenced calls, among other phone functions like gaming and navigation;
- *Magnetometer*: This device is mainly used to determine cardinal directions. Combined with accelerometers, they can operate as a compass and find substantial metallic objects;
- *Proximity Sensor*: Uses electromagnetic fields to detect adjacent objects without making contact. It is essential for preventing unintentional screen touches when on the phone and may be used to recognize hand gestures;
- *Brightness Sensor*: Adjusts screen brightness by detecting ambient light. It can offer the best possible screen viewing while preserving battery life. They are sometimes used in smart homes to change the illumination based on the surroundings;
- *GPS Receiver*: This device receives satellite signals and uses them to pinpoint its location. Useful for tracking, navigation, and location-based suggestions;
- *Other Sensors*: Smartphones come with various other sensors, including thermometers, fingerprint scanners, NFC, and others, but not all are useful for activity identification jobs. The most extensively used sensors in contemporary smartphones are the topic of this overview.

Other Sensors

A modern smartphone can have many other sensors, such as a thermometer, fingerprint sensor, NFC, barometer, heart rate monitor, pedometer, and humidity sensor. However, given the purpose of this work, there are two main reasons why should focus on these sensors. First, sensors such as fingerprints and NFC are not helpful for activity recognition tasks. There is no example of the use of these sensors in literature. Secondly, focusing on the most common sensors embedded in modern smartphones is essential. "Sensor data was gathered from different devices, and the selected sensors were widely adopted in modern smartphones.

Table 2 summarizes the previously explained sensors and reports the measurement units typically adopted. Table 2 summarizes the previously explained sensors, reporting the measure units typically adopted.

Sensor Name	Data Description	Unit of Measurement
Accelerometer	x, y, z axis	m/s^2
Gyroscope	x, y, z axis	rad/s
Magnetometer	x, y, z axis	μT
Proximity Sensor	Distance from object	cm
Brightness Sensor	Illuminance intensity	lux
	Estimated Location Accuracy	m

GPS receiver	Time in UTC concerning the position found	ms
	Latitude detected	$degrees$
	Longitude detected	$degrees$

Table 2 - Data description of the sensors found in the Literature.

2.2 Architecture of a HAR system

A HAR system's architecture comprises three main components: Data Acquisition, Features Engineering, and Classification. Data Acquisition mainly deals with collecting and storing data from all the sensors. At the same time, for the Features Engineering phase, most authors used time-domain features as well as frequency-domain features.

This process involves the following steps:

- *Pre-processing*: Some operations help filter and prepare the raw data to be processed;
- *Features Extraction*: combining existing features to produce a more useful one;
- *Features Selection*: consists in selecting, among the existing, the most valuable features to train on;
- *Classification*: concerns the learning techniques that use the pre-processed data from the previous stage.

The architecture is shown in Figure 1. The main component of the data acquisition phase is the sensors' measure of various attributes, i.e., acceleration, location, audio, temperature etc…

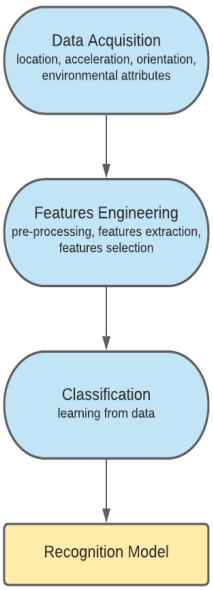


Figure 1 - The general HAR system architecture

2.2.1 Data Acquisition

There are four types of attributes that can be collected:

- *Environmental Attributes:* features that depend on the surrounding environment (e.g., *humidity, temperature, noise, and light levels*). These provide contextual information useful to discriminate the activity performed by the users. For instance, if the user operates in a noisy environment, they are more likely to walk or run rather than sleep. These parameters might be helpful to increase the accuracy of the recognition model, but alone they are insufficient to produce a sufficiently accurate recognition model.
- *Motion and Orientation*: An accelerometer and a gyroscope can be used to build an accurate and reliable recognition model [26]. Triaxial accelerometers are typically used to measure the total acceleration at every instant, requiring a gyroscope to isolate the gravity from the user-initiated acceleration. The gravity magnitude is fixed but can be decomposed along three axes to produce information about the device's Orientation. Instead, the user acceleration is given by the instantaneous acceleration caused by the user's movements. The accelerometer allows us to understand the acceleration on three axes x, y, and z. The axes allow many human activities to be recognized by discriminating walking from the lower to the upper plane, thanks to the z-axis. Moreover, a gyroscope can also measure the rotation rate along the three axes of the device, which is helpful since virtually all activities include swinging movements (e.g., swinging an arm during a run). For this reason, the position of the accelerometer is an essential factor. For instance, placing an accelerometer inside the trouser pocket can help recognize activities like walking and running. However, it can hardly differentiate between activities that do not involve leg movements, like standing still or driving a car.
- *Location*: GPS and GLONASS are the most widely used positioning systems integrated with most modern smartphones. There are two ways to use position sensors in HAR: the first is to use the position to provide contextual information about the activity [27]. For instance, if the user is at home, it is unlikely that he could be swimming or riding a bus, but he might be eating or resting. The second way is to use the longitude variations to calculate the user's speed and course during an activity. This can be useful for recognizing activities where speed and course are vital, like driving a car, running, and walking. Even so, using position sensors comes with many problems: firstly, they are unsuitable for indoor activities or activities performed in areas where the signal is weak; moreover, the GPS/GLONASS systems are expensive in CPU usage and energy consumption [27].
- *Vital Sign*: Heart rate, respiration rate, skin temperature, and more signals can be used to obtain a more accurate recognition model [26]. However, this is not always true, in fact, Tapia et al. [28], which used a triaxial accelerometer and a heart rate monitor, concluded that the heart rate is not helpful in activity recognition, especially during transitions from intense to moderate activities, because the heart rate is slow to fall to normal levels. Therefore, a low-level intensity activity performed after a high-intensity activity can be mistakenly classified as the latter [28].

2.2.2 Features Extraction

Feature extraction is a part of activity recognition. *Frequency-domain* and *Time-domain features* are used to perform this approach. *Time-domain features* use average, range, variance, skewness, kurtosis, median etc. The *frequency-domain features* are Correlation, Peak Power, Spectral Power, Peak Frequency, Different Frequency Bands, and Spectral Entropy.

Some of the common *frequency-domain* features *time-domain* and used for HAR are: The *Time-domain features* include *skewness, kurtosis, variance, range, average, median etc*., are used in HAR papers ([29] [30] [31]. Kim et al. [31] used two simple features in their work applied to the accelerometer: mean and standard deviation [31]:

$$\mu^k = \sum_{i=1}^{T} a_i^k \qquad (1)$$

Moreover, the standard deviation is:

$$\sigma^k = \sqrt{\sum_{i=1}^{T} (a_i^k - \mu_k)^2} \qquad (2)$$

where $a_i^k$and T are the i-th accelerometer data of the k-axis and the length of sequence, respectively. Hnoohom [32] used several time-domain features, among which the Signal Magnitude area. This feature is referred to a statistical measure of the magnitude of a varying quantity, defined as Hnoohom et al., [32]:

$$sma_{xyz} = \frac{1}{3}\left(\sum_{j=1}^{N} |S_{xi}| + \sum_{j=1}^{N} |S_{yi}| + \sum_{j=1}^{N} |S_{zi}|\right) \qquad (3)$$

The *Frequency-domain features* used to frequency spectrum with accelerometer and gyroscope ([29] Jain & Kanhangad, 2018; Ignatov & Strijov, 2015).

$$X(f) = \sum_{i=0}^{N-1} x_i e^{-j2\pi fi/N} \qquad (4)$$

Variable X is frequency spectrum, f is the f-th Fourier coefficient, N is the size of the sliding window:

$$X(f) = \sum_{i=0}^{N-1} a_i + jb_i \qquad (5)$$

With $a_i = x_i \cos\left(\frac{2\pi fi}{N}\right)$ and $b_i = x_i \sin\left(\frac{2\pi fi}{N}\right)$. In HAR system is used the *Power Spectral Density (PSD)* with accelerometer sensor for this activity driving, cycling, running, walking (Abdull Sukor et al. [30]). The *Power Spectral Density* is calculated as follow:

$$P(f) = \frac{1}{N} \sum_{i=0}^{N-1} a_i^2 + b_i^2 \qquad (6)$$

Entropy is generally used in HAR systems with accelerometer sensors Abdull Sukor et al., [30]. This feature is discriminant between activities with the same PSD but different movements patterns Abdull Sukor et al. [30]:

$$H(f) = \frac{1}{N}\sum_{i=0}^{N-1} c_i \log(c_i), \qquad c_i = \frac{\sqrt{a_i^2 + b_i^2}}{\sum_{k=0}^{N-1}\sqrt{a_i^2 + b_i^2}} \quad (7)$$

The DC (PDS at frequency 0 Hz) is used in HAR with an accelerometer, is defined as the sum its squared spectral normalized coefficients [33]:

$$DC = \frac{1}{N}\sum_{i=0}^{N-1} a_i^2 \quad (8)$$

The Correlation between axes is very useful for differentiating activities in only one direction ([34], [32]). The formula is as follows. The HAR system uses Other frequency-domain features based on spectral energy (Abdull Sukor et al., [30]).

$$c_{xy} = \frac{\sum_{i=1}^{N}(S_{xi} - mean(S_x))\left(S_{yi} - mean(S_y)\right)}{\sqrt{\sum_{i=1}^{N}(S_{xi} - mean(S_x))^2 \cdot \left(S_{yi} - mean(S_y)\right)^2}} \quad (9)$$

2.2.3 Features Selection

Feature Selection is another important phase in HAR System. The HAR system's performance depends on the dimensions of the feature space or vector. Shallow learning algorithms need discriminating features. This curse of the dimensionality problem with classifier performance decrease. One way to reduce the data dimensionality problem, an essential topic in Machine Learning and Data Mining, is to apply Feature Selection algorithms. These algorithms are very important because they are used to discard the less discriminating features and keep the most discriminating features for the problem addressed. This procedure aims to reduce the complexity and time of computation and improve the final classification.

There is a critical feature selection named *Principal Component Analysis (PCA)*. That is a linear technique that transports the original features into new mutually uncorrelated features. The PCA (new feature) rearranges the Raw Features into a new low-dimensional space. In this new space, the principal components are arranged from most significant to lowest (then omitted). More articles use PCA ([29] [30] [35] [36] [37] [38]).

Linear Discriminant Analysis (LDA) is associated with principal component analysis (PCA). These two methods find linear combinations of variables [39]. The LDA method is a feature projection in a new space of lower dimensions. This situation minimizing their within-class variability while maximizes the between-class separability.

Independent Component Analysis (ICA) solves *Blind Source Separation (BSS).* This technique is used for non-Gaussian data. The ICA aims to find separate components, and original features can be expressed as a linear combination.

In *Factors Analysis (FA)* the features are grouped by their correlation. FA represents groups of certain highly correlated features that have small correlations regarding some factor with specific features of other groups.

2.2.4 Classification

The feature extracted/selected is essential for classification ML algorithms. In the HAR system, the input data patterns are associated with the activities (classes) under consideration. The definition of Machine Learning has two types of approaches: *Supervised* and *Unsupervised*.

In *Supervised Learning,* generally, the training set includes already labeled features. In contrast, in *Unsupervised Learning*, the training data are not labeled. Activity recognition is usually treated as a supervised machine learning problem since the desired outputs are controlled, and an unsupervised model may result in a more complex model. Supervised learning approaches include many modules, for example, *Decision Tree classifiers such as the J48 and Random Forest (RF), K-Nearest Neighbours (K-NN), Support Vector Machines (SVM), Logistic Regression (LR), Naïve Bayes (NB) and Artificial Neural Networks (ANNs).* Those require entirely labeled activity data. The unsupervised learning approaches, *Hidden Markov Models (HMMs)*, *Gaussian Mixture Models (GMMs), and* Clustering algorithms infer the labels from the data. In the following sections, the classification techniques mentioned above are briefly described.

Classification Model	**References**
J48	[32] [40] [34] [41]
Random Forest	[33] [42] [43] [41]
Support Vector Machines	[33] [30] [44] [45] [34] [43] [38] [41]
K-Nearest Neighbors	[45] [35] [34] [43] [38] [41]
Naïve Bayes	[29] [46] [34]
Logistic Regression	[40] [38]
Gaussian Mixture Models	[43]
Hidden Markov Models	[31] [47] [43] [39] [48]
Convolutional Neural Network	[49] [50] [51] [52] [53] [44] [54] [55] [56] [57] [27] [58] [59] [60]
Long Short-Term Memory	[44] [48] [61] [62] [63] [64] [65] [66]
Multi-Layer Perceptron	[26] [30] [33] [34] [40] [44] [46] [67] [68] [69]

Table 3 - References analyzed in the Classification Phase

K-Nearest Neighbors

K-Nearest Neighbor (KNN) is a supervised learning algorithm for regression and classification. The algorithm calculates the probability that the input data belongs to training data class *"K",* and the class contains the highest likelihood of being selected. KNN predicts the class by calculating the distance between the test data and all training points by choosing the K number of points near the test data.

Ignatov (2015) apply the K-NN classification to differentiate between six human activities (Jogging, Walking, Upstairs, Downstairs, Sitting, and Standing in the WISDM dataset) using both time-domain and frequency-domain features obtained from smartphone three-axial accelerometers. This approach has shown high precision and nearly 96 % recognition accuracy [35].

Wannenburg (2017) [34] make a comparison among different classifiers. They extracted both time-domain and frequency-domain features obtained from the smartphone three-axial accelerometer and used the WEKA framework for the feature selection task with five human activities (Jogging, Laying, Walking, Sitting, Standing). As for the classifiers, they compare SVM, ANN, NB, Tree, k-Star, and K-NN with k equals 1 and k equals to 5. The K-NN gives the best result with k equals 1 achieving 99.01% accuracy, followed by the K-NN with k equals 5, which achieved a 99.0% score for accuracy [34].

Support Vector Machine

Support Vector Machine (SVM) is a statistical learning theory. It is a non-linear classification using kernel methods. The kernel methods protect data from high dimensional space using a non-linear kernel function. SVM is a binary classifier.

Bayat (2014) compared the SVM with other classifiers. Using low-pass filters for features selection, they achieved 72.24% of accuracy, trying to recognize six different activities (*Slow Walking, Fast Walking, Running, Stairs-Up, Stairs-Down, and (Aerobic) Dancing*). Moreover, they combined the SVM with other classifiers to achieve higher accuracy results. Using SVM combined with the Multilayer Perceptron, they obtained 91.15% of accuracy in identifying the same six activities [33].

Abdull Sukor (2018) used the SVM to recognize three different activities (*Standing, Sitting, Laying, Stairs Up, Stairs Down, and Walking*). They extracted both time-domain and frequency-domain features and used the PCA to reduce dimensionality. This approach achieved 92.87% accuracy, outperforming the same configuration without using the PCA for dimensionality reduction (which achieved 90.19% accuracy) [30].

Jain (2018) used the SVM as a classifier and compared achieved accuracy using time-domain, frequency-domain, and both features with this activity Walking, Walking Downstairs, Walking Upstairs, Standing, Sitting, Laying, Stand-to-Sit, Sit-to-Stand, Sit-to-Lie, Lie-to-Sit, Stand-to-Lie, Lie-to-Stand. The result shows that the time-domain works better than frequency-domain features achieving 94.57% accuracy against the 93.82%. However, using both features, the average accuracy in recognizing the six proposed activities increases, achieving 97.12% [45].

Random Forest

Random Forest (RF) combines with the *Bootstrap Aggregation Method (Bagging)* and Randomization in the selection of nodes in the construction of the *Decision Tree*. The output is a majority vote of the different decisions in each tree. The RF model requires huge, labeled data [70].
Xu (2018) proposed a classification methodology to recognize six activities using a wearable device's acceleration data. They compared the classification accuracy of the RF with the ANN and DT classifiers. The result showed that RF achieved the best accuracy, compared with other classifiers, in recognizing walking, jumping, and running activities [42].

Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a probabilistic model used to represent Normally Distributed subpopulations within an overall population. The GMM parameters are estimated using the Expectation-Maximization algorithm (EM).

In HAR systems, separate GMMs can be used for different tasks. GMM does not guarantee convergence to the global minimum many times. The tasks used are Standing, Sitting, Sitting to sitting on the floor, Sitting on the floor, Laying down, Laying down to sitting on the floor, Standing, Ascending on stairs, and Walking with 73% accuracy and 96% specificity.

Markov Chains and Hidden Markov Models

Markov chain (stochastic discrete-time process covering a finite number of states) fits the sequential model data well. It is often used in a more general *Hidden Markov Model (HMM)* model. In this case, each activity is represented by a specific state.

The Hidden Markov Model (HMM) is a stochastic statistical process. It is helpful for modeling time series. One disadvantage of the HMM model is that it often fails to guarantee convergence to the global minimum. Sensor data are time series, and in fact, this approach can help their implementation. This approach observes micro variations that are useful for the final classification [26]. Hidden states are critical to decree an activity in HAR approaches. After training, the most probable sequence of detected activities is decremented by Viterbi's algorithm [47].

Kim (2016) used the HMM extended in the HMM Ensemble Model to improve discriminative power with activity Walking, walking upstairs, walking downstairs, Sitting, Standing, and Laying (UCI HAR Dataset). This latter uses the Decision Template method to integrate the probabilities of multiple HMMs concerning an observation sequence. This method, combined with extracting two simple features (such as mean and standard deviation), only achieved 83.51% accuracy in recognizing seven different activities [31].

Zhang (2016) used HMM in combination with Deep Neural Networks with the activity Riding, Walking, walking upstairs, walking upstairs, Standing, and Walking downstairs. A critical step of training an HMM is to learn the observation model. It could be helpful to use DNNs for modeling the observation probability distribution of HMMs. With this technique, the authors achieved 93.37% accuracy in recognizing five different activities, outperforming other HMM-based classification methods such as HMM-RF and HMM-GMM [47].

Ronao (2017) use the Continuous HMM in a hierarchical architecture. This architecture allows the inherent hierarchical characteristics of the tasks to be exploited. Instead, CHMMs use continuous observation densities such as sensor signals, which are advantageous. Moreover, the hierarchical structure allows different feature subsets for various subclasses while minimizing feature computation time. The authors achieved 92.94% accuracy in identifying the six other activities of the HAR dataset (*Downstairs, Sitting, Standing, Laying, Walking, Upstairs*) and 56.33% accuracy for the twelve different activities of the USC-HAD dataset (*Sleeping, Elevator Up,*

Walking Right, Walking Upstairs, Walking Downstairs, Running Forward, Jumping Up, Walking Forward, Walking Left, Sitting, Standing, Elevator Down) [39].

Artificial Neural Networks and Multilayer Perceptron

Artificial Neural Networks (ANNs) can automatically learn and approximate nonlinear feature combinations to optimally recognize discriminant classes [70]. ANNs attempt to optimize input parameters via neuron passage, activation function, and weights. The objective is to regularize them through back-propagation [36]. Minimizing error by optimizing input parameters without using feature extraction methods practical more Shallow Learning models. In modern HAR systems, ANN can be combined with other classifiers to construct a heterogeneous algorithm [32], [40]. The disadvantage of ANN networks is that they require a large amount of data for the training phase and thus HAR Datasets.

The MLP network is an ANN with a multilayer feedforward architecture. The MLP is based on nonlinear activations for hidden units [26]. The MLP tries to minimize the error function between the outputs it estimates and the desired outputs. The MLP approach is good for nonlinear classifications. In fact, in HAR studies, the MLP network has been used in a wide range of studies:
Ogbuabor (2018) [67] investigated the role of gyroscope and accelerometer sensors in a HAR system using an MLP classifier with six activities (*Walking, Sitting, Laying, Walking Downstairs, walking upstairs and standing*). The accelerometer performs better than the gyroscope, with an overall accuracy of 92%. Combining the accelerometer and gyroscope performed better with 95% accuracy [67].

Wan (2020) [44] propose a smartphone inertial accelerometer-based architecture with typical daily activities (*Nordic walking, Watching TV, Walking, Running, Cycling, Computer work, Laying, Sitting, and Standing*). The preprocessing is done with denoising, and CNN, LSTM, BLSTM, MLP, and SVM models are utilized on the UCI and Pamap2 datasets. With UCI Dataset, the MLP accuracy is 0.8683%, and with Pamap2 Dataset, the accuracy is 0.8207%, better respecting other models [44]. The best model is CNN.

Voicu (2019) [68] propose a HAR system used on three smartphone sensors: an accelerometer, gyroscope, and gravity sensor. The activity is Walking, Running, Sitting, Standing, Upstairs, and Downstairs. The best overall accuracy was obtained with these three experiment configurations, i.e., External implementation 5-s window accelerometer and gyroscope at wrist and in the pocket, External implementation 15-s window accelerometer and gyroscope at the wrist, External implementation 30-s window accelerometer and gyroscope at the wrist. Respectively Overall accuracy equals 94%, 94%, and 97% [68].

Convolutional Neural Network

CNNs, called ConvNets, are widely used extensions of MLPs that have seen great success in computer vision tasks [26]. Convolutions represent a critical difference between CNNs and other NNs in layers of the network that induce weight sharing. Convolutional layers via neurons and deep layers learn and optimize input features. Each neuron is implemented as a receptive field

that connects it to previous layers. CNNs also typically have pooling layers for down sampling, usually performing average or max pooling to reduce the feature maps' spatial resolution. These layers are typically stacked on each other and used in conjunction with dense layers, creating a deep architecture. The architecture of CNNs, which typically refers to the number of layers, the size of each layer, the choice of activation functions etc., is diverse.

A wide variety of studies have been conducted with CNNs, and the three most recent are shown in these examples:

Zhou (2019) propose a pedestrian activities recognition method based on a CNN with nine activities: still, walk, upstairs, up the elevator, up the escalator, down the elevator, down the escalator, downstairs, and turning. The experiments achieve approximately 98% accuracy. The smartphone sensors used are accelerometers, magnetometers, gyroscopes, and barometers collected with various types of smartphones [59].

Wan (2020) propose a smartphone inertial accelerometer-based architecture with typical daily activities (*Cycling, Nordic walking, Laying, Sitting, Watching TV, Computer work, Standing, Walking, and Running*). The preprocessing is done with denoising, and CNN, LSTM, BLSTM, MLP, and SVM models are utilized on the UCI and Pamap2 datasets. With UCI Dataset, the CNN accuracy is 0.9271 %, and with Pamap2 Dataset, the accuracy is 0.9100 %, better respecting other models [44].

Mario (2019) proposes a method to detect activity with a single tri-axial accelerometer with activity Walking, Running, Jumping, Climbing Up, and Climbing down. A CNN is used with a sliding window with 50% overlap to extract 5 seconds of acceleration from a square horizontal-vertical. The results outperform by around 8% concerning the authors of the dataset with p-fold cross-validation [56].

Recurrent Neural Networks (RNNs) and Long Short-Term Memory

Recurrent Neural Networks (RNNs) are built upon feedforward NNs by allowing recurrent edges, i.e., edges that adjacent span timesteps in the network. These results are helpful for problems where data is not independent of time or space, as RNNs can pass specific information across different timesteps [54]; [60]. The most widely used RNNs are Long Short-Term Memory (LSTM) networks [26]. LSTMs introduced the concept of a memory cell replacing the traditional nodes in hidden layers, storing states for given periods. LSTMs and Bidirectional LSTMs have been shown to learn time dependencies and perform well in various tasks successfully.

A wide variety of studies have been conducted with LSTM, and the three most recent are shown in these examples.

Zebin (2018) propose an LSTM model for six life activities (Walking on a level surface, walking upstairs, walking downstairs, sitting, standing, laying) classification with raw accelerometer and gyroscope data. The LSTM achieves 92% average accuracy in a multi-class scenario [61].
Hernandez (2019) propose a HAR system with a smartphone accelerometer and gyroscope data. The model used is bidirectional long short-term memory (Bi-LSTM) network. The activities are sitting, standing, laying, walking, walking upstairs, and walking downstairs. The overall accuracy is 92.67% [62]. Milenkoski (2018) propose a lightweight algorithm for activity detection based on LSTM networks and other models such as J48, Logistic Regression, MLP, and Straw Man with raw accelerometer data. The activity is Downstairs, Jogging, Sitting, Standing, Upstairs, and Walking. The LSTM model only achieves over 95% accuracy with Walking, Sitting, and Standing activities [66].

Other Classification Techniques Used in Activity Recognition

Another paradigm for HAR is one of the fuzzy logic methods derived from fuzzy set theory. The approach is sufficient to identify static postures, such as standing, laying down, and sitting [71]. Few studies have found good performance in fall detection. Fuzzy logic employs methods to construct appropriate membership functions and to combine and interpret fuzzy rules.
Another classification technique is the *Naive Bayes classifier*. For HAR systems, the NB approach shows a similar level of accuracy compared to other classification methods ([29]; Wannenburg & Malekian, 2017; Rodrigues & Mestria, 2016).

2.3 Activities, Datasets, and performances of HAR

This chapter takes on a vital role in the survey. Its importance derives from the in-depth exposition of three key elements: the Activities, the Datasets, and the Performance. Each of these aspects is treated in detail and constitutes a sub-chapter. The first sub-chapter is devoted to the analysis of Activities. The various activities involved in the scope being surveyed were examined and presented in detail. The activities were explored, their distinctive characteristics outlined, and their importance highlighted within the landscape under review. The second sub-chapter is reserved for an in-depth discussion of the Datasets used. The data sources employed for the activities under consideration were exposed. Specifics of the datasets were presented, including size, types of data collected, and collection methods. In addition, the relevance of the datasets in influencing the overall results and evaluations of the activities analyzed was discussed. The third and final sub-chapter deals with Performance. A detailed analysis of Performance related to the activities considered was conducted based on the data provided by the datasets. Metrics and indicators were examined to assess the effectiveness of the methodologies employed in the different activities. Any strengths and limitations of the observed Performance were also identified.

2.3.1 Activities

The section summarizes all the activities found in the Literature, with an in-depth study of all sensors used to recognize each activity. In this section, Table 4 maps the type of activity (or task) with the reference of authors that used that specific task in their study.

Activity	Reference
Walking	[49],[29] [50], [51], [33], [30], [46], [53], [44], [72], [54], [31], [45], [42], [55], [56], [67], [57], [61], [62], [35], [32] [47], [63], [68], [64], [65], [36], [27],[40], [69], [73], [34], [43], [66], , [58], [48], [59], [74], [75], [41], [60], [76], [28], [77], [78], [79], [80], [81], [82], [33], [83], [84], [85], [86], [87], [88], [89] [90] [91] [92] [93] [94] [39] [95] [96] [97] [47]
Jogging	[49], [50], [51], [33], [53], [72], [45], , [35], [65], [36], [40] [34], [66], [58], [60], [77], [81], [88] [90]
Stairs up	[49], [30] , [53], [72], [54], [56], [64], [27], [73], [43], [48], [74], [76], [28], [77], [33], [87]
Stairs down	[49], [30], [53], [44], [72], [54], [56], [64], [27], [73], [43], [48], [74], [76], [28], [77], [33], [87]
Laying Down	[49], [29], [51], [30], [53], [44], [72], [31], [67], [61], [62], [32]), [63], [64], [73], [34], [43], [58], [41], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [39] [95] [97] [47][89]
Sitting	[49], [29]), [51], [30], [46], [53], [44], [72], [54], [31], [55], [67], [61], [62], [35], [32] [63], [68], [64], [36], [40] [73], [34], [43], [66], [58], [48], [74], [41], [76], [28], [77], [78], [79], [80], [81], [82], [83], [84], [55] [42], [85], [86], [87], [88], [89] [90] [91] [93] [94] [39] [96] [97] [47] [95]
Standing	[49], [29], [51], [30], [46], [53], [44], [72], [54], [31], [55], [42], [67], [57], [61], [62], [35], [32] [47], [63], [68], [64], [36], [40], [73], [34], [43], [66], [58], [48], [59], [74], [41], [60], [28], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89] [93] [94][39] [95] [96] [97] [47]
Upstairs	[29], [50], [51], [33], [46], [72], [31], [45], [55], [42], [67], [61], [62], [35], [32] [47], [63], [68], [65], [36], [40] [69], [66], [58], [59], [41], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89] [94][39] [95] [96] [97] [47]
Downstairs	[29] [50], [51], [33], [72], [31], [45], [55], [42], [67], [61], [62], [35], [32] [47], [63], [68], [65], [36], [40] [69], [58], [59], [41], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89] [94][39] [95] [96] [97] [47]
Falling	[50] [90] [91]
Running	[50], [51], [33], [53], [44], [55], [56], [57], [68], [64], [65], [69], [75], [60], [76], [28], [77] [90] [91] [96] [47]
Jumping	[50], [51], [53], [55], [56], [65], [41], [77], [81], [88] [90] [91]
Aerobic Dancing	[33]
Sleeping	[96]
Elevator up, Elevator down	[96], [59]
Waist bends forward, Relaxing, Frontal elevation of arms, Knees bending	[53]
Cycling	[53], [44], [54], [47], [48], [74], [60], [28]
Nordic walking	[44]
Watching TV, Computer work, Vacuum cleaning, Folding laundry, House cleaning	[44], [64]
Car driving	[44], [60] [92]
Ironing	[44], [64]
Playing soccer, Rope jumping	[44]
Write notes, Open engine hood, Close engine hood, Check door gaps, Open door, Close door, Open/close two doors, Check trunk gap, Open/close trunk, Check the steering wheel	[49], [98]
Open then close the fridge, Open then close the dishwasher, Open then close three drawers (at different heights), Open then close door 1/2, Open then close	[49], [96]

door 2, Toggle the lights on then off, Clean the table, Drink while standing, Drink while seated	
Pour milk into cup, Put kettle onto charging point, Reach out for the power switch on the wall, Drink a glass of water while waiting for kettle to boil, Reach out to switch off the kettle, Pour hot water from the kettle into cup, Fetch milk from the shelf, Pour milk into cup, Put the bottle of milk back on shelf, Have a sip and taste the drink, Have another sip while walking back to desk, Unlock drawer, Retrieve biscuits from drawer, Eat a biscuit, Lock drawer, Have a drink, Fetch cup from desk, Place cup on kitchen surface, Fetch kettle, Pour out extra water from kettle, Fetch cup from kitchen surface,	[52]
Brush own teeth, Comb own hair, get up from the bed, lie down on the bed, sit down on a chair, Stand up from a chair, Drink from a glass, Pour water into a glass, Use the telephone	[27]
Sawing, Hammering, turning a wrench, Loading, Hauling, Unloading, Returning	[38]

Table 4 - Activities found in Literature with the corresponding reference.

The graph below summarizes the information shown in the previous table. All the activities with a frequency equal to or less than 3 activities are not reported (Figure 2).

- *Total number of activities*: 29;
- *Total number of references provided*: 437;

- Averages and statistics:
- *Average number of referrals per activity*: 437 / 29 ≈ 15.07 (rounded to two decimal places)
- *Maximum number of references for an activity*: 72 (for "Laying Down," "Sitting," "Standing," "Upstairs," "Downstairs")
- *Minimum number of references for an activity*: 1 (for several activities)

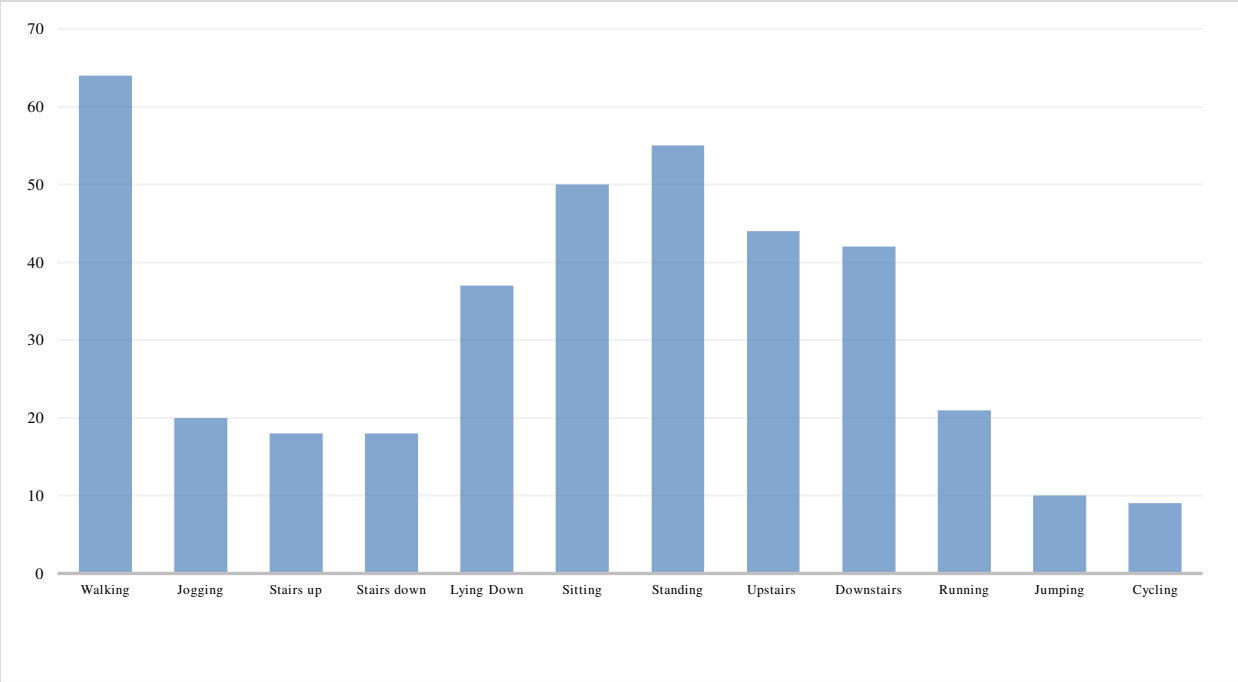


Figure 2 - Frequency of activities found in Literature. The x-axis shows the human activities, while the y-axis shows the frequency of use of the activities in the scientific papers analyzed.

Moreover, the sensors used to recognize the activities shown in the previous table are listed below (Table 5).

Activity	Sensor	Reference
Walking	Accelerometer	[49], [50], [51], [33], [30], [46], [44], [72], [31], [55], [42], [56], [57], [35], [32] [47], [99], [65], [36], [27], [40], [69], [34], [66], [58], [48], [74], [75], [41], [77], [87], [90] [91] [96] ,[47]
	Accelerometer + gyroscope	[29], [53], [54], [45], [67], [61], [62], [63], [64], [73] , [78], [79], [80], [82], [83], [84], [85], [86], [87], [89] [94] [100] [93] [39] [95] [97]
	Accelerometer + gyroscope + magnetometer	[68], [43], [60], [81], [88]
	Accelerometer + gyroscope + magnetometer + barometer	[59]
	Accelerometer + vital sign	[76], [28]
Jogging	Accelerometer	[49], [50], [51], [33], [72], [55], [35], [65], [36], [40], [34], [66], [58], [77], [87] [90] [91]
	Accelerometer + gyroscope	[53], [45]

	Accelerometer + gyroscope + magnetometer	[60], [81], [88]
Stairs up	Accelerometer	[49], [30], [44], [72], [56], [27], [48], [74], [77], [87]
	Accelerometer + gyroscope	[53], [54], [64], [73]
	Accelerometer + gyroscope + magnetometer	[43]
	Accelerometer + vital sign	[76], [28]
Stairs down	Accelerometer	[49], [30], [44], [72], [56], [27], [48], [74], [77], [87]
	Accelerometer + gyroscope	[53], [54], [64], [73]
	Accelerometer + gyroscope + magnetometer	[43]
	Accelerometer + vital sign	[76], [28]
Laying Down	Accelerometer	[49], [51], [30], [44], [72], [31], [32] [34], [58], [41], [77] [47]
	Accelerometer + gyroscope	[29], [53], [67], [61], [62], [63], [64], [73], [63], [79], [80], [82], [83], [84], [85], [86], [87], [89][39] [95] [97]
	Accelerometer + gyroscope + magnetometer	[43], [81], [88]
Sitting	Accelerometer	[49], [51], [30], [46], [44], [72], [31], [55], [35],[32] [36], [40], [34], [66], [58], [48], [74], [41], [77], [87] [90] [99] [47]
	Accelerometer + gyroscope	[29], [53], [54], [67], [61], [62], [63], [64], [73], [78], [79], [80], [82], [83], [84], [85], [86], [87], [89] [100] [93] [94][39] [95] [97]
	Accelerometer + gyroscope + magnetometer	[68], [43], [81], [88]
	Accelerometer + vital sign	[76], [28]
Standing	Accelerometer	[49], [51], [30], [46], [44], [72], [31], [55], [42], [57], [35], [32] [47], [36], [40], [69], [34], [66], [58], [48], [74], [41], [77], [87] [90] [91] [96] [99]
	Accelerometer + gyroscope	[29] [53], [54], [67], [61], [62], [63], [64], [73], [78], [79], [80], [82], [83], [84], [85], [86], [87], [89] [93] [94][39] [95] [97]
	Accelerometer + gyroscope + magnetometer	[68], [43], [60], [81], [88]
	Accelerometer + gyroscope + magnetometer + barometer	[59]
	Accelerometer + vital sign	[28]
Upstairs	Accelerometer	[50], [51], [33], [46], [72], [31], [55], [42], [35], [32] [47], [65], [36], [40], [69], [66], [58], [41] [96] [47]
	Accelerometer + gyroscope	[29] [45], [67], [61], [62], [63], [78], [79], [80], [82], [83], [84], [85], [86], [87], [89] [94][39] [95] [97]
	Accelerometer + gyroscope + magnetometer	[68], [81], [88]
	Accelerometer + gyroscope + magnetometer + barometer	[59]

Downstairs	Accelerometer	[50], [51], [33], [72], [31], [55], [42], [35], [32] [47], [65], [36], [40], [69], [58], [41] [96] [47]
	Accelerometer + gyroscope	[29] [45], [67], [61], [62], [63], [78], [79], [80], [82], [83], [84], [85], [86], [87], [89] [94][39] [95] [97]
	Accelerometer + gyroscope + magnetometer	[68], [81], [88]
	Accelerometer + gyroscope + magnetometer + barometer	[59]
Falling	Accelerometer	[50] [90] [91]
Running	Accelerometer	[50], [51], [33], [44], [55], [56], [57], [65], [69], [75], [77] [90] [91] [96] [47]
	Accelerometer + gyroscope	[53], [64] [100]
	Accelerometer + gyroscope + magnetometer	[68], [60]
	Accelerometer + vital sign	[76], [28]
Jumping	Accelerometer	[50], [51], [55], [56], [65], [41], [77] [90] [91]
	Accelerometer + gyroscope	[53] [100]
	Accelerometer + gyroscope + magnetometer	[81], [88]
Aerobic Dancing	Accelerometer	[33]
Relaxing	Accelerometer + gyroscope	[53]
Sleeping	Accelerometer	[96]
Elevator up	Accelerometer	[96]
	Accelerometer + gyroscope + magnetometer + barometer	[59]
Elevator down	Accelerometer	[96]
	Accelerometer + gyroscope + magnetometer + barometer	[59]
Waist bends forward, Frontal elevation of arms, Knees bending	Accelerometer + gyroscope	[53]
Cycling	Accelerometer	[44], [47], [99], [48], [74]
	Accelerometer + gyroscope	[53], [54]
	Accelerometer + gyroscope + magnetometer	[60]
	Accelerometer + vital sign	[28]
Nordic walking	Accelerometer	[44]
Watching tv	Accelerometer	[44]
	Accelerometer + gyroscope	[64]
Computer work	Accelerometer	[44]
	Accelerometer + gyroscope	[64]

Car driving	Accelerometer	[44] [92]
	Accelerometer + gyroscope + magnetometer	[60]
Vacuum cleaning	Accelerometer	[44]
	Accelerometer + gyroscope	[64]
Ironing	Accelerometer	[44]
	Accelerometer + gyroscope	[64]
Folding laundry	Accelerometer	[44]
	Accelerometer + gyroscope	[64]
House cleaning	Accelerometer	[44]
	Accelerometer + gyroscope	[64]
Playing soccer, Rope jumping	Accelerometer	[44]
Write notes	Accelerometer	[49]
	Accelerometer + gyroscope	[98]
Open engine hood	Accelerometer	[49]
	Accelerometer + gyroscope	[98]
Close engine hood	Accelerometer	[49]
	Accelerometer + gyroscope	[98]
Check door gaps	Accelerometer	[49]
	Accelerometer + gyroscope	[98]
Open door	Accelerometer	[49]
	Accelerometer + gyroscope	[98]
Close door	Accelerometer	[49]
	Accelerometer + gyroscope	[98]
Open/close two doors	Accelerometer	[49]
	Accelerometer + gyroscope	[98]
Check trunk gap	Accelerometer	[49]

	Accelerometer + gyroscope	[98]
Open/close trunk	Accelerometer	[49]
	Accelerometer + gyroscope	[98]
Check steering wheel	Accelerometer	[49]
	Accelerometer + gyroscope	[98]
Open then close the fridge, Open then close the dishwasher, Open then close 3 drawers (at different heights), Open then close door 1/2, Toggle the lights on then off, Clean the table, Drink while standing, Drink while seated	Accelerometer	[49], [96]
Fetch cup from the desk, Place cup on kitchen surface, Fetch kettle, Pour out extra water from kettle, Put kettle onto charging point, Reach out for the power switch on the wall, Drink a glass of water while waiting for kettle to boil, Reach out to switch off the kettle, Pour hot water from the kettle into cup, Fetch milk from the shelf, Pour milk into cup, Pour milk into cup, Put the bottle of milk back on shelf, Fetch cup from kitchen surface, Have a sip and taste the drink, Have another sip while walking back to desk, Unlock drawer, Retrieve biscuits from drawer, Eat a biscuit, Lock drawer, Have a drink.	Accelerometer	[52]
Brush own teeth, Comb own hair, get up from the bed, lie down on the bed, sit down on a chair, stand up from a chair, drink from a glass, pour water into a glass, use the telephone	Accelerometer	[27]
Sawing, Hammering, turning a wrench, Loading, Hauling, Unloading, Returning	Accelerometer + gyroscope	[38]

Table 5 - Co-occurrences between activities and sensors found in Literature with the corresponding reference.

It is noticeable from Tables 4 and 5 that some activities occur more often in the literature. Moreover, it is possible to see that some sensors, or combinations of sensors, have been investigated much more than others. To get a clearer idea of the situation, consult the following graphs (Figure 3).

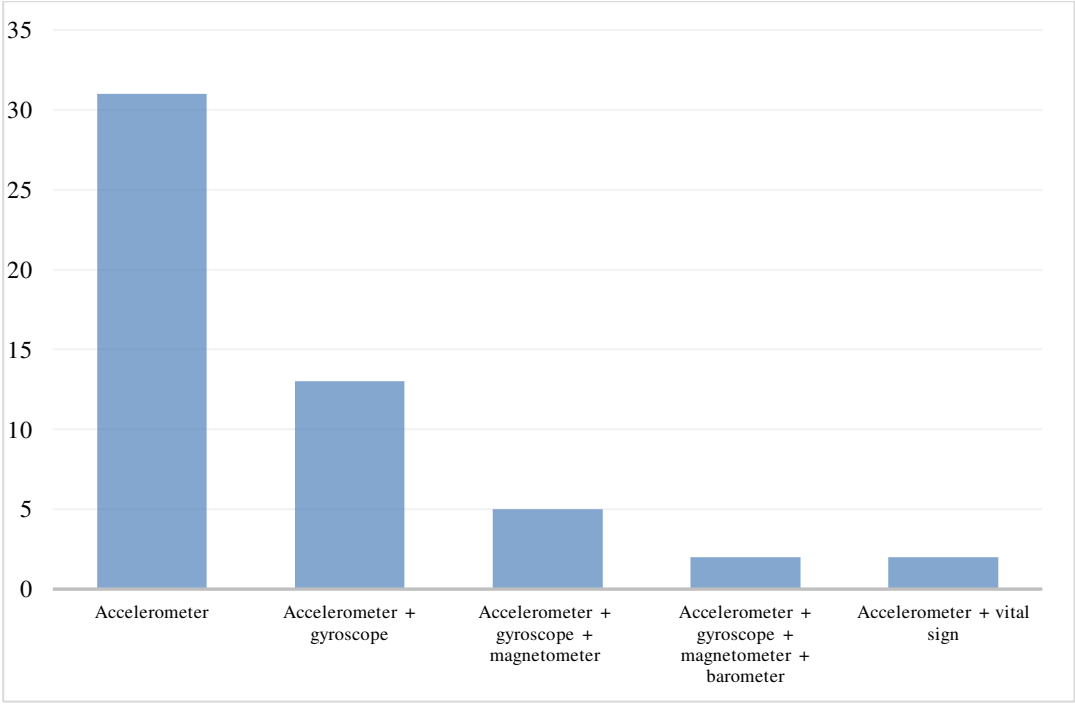


Figure 3 - Frequency of sensors found in Literature. The x-axis shows the sensors, while the y-axis shows the frequency of use in the scientific papers analyzed.

2.3.2 Dataset

Although different authors have adopted various datasets, Table 6 provides a comprehensive overview of the most widely used and publicly available datasets. A brief description of the three most utilized availability datasets is shown below:

WISDM: The following dataset contains data collected by the WISDM research team through controlled laboratory conditions. The team gathered a population of users with Android smartphones while carrying out daily activities. The smartphone was placed in the front trouser pocket. The activities that have been detected are *Downstairs, Sitting, Walking, Jogging, Upstairs, and Standing*. Data collection was supervised to ensure data quality. The data was collected thanks to an application created by the authors and performed on each participant's Android phone. The application allows you to collect sensor data (*e.g., GPS, accelerometer*). The authors collected accelerometer data every 50 Hz in all cases.

Actitracker: This dataset was created by the WISDM research team, the same as the previous dataset. In principle, these two datasets are very similar since the number of subjects that performed the activities and the recorded sensor used is the same as the previous dataset. The main difference with the WISDM dataset is that this dataset contains *"real-world"* data, whereas in the previous one, the data was collected

through controlled laboratory conditions. Moreover, 225 subjects were involved in the activity recording phase. The activities are sitting, standing, walking, jogging, stairs, Laying down. This dataset provides raw, labeled, unlabeled, and transformed data labeled and unlabeled.

Har using a smartphone: The study team collected data considering a statistical population of 30 volunteers (19 to 48 years old). The experiments were done by manually labeling the data. The device was a Samsung Galaxy S2, and the detected activities were climbing stairs, Lying down, descending stairs, sitting, walking, and standing with the smartphone positioned at the waist. The study team captured the triaxial linear acceleration and triaxial angular velocity at a constant 50Hz speed of the accelerometer and gyroscope. The signals have been pre-processed by applying noise filters. Sampling was done in sliding windows with a fixed of 2.56 seconds and 50% overlap (128 reads/window). A feature vector was obtained from each window by calculating time and frequency domain variables.

For each record, the authors provide the following:

- Triaxial acceleration (total acceleration) and estimated body acceleration (accelerometer);
- Triaxial angular velocity (gyroscope);
- A vector of 561 characteristics with time and frequency domain variables;
- The label of the performed activity;
- Subject ID.

D	NoD	S	A	L/RW	RD/PP	Av
UniMiB SHAR	Unknown (30 subjects)	Accelerometer	Falling (Falling Forward, Backward, Right, Left, Hitting An Obstacle, With Protective Strategies, Backward Without Protective Strategies, Syncope), Walking, Running, Climbing Stairs, Descending Stairs, Jumping, Lying Down From Standing, Sitting	L	RD	Public and available online https://paperswithcode.com/dataset/unimib-shar
USC - HAD	MotionNode (14 users)	Accelerometer; Gyroscope; Magnetometer	Running forward, Jumping, Sitting, Standing, Sleeping, Walking forward, Walking left, Walking right, ElevatorUp ElevatorDown, Walking upstairs, Walking downstairs	RW	RD	Public and available online https://sipi.usc.edu/had/
HAR using smartphone	Samsung Galaxy S2 (30 users)	Accelerometer Gyroscope	Walking downstairs, Sitting, Standing, Laying, Walking, Walking upstairs	L	PP	Public and available online https://archive.ics.uci.edu/ml/datasets /Human+Activity+Recognition+Using+Smartphones [101]
Real-world HAR	Unknown (9 users)	Accelerometer; GPS; Gyroscope; Light sensor; Magnetometer; Sound Level Data	Walking, Stairs up, Stairs down, Sitting, Standing, Laying, Running, Jogging, Jumping	RW	RD	Public and available online https://sensor.informatik.uni-mannheim.de/
Unknown	Unknown	Accelerometer	Check_trunk_gap, Open/close trunk, Check steering wheel, Write notes, Open_engine hood, Close_engine_hood, Check_door_gaps, Open_door, Close_door, Open/close two doors	L	PP	Public and available online [102]

WISDM: Wireless Sensor Data Mining	Nexus One; TC Hero; Motorola Backflip (36 users)	Accelerometer	Walking, Jogging, Upstairs, Downstairs, Sitting, Standing	L	D/PP	Public and available online [103]
Actitracker	Android smartphone (29 users)	Accelerometer	Walking, Jogging, Stairs, Sitting, Standing, Laying down	RW	RD/PP	Public and available online [73]
Opportunity	Unknown device name (4 users)	Gyroscopes; Accelerometers; GPS; objects accelerometers; 2Dgyroscopes; switches	Prepare coffee, Drink coffee, Start, Groom Relax, Prepare sandwich, Eat sandwich, CleanupBreak	RW	PP	Public and available online https://archive.ics.uci.edu/ml/datasets /opportunity+activity+recognition [104]
Pamap2	Colibri wireless inertial measurement units	Accelerometer; Gyroscope Magnetometer	Watching TV, Computer work, Car driving, Ascending stairs, Descending stairs, Vacuum cleaning, Ironing, Folding laundry, House cleaning, Playing soccer, Rope jumping, Laying, Sitting, Standing Walking, Running, Cycling, Nordic Walking	RW	RD	Public and available online https://archive.ics.uci.edu/ml/datasets/ pamap2+physical+activity+monitoring
	Unknown device name	Heart rate monitor				
HHAR	LGWatches; Samsung-Galaxy; GearSamsung Galaxy S3 mini; Samsung Galaxy S3; LG Nexus 4; Samsung Galaxy S+ (9 users)	Accelerometer; Gyroscope	Biking Sitting Standing Walking Stair up Stair down	RW	RD	Public and available online https://archive.ics.uci.edu/ml/ datasets/heterogeneity+activity+recognition [105]

Table 6 - Summary and comparison among the different datasets found in Literature. Legend: Name of Devices(D), Number of users (NoD), Sensor(S), Activities(A), Laboratory/Real(L/RW), Raw data/pre-processed (RD/PP), Availability (Av)

2.3.3 Performance

This section is devoted to reporting performance obtained by different systems when adopting solutions previously described. Although many results cannot be compared because they adopted other testing conditions even if the adopted dataset was the same, Table 7 summarizes the most performing. Another view is the graph in Figure 4.

Paper	PP or FE	Classification	Dataset	Sensor	Performances	Year of publication
[90]	Sliding Window, Raw Data	CNN, Bi-LSTM	**Unimib SHAR** (Falling (Falling Forward, Backward, Right, Left, Hitting An Obstacle, With Protective Strategies, Backward Without Protective Strategies, Syncope), Walking, Running, Climbing Stairs, Descending Stairs, Jumping, Lying Down From Standing, Sitting) **Dataset HAR Uniba** (Walking, Running, Jumping, Sitting, Falling (Forward, Backward, Right, And Left))	Accelerometer	**Average Accuracy Dataset HAR Uniba :** 91% CNN 90% Bi-LSTM **Average Accuracy UnimibShar:** 97% CNN 76% Bi-LSTM	**2023**
[91]	Sliding Window, Discrete Fourier Transform	SVM.NuSVC, Bernoulli Naïve Bayes , Random Forest Classifier , Bayesian Ridge, CNN, LSTM, Bi - LSTM	**Dataset HAR Uniba** (Walking, Running, Jumping, Sitting, Falling (Forward, Backward, Right, And Left))	Accelerometer	**Average Accuracy Dataset HAR Uniba :** 91% CNN	

[92]	Sliding Window	CNN separabile in profondità (DS-CNN) e LSTM bidirezionale (Bi-LSTM)	**A Public Domain Dataset For Real-life Human Activity Recognition Using Smartphone Sensors:** Inactive: not carrying the mobile phone. Active: carrying the mobile phone, moving, but not going to a particular place. Walking Driving: Moving in a means of transport powered by an engine. This would include cars, buses, motorbikes, trucks and any similar.	Accelerometer, Gyroscope, Magnetometer and GPS	**Accuracy:** 94.80% (DS-CNN)-LSTM	
[100]	Sliding Window, Raw Data	LSTM RNN	**Ibigworld Dataset** (Walk, Run, Squat, Jump, Lie Down, Swing Arms, Sit and Stand)	Accelerometer Gyroscope	**Accuracy**: 99.59% LSTM RNN	
[93]	Sliding Window, Mean, Median, Variance and Standard Deviation	LSTM, Random Forest, Decision Tree, KNN, SVM	**KAU-COVID19-AR-Dataset (**Walking, Handwashing, Standing, Sitting, Hand Sanitizing , Nose–Eyes Touching , Handshake Drink water)	Accelerometer, Gyroscope, Speed and GPS	**Accuracy:** 97,33% Random Forest	**2022**
[94]	Sliding Window, Overlapping	Ensemble (ELA): GRU, CNN, DNN	**UCI HAR** (Sitting, Standing and Laying down, Walking, Walking upstairs, Walking downstairs)	Accelerometer Gyroscope	**Average Accuracy:** 96.7%	
[85]	Sliding Window; Z-Score Standardization FFDRT	Random forest	**HAR using smartphone** (Walking downstairs, Sitting, Standing, Laying, Walking, Walking upstairs)	Accelerometer Gyroscope	**Accuracy** 92,72%	
[87]	Sliding Window;	CNN+LSTM CNN+RF	**HAR using smartphone** (Sitting, Standing, Laying, Walking, Walking upstairs, Walking downstairs) **WISDM** (Jogging, Walking, Upstairs, Downstairs, Sitting, Standing)	Accelerometer (WISDM); Accelerometer Gyroscope	**Accuracy WISDM**: 94% CNN+LSTM 97,77% CNN+RF **Accuracy HAR**: 97% CNN+LSTM 98,2% CNN+RF	**2021**
[83]	Sliding Window; FFT	1D CNN 2D CNN	**HAR using smartphone** (Walking downstairs, Sitting, Walking upstairs, Standing, Laying)	Accelerometer Gyroscope	**Accuracy** 90,51% 1D CNN 95,69% 2D CNN	
[44]	Sliding Window	CNN	**HAR using smartphone** (Walking downstairs, Sitting, Standing, Laying, Walking, walking upstairs) **Pamap2** (Vacuum cleaning, Ironing, Folding laundry, House cleaning, Playing soccer, Rope jumping, Laying, Sitting, Standing, Walking, Running, Cycling, Nordic Walking, Watching TV, Computer work, Car driving, Ascending stairs, Descending stairs)	Accelerometer	**Accuracy** 92,71% HAR 91,00% Pamap2	**2020**
		LSTM			**Accuracy** 89,01% HAR 85,86% Pamap2	
		BLSTM			**Accuracy** 89,4% HAR 89,52% Pamap2	
		MLP			**Accuracy** 86,83% HAR 82,07% Pamap2	
		SVM			**Accuracy** 90,5% HAR 84,07% Pamap2	
[86]	Sliding Window;	CNN+LSTM	**HAR using smartphone** (Walking downstairs, Sitting, Walking, Walking upstairs, Standing, Laying)	Accelerometer; Gyroscope	**Accuracy** 93,40%	**2019**
[89]	Sliding Window;	LSTM	**HAR using smartphone** (Walking, walking upstairs, walking downstairs, Sitting, Standing, Laying)	Accelerometer; Gyroscope	**Accuracy** 93,13%	

[45]	Sliding Window; Magnitude Signals; Histogram Of Gradient; Fourier Descriptor	SVM	**HAR using smartphone** (Sitting, Standing, Laying, Walking, Walking upstairs, Walking downstairs)	Accelerometer Gyroscope	**Accuracy** 97,12%	
		K-NN			**Accuracy** 91,75%	
[55]	Sliding Window; Z-Score Standardization	CNN	**WISDM** (Jogging, Walking, Upstairs, Downstairs, Sitting, Standing)	Accelerometer	**Accuracy** 91,97%	
[67]	Sliding Window; Magnitude Signals;	MLP	**HAR using smartphone** (Laying, Walking, Walking upstairs, Walking downstairs, Sitting, Standing)	Accelerometer Gyroscope	**Accuracy** 92%Accelerometer 80% Gyroscope 95% Both	**2018**
[48]	Sliding Window; Z-Score Standardization	LSTM	**HHAR** (Biking, Sitting, Standing, Walking, Stair up, Stair down)	Accelerometer	**F-score**: 87,3	
[78]	Sliding Window;	LSTM	**HAR using smartphone** (Walking downstairs, Sitting, Standing, Laying, Walking, Walking upstairs)	Accelerometer Gyroscope	**Accuracy** 94,34%	
[58]	Sliding Window	CNN	**WISDM** (Jogging, Walking, Upstairs, Downstairs, Sitting, Standing) **HAR using smartphone** (Walking, Walking upstairs, Walking downstairs, Sitting, Standing, Laying)	Accelerometer	**Accuracy** 93,32% WISDM 94,35% HAR	
[63]	Sliding Window	Bi-LSTM	**UCI HAR** (Sitting, Standing and Laying down, Walking, Walking upstairs, Walking downstairs)	Accelerometer Gyroscope	**Accuracy**: 93,79%	
[51]	Sliding Window Mean	CNN	**Real-World HAR** (climbing downstairs, climbing upstairs, jumping, lying, standing, sitting, running/jogging, and walking)	Accelerometer	**Precision 0,79**	
[64]	Sliding Window	LSTM	**Pamap2** (Laying, Sitting, Nordic Walking, Watching TV, Computer work, Car driving, Vacuum cleaning, Ironing, Folding laundry, House cleaning, Playing soccer, Ascending stairs, Descending stairs, Rope jumping, Standing, Walking, Running, Cycling)	Accelerometer Gyroscope	**Accuracy**: 82.57%	**2017**
[39]	Sliding Window; Z-Score Standardization Correlation PCA LDA	HMM	**HAR using smartphone** (Walking, Walking upstairs, Walking downstairs, Sitting, Standing, Laying) **USC-HAD** (Walking downstairs, Running forward, Jumping, Sitting, Standing, Sleeping, Elevator up, Elevator down, Walking forward, Walking left, Walking right, Walking upstairs)	Accelerometer Gyroscope	**Accuracy** 93,18% HAR 67,07% USC-HAD	
[98]	Sliding Window	CNN	**Skoda** (Write notes, Open engine hood, Close engine hood, Check door gaps, Open door, Close door. Open/close two doors, Check trunk gap, Open/close trunk, Check steering wheel)	Accelerometer Gyroscope	**Accuracy** 97,92%	
[73]	Sliding Window; PCA	MLP	**HAR using smartphone** (Walking, Walking upstairs, Walking downstairs, Sitting, Standing, Laying)	Accelerometer Gyroscope	**Accuracy** 96,17%	
[96]	Sliding Window	DNN	**Opportunity** (Start, Drink coffee, Prepare sandwich, Eat sandwich, Cleanup, Break, Groom, Relax, Prepare coffee) **USC-HAD** (Walking downstairs, Running forward, Jumping, Sitting, Standing, Sleeping, Elevator up, Elevator down, Walking forward, Walking left, Walking right, Walking upstairs)	Accelerometer	**Accuracy** 82,3% Opportunity 91,17% USC-HAD	**2016**
[95]	Sliding Window; FFT	Deep CNN	**HAR using smartphone** (Walking, Walking upstairs, Walking downstairs, Sitting, Standing, Laying)	Accelerometer Gyroscope	**Accuracy** 95,75%	

[35]	Event-Defined Window (Custom Algorithm For) Segmentation	K-NN	**Actitracker** (Walking, Jogging, Stairs, Sitting, Standing, Laying down)	Accelerometer	**Accuracy**: 96%	
[31]	Sliding Window	HMM	**HAR using smartphone** (Sitting, Standing, Laying, Walking, Walking upstairs, Walking downstairs)	Accelerometer	**Accuracy** 83,51%	
	Sliding Window	Ensemble learning with vote (J48, Logistic Regression, MLP)	**WISDM** (Jogging, Walking, Upstairs, Downstairs, Sitting, Standing)	Accelerometer	**Accuracy** 91,62%	
[36]	Fft; Autoencoder (Pre-Trained Network For F.E)	(4-layers) ANN	**HAR using smartphone** (Sitting, Standing, Laying, Walking, Walking upstairs, Walking downstairs) **WISDM** (Jogging, Walking, Upstairs, Downstairs, Sitting, Standing)	Accelerometer	**Precision** 77,01% HAR 80,02% WISDM	**2015**
[49]	Sliding Window	CNN	**Skoda** (Open/close two doors, Check trunk gap, Open/close trunk, Write notes, Open engine hood, Close engine hood, Check door gaps, Open door, Close door, Check steering wheel) **Actitracker** (Sitting, Standing, Lying down, Walking, Jogging, Stairs) **Opportunity** (Drink coffee, prepare sandwich, Eat sandwich, Cleanup, Start, Groom, Relax, prepare coffee, Break)	Accelerometer	**Accuracy** 86,18% Skoda 91,00% Actitracker 75,83% Opportunity	
[97]	Sliding window; Z-score standardization RF features selection	HMM	**HAR using smartphone** (Sitting, Standing, Laying, Walking, Walking upstairs, Walking downstairs)	Accelerometer Gyroscope	**Accuracy** 91,76%	**2014**

Table 7 - Summary of the experimentation settings with performance scores found in the Literature, sorted concerning the year of publication

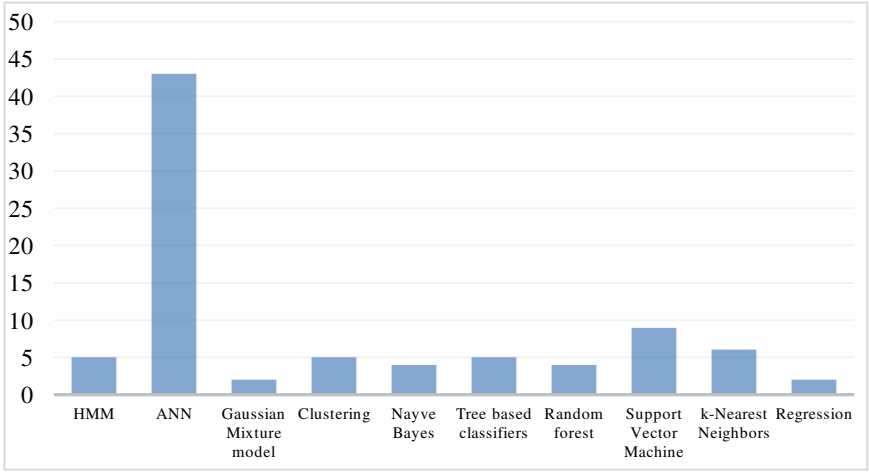


Figure 4 - Shallow Learning and Deep Learning Classification Algorithms. The x-axis shows the Machine Learning Shallow/Deep Models, while the y-axis shows the frequency of use of the shallow/deep models in the scientific papers analyzed correlated to the HAR approach

This research has suggested several techniques and structures (Table 7) to enhance the precision of contemporary HAR models. One among them is the *Fast Feature Dimensionality Reduction Technique (FFDRT)*, which reduces feature dimensionality by overcoming the issues of power limitations of smartphones without losing degrees of accuracy [85].

In the literature, other models prefer the synergy of CNNs with different architectures, such as LSTMs and GRUs, creating more complex networks that increase performance [87]. CNN-LSTM integration has found superior temporal understanding helpful for dynamic HAR recognition [86]. CNN architectures with 1D and 2D kernels capture sensor data dependencies. Instead, Fast Fourier Transform (FFT) powerfully amplifies the results [83]. Stacked LSTM networks were considered to help with smartphone behavior identification for incremental learning [89]. Deep convolution neural network models have also been proposed to emphasize location-independent recognition for high accuracy of activity and location [51].

For the *"Device heterogeneity challenge,"* various feature extraction, normalization, and deep CNN techniques have been proposed to improve HAR among different devices [48]. Then, *Continuous Hidden Markov Models (CHMMs)* were introduced to exploit activity hierarchy and achieve better classification performance [39]; [97]. To enrich the methods employed and capture temporal and spatial dependencies in sensor data, multimodal CNNs with 2D kernels have been presented, outperforming traditional methods [98]. However, dimensionality reduction through PCA and neural networks such as MLP provide excellent performance without compromising accuracy [73].
Although deep learning techniques (CNN-LSTM) have dominated the HAR landscape, there is also a modern inclination toward ensemble and hybrid approaches. These methods leverage the strengths of multiple models to improve recognition accuracy [40]; [31].

The approaches, strategies, and models used in 2022–2023 scientific articles are the same as those used in prior years. On the other hand, they have presented several concepts that might improve current research and future advancements. In particular, *Human Activity Recognition (HAR)* architectures have been used by [90]; [91] to detect bullying and cyberbullying even in the setting of *Behavioral Biometrics*. The first years these models may be used on contemporary smartphones are 2023 and 2024.

2.4 Discussion

This chapter covers a discussion focusing on the central aspects of the study, including *"Points in Common"* about the scientific papers reviewed in the survey. The *"Advantages"* and *"Disadvantages"* arising from these commonalities and differences are examined in depth, shedding light on their implications for the overall goals of the research. As it progresses to "*Future Developments,"* it is necessary to extend the research. At the same time, a meticulous assessment of the associated "*Risks*" is presented, pointing out potential pitfalls that must be considered in subsequent undertakings. A mention is made of the *"Best"* and *"Worst"* results regarding accuracy. By meticulously dissecting the results, decipher how much the research aligns with the goals set. To tangibly illustrate these notions, "*Realistic Examples*" are introduced, clarifying the practical implications of the study results in real-world scenarios. By consolidating these examples, it was provided an essential reference for subsequent attempts to develop the present work.

Points in Common: There is a similar theme in the reality of Human Activity Recognition (HAR) throughout several study projects with unique aspects. Accelerometer and Gyroscope data are the main emphases since they are essential for understanding user movements and activities in real-time. These sensors are widely used to measure everyday activities, fitness, and health. They are ubiquitous in smartphones and wearable devices. These works include a variety of machine-learning techniques, such as ensemble approaches and Hidden Markov Models. The methods include Principal Component

Analysis (PCA), Multilayer Perceptron's (MLPs), and Convolutional Neural Networks (CNNs), which are all carefully used to improve the accuracy of activity identification. Temporal factors are crucial because Activity Detection relies heavily on the temporal element of sensor data. Various papers examine the temporal interdependence and sequential patterns in time series data, and they use CNNs, HMMs, and continuous HMMs to use these temporal features for better recognition performance.

The hierarchical approach (leveraging two-level models and continuous HMMs, exploring complex actions into simpler components) enhances overall recognition accuracy and facilitates more precise activity classification.

Datasets like the UCI HAR and WISDM datasets play a crucial role, offering standardized frameworks for assessing the efficacy of various strategies. These datasets support the joint endeavor to develop HAR using thorough and equivalent evaluations. The suggested methods emphasize usability and real-time detection, which aligns with the viability of using smartphones as data collection instruments in real-world situations.

Advantages: The models adaptability, which enables application across several domains and use cases, is one of its significant strengths. Numerous potential use cases, ranging from health monitoring to fitness tracking, demonstrate the suggested models' versatility and wide range of applications. High levels of accuracy are achieved when the *CNN-based strategy* and the *two-stage HMM method* are combined. The improved privacy these methods offer by doing away with the requirement for personally identifiable data is an important issue they address. According to [16], a privacy-centric strategy protects sensitive data and improves the models' applicability to various user groups. In modern settings, when data security and user confidentiality are critical concerns, this emphasis on privacy is essential.
Another noteworthy strength is applying Convolutional Neural Networks (CNNs) to capture local relationships in the data successfully. It also includes a Hidden Markov Model (HMM) with two stages. This variety of techniques offers a variety of ways to portray and understand data.

Disadvantages: There are several disadvantages to using sophisticated models like convolutional neural networks (CNNs). First, its implementation necessitates significant resources, which restricts deplorability in some situations. Furthermore, the complexity of these models could increase the chance of overfitting, reducing their generalizability.
The lack of publicly accessible datasets designed for Human Activity Recognition (HAR) is another drawback. This constraint impedes the advancement of this subject by making it more challenging to design and evaluate HAR models. Moreover, depending too much on essential characteristics to extract discriminative information may hinder the method's ability to identify nuanced patterns in the data. This constraint highlights the need for more advanced methods to understand human activities completely. Using sophisticated models has risks such as the possibility of overfitting to datasets, decreasing generalizability, and affecting dependability in both theoretical and real-world scenarios and real-world situations.

Future Developments: In considering future developments in Human Activity Recognition (HAR), various avenues have been explored to enhance the applicability of methods in daily life.

- Extending the scope to activities such as cooking and sweeping has been suggested, broadening these systems' utility and contributing to a deeper understanding of human behavior. Moreover, integrating HAR models into mobile devices has been proposed to improve accessibility and real-time responsiveness. This potential direct implementation on smartphones aligns with the prevalence of mobile device use in everyday life, emphasizing the need for prompt and effective activity detection.

- Recommendations for enhancing feature extraction methods have been put forth, offering the potential for gaining fresh insights into interpreting time-series data. Exploring comprehensive feature extraction methods, manually created and automatically learned, remains a common theme across publications. Techniques such as genetic programming are considered to pursue ideal feature representations continuously.
- An essential area of study centers around the feasibility of embedding recognition models directly into smartphones for real-time activity detection. This aligns with the increasing demand for efficient identification methods in the context of widespread mobile device usage.
- Several research proposals aim to advance the HAR field, including projects focused on dataset expansion and testing on confidential data. These endeavors seek to bolster the robustness and versatility of recognition systems, enabling their application across diverse tasks.
- A promising avenue for investigation involves the integration of model unsupervised pretraining and optimization approaches, potentially contributing to increased recognition precision and the discovery of novel ways to describe features. These proposed strategies open new possibilities for refining HAR methodologies and addressing challenges in this evolving field.

Best/Worst Results Regarding Accuracy: The analysis of best and worst results concerning accuracy in human activity recognition reveals intriguing findings.

Two-stage *Hidden Markov Model (HMM)* frameworks performed best, while the state-of-the-art CNN-based method showed impressive accuracy. The utilization of sophisticated techniques, including ensemble methods, Hierarchical Hidden Markov Models, Convolutional Neural Networks, and Deep Neural Networks, is responsible for the success of these models. The discipline has significantly benefited from the novel methodologies, which constantly produce better or at least comparable results, indicating noteworthy progress in identifying human activity.

The two-stage *Continuous Hidden Markov Model (CHMM)* stands out for its hierarchical classification, enabling efficient feature utilization and achieving the highest reported accuracy. On the other hand, the *CNN-based approach* showcased strong discriminative feature extraction capabilities, surpassing state-of-the-art methods and affirming its cutting-edge efficacy. In contrast, the HMM approach [11], exhibited a lower accuracy of 83.51% when compared to innovative methods, positioning it as the least favorable option in terms of performance within the context of this analysis.

Realistic Examples:
Health Monitoring: The HAR study can provide clinicians with a tool to monitor daily activities and detect abnormalities during the patient's routine. HAR is also helpful in Parkinson's disease screening to accurately identify symptoms and monitor the course of the condition by tracking its evolution described in terms of the severity of the disease during a time [67]. Additionally, the HAR monitors patients' rehabilitation to provide a more accurate status of their health condition and assist them in assessing their improvements [67]. Real-world examples of "*Health Monitoring*" are focal in HAR. These models, for example, can monitor whether a patient in accident rehabilitation is performing the activities (exercises) prescribed by the physician; Can be applied to patient care, therapy, and fitness tracking. Reflects wide-ranging impact on wellness and medical fields;

Safety: HAR systems can automatically send alerts if the user falls [30]. In this case, an automatic message can be sent to their relatives, or an automated emergency call could be triggered to save the patient. Smartphones could be used instead of specialized wearable systems ([68]), they have greater flexibility because they do not need to be tied to a receiver and have many response methods, such as sending a text message, an email, or an automatic call. Additionally, smartphones are cheaper, and users can afford a greater audience. Chen et al. proposed a pervasive fall detection system based on mobile phones. Yao et al. proposed a system that automatically detecting dangerous vehicle

maneuvers to prevent car accidents [30]. The solution required a mobile phone placed in the vehicle, whose accelerometer, gyroscope, and magnetometer are used to collect data and compare it against typical dangerous driving patterns obtained from real driving tests [33]. Real-world examples of "safety" can be related to the person's safety. These HAR models can monitor bullying activities with the purpose of prevention. It may be helpful to send a smartphone alert to caregivers immediately when a fall, push, or run activity is detected;

Context-aware Behavior: HAR can be used to customize the device's behaviors according to the high-level activity that the user is performing, such as *"at work"* or *"walking home"*; for instance, the system may be able to turn off incoming calls or set the device mode to silent while the user is working, or if the user is exercising it may play music from a pre-selected playlist. When the user moves, reading the screen is more complex and dangerous. This problem can be overcome by recognizing the user context to automatically adapt the interface and the display methods, thus improving the user experience. Also, the phone volume may be automatically adjusted and increased while the user is walking, and it is subjected to extra noise caused by motion. Real-world examples in "*Context-aware behavior*" can be related to behavioral biometrics topics. Training HAR models is helpful in continuous authentication approaches. They can be related to Touch Dynamics approaches and can thus enhance smartphone security;

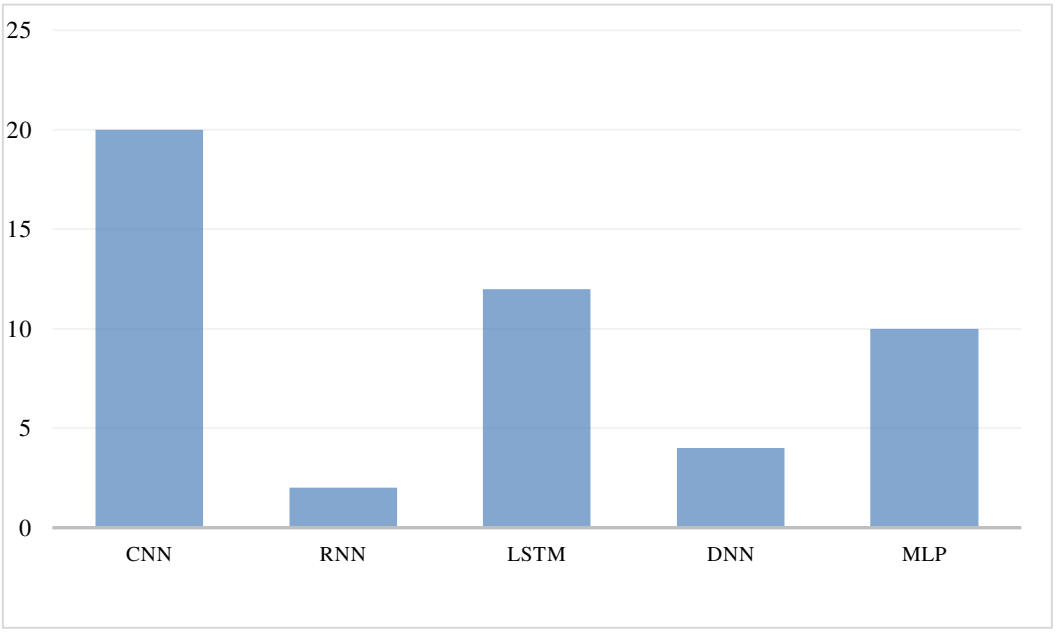
Fitness Tracking: Fitness tracking monitors metrics such as steps taken, calories burned, and heartbeat. Some advanced systems can observe much more sophisticated metrics, such as climbing floors and sleep quality. Modern smartphones include a built-in accelerometer, a gyroscope, and a global positioning system such as GPS or GLONASS, making fitness tracking available to a broad audience. While specialized devices such as wristbands or sports watches are of a particular purpose, a smartphone is instead a general-purpose device not purchased explicitly by users exclusively interested in activity tracking. Real-world examples in "fitness tracking" can be focal to detect whether one is performing an exercise well. Training HAR models helps monitor parameters such as steps taken, calories burned, and heart rate.

2.5 Conclusions

Human Activity Recognition (HAR) is defined as the problem of identifying a physical activity performed by an individual as a function of a motion-tracked in each environment. This work is an extensive survey regarding the most widely used sensors, detectors, datasets, and classifiers to identify the most performing state-of-the-art techniques for HAR. Since the context of this work is to start implementing HAR methodologies, it is necessary to consider the most natural position for carrying the smartphone. When not used, the most natural place to store a smartphone is in your pocket. Moreover, the literature shows that the best position to perform the HAR task is precisely the waist ([34]; [53]; [54]). For this reason, it is necessary to consider all datasets that contain records of devices brought in the waist. Several studies have been made to understand the differences in classification performances when smartphones are carried in hand and waist ([13]; [28]). The classification performances with the smartphone carried in the waist were generally better. The literature also shows that accelerometers and gyroscopes are the most used smartphone sensors. The accelerometer generally performs better than the gyroscope, but by combining these two sensors, it is possible to get better classification results [9] [45] [12]. In most documents, the most easily distinguishable activities are Standing, Laying, Walking Downstairs, Sitting, Walking, and Walking Upstairs. Many activities need to be more easily differentiated from the three axes of the accelerometer. To favor them, the gyroscope could accumulate in the Literature. The most recurrent Datasets in Literature are the WISDM Lab research team, WISDM, Actitracker, and the UCI research team dataset Har using a smartphone. These three datasets contain waist-recorded data. The first two have only accelerometer-recorded data, while the last includes accelerometer and gyroscope data. For what concerns the classification task, it is possible to see that Convolutional Neural Networks (CNNs) and Long Short-Term Memory

Recurrent Neural Networks (LSTM) are the most frequent in the Literature and the ones that seem to lead to the best results (see Figure 5). More implications and analysis have been added in chapter *"Discussion."*

Figure 5 - Deep Learning Classification Algorithm. The x-axis shows the Machine Learning Deep Models, while the y-axis shows the frequency of use of the deep models in the scientific papers analyzed correlated to the HAR approach



Moreover, unlike other technologies from Literature, the *Neural Network* performs well without using complex feature extraction techniques. This can be an advantage since the proposed scenario requires subjects to use their smartphones, each of which can have different kinds of the previously presented sensors. For this reason, a novice reader might delve further into these technologies to better understand how they perform under other conditions by using the previously mentioned datasets in different ways.

3. Cyberbullying from Detection to Evaluation

3.1 BBQuestionnaire Design

The subchapter introduces the pivotal research instrument known as the *"BullyBuster Questionnaire,"* which serves a foundational role in the study and analysis of data central to this investigation. The primary aim of this chapter is to offer an in-depth overview of the phases inherent to the questionnaire's application, presenting comprehensive guidelines to ensure its correct deployment.

Sub-chapter 3.1.1 Requirements Specification*,* delves into the questionnaire's requirements, covering the design process, the selected questions, and the methodologies employed. This phase is essential to ensure that the questionnaire effectively gathers pertinent information to address key research questions, upholding both its validity and reliability.

Following this, in Sub-chapter 3.1.2 Design*,* the design phase of the BB Questionnaire is thoroughly examined. Here, the structure of the questionnaire is scrutinized, with careful selection of criteria and grouping of questions to enhance participant comprehension and streamline data analysis. Additionally, the rationale behind the development of both smartphone and web application versions of the questionnaire is discussed, highlighting their roles in accessibility and user engagement.

3.1.1 Requirements Specification

In this subchapter, a pivotal phase in the analysis of requirements is explored, underscoring a foundational process integral to the success of any software development endeavor. This phase acts as the critical juncture between the conceptualization of the initial idea and the practical implementation of the system, serving an essential function in delineating the objectives the software must achieve and how it will accomplish them.

The analysis commenced with an in-depth examination of the Requirements Specifications (3.2.1 Requirements Applications), detailing all the functionalities and attributes essential for the system's fulfillment of objectives. Both functional and non-functional requirements were thoroughly assessed, ensuring alignment with the anticipated software goals. Following this, Class Diagrams, UML, and Use Cases (3.2.2 Class diagram, UML and use cases) were meticulously evaluated. These tools facilitated a clear and coherent depiction of the system's architecture and structure, highlighting the foundational concepts of objects, relationships, and interactions among the various software components. Specifically, the Class Diagram offered a high-level perspective on system entities and their interrelations, while Use Cases provided deeper insights into the interaction scenarios between users and the system.

The proper analysis of requirements, coupled with the effective utilization of these powerful graphical representations, furnished a solid and clear foundation for the continuation of the development project. Only through thorough and precise analysis could the accurate implementation of the required functionalities and the complete satisfaction of the end-user's needs be ensured.

3.2.1 Requirements Applications

In the present sub-chapter, the detailed analysis of the requirements specifications was focused on. The distinction between functional and non-functional requirements was comprehended, emphasizing the crucial importance that both hold in the product development process. Functional requirements constituted the system's backbone, delineating its primary actions, functions, and services. They were presented clearly and concisely to ensure stakeholders could fully understand them and assess their relevance to their needs. On the other hand, due prominence was given to non-functional requirements, often overlooked but equally significant. These criteria defined the system's performance, security, usability, and scalability features, ensuring that the final product was reliable, efficient, and capable of meeting the required quality standards.

Functional Requirements:

1. Access to application functionality.
   - The user must be able to access the application.
2. Acceptance of smartphone permissions.
   - The application must request permission from the user to access device-specific features necessary to function correctly, such as access to touch and smartphone hardware sensors.
3. Cyberbullying video viewing.
   - The user must be able to view four videos related to cyberbullying.
4. After each video, the user must be able to answer an emotional question associated with the video.
   - Access to Questionnaire.
5. The user must have access to a questionnaire containing 99 questions.
   - The user must be able to fill out the questionnaire questions sequentially.
6. End of Test.
   - The user must be able to uninstall the application after completing the test.

Nonfunctional Requirements:

1. Security
   a. The application must ensure the security of user data.
   b. Information related to test results and questionnaire responses must be protected from unauthorized access.
2. Usability
   c. The application must be designed with an intuitive and easy-to-use user interface.
   d. The videos and questionnaires must be presented in a clear and accessible format.
3. Performance
   e. The application must have smooth and responsive performance while viewing the videos and completing the questionnaire.
4. Compatibility
   f. The BullyBuster Questionnaire smartphone application is implemented exclusively for Android operating systems. For the app's features to work correctly, it is necessary for the SDK version installed on the smartphone to be 24 or higher, which is the Android 7.0 version, more commonly known as Android Nougat. The Web application is also accessible for IOS devices.
5. User Consent
   g. The application must explicitly ask for the user's consent to access the device features and to participate in the cyberbullying test. Informed consent was included in the Android application, agreed upon, and written by the project lawyers.
6. Localization and supported languages
   h. This version of the app is not designed for international users. Italian, to date, is the only language set.
   a. Data backup and restore (if needed)
   7. scalability
   b. Code maintainability and the constant support and updating of the application.
7. Online/offline operation: The app sent all functionality to the server only when connected to the Internet (Mobile or Wi-Fi). In offline mode, the app did not send any data to the server.
8. Connection to server: All features extracted from cell phones and tablets were saved on the server. This server had accounts and passwords in the custody of system administrators;
9. User registration: No user registration is contemplated in the following app. Every detected functionality contemplates privacy regulation, and every data detected and saved is kept anonymous.

App Features:

BullyBuster Questionnaire is an app that aims to collect data to prevent bullying and cyberbullying. The app contemplates five main and consecutive steps to be able to perform the entire test:

- Grant the permissions that the app requires, i.e., access to location and Sensor and Keylogger services;
- View four videos depicting situations of cyberbullying among youth through animated *"cartoon"* scenes. An emotional question is asked at the end of each video to record the emotion experienced after viewing it;
- Fill out a questionnaire made with the collaboration of psychologist Grazia Terrone to identify the user's attitude to the subject matter. Each question has a mandatory answer. Otherwise, it is impossible to continue;
- Open a debate on an ad-hoc created Telegram group among the test participants;
- Conclude the test and uninstall the app. Uninstalling the app also involves ceasing all data acquisition;
- The timeline for completion of the entire test has been estimated at 30 minutes; everything depended on the final debate.

Data Acquisition: During the test, the app recorded the following features that were saved to the server:

- *KeyLogger*: Everything written on the keyboard;
- *Touch and Multi-Touch coordinates* on cell phone display during the test (e.g., Play on video, scrolling back and forth of video, etc....);
- Answers to the questionnaire (date and time, question, answer);
- *Sensor values*: Gyroscope, Accelerometer, Proximity, Atmospheric Pressure, Magnetometer, *Ambient Brightness, StepDetector, GPS* (Some cell phones do not have all sensors);

3.2.1.1 Informed consent

This sub-chapter will elaborate on the critical aspect of "*Disclosure under Art. 13 EU Regulation No. 679/2016 and the relevant Italian legislation as amended*". The following disclosure has been added to the Android application.

Last updated October 2022

This privacy policy clearly explains what information we collect, how we use it, and what rights we have in retrieving it. The BullyBuster Questionnaire application is part of the PRIN2017 - BullyBuster Project activities - A framework for bullying and cyberbullying action detection by computer vision and artificial intelligence methods and algorithms.

Please read carefully what is stated in the indications that will be released on the use and how to enter information through the app and decide, in total freedom, whether or not to participate in this study that has scientific research purposes and is promoted by the Universities of Naples, Cagliari, Bari, and Foggia. There are no risks or contraindications in participating. If there are terms in this privacy policy with which you disagree, you can abruptly stop using our Services by uninstalling the app.

Summary

7. USE OF THE APP
8. UPDATES TO THIS NOTICE
9. CONTACTS

1. MOBILE SOFTWARE
This notice grants you a license to use a complete copy of the BullyBuster Questionnaire Software code associated with the account you own or lease. This license is non-exclusive, non-transferable, revocable, and intended for personal use only. It is prohibited to:
(i) Modify, decompile, or reverse engineer the Mobile Software, except as otherwise explicitly provided by law;
(ii) Rent, lease, lend, resell, sublicense, distribute, or otherwise transfer to third parties the BullyBuster Questionnaire Software or share the Mobile Software in any way with third parties;
(iii) Copy the Mobile Software;
(iv) Remove, circumvent, disable, damage, or otherwise interfere with security features of the Mobile Software, features that prevent or restrict the use or copying of any content accessed through the Mobile Software, or features that enforce limits on the use of the Mobile Software;
(v) Remove copyright or other notices showing ownership rights in the Mobile Software. Update versions of the Mobile Software will be released periodically, and such versions may automatically update the version of the BullyBuster Questionnaire Software in use on your device. You authorize the automatic update on your mobile device and agree that the terms and conditions of this agreement will also apply to the updates. Any third-party code string included in the BullyBuster Questionnaire Software is covered by the third-party or open-source license in the EULA, which authorizes using that code string. The license granted is not a sale of the BullyBuster Questionnaire Software or any copy thereof, and all rights and interests in the BullyBuster Questionnaire Software (and any copy thereof) remain with the partners. Any attempt to transfer the underlying rights, duties, or obligations, except as expressly provided in this Agreement, is void.

2. PURPOSE OF PROCESSING.
The processing of the acquired data is for scientific research in the characterization and identification of activity attributable to bullying and cyberbullying. It is reiterated that no personal information will be extracted or requested, meaning all information that is used to identify and contact a person, such as name, e-mail address, username, address, and phone number. The data relating to the USER's interaction with the app will be collected. They will be unknown to the data controller and recorded as USER 1, USER 2, USER 3, and so on.

3. INFORMATION COLLECTED
Information is collected only and exclusively while the user is interacting with the running app. The information collected is as follows:
- Device Functionality: While using the app, access to certain device functionality is collected, such as recording Keylogger activity (every single click action and every single word typed from the keyboard by the user), cell phone sensoristics, device hardware and software information on servers. Behavioral information will also be collected. All the information content of the experiment will be saved on servers.
- Mobile device data: While using the app, information about the device (model and manufacturer), operating system, and version information is automatically collected to maintain the app.
This information is needed to maintain the security and operation of the App, for troubleshooting, and for internal research purposes of data collection and error analysis.

4. METHODS OF PROCESSING.

User interactions with the app will be processed for research purposes to study the phenomenon of bullying and cyberbullying. The app is intended to collect data that will later be processed for statistical and research purposes.

5. INFORMATION RETENTION TIME
Unless otherwise required by law, we will retain user information for as long as necessary to fulfill the purposes described in this privacy policy.

6. SECURE INFORMATION
WHEREAS the user's identity cannot be traced (except to the extent of USER1, USER 2, USER 3 and so on), we also aim to protect user iteration information through a system of organizational and technical security measures.
Appropriate technical and organizational security measures have been implemented to protect the security of the information being extracted from the app. However, despite safeguards and efforts to protect the information, no electronic transmission over the Internet or information storage technology can be guaranteed to be 100% secure, so it is not possible to promise or guarantee that hackers, cybercriminals, or other unauthorized third parties will not be able to defeat our security and improperly collect, access, steal, or modify the information.

7. USE OF THE APP.
The information listed in Step 3 will only be collected while you are using the app. You can uninstall the app at any time. From then on, no more information will be collected. If the app is reinstalled on the same device, the system will assign a new user ID to that instance.
8. UPDATES TO THIS NOTICE
The following notice will be updated and modified by relevant laws.
The updated version will be indicated by an updated "Revised" date and will go into effect as soon as it is accessible. You are encouraged to review and re-read this privacy notice.

9. CONTACT
If you have any questions or comments about this notice: mailinfobully@yahoo.com

3.2.2 Class diagram, UML and use cases
The **Class Diagram** stands as one of the core diagrams in software engineering and object-oriented programming. It offers a graphical representation of the classes within a software system and illustrates the relationships among them. This diagram is particularly prevalent in the realm of Java programming.

A Java class diagram provides a visual depiction of the static architecture of a Java application, displaying the classes, attributes, and methods that compose the system, as well as the interrelations among various classes. This diagram forms an integral part of the Unified Modeling Language (UML), a standardized language for modeling software systems. Notably, the class diagram is specific to the Android application, as it is developed in Java, whereas the UML diagram encompasses both Android and iOS platforms.

The diagram is shown in Figure 6.

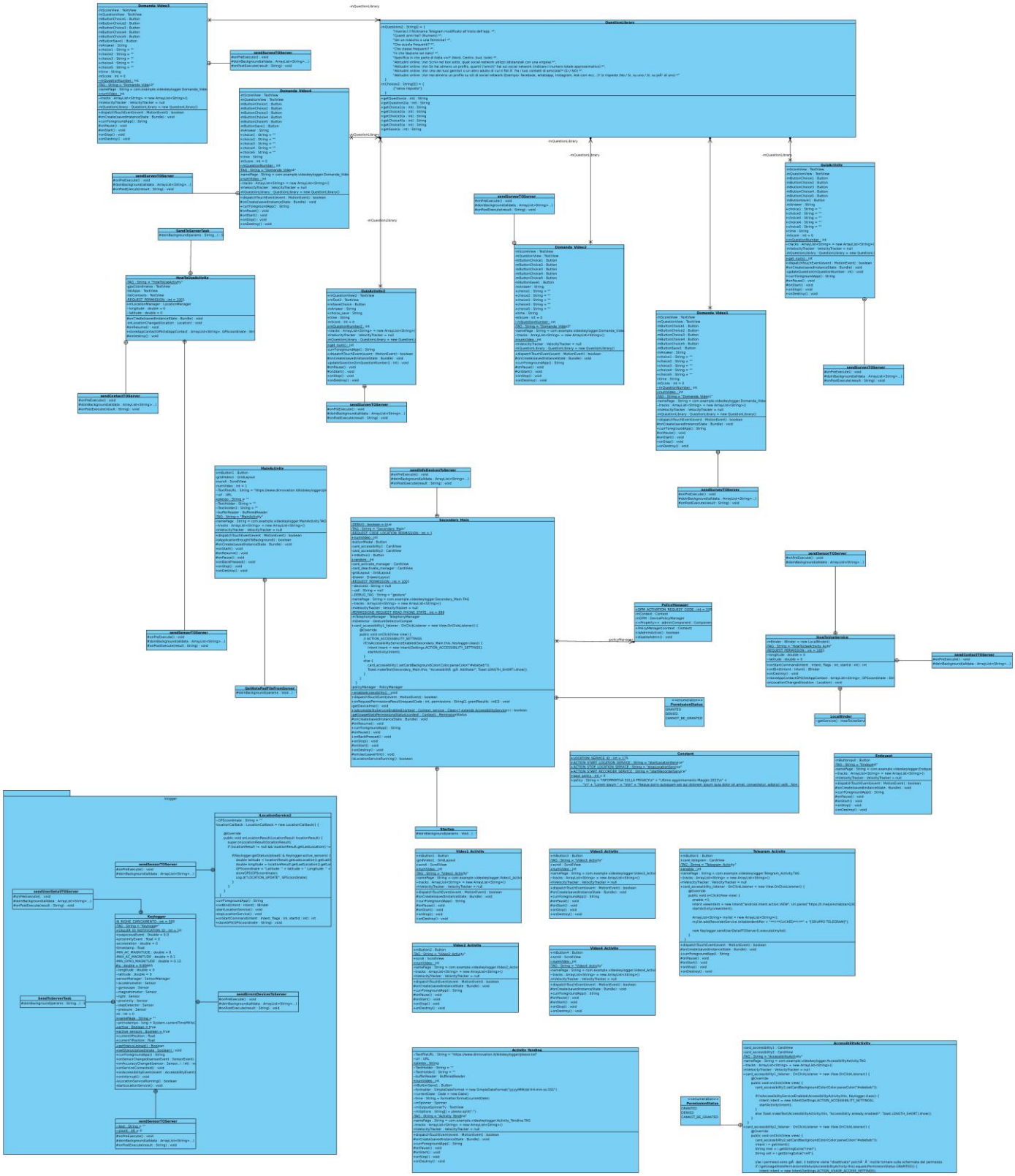


Figure 6 - Diagram of classes Android Application

Instead, *Unified Modeling Language (UML)* diagrams are a graphical standard for modeling and documenting software systems and business processes. They provide a visual representation of various aspects of a system, enabling developers, analysts, and stakeholders to understand better the system's structure, behavior, and interactions. UML is widely used in software engineering and development

processes to improve understanding of projects and facilitate communication between different stakeholders.

Use cases represent a distinct category of UML diagrams designed to capture and articulate the interactions between a system and the actors—be they users, other systems, or external components—that engage with it. These diagrams emphasize the system's functionality from the viewpoint of the participating actors, thus aiding in the identification of the system's functional requirements. Presented below is a comprehensive outline of the smartphone application questionnaire addressing cyberbullying (Figure 7):

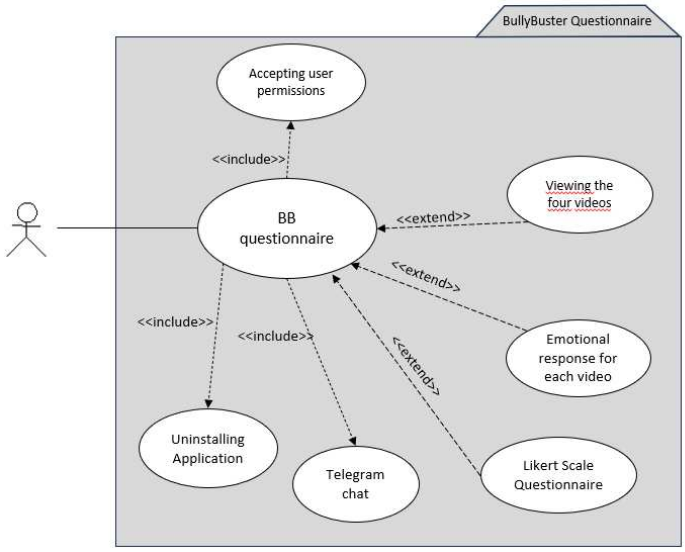


Figure 7 - BullyBuster Questionnaire UML Diagram

The leading actor in the system is the general user who uses the application. In our case, the student. Figure 7 specifies this defined semantics:

1. **Access to application functionality**
   - Description: The user accesses the application to use all its features.
2. **Accepting smartphone permissions**
   - Description: The app requires permission to access certain device features. The purpose is to detect smartphones and touch sensors. Without this permission, it is impossible to do so.
3. **Viewing Cyberbullying Videos**
   - Description: The user views four videos related to cyberbullying, each followed by an emotional question.
   - Subcases:
   - Viewing the first video
   - Response to the emotional question in the first video
   - Viewing the second video
   - Response to the emotional question of the second video
   - Viewing the third video
   - Response to the emotional question of the third video
   - Viewing the fourth video
   - Response to the emotional question of the fourth video
4. **Accessing the Questionnaire**
   - Description: The user accesses the questionnaire to answer the questions.
   - Subcases:
   - Filling in questions through buttons on a 5-Likert scale.
5. **Telegram group access.**

- Description: The user logs into the Telegram group related to the application.
- Subcases: No subcases.

6. **End Test**
   - Description: The user uninstalls the application after completing the cyberbullying test.
   - Subcases: No subcases.

3.1.2 Design

This chapter explores the key design components that helped bring this innovative application to life.

Section 3.1.2.1 BullyBuster Questionnaire, delves into the semantics underlying the "BB Questionnaire" application. This section provides a comprehensive overview of the application and its core functionality, fostering a deeper understanding of its purpose and utility. Subsection 3.1.2.1.1 Categorization Questionnaire, discusses the pivotal process of Questionnaire Categorization, illustrating how this approach is instrumental in structuring and processing the questionnaires within the application to generate the personality index, also referred to as categorization.

Continuing to Section 3.1.2.2 Android Application Technologies, this part examines the specific technologies implemented for the application's development on the Android platform, demonstrating how these technologies ensure an optimal user experience and smooth integration with the Android operating system. Lastly, Section 3.1.2.3 Web Platform Application Technologies, investigates the technologies employed in developing effective web applications for Apple devices, underscoring how the BB Questionnaire app was crafted for accessibility on various platforms, thus broadening its reach.

3.1.2.1 BullyBuster Questionnaire

The application aims to administer a questionnaire consisting of 99 questions proposed by psychological researchers.

Why was an application introduced?

Through it and the subsequent completion of the questionnaire, it was possible to extrapolate the essential characteristics that belong to the world of behavioral biometrics. The smartphone sensors, gyroscope, accelerometer, proximity sensor, atmospheric pressure, magnetometer, ambient brightness, and Step Detector were extrapolated. In particular, the accelerometer was considered in this study as one of the most advanced discriminative sensors and, for this reason, was chosen early. In addition, features inherent to touch dynamics were also extrapolated for possible future work.

The graphic layout of the existing app is simple and essential. Every choice was made to make the graphical interface simple and appealing to the end-user and, in particular, to eliminate or at least drastically reduce the possibility that the user during the activities may make mistakes, have doubts about the actions to be performed, or other problems in general.

The application contemplates five basic steps:

1. *Access to the functionality of the app;*
2. *Access to the four Cyberbullying videos. For each video, an emotional question;*
3. *Access to the Questionnaire (more than 30 screens) that is quick to fill out;*
4. *Access to Telegram group (This step is implemented only in the Android app)*
5. *End Test.*

The Questionnaire can be ideally divided into two parts. The first part shows four videos representing animated skits about bullying and cyberbullying. These animated skits are introduced to impersonate the person being tested in their observation scene. Each video shows different difficulties that a victim

might experience while and after being attacked. After each video, a question is asked to extrapolate the individual's degree of emotion after watching the video. The question asks, *"What emotion did you feel while watching video 1?"* and response buttons are assigned based on the five basic emotions of the 5-Likert scale (*sadness, happiness, etc.*) [108]. The application aims to administer a questionnaire consisting of 99 questions proposed by psychological researchers.

Specifically, the asset of the **first part** is as follows:

1. ***Video1Activity*** *("VIRTUAL ACTIONS, REAL CONSEQUENCES," https://www.youtube.com/watch?v=x2AxcllGLJg&t=4s&ab_channel=TabbyEUproject* [5]*)*
2. ***Question_Video1Activity*** *(5-point Likert scale emotional question);*
3. ***Video2Activity*** *("ANYONE CAN BE ANYONE," https://www.youtube.com/watch?v=z3N24DpD64c* [5]*)*
4. ***Question_Video2Activity*** *(5-point Likert scale emotional question);*
5. ***Video3Activity*** *("INTERNET = EVERYONE, FOREVER," https://www.youtube.com/watch?v=K31Kuc5pTXM&t=42s&ab_channel=TabbyEUproject* [5]*)*
6. ***Question_Video3Activity*** *(5-point Likert scale emotional question);*
7. ***Video4Activity*** *("VIRTUAL VENDETTA (joke or crime?)," https://www.youtube.com/watch?v=FpBVBwv6UQ4&ab_channel=TabbyEUproject* [5]*)*
8. ***Question_Video4Activity*** *(5-point Likert scale emotional question);*
   The **second part** of the questionnaire includes questions designed to label the primary classes (Personality Index), namely (*Bullying-Bully, Bullying-Victimization, Cyberbullying-Cyberbully, and Cyberbullying-Victimization* [109], [110], [111]).

The twenty-four preliminary questions are:

1. *What emotion did you feel while watching VIDEO 1: VIRTUAL ACTIONS, REAL CONSEQUENCES? **
2. *What emotion did you feel watching VIDEO 2: WHO COULD BE WHO? **
3. *What emotion did you feel watching VIDEO 3: INTERNET = EVERYONE, FOREVER? **
4. *What emotion did you feel watching VIDEO 4: VIRTUAL VENDETTA (joke or crime)? **
5. *Select your plexus from the drop-down menu. **
6. *Enter the modified Telegram Nickname at the beginning of the app. **
7. *How old are you (Number) **
8. *Are you a boy or a girl? **
9. *What school do you attend? **
10. *What grade do you attend? **
11. *In what country were you born? **
12. *Specify what part of Italy you live in* (North, Center, South, Islands) **
13. *Online Habits: Write in the box below which social networks you use (space them with a comma) **
14. *Online Habits: If you have at least one profile, how many friends do you have on social networks (give approximate total number) **
15. *Online Habits: Is one of your parents or another adult you trust among your friends?* (YES / NO) **
16. *Online Habits: Do you have at least one profile on social networking sites (Example: Facebook, WhatsApp, Instagram, Ask.com, etc...)? answers (No / Yes, on one / Yes, on more than one) **
17. *Online Habits: Do you personally know all your friends you have in your Internet profiles? **
18. *Online Habits: Do you find it difficult to stop using the Internet when you are online (e.g., stop or stop activity...)? **
19. *Online habits: Do others (e.g., friends, partners, children, parents...) tell you that you should use the Internet less? **
20. *Online Habits: Do you sleep less because of Internet use? **

21. *Online Habits: Do you look forward to the next occasion when you use the Internet? **
22. *Online Habits: Have you tried to spend less time online without succeeding? **
23. *Online Habits: Do you neglect your daily commitments (work, school, or family life...) because you prefer to go on the Internet? **
24. *Online Habits: Do you go online when you feel down in the dumps (are you sad or melancholy...)? **

The questions defining user categorization are as follows (Table 8). Subchapter 2.2.1.1 Categorization Questionnaire explained the statistical analysis employed [5]. The following scientific papers illustrate mathematical calculations that allow us to decree categorization [109], [110], [111].

Question Bully	Question Cyberbullying
BULLISM: The following questions are about your experience; answer them truthfully. Let's say that a boy/girl gets bullied when another boy/girl or a group of boys/girls:	CYBERBULLYING: The following questions are about your experience, answer truthfully. Cyberbullying is a new form of bullying that makes use of:
BULLISM: The following questions are about your experience; answer them truthfully. How many times have you been bullied?	CYBERBULLYING: The following questions are about your experience, answer truthfully. How many times have you been Subjected to cyberbullying incidents?
BULLISM: Indicate how often you have SUBJECTED to bullying a) I have been beaten	CYBERBULLYING: Indicate how often you have SUBJECTED acts of cyberbullying 1. Of receiving text messages with threats and insults
BULLISM: Indicate how often you have SUBJECTED to bullying b) I have been called nasty names	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 2. Of receiving videos/photos/pictures of assaults and violence via cell phone
BULLISM: Indicate how often you have SUBJECTED to bullying c) I have been teased	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 3. Of receiving threats and insults on the internet (websites, chat rooms, blogs, text messages, Facebook, Twitter, etc.).
BULLISM: Indicate how often you have SUBJECTED to bullying d) my classmates have ignored me	
BULLISM: Indicate how often you have SUBJECTED to bullying e) I have been threatened	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 4. Of receiving silent phone calls
BULLISM: Indicate how often you have SUFFERED bullying f) I have been excluded from activities	
BULLISM: Indicate how often you have SUBJECTED to bullying g) I have been kicked and punched	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 5. Of receiving emails with threats and insults
BULLISM: Indicate how often you have SUFFERED bullying h) They spread rumors about me	
BULLISM: Indicate how often you have SUBJECTED to bullying i) I have been teased because of the color of my skin or my culture	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 6. Of receiving videos/photos/pictures of embarrassing or intimate situations via cell phone
BULLISM: Indicate how often you have SUBJECTED to bullying j) I have had items stolen and damaged	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 7. Of receiving phone calls with threats and insults
BULLISM: Indicate how often you have SUBJECTED to bullying k) I have been teased because of a disability I have	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 8. Receiving videos/photos/pictures of assaults and violence online (e-mail, websites, YouTube, Facebook...)
BULLISM: Indicate how often you have SUBJECTED to bullying l) I have been teased because of my religion	
BULLISM: Indicate how often you have SUBJECTED to bullying m) I have been pushed and shoved	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 9. That you have received untrue rumors about yourself via telephone
BULLISM: Indicate how often you have SUFFERED bullying n) I have been called gay or lesbian	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 10. Receiving videos/photos/pictures of embarrassing or intimate situations on the internet (e-mail, websites, YouTube, Facebook...)
BULLISM: Indicate how often you have SUBJECTED to bullying o) I have been surrounded	
BULLISM: Indicate how often you have SUBJECTED to bullying p) I have been isolated	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 11. That someone has manipulated personal and private material and then reused it
BULLISM: Indicate how often you have SUFFERED bullying q) I have been grabbed	
BULLISM: Indicate how often you have SUBJECTED to bullying r) They moved in groups toward me	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 12. Of being deliberately ignored in online groups (chats, forums, Facebook groups...)
BULLISM: Indicate how often you have SUBJECTED to bullying s) They moved away quickly	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 13. That someone has taken possession of personal information or material (e.g., pictures, photos...) and then reused it
BULLISM: The following questions are about your experience, answer truthfully. Have you ever been involved in bullying other boys/girls?	CYBERBULLYING: Indicate how often you have SUBJECTED acts of cyberbullying 14. That you have received untrue rumors about yourself on the internet
BULLISM: Indicate how often you have DONE bullying a) I have hit someone	
BULLISM: Indicate how often you have DONE bullying b) I have called someone bad names	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 15. That someone has appropriated and used your password and account (e-mail, Facebook...) under a false identity
BULLISM: Indicate how often you did bullying c) I teased someone	
BULLISM: Indicate how often you did bullying d) I ignored some of my classmates	CYBERBULLYING: Indicate how often you have SUBJECTED acts of cyberbullying 16. Being excluded or left out of online groups (chats, forums, groups on Facebook, etc.)

BULLISM: Indicate how often you have DONE bullying e) I have threatened someone	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 17. That someone has appropriated and used your cell phone address book under a false identity
BULLISM: Indicate how often you have DONE bullying f) I have excluded others from activities	CYBERBULLYING: Indicate how often you have SUBJECTED to acts of cyberbullying 18. That someone blocked you in chat or on Facebook to exclude you from the group
BULLISM: Indicate how often you have DONE bullying g) I have kicked and punched someone	CYBERBULLYING: The following questions are about your experience, answer truthfully. Have you ever been involved in cyberbullying other kids/and?
BULLISM: Indicate how often you have DONE bullying h) I have spread rumors about others	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 1. Sending text messages with threats and insults
BULLISM: Indicate how often you have DONE bullying i) I have teased someone because of their skin color or culture	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 2. Sending videos/photos/pictures of assaults and violence via cell phone
BULLISM: Indicate how often you have DONE bullying j) I have stolen and damaged items	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 3. Sending threats and insults over the internet (websites, chat rooms, blogs, SMS, Facebook, Twitter...)
BULLISM: Indicate how often you have DONE bullying k) I have teased someone because of their disability	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 4. Making silent phone calls
BULLISM: Indicate how often you have DONE bullying l) I have teased someone because of their religion	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 5. Sending emails with threats and insults
BULLISM: Indicate how often you have DONE bullying m) I have pushed and shoved someone	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 6. Sending videos/photos/pictures of embarrassing or intimate situations via cell phone
BULLISM: Indicate how often you have DONE bullying n) I have called someone gay or a lesbian	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 7. Making phone calls with threats and insults
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 8. Sending videos/photos/pictures of assaults and violence online (e-mail, websites, YouTube, Facebook...)
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 9. Spreading untrue rumors about someone over the phone
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 10. Sending videos/photos/pictures of embarrassing or intimate situations over the internet (e-mail, websites, YouTube, Facebook...)
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 11. Manipulating personal and private material and then reusing it
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 12. Deliberately ignoring someone in online groups (chat rooms, forums, Facebook groups...)
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 13. Appropriating personal information or material (e.g., pictures, photos...) and then reusing it
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 14. Spreading untrue rumors about someone on the Internet
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 15. Appropriating and using someone else's password and account (e-mail, Facebook...) under a false identity
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 16. Excluding or leaving someone out of online groups (chats, forums, Facebook groups, etc.)
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 17. Appropriating and using someone's cell phone address book under a false identity.
	CYBERBULLYING: Indicate how often you have DONE acts of cyberbullying 18. Blocking someone in chat or on Facebook to exclude them from the group

Table 8 - Focal Question Bully/Cyberbully [109], [110], [111]

Underage and overage students from different schools and universities conducted the test via the Android app-web platform. The test was conducted so that nothing was anticipated to the people who then took the test. This made the questionnaire data much more truthful. As soon as the test was over, everyone was asked to comment sincerely on the experiment, especially on the relevance of the topic approved by all participants. Finally, everyone uninstalled the application.

3.1.2.1.1 Categorization Questionnaire

This survey instrument was a crucial pillar in the research, as it allowed an in-depth exploration and understanding of the dynamics of bullying, cyberbullying, and young people's perceptions of victimization. The questionnaire significantly contributed to understanding these dynamics, enabling the adoption of targeted and effective intervention strategies. This sub-chapter focused more on the statistics of categorizations or 'personality indices' easily extracted from the questionnaire related to bullying and cyberbullying. Finally, the CIUS-7 evaluation was addressed. This subchapter provided mathematical calculations that enabled the final categorization. The following tables denote the selected questions and scores for active and passive bullying and cyberbullying classes (Table 9 - Table 10 - Table 11- Table 12).

Bullying

Bullying Victimization	**Never**	**Only 1 or 2 times**	**2-3 times a month**	**1 time a week**	**Several times a week**
a) I have been beaten	0	1	2	3	4
b) I have been called bad names	0	1	2	3	4
c) I have been teased	0	1	2	3	4
d) I have been ignored by my peers	0	1	2	3	4
e) I have been threatened	0	1	2	3	4
f) I have been excluded from activities	0	1	2	3	4
g) I have been kicked and punched	0	1	2	3	4
h) They have spread rumors about me	0	1	2	3	4
i) I have been teased because of the color of my skin or because of my culture	0	1	2	3	4
j) I have had items stolen and damaged	0	1	2	3	4
k) I have been teased because of my disability	0	1	2	3	4
l) I have been teased because of my religion	0	1	2	3	4
m) I have been pushed and shoved.	0	1	2	3	4
n) I have been called gay or lesbian	0	1	2	3	4

Table 9 – Questionnaire Bullying Victimization table

Bullying_Bully	**Never**	**Only 1 or 2 times**	**2-3 times a month**	**1 time a week**	**Several times a week**
a) I have beaten someone	0	1	2	3	4
b) I have called someone bad names	0	1	2	3	4
c) I have teased someone	0	1	2	3	4
d) I have ignored some of my classmates	0	1	2	3	4
e) I have threatened someone	0	1	2	3	4
f) I have excluded others from activities	0	1	2	3	4
g) I have kicked and punched someone	0	1	2	3	4
h) I have made fun of rumors about others.	0	1	2	3	4
i) I have teased someone because of the color of their skin or because of their culture.	0	1	2	3	4
j) I have stolen and damaged items	0	1	2	3	4
k) I have teased someone because of their disability	0	1	2	3	4
l) I have teased someone because of their religion.	0	1	2	3	4
m) I have pushed and shoved someone.	0	1	2	3	4
n) I have called someone gay or a lesbian.	0	1	2	3	4

Table 10 – Questionnaire Bullying Bully table

Therefore, all the models tested have a starting configuration based on 10 items:
(a.) Physical behaviours items: a, g, j, m.
(b.) Verbal behaviours items: b, c, n
(c.) Indirect-Relational behaviours items: d, f, h.

The topics of the calculations concern the division of some behaviors into different categories (second-order factors) and the recording of these behaviors into thresholds defined based on a single item. The division of behaviors is done according to three main categories: *physical behaviors, verbal behaviors, and indirect relationship behaviors* [112].

Second-order factors:

- VB is the sum of several sub-categories of behaviors: VB_a, VB_g, VB_j, VB_m, VB_b, VB_c, VB_n, VB_d, VB_f, VB_h.
- BB is the sum of several subcategories of behaviors: BB_a, BB_g, BB_j, BB_m, BB_b, BB_c, BB_n, BB_d, BB_f, BB_h.

Subscales of behaviors:

- VB_physical represents the sum of the subscales of physical behaviors: VB_a, VB_g, VB_j, VB_m.
- BB_physical represents the sum of the subcategories of physical behaviors: BB_a, BB_g, BB_j, BB_m.
- VB_VERBAL represents the sum of the subcategories of verbal behaviors: VB_b, VB_c, VB_n.
- BB_VERBALS represents the sum of the subcategories of verbal behaviors: BB_b, BB_c, BB_n.
- VB_RELATIONS represents the sum of the subcategories of indirect relationship behaviors: VB_d, VB_f, VB_h.
- BB_RELATIONS represents the sum of the subcategories of indirect relationship behaviors: BB_d, BB_f, BB_h.

Recoding in defined thresholds:

- Recoding is done based on the value of the single item. Three values are defined: **0 normal range, 1 risk range, 2 pathology range.**
- VB_S2 if VB (Lowest thru 9=0) (Lowest thru 15=1) (16 thru Highest=2)
- BB_S2 if BB (Lowest thru 7=0) (Lowest thru 13=1) (14 thru Highest=2)

Cyberbullying

Cyberbullying_Cybervictim	Never	Only 1 or 2 times	2-3 times a month	1 time a week	Several times a week
1.Receiving sms with threats and insults	0	1	2	3	4
2.Receiving videos/photos/images of assaults and violence via cell phone	0	1	2	3	4
3.Receiving threats and insults on the internet (websites, chat rooms, blogs, instant messaging (MSN), Facebook, Twitter, Myspace...)	0	1	2	3	4
4.Receiving silent phone calls	0	1	2	3	4
5.Receiving e-mails with threats and insults	0	1	2	3	4
6.Receiving videos/photos/images of embarrassing or intimate situations via cell phone	0	1	2	3	4
7.Receiving phone calls with threats and insults	0	1	2	3	4
8.Receiving videos/photos/images of assaults and violence on the internet (e-mails, websites, YouTube, Facebook...)	0	1	2	3	4
9.Receiving untrue rumors about you via phone	0	1	2	3	4
10.Receiving videos/photos/images of embarrassing or intimate situations on the internet (e-mails, websites, YouTube, Facebook...)	0	1	2	3	4
11.That someone has manipulated personal and private material and then reused it	0	1	2	3	4
12. Of being deliberately ignored in online groups (chat, forum, Facebook groups, etc.)	0	1	2	3	4
13. That someone has taken possession of personal information or material (e.g. images, photos, etc.) and then reused them	0	1	2	3	4
14. Of having received false rumors about you on the Internet	0	1	2	3	4
15. That someone has appropriated and used your password and account (e-mail, Facebook, etc.) under a false identity	0	1	2	3	4
16. Of being excluded or left out of online groups (chat, forum, Facebook groups, etc.)	0	1	2	3	4
17. That someone has appropriated and used your mobile phone address book under a false identity	0	1	2	3	4
18. That someone has blocked you in chat or on Facebook to exclude you from the group	0	1	2	3	4

Table 11 – Questionnaire Cyberbullying_Cybervictim table

Cyberbullying_Cyberbully	Never	Only 1 or 2 times	2-3 times a month	1 time a week	Several times a week

1. Sending sms with threats and insults	0	1	2	3	4
2. Sending videos/photos/images of assaults and violence via mobile phone	0	1	2	3	4
3. Sending threats and insults on the internet (websites, chat rooms, blogs, instant messaging (MSN), Facebook, Twitter, Myspace...)	0	1	2	3	4
4. Making silent phone calls	0	1	2	3	4
5. Sending e-mails with threats and insults	0	1	2	3	4
6. Sending videos/photos/images of embarrassing or intimate situations via mobile phone.	0	1	2	3	4
7. Making phone calls with threats and insults	0	1	2	3	4
8. Sending videos/photos/images of assaults and violence on the internet (e-mail, websites, YouTube, Facebook, etc.)	0	1	2	3	4
9. Spreading untrue rumors about someone via telephone	0	1	2	3	4
10. Sending videos/photos/images of embarrassing or intimate situations on the internet (e-mail, websites, YouTube, Facebook, etc.)	0	1	2	3	4
11. Manipulating personal and private material and then reusing it	0	1	2	3	4
12. Deliberately ignoring someone in online groups (chats, forums, Facebook groups, etc.)	0	1	2	3	4
13. Stealing personal information or material (e.g. images, photos, etc.) and then reusing it	0	1	2	3	4
14. Spreading untrue rumors about someone on the internet	0	1	2	3	4
15. Stealing and using someone else's password and account under a false identity (e-mail, Facebook, etc.)	0	1	2	3	4
16. Excluding or leaving someone out of online groups (chat, forum, Facebook groups, etc.)	0	1	2	3	4
17. Taking over and using someone's phone book under a false identity.	0	1	2	3	4
18. Blocking someone in chat or on Facebook to exclude them from the group	0	1	2	3	4

Table 12 – Questionnaire Cyberbullying_Cyberbully table

- WRITTEN VERBAL CB_WV (CB1;CB3; CB5; CB7)
- VISUAL CB_VIS (CB6; CB8; CB10)
- INTERPERSONATION CB_IMP (CB11; CB13; CB15; CB17)
- EXCLUSION CB_ESC (CB12; CB16; CB18)

The calculations presented concern the subdivision of certain behaviors into different categories (second-order factors) and the recording these behaviors into thresholds defined on a single item. The division of behaviors is done according to three main categories: verbal writing *(WRITTEN VERBAL)*, visual writing *(VISUAL)*, *IMPERSONATION,* and *ESCLUSION* [110].

Second-order factors:

- CV (Variable Behaviors) is the sum of several sub-categories of behaviors: CV_1, CV_3, CV_5, CV_7, CV_6, CV_8, CV_10, CV_11, CV_13, CV_15, CV_17, CV_12, CV_16, CV_18.
- CB (Basic Behaviors) is the sum of several sub-categories of behaviors: CB_1, CB_3, CB_5, CB_7, CB_6, CB_8, CB_10, CB_11, CB_13, CB_15, CB_17, CB_12, CB_16, CB_18.

Subscales of behaviors:

- CV_WV represents the sum of verbal and written behaviors subscales: CV_1, CV_3, CV_5, CV_7.
- CB_WV represents the sum of the subcategories of verbal and written behaviors: CB_1, CB_3, CB_5, and CB_7.
- CV_VIS represents the sum of the sub-categories of visual behaviors: CV_6, CV_8, CV_10.
- CB_VIS represents the sum of the subcategories of visual behaviors: CB_6, CB_8, CB_10.
- CV_IMP represents the sum of the sub-categories of impersonation behaviors: CV_11, CV_13, CV_15, CV_17.
- CB_IMP represents the sum of the subcategories of impersonation behaviors: CB_11, CB_13, CB_15, CB_17.
- CV_ESC represents the sum of the subcategories of exclusion behaviors: CV_12, CV_16, CV_18.
- CB_ESC represents the sum of subcategories of exclusion behaviors: CB_12, CB_16, CB_18.

Recoding in defined thresholds:

- Normality range (0): for the lowest single item values (CV and CB).
- Risk range (1): for the intermediate values of the single item (CV and CB).
- Pathology range (2): for the highest values of the single item (CV and CB).

Recoding into defined thresholds:

Recoding is done based on the value of the single item. Three values are defined: **0 Range Of Normal, 1 Range Of Risk, and 2 Range Of Pathology.**

- *CV_S2 = CV (Lowest thru 9= 0) (Lowest thru 15 = 1) (16 thru Highest=2)*
- *CB_S2 = CB (Lowest thru 7= 0) (Lowest thru 13 = 1) (14 thru Highest=2)*

CIUS-7

The **CIUS-7** questionnaire is an assessment tool used to measure levels of *"compulsive Internet use" (CIU)*, commonly known as Internet addiction. CIU refers to an inability to control Internet use, which can lead to several negative consequences in a person's life, such as mental health problems, reduced social interactions, poor productivity, and more.

The **CIUS-7** is an abbreviated version of the Compulsive Internet Use Scale (CIUS), initially developed by Meerkerk, Van Den Eijnden, and Garretsen in 2009. This abbreviated version contains seven questions covering various aspects of compulsive Internet use. Individuals participating in this questionnaire must answer the questions by assessing their behavior and feelings regarding Internet use.

(1) Do you find it difficult to stop using the Internet when you are online (e.g., stop or discontinue activity...)?
(3) Do others (e.g., partner, children, parents...) tell you that you should use the Internet less?
(5) Do you sleep less because of your Internet use?
(7) Do you look forward to the next occasion when you use the Internet?
(9) Have you tried to spend less time on the Internet without succeeding?
(11) Do you neglect your daily commitments (work, school, or family life...) because you prefer to go on the Internet?
(12) Do you go on the Internet when you feel down in the dumps (you are sad, melancholy...)?

Scoring: For each item, participants are asked to select a score on the following scale:
(0) Never
(1) Rarely
(2) Sometimes
(3) Often
(4) Very often

Based on the participants' responses, scores were assigned to each item according to the above scale.

- **CIUS 7**: The variable "CIUS 7" represents the sum of the scores of items 1, 3, 5, 7, 9, 11 and 12.
- Second-order factors:
- **CIUS_TOTAL7**: The variable "**CIUS_TOTAL7**" is the sum of the scores obtained from the individual items of CIUS 7. Thus, it is the sum of the scores obtained from the seven items mentioned above.
- **CIUS_TOTAL5**: The variable "**CIUS_TOTAL5**" results from the sum of the scores obtained from items 1, 3, 5, 11, and 12. In other words, it is the sum of the scores obtained from the five mentioned items.

The ***"CIUS7_S2"*** is a second-order variable representing a sum or average of some scores. The description only indicates that the possible values for this variable range from 0 (if **CIUS_TOTAL7** is less than or equal to 14) to 1 (if **CIUS_TOTAL7** is greater than 15).

3.1.2.2 Android Application Technologies

The Android application was created using the Android Studio IDE, Java as the programming language, and PHP for the link to the Aruba server. Android Studio is a popular *Integrated Development Environment (IDE)* used primarily for Android application development. Google developed it and provided many tools and features to simplify the Android app development process.

The main features of Android Studio include:

- *Advanced code editor*: Android Studio offers an intelligent code editor that supports code autocompletion, fast navigation, and error detection.
- *Android Emulator*: The IDE includes an Android emulator that allows developers to run and test their apps on different virtual devices.
- *Gradle Build System*: Android Studio uses Gradle as its build system, simplifying the configuration of dependencies and managing app builds.
- *Debugging Tools*: Android Studio provides advanced tools for debugging apps, such as a debugger, log viewing, and performance monitoring.
- *Layout Editor*: A visual layout editor is provided that allows developers to create user interfaces for their apps by dragging and dropping elements.
- *Testing Tools*: Android Studio supports unit, integration, and UI testing to ensure app quality.
- *Support for Kotlin and Java*: Android Studio supports both the Kotlin and Java programming languages, allowing developers to choose their preferred language for app development.
- *Integration with Android SDK*: The IDE is tightly integrated with the Android Software Development Kit (SDK) and provides access to all APIs and resources needed for Android app development.
- *Publishing support*: Android Studio simplifies the process of publishing apps to the Google Play Store by providing tools for app signing and generating distribution packages.
- *Android Studio* is free and ideal for many Android app developers because of its robustness, active support from Google, and large developer community.

The Android app was developed using the Android framework, which provides a development environment for creating Android applications. Activities are the core components of the Android application that manage the user interface and user interaction with the application. Intents and Intent Filters enable communication between different components of the application or between different Android applications. Broadcast Intent Receivers allow the application to receive and respond to broadcast messages sent by the system or other applications. Content Providers provide an interface for securely sharing data between different Android applications. Services are components that perform operations in the background without a visible user interface. Permissions allow the application to request user permission to access certain features or resources on the device. Sensors allow the application to detect and respond to environmental changes, such as orientation, motion, light, and other sensor data. Classes are basic units of an Android program used to define objects and behaviors. PHP is a server-side scripting language widely used to develop web applications and interact with databases. In the Android application, these technologies have been used synergistically to create an engaging and functional user experience, enabling the application to perform various tasks, communicate with other applications, and use the device's sensors and data effectively. Let's start by dissecting each component:

Android: Android is the most widely used operating system in the world today: it is attested that 62.94% of mobile devices, including car stereos, smartwatches, televisions, and IoT products, use Android as their operating system or an Android-based operating system, each of which has a dedicated graphical user interface such that the user experience is highly performant. Android is an operating system run by Google LCC based on a Linux kernel. Among the features of Android, particular attention deserves its being Open Source, that is, giving anyone the ability to access the system's source code, except for some proprietary parts, thus exclusive access. The source code is in a Git repository managed by Google, AOSP (Android Open Source Project). The spread of Android and open source has had two effects: the first is related to the emergence of custom ROMs, i.e., a third-party modified stock ROM based on AOSP (Android Open Source Project), which is smoother and lighter; the second is related to the security of the system itself. Android is one of the favorite victims of malicious attackers, who look for vulnerabilities to attack the system. Its open source exposes it to more risks and faster resolution of identified flaws. The Android architecture is a layered architecture whose functionality starts from the lower layers, which must provide services to the higher layers, offering a higher degree of abstraction. The following is the structure (Figure 8):

- *Application;*
- *Application Framework;*
- *Android Runtime;*
- *Libraries;*
- *Linux Kernel.*

Figure 8 - Android system architecture

The lowest layer present in the architecture is the Linux Kernel layer, as shown in the figure. The choice of a Linux kernel directly results from the need for an OS that provides several important features, most notably security, memory management, and process management. The Linux Kernel is also considered one of the most portable: it is nothing more than an operating system that provides low-level tools for virtualizing the underlying hardware by defining various drivers. The Libraries layer, positioned above the Linux Kernel, contains a set of native libraries representing a microprocessor's core, the processing core.
These include:

- *Surface Manager*: is responsible for managing and coordinating views, i.e., what a graphical user interface is composed of, commonly known as windows, and placing them in a buffer with the aim of not having views overlapping uncoordinatedly on the display;
- *Media Framework*: responsible for managing the different CODECs for the various audio and video capture and playback formats;
- *SSL* is a library for handling Secure Socket Layers.

The second layer in the architecture is the Application Framework, which consists of a set of APIs or components for executing certain functionality needed for applications.

These include:

- Activity Manager has served as the fundamental tool through which the user has interacted with the application. The responsibility of this component has been to organize the various screens of an application in a stack, according to the order in which they have been displayed on the screens of different devices;
- Package Manager: is responsible for managing the application lifecycle in devices;
- View System: is responsible for managing the rendering of components and associated events;
- Notification Manager: provides a set of tools that the application can use to send a particular notification to the device, which must present itself to the user with known mechanisms such as vibration, an LED, or an icon.

The Android environment allows, in the build phase, to compile Java code into bytecode and then convert it into another type of intermediate code, dex, which consists of the code executed by the new VMs, capable of taking full advantage of the features of the various hardware. The main components that form the basis of the architecture of an Android application are:

- *Activity;*
- *Intent and Intent Filter;*
- *Broadcast Intent Receiver;*
- *Content Provider;*
- *Services.*

Activity

The Activity is one of the central elements of any Android application and is responsible for managing the interface and user actions. A typical application usually consists of several screens that exchange information. For this reason, Android organizes activities according to a "stack" structure where the activity at the top is always the one that is active at that particular moment. The display of a new screen corresponds to the start of a new Activity: the latter is brought to the top of the stack, pausing the previous ones. In a logic of resource optimization, it is necessary to provide that an Activity that is not displayed, thus not at the top of the stack, can be deleted from the system and possibly restored later. Using any app, instances of the various Activities navigate through different states during their lifecycle. The Activity class provides several callback methods that allow it to know the state of the Activity itself and thus determine whether or not it has changed. In these callback methods, actions are defined to be performed, given the state of the Activity, thus preventing any crashing of an app or consumption of valuable resources.

There are six callback methods, and they are:

- *onCreate();*
- *onStart();*
- *onResume();*
- *onPause();*
- *onStop();*
- *onDestroy().*

Depicted below is the life cycle of an Activity (Figure 9).

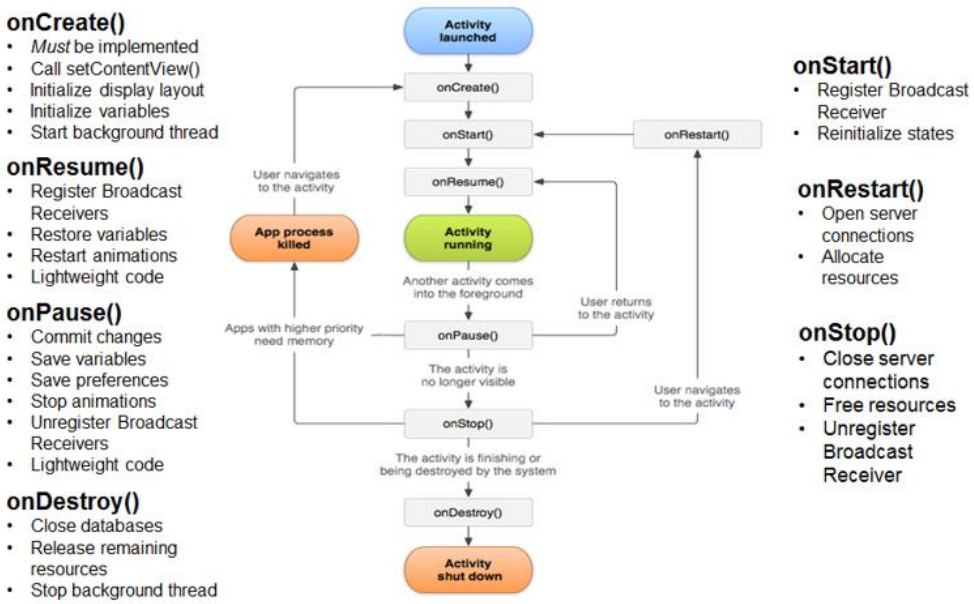


Figure 9 - Life cycle of an Activity

When an Activity is requested to execute, the system creates an instance of the corresponding class. The first callback method executed is onCreate(), in which all those operations are executed once for the entire duration of the Activity. This method receives the saveInstanceState parameter, a Bundle object containing the Activity's previously saved state. If the Activity has never existed before, the value of the Bundle object is null. Upon completion of the execution of the onCreate() method, the Activity enters the STARTED state upon which the system calls the following two callback methods, namely, onStart() and onResume().

The onStart() method provides the layout display of the Activity by preparing it to be interactive. As in the case of the previous method, this method ends quickly by calling the next one.
The actual possibility of interaction between the user and Activity is integrated with the onResume() method, against which the Activity is in the RUNNING state; if an event occurs that leads to the interruption of the Activity in the foreground, it enters the paused state via the onPause() callback method. By calling the onResume() method again, the Activity can be restored to its previous state.
To exit the app by pressing the back button, the system handles executing three callback methods, symmetrical to the previous three. The first method executed is the onPause() method, indicating that the Activity is no longer in the foreground and, therefore, is no longer at the top of the stack; this is the symmetric method corresponding to onResume(). The onStop() method, invoked next, is the symmetric method to onStart(): when the app executes this method, it receives the ON STOP event, prompting it to release resources that are no longer needed. The last method, symmetrical to onCreate(), is onDestroy(): the Activity receives the ON DESTROY event, and any remaining resources not released in the onStop() method are released.

Intent and Intent Filter
The Intent is a simple request made to the system to use a resource or a component that can handle it. It is a passive data structure containing an abstract description of an action to be performed. There are three primary use cases:

- Starting an Activity;
- Starting a Service;
- Sending Broadcast Messages;
  Intents can be classified as:
- Explicit.

- Implicit;

The former specified which application satisfied the Intent by providing the package name of the target app or a class name of the entire component. They were typically used to start a component in an app because the class name of the task or service to be started was known. Seconds did not define a specific component but declared a general action to be performed, allowing a component in another app to handle it. Here, the Android system performed an Intent Resolution operation, which consisted of checking, via the Intent Filters defined in the AndroidManifest.xml files, which apps were capable of handling that Intent. If compatibility with various Intent Filters was found, Android showed the user a dialog box, allowing them to choose which app to run the Intent with.

The leading information related to Intents are:

- *Action*: the generic action to be performed, specified with the action constants of the Intent class, e.g., ACTION VIEW, ACTION MAIN;
- *Data*: the data to operate on expressed as a URI object.
- Secondary information, on the other hand, is:
- *Category*: provides additional information about the action to be performed. For example, CATEGORY LAUNCHER means that it should appear in the Launcher as a top-level application, while CATEGORY ALTERNATIVE means that it should be included in a list of alternative actions that the user can perform on a data item;
- *Type*: specifies an explicit type (a MIME type) of the data, which is usually inferred from the data itself;
- *Component*: specifies an explicit name of a component class to be used;
- *Extra*: represents a packet of additional information.

Broadcast Intent Receiver
In Android, the handling of external events (the arrival of a phone call, a message, etc.) can be accomplished through the definition of BroadcastReceivers, which allows the implementation of specific handlers of the events to which they are registered. Broadcast Intent Receivers are then able to activate themselves following the launch of a particular Intent, which is called a broadcast Intent, they do not have a *UI (User Interface)* and are associated with a particular set of IntentFilters corresponding to as many broadcast Intents. Their task is to activate at particular events, gathering information to be used later to execute more complex operations. Registration at a particular event can be done through the configuration file (the AndroidManifest.xml) or the appropriate API.

Content Providers
Content Providers represent an official system mechanism for sharing data. They can manage access to stored data, share it with other apps, encapsulate it, and provide mechanisms for defining data security. Content providers make it possible to read and modify data in a given source by calling a resource's unique address, also called a URI. The four operations that can be performed on databases are CRUD operations: Create, Read, Update, and Delete.

Content providers are used in a variety of areas:
- To share general-purpose databases found in the operating system (think Contacts, User Dictionary, or Calendar);
- To make it possible for multiple applications to share the same data, which often happens when multiple apps come from the same manufacturer;
- As an internal architectural element of an articulated app in which multiple components access the same data.

The implementation of a content provider has many advantages. These can be defined and used even if no data sharing is involved, as they can improve the system's abstraction and thus allow changes to be made to the app's data storage implementation without affecting other existing apps that rely on data access. Content providers offer granular control over data access permissions, including, for example, allowing general permission to access data from other applications or configuring different permissions for reading and writing data.

Service

The service component was introduced to satisfy the need for a mechanism to "keep alive" as many objects as possible, also defined as threads, without risking their being deleted for the sake of a resource optimization policy. The Service does not have a graphical interface; its task is to remain running in the background independent of what is possibly displayed on the display, thus independent of the Activity at the top of the stack, and thus independent of what the user, at that moment, is interacting with, and to communicate with applications that intend to do so. An Activity, or possibly another component, can connect to a particular service and, if it has the right to do so, start or stop it. The Activity intending to use the service's functionality can perform a bind and access the interface (API).

Services can be classified based on type, as shown below (Figure 10):

- **Foreground**: A service in the foreground performs some apparent operations to the user. For example, an audio app uses a Service in the foreground to play an audio track. Services in the foreground must display a notification. Services in the foreground continue to execute even when the user does not interact with the app;
- **Background**: A service in the background performs an operation that the user does not directly notice. For example, an app uses a Service to compact its storage space; beware, however: if the app is targeting API level 26 or higher, the system imposes restrictions on the execution of background Services when the app itself is not in the foreground. In most cases like this, the app should use a scheduled process instead.
- **Bound**: A service is inbound when an application component associates with it by calling bindService(). A Service inbound provides a client-server interface that allows components to interact with the Service, send requests, receive results, and even do so through processes with interprocess communication (IPC). Service inbound runs only as long as another application component is associated. Multiple components can connect to the Service simultaneously, but when they all disband, the Service is destroyed.

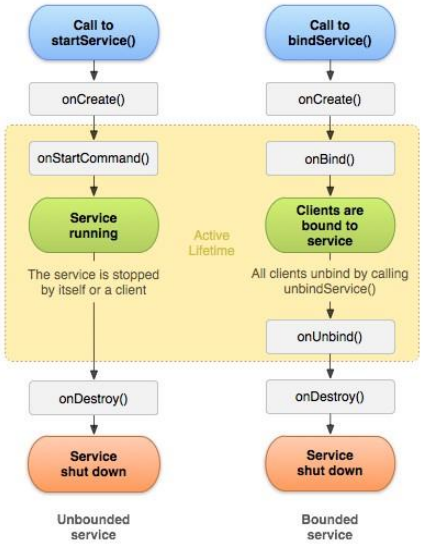


Figure 10 - Life cycle of a Services

Permissions

An app that intends to use one of these resources must necessarily indicate this within its AndroidManifest.xml file. In versions before 6.0, the operating system used this information during the installation phase by showing a report to the user asking for bulk acceptance of all permissions or massive rejection failing the app installation procedure. From version 6.0 onward, permissions management has changed: The user has not had to accept all permissions at once but has evaluated each permission individually upon approaching a feature that requires it. This has required the app to handle cases where the user does not accept a particular permission. The platform API has provided a set of predefined permissions, yet the same mechanism has enabled the creation of custom permissions. Custom permissions have been defined within the AndroidManifest.xml file using the appropriate element.

This element provides several attributes, some of which are mandatory, such as the name of the permission and the protection level, which can be of four different types:

- *Normal*: This is the lowest level of protection and is granted automatically by the system during the app installation;
- *Dangerous*: this level of protection covers those permissions that allow access to sensitive user data. The user must explicitly accept these types of permissions;
- *Signature*: these are permissions that can only be granted to apps signed with the same certificate. In case the certificate is the same, they are granted automatically by the system;
- *Signature Or System*: permission related to permissions granted to system APIs or otherwise to apps signed with the same certificate.
- Other attributes that can be used within the ¡permission/¿ element are the non-mandatory attribute permissionGroup that indicates to which permission group the specific permission belongs, and the non-mandatory attributes label and description that allow giving a name and description to the permission, respectively. This information is then shown to the user when they are asked for permission.

To verify that the user has granted the requested permission, use the checkSelfPermission method of the ContextCompat class. The values that this method can return are:

- *PackageManager*: *PERMISSION GRANTED*;
- *PackageManager*: *PERMISSION DENIED*.

In case the user had granted permission, the execution of the various operations proceeded; otherwise, the user was informed of the reasons for the request so that he could decide whether to grant it or not. Android provided the method shouldShowRequestPermissionRationale(), which allowed the determination of whether showing the message to the user was appropriate.
This is to avoid the display of a large number of messages. Next, permission is requested from the user via the requestPermission method of the ActivityCompat class, and the result is notified via the Callback *onRequestPermissionResult*() method. In the element, some attributes related to permissions can be used, such as the Permission attribute, which defines whether that specific Activity can only be used by apps with the given permission. On the other hand, services are deprived of them by default; nevertheless, they can be inserted via the Exported attribute. As with Activities, a Permission attribute for Services governs service access. The *BroadcastReceiver* is always defined as public even though permissions allow you to decide which app can send Intents. Failure to match the required permissions with those held does not lead to raising an exception but simply to the non-receipt of the sent Intent. Two attributes are made available in the ContentProvider, readPermission and writePermission, designed to indicate what permissions are required for applications to read and write to the repository.

Sensors

Most Android devices predispose a set of sensors, which provide raw data that enjoy high accuracy and precision, suitable for measuring motion, orientation, or even environmental conditions. The Android platform supports three broad categories of sensors:

- ***Motion sensors***: measure acceleration and rotational forces along three axes. These include accelerometers, gravity sensors, gyroscopes, and rotational vector sensors;
- ***Environmental sensors***: measure environmental parameters, such as temperature, air pressure, and illumination. These include barometers, photometers, and thermometers;
- ***Position sensors***: measure the physical position of the related device. These include orientation sensors and magnetometers.
  The sensor framework through which access is managed offers various classes and interfaces that allow for a variety of tasks, such as, for example, determining the sensors available in the device. Sensors are divided into two types: hardware and software. The former consists of physical components built into the device, and the latter consists of virtual sensors that obtain data by combining hardware-type sensors. The Android. The hardware package includes the following classes and interfaces:
- *SensorManager*: class used to create an instance of the sensor service;
- *Sensor*: class used to create an instance of a specific sensor;
- *SensorEvent*: class used to create an object that provides information about a sensor event containing data and timestamp;
- *SensorEventListener*: an interface that creates two callback methods that receive notifications when sensor values or accuracy changes.

The Android classes that played a crucial role in achieving the design goals are described below.

AccessibilityServices

Accessibility services should only be used to assist disabled users using Android devices and apps. They run constantly in the background, resist system reboots, and are invoked by the system when AccessibilityEvents are triggered. The latter is a class that represents accessibility events sent by the system the moment a change occurs in the user interface, some examples being actions such as clicking on a button, long-click, or scroll action relative to a screen, or even changes in typed text. Each type of event is associated with a constant and a subset of properties exposed by this class. The primary purpose of an accessibility event is to communicate changes in the user interface to an accessibility service. The service can then inspect, if necessary, the user interface by examining the View hierarchy, represented by an AccessibilityNodeInfo tree, which can be used to explore the View content. Access to the View's content must be explicitly requested. Otherwise, the event did not contain references to its source.

AsyncTask

AsyncTask is designed to be a support class for Thread and Handler and is not a generic threading framework. An asynchronous task is defined by a computation that runs on a background thread whose result is published to the UI Thread. Three generic types called define an asynchronous activity:

- *Params;*
- *Progress;*
- *Result; and four steps, called:*
- *onPreExecute;*
- *doInBackground;*
- *onProgressUpdate;*
- *onPostExecute.*

DeviceAdminReceiver

DeviceAdminReceiver is the class appropriate for implementing a device administration component.

DevicePolicyManager
DevicePolicyManager is a public interface for managing the policies applied to the device.

LocationManager
LocationManager is the class that provides access to the system's location services. These services allow applications to obtain periodic updates regarding the device's location.

NotificationManager
NotificationManager is the class introduced to inform users of events occurring in the background. Notifications can take several forms:

- Persistent icon in the status bar;
- Turning on or flashing of the LEDs on the device;
- Activation of the backlight followed by a sound or vibration playback.

PHP
PHP is a programming language introduced in 1995 and used primarily for creating dynamic Web pages. The acronym PHP stands for Hypertext Preprocessor, although in the early days of the language, it stood for Personal Home Pages. Because it is simple and flexible, it is suitable for both the development of simple applications and more complex systems. Among the features that can be implemented with the said language are:

- *file upload;*
- *registration and login forms;*
- *file management.*

The following are the main features:

- *It is an interpreted scripting language;*
- *It is an HTML-embedded language in that PHP code can be inserted within a page containing HTML code;*
- *It has weak typing;*
- *It integrates seamlessly with popular DBMSs such as MySQL and PostgreSQL;*
- *It allows interfacing with numerous libraries: cURL, GD, OpenSSL;*
- *It is supported by the most popular web servers, such as Apache and nginx.`*

PHP is the fastest of the scripting languages and contains numerous features that enable the writing of robust and secure code. It has been released under a specific open-source license, the PHP license, which includes some restrictions on the name "PHP."

3.1.2.3 Web Platform Application Technologies

The Web Platform application has been created using JavaScript and Python languages to add functionality and for the user interface, HTML, CSS, and PHP. The entire application is programmed directly from the PythonAnywhere platform.

PythonAnywhere
PythonAnywhere is a web hosting and development platform that allows users to run, write, and host Python applications online. It is a cloud-based service that provides an integrated development environment for Python, enabling developers to write Python code directly in the browser and execute scripts or applications without configuring a server or local development environment.

Some key features of *PythonAnywhere* include:

- *Code editor*: It provides an integrated Python code editor that supports error detection, syntax highlighting, and other typical development environment features.
- *Interactive console*: Users can run Python commands directly from the interactive console.
- *Web hosting*: PythonAnywhere offers the ability to host Python web applications easily and quickly.
- *Multiple Python versions*: It supports various Python versions, allowing users to run their code on a specific Python version.
- *Preinstalled packages*: PythonAnywhere includes numerous preinstalled Python packages and libraries, simplifying dependency management.
- *Task scheduling*: It allows the execution of Python scripts at specific times.
- *File system access*: Users can upload, download, and manage files directly through the interface.
- PythonAnywhere is particularly useful for those who want to develop, test, or run Python applications without configuring a complex local environment. It is also ideal for students, as it provides an online learning environment for Python without needing to install it on their computers. However, it should be noted that some features may be limited in the free version of the service, and a paid plan with additional features and higher resources is available.

HTML (HyperText Markup Language):

HTML is the markup language used to create the structure and content of a web page. You can define elements and their arrangement on the page with HTML, such as text, images, hyperlinks, etc. Elements are marked with tags (e.g., <p> for a paragraph, <img> for an image) that specify the type of content they contain.

CSS (Cascading Style Sheets):

CSS is a language used to describe the appearance and layout of a web page created in HTML. It allows you to define the presentation of different HTML elements, such as color, size, position, font type, background, and more. The goal of CSS is to separate the content structure (defined in HTML) from its presentation, enabling greater flexibility and ease of website maintenance.

JavaScript:

JavaScript is a programming language that adds interactivity and dynamic functionality to web pages. It is a client-side scripting language, meaning that JavaScript code is executed directly in the visitor's browser and can interact with the Document Object Model (DOM) to modify page content, respond to user events, and communicate with a server to retrieve or send data without reloading the entire page. In summary, HTML defines the structure and content of the page, CSS manages the appearance and layout, while JavaScript adds interactivity and dynamic behavior to the page. These three languages work together to create rich and engaging web experiences.

PHP:

PHP is a programming language introduced in 1995, mainly used for creating dynamic web pages. The acronym PHP originally stood for Hypertext Preprocessor, although in the early days of the language, it referred to Personal Home Pages. Due to its simplicity and flexibility, PHP is suitable for developing simple applications and more complex systems.

Some functionalities implemented with PHP include file uploads, registration and login forms, and file management. The main features of PHP are:

- It is an interpreted scripting language.
- It is an HTML-embedded language, as you can insert PHP code within a page containing HTML code.
- It has weak typing.

- It integrates seamlessly with popular database management systems such as MySQL and PostgreSQL.
- It allows interfacing with numerous libraries, including cURL, GD, and OpenSSL.

PHP is supported by widely used web servers such as Apache and Nginx. PHP is the fastest among scripting languages and contains numerous features that enable the writing of robust and secure code. It has been released under a specific open-source license called the PHP license, which includes some restrictions on the name *"PHP."*

Python:
Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It is used for application development and as a scripting language. Its simple and easy-to-learn syntax emphasizes readability, reducing program maintenance costs. Python supports modules and packages, encouraging program modularity and code reuse. The edit-test-debug cycle was of great importance due to its incredible speed. Debugging Python programs proved straightforward: a bug or incorrect input never caused a segmentation fault but instead resulted in an exception. The interpreter prints a stack trace when the program does not catch the exception. A source-level debugger allows inspection of local and global variables, evaluating arbitrary expressions, setting breakpoints, stepping through code one line at a time, and more. The debugger is written in Python itself. Python is used not only for traditional development but is also the primary language for developing systems for emerging data science sectors, including data analysis, artificial intelligence, and machine learning.

3.2 BBQuestionnaire Implementation

This subchapter delves into the pivotal phase of implementing the proposed system within the project framework. Implementation represents the transition where theoretical concepts and ideas from the previous chapter are materialized into a fully functional and operational product. Here, the implementations of the Android smartphone application and the web-based questionnaire platform are comprehensively examined.

In Section 3.2.1 Android Smartphone Application Implementation, the focus is on the creation of the Android application, designed to deliver a seamless and intuitive user experience. This section explores the technologies employed, the application's architecture, and the critical steps involved in developing an efficient, high-performance mobile platform. Key topics include data management, user interface design, and the primary functionalities that constitute the core of the application.

Section 3.2.2 Questionnaire Web Platform Implementation, shifts attention to the web-based platform for administering the questionnaire. This section outlines the architecture of the platform, as well as the technologies selected for its development. It addresses aspects of data management, user interface considerations, and the essential functionality integral to the platform's operation.

Both components were implemented through rigorous processes of development, testing, and optimization to ensure readiness for end-user deployment. The in-depth analysis of these two implementations underscores the system's effectiveness and its capacity to offer practical and functional solutions for the users. The Android application and web platform were each developed with an emphasis on scalability, usability, and flexibility to meet the diverse needs of a wide range of users and operational contexts.

3.2.1 Android Smartphone Application Implementation

The versioning of these applications was designed and implemented during the six-month period with Digital Innovation srl (Agreement from 01/05/2023 to 31/10/2023).

Subchapter 3.2.1 Android Smartphone Application Implementation, dealt with the implementation of the Android smartphone application, BB Questionnaire. This section explored the various critical aspects related to the application's implementation, focusing on three key points: the different versions of the application, the libraries used, and the development process of the application itself.

In the first section, the different versions of the application developed were discussed. Each version included new features, improvements, bug fixes, or optimizations. The changes between versions were examined to provide a better understanding of the application's evolution and how it was adapted to users' needs. The second section focused on the libraries used during the application's development. Libraries are essential tools for simplifying and improving the development process by providing predefined functionality and reducing the amount of manually written code. The libraries adopted, their key functionalities, and their role in the application's implementation were explained. Finally, in the third section, the development process of the BB Questionnaire application was explored. This section provided an overview of the challenges faced, decisions made, and best practices followed during the development.

3.2.1.1 Application Versions

The app used to extrapolate Touch Interactions and Interaction Dynamics has undergone multiple modification stages, from version *"BullyBuster Questionnaire V. 1.0"* to the latest one named *"BullyBuster Questionnaire V. 4.0."* The latest version *("BullyBuster Questionnaire V. 4.0")* results from

multiple project meetings that consequently decreed the final screens of the app. With it, it was possible to run tests with users.

First Phase

An application was created to perform "Parental control" activities. Parental control is the system that allows parents or guardians to monitor a minor's access to certain activities and to set the time of use of Social Networks. This first app aims to extract media components from Social Network apps (Table 13).

Phase	Version	Features Version
First Phase	BullyBuster Questionnaire V. 1.0	Keylogger (Click, Focus, Text); SocialMedia groups screen recording with timer; Display of accelerometer, contacts, GPS, and app list (unsaved).
Two-Phase	BullyBuster Questionnaire V. 2.0	Keylogger (Click, Focus, Text); Screen recording Group-Wapp test; Display of accelerometer, contacts, GPS, and Apps (saved); Mobile log saving Android version, cellular model.
Third Phase	BullyBuster Questionnaire V. 3.0	Keylogger (Click, Focus, Text); New app design structure with video, questionnaire, and telegram chat; Presence of a single button in the main accessibility screen, "Go to Accessibility"; Draft privacy policy; Preliminary questionnaire questions; Data capture even outside the app; Mobile log saving Android version, cellular model.
Four Phase	BullyBuster Questionnaire V. 4.0	Keylogger (Click, Focus, Text); Implementation Total Questionnaire; Presence of project logos; Data acquisition only within the app; Permissions with controls; Full Privacy Policy; Insert Telegram nickname; Marquee questions in the questionnaire inherent to the plexuses; Informational pop-ups; Addition of an uninstall screen for the app; Presence of emotional questions at the end of each video; Display of accelerometer, contacts, GPS, and apps (saved); Mobile log saving Android version, cellular model.

Table 13 – Versioning BullyBuster Android Application (Digital Innovation srl version)

This version of the application has the following features:

- Request access permission to GPS location (saving it on the Aruba server);
- Requesting access permission to the contacts in the smartphone address book (without saving them on the Aruba server);
- Request access permission to the applications installed on the device (without saving them on the Aruba server);
- Use a service that implements a KeyLogger for logging the touches made on the screen and the text typed and save them on the Aruba server;
- Effmake a recording of the device screen when the child opens specific applications (messaging apps, social networks) and send the video to a Firebase DB;
- Extract the sensors on the phone by recording to .txt files generated by the Keylogger service, saving them later to the Aruba server.

When the app starts, the first item on the screen is a Dialog containing a welcome message and instructions to follow to activate the permissions that the app needs. After the Dialog screen, the app presented a screen with four different Card objects (four graphical buttons):

- **Go to Accessibility,** which allows access to the accessibility section of the device to allow the permissions that the app requires by enabling Keylogger and smartphone sensor extraction;

- **ActivateManager** enables the DPM option, which stands for Device Policy Manager, an Android class that makes the app in question a system app, thus making it impossible to uninstall.
- **DeactiveManager,** which allows you to disable and make it possible to uninstall the app;
- **Info Device** allows you to show some information about the device. After this graphical button is clicked, a call is made to an Intent that displays some extra information about the supervised device (Latitude, Longitude, list of installed applications, list of phone contacts with phone number and contact name). In this version, the information displayed in the cards is only displayed and not saved on any server.

Regarding permissions, a Dialog displayed asked for consent to allow the app to collect information, such as the list of applications installed on the device and the list of phone numbers saved in the address book. Once installed, it was possible to decide whether to make the app uninstallable by clicking on the *"Activate Manager"* card or the *"Go to Accessibility"* card, as instructed in the Welcome Dialog, by accessing the Accessibility section. Within this section, the keylogger service needed to be located and activated.

The Keylogger service extracted three fundamental constants:
- **Type View Text Changed**: Represents the text change event in an EditText; most commonly, it performs keylogger functionality. Each time the cell phone's writing keyboard is activated, every single typed letter is saved;
- **Type View Focused**: Represented the View display event. Whenever an email read notification was triggered, the text of a web page was displayed, and the entire content was saved;
- **Type View Clicked**: represents the click event on any clickable component within the View, e.g., Button, ListView, and RecyclerView.

By enabling the previous permissions, it is possible to extract the sensors of the cellular device, these are the basis of a comfortable user experience used to collect habits and movements to draw up a constantly updated profile of it suitable for its protection. BullyBuster Questionnaire v.0.1 extracts the following sensors:
- *Accelerometer;*
- *Gyroscope;*
- *Magnetometer;*
- *Proximity sensor;*
- *Light sensor;*
- *GPS.*

Each value the individual sensors found was saved in an appropriate text file allocated on the Aruba server. The writing of the values occurred each time they changed, accurately tracking the user's progress throughout the day. If a device did not have one of the listed sensors, such data was not retrieved and, therefore, not monitored. Once BullyBuster Questionnaire V. 1.0 permissions and personal data access permissions had been enabled, it was possible to close the application (resize it to Background). In this way, this service continued its execution even after shutting down or restarting the device. From the moment of activation, the Keylogger intercepted any touch on the device screen from when the smartphone was unlocked, including text typing. All the information collected by the Keylogger was recorded in a text file and sent to the Aruba server to enable later content analysis.

BullyBuster Questionnaire V. 1.0 monitored the following applications:
- WhatsApp;
- Telegram;
- Instagram;

- Facebook;
- Twitter.

When one of these individual applications is opened, the device screen is recorded from the background. Due to restrictions related to the Android system architecture and its security protocols, which are constantly being updated and improved, such an action is made impossible unless there is in the foreground (at the top of the stack) a clear and direct reference to the application or service being processed. By doing so, Android makes it possible to avoid the propagation of malicious apps across devices, as the user is constantly aware and conscious of what is happening on their device. A notification was necessary to prevent the expressed problem, visible to the user when clicking on one of the previously listed apps.

Thus, the function introduces the NotificationManager class, which is responsible for starting the notification service, creating a communication channel for notifications related to the service, and generating the notification itself. Such a notification is intended to trick the user; it contains a simple message informing that the clicked app is open and the video recording starts. It is configured not to be deleted unless the user interacts directly with it through a click. Screen recording is handled initially by the *MediaProjectionManager* class to manage the session and later by the *MediaRecorder* class to manage the configurations of the video itself. On first use, after interacting with the notification, the user is asked for permission to record everything that takes place on the device. By checking the "*Don't ask again*" box, this dialog no longer displays.

This recording varies in duration depending on the time of day; this duration was defined after conducting a study on app usage by users at different times of the day. Below is the duration of the video, depending on the time of day.
- *00:00 - 06:59 duration 10 seconds;*
- *07:00 - 08:59 duration 25 seconds;*
- *09:00 - 12:59 duration 10 seconds;*
- *13:00 - 14:59 duration 35 seconds;*
- *15:00 - 18:59 duration 15 seconds;*
- *19:00 - 20:59 duration 25 seconds;*
- *21:00 - 21:59 duration 10 seconds;*
- *22:00 - 23:59 duration 35 seconds.*

After the defined time elapsed, the screen recording stopped, proceeding with its saving. If the user locked the device before the preset time, the recording stopped immediately, again proceeding with its saving. Initially, the recording was saved locally, in the device's memory, and then sent to the Firebase server via an upload function. Since numerous videos could be recorded throughout the day, taking up significant memory relative to availability, BullyBuster Questionnaire V. 1.0 stored the videos with a standard, constant name, allowing for continuous overwriting of the content. The Android code from the first phase served as the basis for subsequent improvements used in later phases.

Second Phase

The second phase of the app is also called BullyBuster Questionnaire V. 2.0 and has some improvements over the first phase. This version is quickly explained, specifying the updated and modified sections. The Second Phase app is structured into two main packages:

P: klogger
C: Keylogger
C: LocationService2
P: videoRecording

C: RecorderService
C: RecordingConfig
C: RecordingSession
C: Secondary
C: Constant
C: ContactVO
C: HowToUseActivity
C: MainActivity
C: PolicyManager
C: SampleDeviceAdminReceiver
Legend: C: Class, P: Package

The BullyBuster Questionnaire V. 2.0 application was executed thanks to the onCreate parent function of the C: MainActivity, which initialized the system variables of the app and ran its graphics part. In it, some functions responded to button listeners within the app, including the card_accessibility_listener, which allowed access to the mobile accessibility setting and set BullyBuster Questionnaire V. 2.0 services to "ON." Without the accessibility services turned on, the app would not perform all its functions correctly. A specific type of button affected the C: HowToUseActivity class, activated by the button labeled "Info Devices" in the app's graphics. Through special functions, it extracted the phone contacts and apps of the mobile device, displayed them on the screen, and stored them on the server.

After using the app, it may well be resized to use its other features. Thus, talking about the keylogger functionality that has always been active up to this point, every single user action on the device has been recorded. It runs in the background and extracts and saves to the server the text typed by the user, the user's focus in text format, and the icons or keys clicked. These actions are essentially addressed in the "klogger" package.

So, describing the C: Keylogger class, first, the initialization of the variables that affected the sensors was noted; in this implementation, the accelerometer values were recorded thanks to the onSensorChanged function, where a threshold was set that, if exceeded, recorded the acceleration. In addition, the sendUserDetailTOServer function allowed the connection to the server and then logged the actions of the users who used the app. The primary function of the C: Keylogger was onAccessibilityEvent, which had the peculiarity of continuously listening by detecting user AccessibilityEvent-type events. If the user wrote something, the TYPE_VIEW_TEXT_CHANGED event was triggered. If the user stayed still in an app and looked at something, the TYPE_VIEW_FOCUSED event was triggered, extracting the text of the specific thing they were reading at that moment. More importantly, if the user clicked on an app, the TYPE_VIEW_CLICKED event was triggered to record the textual name of the app being opened. Unlike other events, if they clicked on messaging apps such as Facebook, Instagram, WhatsApp, or Telegram, another Activity named C: Secondary was triggered, which was part of Package P: videoRecording.

Before understanding how the C: Secondary class works, let's explain the meaning of the package *"videoRecording."* It was responsible for starting video recording whenever the user clicked on a preset messaging app, be it Telegram, Instagram, WhatsApp, or Facebook. The video had the duration of the user's stay on the open messaging app, as the recording was stopped automatically when the app was closed. After recording the video, if there was a Wi-Fi network, it was forwarded to Firebase; otherwise, the video was stored locally. The video was forwarded to Firebase and deleted from the local storage when a Wi-Fi connection became available. This was an improvement over the previous version.

The C: Secondary class was called by the user's click action on a messaging app; it established permissions and started the C: RecorderService activity that handled, in the onStartCommand function, the start of recording (startRecorder function), end of recording at the appropriate time (stopRecorder function), and sending the recorded video to Firebase (sendVideoToFirebase function). Regarding the start of recording (startRecorder function), objects of the classes that supported settings and properties useful for recording and saving the newly recorded video to the cell phone were instantiated, namely C: RecordingConfig and C: RecordingSession. The C: RecordingConfig class presented focal and essential settings for configuring the video, such as video quality. The C: RecordingSession class presented configurations useful for momentary saving of the video, which saved to a specific preset location on the cell phone, then was found by the program, deleted, and sent to Firebase. Finally, class C: LocationService2 was a specific activity that continuously recorded the user's GPS location. The service with priority "PRIORITY_BALANCED_POWER_ACCURACY" was used to request the accuracy of the cell phone location. A sendSensorTOServer function in this module sent this information to the server via a php file.

Auxiliary classes:
C: ContactVO initialized objects that helped perform tasks involving user contact information. It was used for displaying contacts by the C: HowToUseActivity class. The classes C: PolicyManager and C: SampleDeviceAdminReceiver consisted of Android app system variables, which were recommended to remain unchanged. The C: Constant class had service variables for registration and Location service. The app was executable only on Android versions between 6 and 10. Tests were performed on a Huawei P10 lite with EMUI version 8.0.0 and Android 8.0.0.

Third Phase
The third phase of the app is called *"BullyBuster Questionnaire V. 3.0"*. This app featured a different design than the previous two versions. It resulted from countless project calls in which graphic and semantic designs of the screens to be implemented were verbally defined. A draft privacy policy was added in this version, which was to be explored further in the fourth phase of the app.

On the main screen, four buttons were changed to one that only allows acceptance of user permissions. The remaining buttons were deemed excessive, especially "Info Devices," which displayed and captured phone contacts and usernames because the app gained all the privacy policies protecting the testing user.

Earlier app versions considered the possibility of extrapolating chat videos from social media. This procedure for privacy discourse was removed and also considered wasteful because of the amount of video that was being saved on servers. In addition, it was deemed essential to focus the study within the app by adding animated videos outlining bullying and cyberbullying actions that could elicit mixed feelings and emotions among participants. In addition, a questionnaire is implemented to identify user profiling, which Dr. Grazia Terrone has researched and studied in depth.

The new Design contemplates five basic steps:
1. *Access to app functionality (3 screens): This step is the same as in the previous sections identified by the "Go to Accessibility" button. It is essential in all versions to give permissions to the app to start Keylogger and acquisition of Smartphone Sensors;*
2. *Access the four Cyberbullying videos (4 screens);*
3. *Access to the Preliminary Questionnaire (10 screens);*
4. *Access to the Telegram group (1 screen). In this group, making comments regarding the test taken will be possible.*
5. *End Test (1 screen).*

Fourth Phase

The Fourth and final phase of the app is called " *BullyBuster Questionnaire V. 4.0*." The implementation constructs illustrated in Phase Three were extended by enriching the app's implementation.

The following features were added and improved:

- An emotional question was added after each video screen that extrapolates the user's emotion immediately after viewing each of the four selected videos;
- Graphic buttons inherent to the main screen have been implemented with user controls. If the user has not agreed to the privacy policy and has not consented to the required permissions, the app has been unable to start the test. In previous versions, this was allowed at the risk of not receiving any data on the server.
- In this version, the questionnaire questions have been extended, making it compatible with School Student and adults. In addition, a drop-down menu has been added so that the acquisition school departments can be stratified;
- Several informational pop-ups were added in this version, making the interface more User-Friendly and more usable;
- BullyBuster project logos have been added to this version, and the privacy policy has been expanded;
- In this version, data are captured only within the "BullyBuster Questionnaire" app. Which in previous versions was also done outside the app;
- In this version, again, to enrich the privacy discourse, a final screen has been added that allows the overall uninstallation of the app.

As with previous versions, the app aims to capture data through text, touch values, and sensors. The app contemplates six basic steps:

1. *Access to the functionality of the app (3 screens);*
2. *Accessing the four Cyberbullying videos (4 screens) for each video, an emotional question (4 screens);*
3. *Access to the Questionnaire (more than 30 screens) that is quick to fill out;*
4. *Access to Telegram group (1 screen);*
5. *End Test (1 screen);*
6. *Uninstall app (1 screen).*

The app sent all functionality to the server only if you were connected to the internet (Mobile or Wi-Fi). In offline mode, the app did not send any data to the Aruba server. All functionality extracted from cell phones and tablets was saved on the Aruba server. This server had accounts and passwords under the custody of system administrators. No user registration was contemplated in the app. Every detected functionality adhered to privacy regulations, and every detected and saved data was anonymous. The entire test was estimated at 30 minutes, depending on the final debate.

During the test, the app recorded the following features that were saved on the server:

- *KeyLogger*: Everything that is typed on the keyboard;
- *Touch and Multi-Touch coordinates* are on the cell phone display during the test (e.g., Playing video, scrolling back and forth on video, etc.). Specifically, *ACTION_DOWN, ACTION_POINTER_DOWN, ACTION_POINTER_UP, ACTION_MOVE* (*X, Y, velocityX, velocity, pressure, surface*);
- *Questionnaire answers* (date and time, question, answer);
- *Sensor Values*: Gyroscope (x, y, z), Accelerometer (x, y, z), Proximity, Atmospheric Pressure, Magnetometer, Ambient Brightness, StepDetector (Some cell phones do not have all sensors).

3.2.1.2 Libraries

This set of imports defines the libraries and classes needed to develop an Android application with certain features, including accessibility, device usage monitoring, hardware sensors, touch event handling, and more. Here is a brief description of the leading android libraries used (not all):

1. **import android.accessibilityservice.AccessibilityService**: This allows you to create an accessibility service that can interact with other applications' UI content.
2. **import android.app.ActivityManager**: Allows you to access and manage the activities running in the Android system, including the ability to get information about featured activities.
3. **import android.app.usage.UsageStats**: Represents information about application usage statistics, such as the time of use of a specific app.
4. **import android.app.usage.UsageStatsManager**: Gets an instance of "UsageStatsManager," which provides access to app usage statistics.
5. **import android.content.Context**: Provides access to system resources and services.
6. **import android.content.Intent**: Used to communicate with other applications or system components.
7. **import android.hardware.Sensor**: Represents a hardware sensor on the device, such as an accelerometer, gyroscope, etc.
8. **import android.hardware.SensorEvent**: Encapsulates a sensor event containing data about the specific sensor.
9. **import android.hardware.SensorEventListener**: Interface to receive data updates from sensors.
10. **import android.hardware.SensorManager**: Provides methods to access and manage the hardware sensors on the device.
11. **import android.os.AsyncTask**: Class used to perform asynchronous operations in the background.
12. **import android.os.Build**: Provides information about the version and features of the running Android device.
13. **import android.util.Log**: Used for logging debug messages and information.
14. **import android.view.accessibility.AccessibilityEvent**: Encapsulates system-generated accessibility events.
15. **import android.view.accessibility.AccessibilityNodeInfo**: Provides information and methods for interacting with nodes in the accessibility view tree.
16. **import android.app.AlertDialog**: Provides dialog boxes for user interactions.
17. **import android.content.ComponentName**: Represents a component name (such as an activity or service) within the Android system.
18. **import android.content.DialogInterface**: Interface to implement button behavior in dialog boxes.
19. **import android.content.pm.PackageInfo**: Provides information about the package of an installed application, such as the package name and version.
20. **import android.content.pm.PackageManager**: Provides methods for accessing application package information.
21. **import android.os.Bundle**: Provides a container for data that can be passed between Android components.
22. **import android.view.MotionEvent**: This represents a touch input event on the screen.
23. **import android.view.VelocityTracker**: Used to monitor the speed of a touch event.
24. **import android.view.View**: Represents a view in the Android user interface.
25. **import android.widget.Button**: Represents a button in the Android user interface.
26. **import android.widget.GridLayout**: Layout that organizes its views in a grid.
27. **import android.widget.ScrollView**: Provides a scrollable view for content more significant than the screen.

3.2.1.3 BB Questionnaire App Development

I have thoroughly examined the specialized context of the application, analyzed the state-of-the-art methodologies relevant to typical implementations that address the problem at hand, and focused on developing the application to record sensor data, with a particular emphasis on accelerometer and touch acquisition.

In the implementation phase, the architectural scheme shown in Figure 11 was outlined.

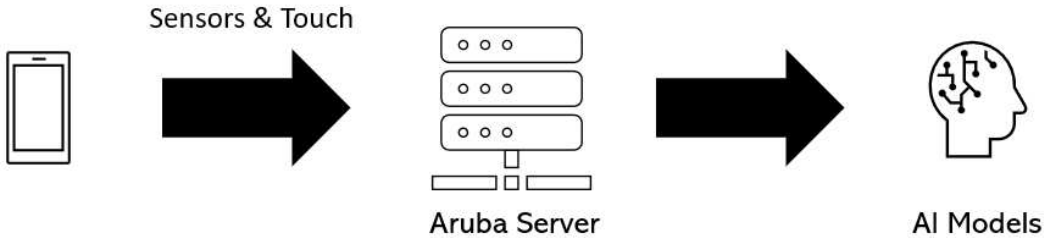


Figure 11 - System design architecture

The diagram depicts a typical design architecture that establishes the connection points between various hardware and software architectures. The focal point of the design architecture is the implementation of the Android mobile application, which is designed to conduct the questionnaire by sending the detected sensor data and touch coordinates to the designated server. The secure storage of data can then be carried out on Aruba servers. This method of data storage is among the most secure and practical for projects addressing this subject matter. Ultimately, these data have been made downloadable in an anonymous format, proving valuable in the implementation phase of AI algorithms in offline mode.

The initial phase involved the development of an Android app using Android Studio, written in Java, structured as illustrated in the image below. As delineated in 3.2.1.1 Application Versions, the fourth phase is described in the figure below. The classes utilized are self-explanatory and have been extensively described in Subsection 3.2.1.1 Application Versions (Figure 12).

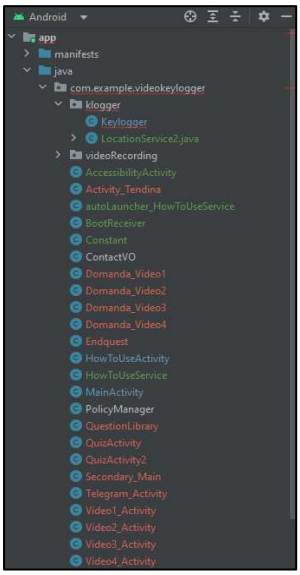


Figure 12 - Classes and Package

After being installed from the app store, the application has a project icon that reads "*BullyBuster Questionnaire*" (Figure 13).

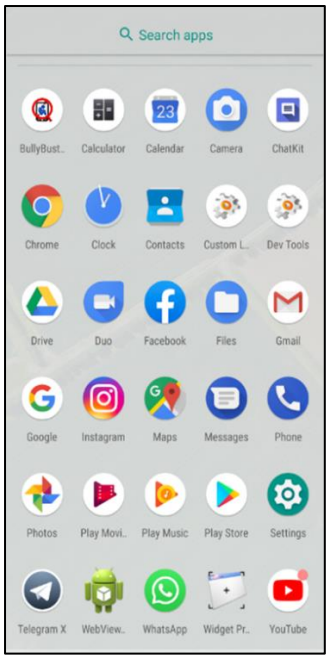


Figure 13 - Project Icon

After clicking on the icon, the **C: MainActivity** main screen is displayed. During the first start of the application, the Activity called is the **C: MainActivity**, and according to the lifecycle pattern of an Activity, the first function invoked is the *OnCreate()*. Within the *OnCreate()* function, the layout to be shown on the screen is defined by calling the *setContentView()* function. Click the *"Go to Bully-Buster Test"* button (Figure 14). Clicking on this button loads the next class, the **C: SecondaryMain**.

Figure 14 - C: MainActivity main screen

After loading the **C: SecondaryMain** screen, this, in turn, invokes the **C: Constant** class, which displays the privacy policy. This procedure is essential because without clicking the *"I agree"* button, you cannot use the application's full functionality and continue using it. The user must provide explicit consent to the privacy policy to ensure proper and legal handling of personal information in the application (Figure 15).

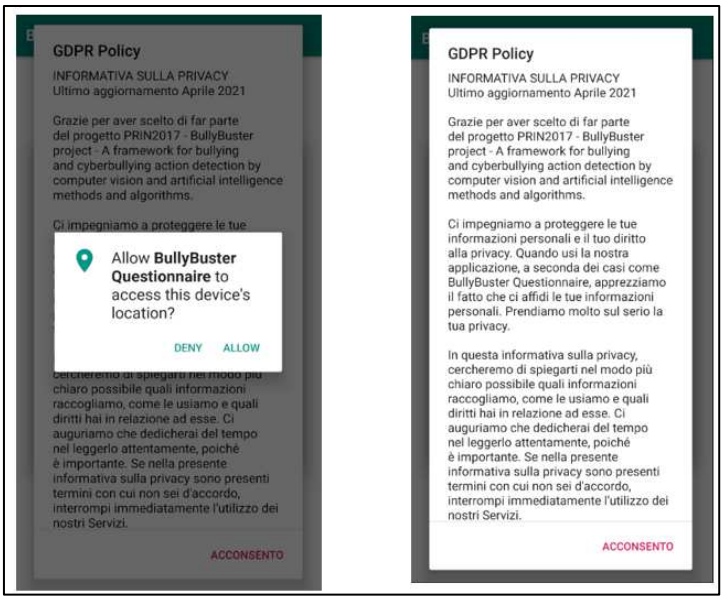


Figure 15 - C: SecondaryMain screen

Once the user has clicked *"I agree"* to consent to the privacy policy, a small disclosure outlining the basics of granting the necessary consent to access the device's touch and sensor hardware (Figure 16). The information lets the user know the data the application collects and uses to provide the desired functionality.

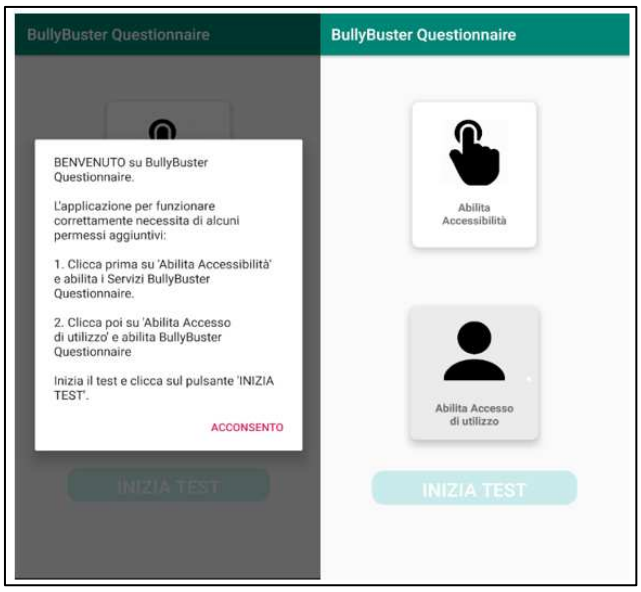


Figure 16 - Disclosure Granting

After obtaining the user's consent regarding privacy, the application executes the **C: AccessibilityActivity** class, which initiates a request to the system to access the device's accessibility settings using the constant *ACTION_ACCESSIBILITY_SETTINGS*. This allows the user to enable the permission needed for the system to acquire the relevant information.

From this point on, the **C: Keylogger** class is activated, which runs in the background and detects various information:

1. *TYPE_VIEW_TEXT_CHANGED*: Represents the text editing event in an EditText as a keylogger.
2. *TYPE_VIEW_FOCUSED*: Represents the display event of the View.

3. *TYPE_VIEW_CLICKED*: Represents the click event on a clickable component within the View, such as buttons, ListView, and RecyclerView.
4. *SENSOR_EVENT*: Detects data from the device's sensors, including gyroscope, accelerometer, proximity, atmospheric pressure, magnetometer, ambient brightness, and StepDetector.
5. *TOUCH EVENT (MOTIONEVENT): ACTION_DOWN (x,y,time), ACTION_POINTER_DOWN(x,y,time), ACTION_POINTER_UP (x,y,time), ACTION_MOVE (x, y, p, s, vX mVelocityTracker.getXVelocity, vY mVelocityTracker.getYVelocity), ACTION_UP (x,y,time)*

All this information is sent to the Aruba localhost server using the Android AsyncTask class. Specifically, the *onServiceConnected*() functionality is part of the *AccessibilityService* lifecycle and is the first to be called when the system and the service are successfully associated. Next, the *onAccessibilityEvent*() function is called at each new accessibility event, i.e., each change in the View. This function uses an event parameter of *AccessibilityEvent* to classify the type of event and sends the information to the server via *AsyncTask.execute()*. Note that the application performs monitoring functions, including logging keystrokes (keylogging) and detecting sensor data. It is important to note that these features may raise concerns about the privacy and ethical use of collected information. These features are used with the user's full consent and by privacy laws and ethical use of data (Figure 17).

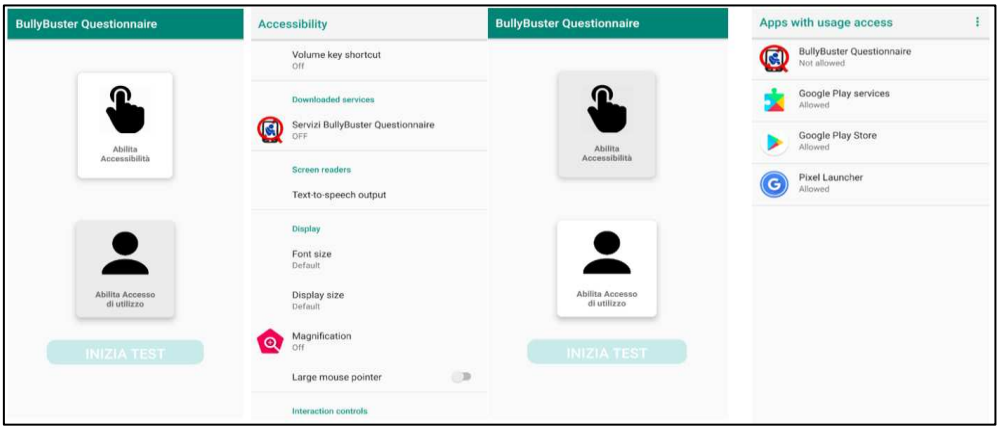


Figure 17 - C: AccessibilityActivity class Event

After providing all the required consents, the Keylogger has been active only within the application and has not interfered with other activities on the device.

Now, you can click the *"Start Test"* button (Figure 18) to start the test or the specific activity planned in the application. During the test, the application may continue to collect data via the Kaylogger and other features if necessary for the test to work or to provide the intended features.

Figure 18 - Start Test button

The actual test starts with the display of the first video, called Video1. The class **C: Video1_Activity** is loaded. The emotional question is displayed after the user clicks the "next" button. The class **C: Question_Video1** is loaded next. This Iter is executed with all the Video Activities (*C:Video2_Activity , C:Video2_Activity, C:Video2_Activity*) and the various Video Questions (C*:Question_Video2, C:Question_Video3, C:Question_Video4*). In (Figure 19) the graphic part of the screens is shown.

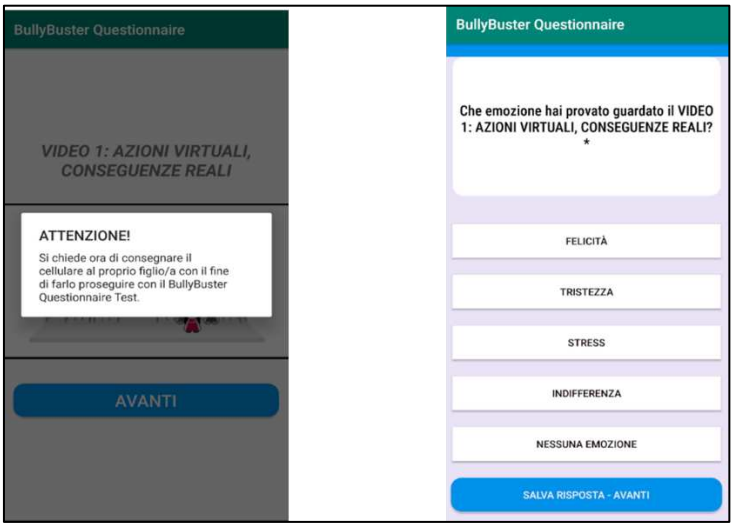


Figure 19 - C: Video_Activity and C: Domanda_Video

After class **C:Question_Video4,** the question section is loaded and divided between questions with textual input and 5-likert scale questions. Class **C: QuizActivity2** loads the questions with textual input, which chronologically comes first. The class **C: QuizActivity** loads the 5-Likert scale questions that allow us to do the future categorization of the personality index. While class **C: QuestionLibrary** does not contain all the questions and answers in text format, the front end must populate the screens with new questions. In (Figure 20) the graphical part of the screens is shown.

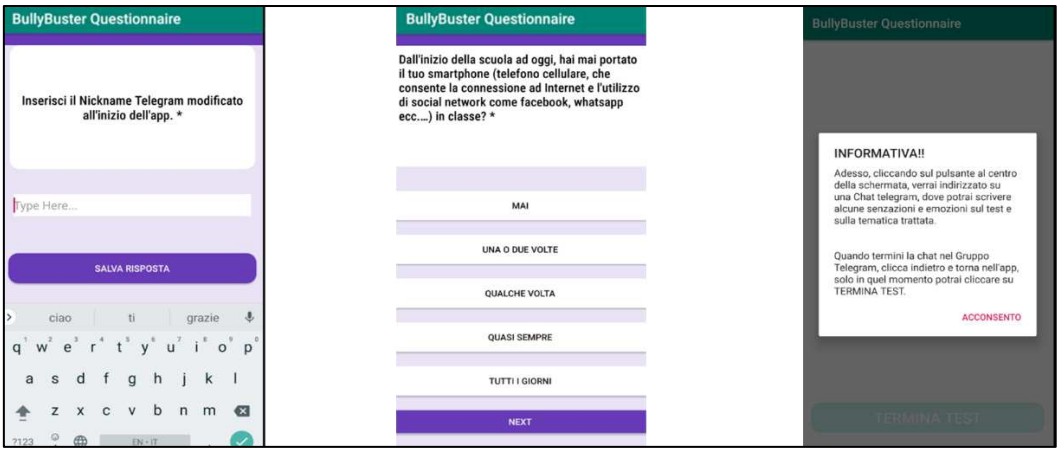


Figure 20 - C:QuizActivity2 and C:QuizActivity2 screen

At the end of the 5-likert scale questions, brief information alerted the user that they could access the Telegram group by clicking the middle button. The user exited the main application, all data acquisition ceased, and they accessed the Telegram group. The purpose of the Telegram group was to leave an opinion about the test performed and the importance of prevention. The class that was loaded was C: TelegramActivity.

In (Figure 21) the graphical part of the screens is shown.

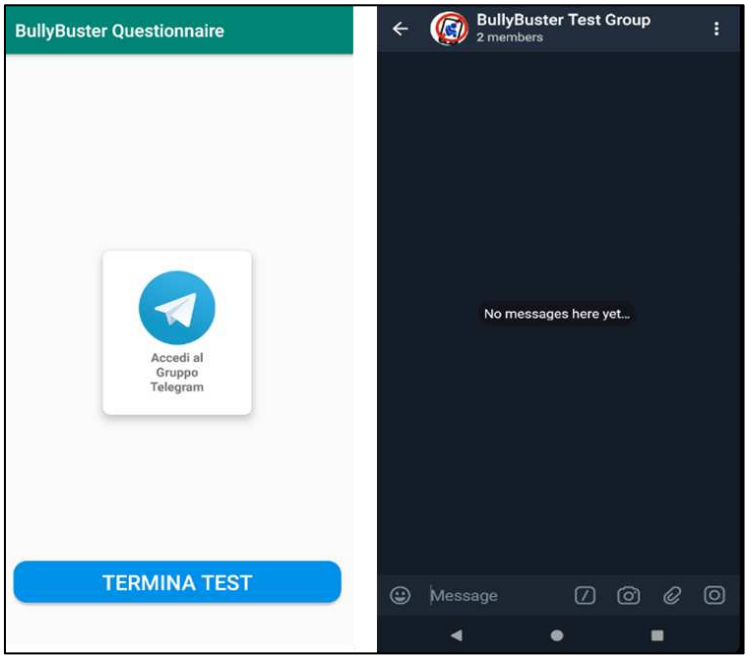


Figure 21 - C: TelegramActivity screen

The test ends with the last schematic shown in (Figure 22). The acquisition ceased, even in the main app, since then. After the screenshot in (Figure 22) the system directs the user to the uninstall app screen.

Figure 22 - Test Ends screen

The front end of the application has been shown in the previous screenshots. Examples of functions in Java that have been considered as 'core functions' of the system have now been shown.The code represents a part of an Android application that uses the SensorManager to register various device sensors (e.g., accelerometer, gyroscope, magnetometer, light, proximity, etc.) and listen to their events. The code starts by obtaining an instance of the SensorManager using the getSystem-Service(Context.SENSOR_SERVICE) method. A reference for each sensor type of interest is obtained using the SensorManager's getDefaultSensor(Sensor.TYPE_XYZ) methods, where "XYZ" represents the sensor type (e.g., Sensor.TYPE_ACCELEROMETER, Sensor.TYPE_GYROSCOPE, etc.). The code was registered as a listener for the various sensors by calling sensorManager.register-Listener(this, sensor, delay), where this indicated that the current object (presumably the activity or class implementing the SensorEventListener interface) handled the specified sensor events. The delay parameter specifies how often the system should update the listener object. Log.d() is used to write debug messages to the Android system logs, reporting the start of service and other sensor-related information.

```
@KayLogger
public void onServiceConnected() {
    Log.d("Keylogger," "Starting service");
    // Give some tag
    Log.d("SENSOR M," "onCreate: Initializing Sensor Services");
    // Get the system services, we need to get the system manager services
    // Get the permission to use the sensor
    sensorManager=(SensorManager) getSystemService(Context.SENSOR_SERVICE);
    // Now need to get a sensor type
    accelerometer=sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    gyroscope=sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
    magnetometer=sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
    light=sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
    proximity=sensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);
    stepDetector = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_DETECTOR);
    pressure = sensorManager.getDefaultSensor(Sensor.TYPE_PRESSURE);
    // Now we need to register a listener
    //You can specify other data delays, such as SENSOR_DELAY_GAME (20,000 microsecond delay 50hz), SENSOR_DELAY_UI (60,000 micro-
second delay), or SENSOR_DELAY_FASTEST (0 microsecond delay), SENSOR_DELAY_NORMAL (200,000 microsecond delay)
    sensorManager.registerListener(this, accelerometer, SensorManager.SENSOR_DELAY_UI);
    sensorManager.registerListener(this, gyroscope, SensorManager.SENSOR_DELAY_UI);
    sensorManager.registerListener(this, proximity, SensorManager.SENSOR_DELAY_NORMAL);
    sensorManager.registerListener(this, magnetometer, SensorManager.SENSOR_DELAY_UI);
    sensorManager.registerListener(this, light, SensorManager.SENSOR_DELAY_NORMAL);
```

```
    sensorManager.registerListener(this, stepDetector, SensorManager.SENSOR_DELAY_NORMAL);
    sensorManager.registerListener(this, pressure, SensorManager.SENSOR_DELAY_NORMAL);

    Log.d("SENSOR M 2", "onCreate: Registered accelerometer listener");
}
```

Instead, this code is a set of touch input event handlers for a class that records users' touch actions and sends them to a server via a keylogger service:

1. **MotionEvent.ACTION_DOWN**: This is executed when the first finger is pressed on the screen. It records the initial touch position and sends the data to the server.
2. **MotionEvent.ACTION_POINTER_DOWN** occurs when an additional finger is pressed on the screen while one or more fingers are already in motion. It records the position of the new touch and sends the data to the server.
3. **MotionEvent.ACTION_POINTER_UP** occurs when one of the moving fingers is released from the screen. It records the position of the released finger, sends data to the server, and removes the finger from the trace list.
4. **MotionEvent.ACTION_MOVE**: Occurs when a moving finger moves on the screen. It records each finger's position and speed of movement in the trace list and sends the data to the server.
5. **MotionEvent.ACTION_UP**: This occurs when all fingers are removed from the screen. It records the position of the removed finger, sends the data to the server, and removes the finger from the trace list.

Each event handler creates a list of data to send to the server with information such as finger ID, coordinates (x, y), pressure, and velocity.

```
Case MotionEvent.ACTION_DOWN:{
    if(mVelocityTracker == null)
        mVelocityTracker = VelocityTracker.obtain();
    else
        mVelocityTracker.clear();

    tracks.add(String.valueOf((event.getPointerId(event.getActionIndex()))));
    final float x = event.getX(tracks.size()-1);
    final float y = event.getY(tracks.size()-1);

    ArrayList<String> mylist = new ArrayList<String>();
    mylist.add(RecorderService.reliableIdentifier + "**!!**" + time + "**!!**ACTION_DOWN_"+namePage+"**!!**" + millis + "**!!**" +
"id:"+tracks.get(tracks.size()-1)+" x:" + String.valueOf(x) + " y:" + String.valueOf(y) + " - " + time);
    new Keylogger.sendSensorTOServer().execute(mylist);
    Log.d("touch", "DOWN trak:"+tracks.get(tracks.size()-1)+" x:"+x+ " y="+y);
    break;
}

case MotionEvent.ACTION_POINTER_DOWN:{
    tracks.add(String.valueOf((event.getPointerId(event.getActionIndex()))));
    final float x = event.getX(tracks.size()-1);
    final float y = event.getY(tracks.size()-1);

    ArrayList<String> mylist = new ArrayList<String>();
    mylist.add(RecorderService.reliableIdentifier + "**!!**" + time + "**!!**ACTION_POINTER_DOWN_"+namePage+"**!!**" + millis +
"**!!**" + "id:"+tracks.get(tracks.size()-1)+" x:" + String.valueOf(x) + " y:" + String.valueOf(y) + " - " + time);
    new Keylogger.sendSensorTOServer().execute(mylist);
    Log.d("touch", "POINTER_DOWN trak:"+tracks.get(tracks.size()-1)+" x:"+x+ " y="+y);
    break;
}

case MotionEvent.ACTION_POINTER_UP:{
    final int index = tracks.indexOf(String.valueOf(event.getPointerId(event.getActionIndex())));
    final String id = String.valueOf(event.getPointerId(event.getActionIndex()));
    if(index >= 0){
        final float x = event.getX(index);
        final float y = event.getY(index);
        ArrayList<String> mylist = new ArrayList<String>();
        mylist.add(RecorderService.reliableIdentifier + "**!!**" + time + "**!!**ACTION_POINTER_UP_"+namePage+"**!!**" + millis + "**!!**"
+ "id:"+id+" x:" + String.valueOf(x) + " y:" + String.valueOf(y) + " - " + time);
        new Keylogger.sendSensorTOServer().execute(mylist);
        Log.d("touch", "POINTER_UP trak:"+tracks.toString()+" id:"+event.getPointerId(event.getActionIndex()));
```

```
        tracks.remove(index);
    }
    break;
}

case MotionEvent.ACTION_MOVE:{
    for (int i =0; i<tracks.size(); i++){
        if(i==0){
            mVelocityTracker.addMovement(event);
            mVelocityTracker.computeCurrentVelocity(1);
        }
        final float x = event.getX(i);
        final float y = event.getY(i);
        final float p = event.getPressure(i);
        final float s = event.getSize(i);
        final float vX = mVelocityTracker.getXVelocity(i);
        final float vY = mVelocityTracker.getYVelocity(i);
        ArrayList<String> mylist = new ArrayList<String>();
        mylist.add(RecorderService.reliableIdentifier + "**!!**" + time + "**!!**ACTION_MOVE_"+namePage+"**!!**" + millis + "**!!**"
+"id:"+tracks.get(i)+ " x:" + String.valueOf(x) + " y:" + String.valueOf(y) +" vX:"+vX+" vY:"+vY+ " p:" + String.valueOf(p) + " s:" + String.val-
ueOf(s) + " - " + time);
        new Keylogger.sendSensorTOServer().execute(mylist);
        Log.d("touch", "MOVE id:"+tracks.get(i)+" x:"+x+ " y="+y+" p:"+p+" s:"+s+" vX:"+mVelocityTracker.getXVelocity(i)+" vY:"+mVeloci-
tyTracker.getYVelocity(i));
    }
    break;}

case MotionEvent.ACTION_UP:{
    if(tracks.size()>0){
        final int index = tracks.indexOf(String.valueOf(event.getPointerId(event.getActionIndex())));
        final String id = String.valueOf(event.getPointerId(event.getActionIndex()));

        final float x = event.getX(index);
        final float y = event.getY(index);

        ArrayList<String> mylist = new ArrayList<String>();
        mylist.add(RecorderService.reliableIdentifier + "**!!**" + time + "**!!**ACTION_UP_"+namePage+"**!!**" + millis + "**!!**" + "id:"+id+"
x:" + String.valueOf(x) + " y:" + String.valueOf(y) + " - " + time);
        new Keylogger.sendSensorTOServer().execute(mylist);
        Log.d("touch", "UP" +tracks.toString()+" get(index):"+tracks.get(index)+" index: "+index+" id:"+ id);
        tracks.remove(index);}
    break;}
```

This code represents a Java class that extends AsyncTask and is responsible for sending sensor data to a remote Aruba server.

The *sendSensorTOServer* class is a subclass of AsyncTask, which allows it to perform background operations asynchronously to the main thread of the Android application. The code sends a string of data about a given sensor to the Aruba server using a POST request. The post request contains all the data the .php file handles and saves in the appropriate folders on the Aruba server. The useful libraries *DefaultHttpClient* and *HttpPost* are efficient for making *HTTP requests* in Android.

```
public static class sendSensorTOServer extends AsyncTask< ArrayList<String>, Void, String> {
    @Override
    protected void onPreExecute() {
    }

    protected String doInBackground(ArrayList<String>... alldata) {
        String result = "";
        ArrayList<String> passed = alldata[0]; //get passed arraylist
        String value = passed.get(0);

        InputStream is = null;
        try {
            ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
            nameValuePairs.add(new BasicNameValuePair("value", value));

            HttpClient httpclient = new DefaultHttpClient();
            HttpPost httppost = new HttpPost("__ArubaServer__write_sensor_log.php");
            httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs,"UTF-8")); // UTF-8  support multi language
            HttpResponse response = httpclient.execute(httppost);
            HttpEntity entity = response.getEntity();
            is = entity.getContent();
```

```
            Log.e("pass 1", "connection success ");
        }
        catch (Exception e)
        {
            Log.e("Fail 1", e.toString());
        }
        return result;
    }
}
```

This is similar to the "*sendSurveyTOServer*" class that sends the questionnaire string to the Aruba server via the .php file. That is time, question, answer.

```
public class sendSurveyTOServer extends AsyncTask<ArrayList<String>, Void, String> {
    @Override
    protected void onPreExecute() {}
    protected String doInBackground(ArrayList<String>... alldata) {
        String result = "";
        ArrayList<String> passed = alldata[0]; //get passed arraylist
        String value = passed.get(0);

        InputStream is = null;
        try {
            ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
            nameValuePairs.add(new BasicNameValuePair("value", value ));

            HttpClient httpclient = new DefaultHttpClient();
            HttpPost httppost = new HttpPost("_ServerAruba__write_log_survey.php");
            httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs ,"UTF-8"));
              // UTF-8  support multi language
            HttpResponse response = httpclient.execute(httppost);
            HttpEntity entity = response.getEntity();
            is = entity.getContent();
            Log.e("pass 1", "connection success ");
        }
        catch(Exception e)
        {
            Log.e("Fail 1", e.toString());
        }
        return result;
    }
    @Override
    protected void onPostExecute(String result) {}
}
```

The communication between the app and the server is handled by the *"org.apache.http.legacy"* library; the client, therefore, executes a call to a script inside the server, developed in PHP, and sends in POST the collected data. The script processes the received information and concludes its life cycle by generating a log file saved in the root of the project repository, also within the same server, containing the event timestamp, type, and relevant information. On the Aruba server, the folder "Kidskeylogger" was created to contain all operational files and various save folders (Figure 23). The critical folders include "info_devices, log_device, log_survey, sensor_log_device." All files with the previously mentioned data appear in these folders.

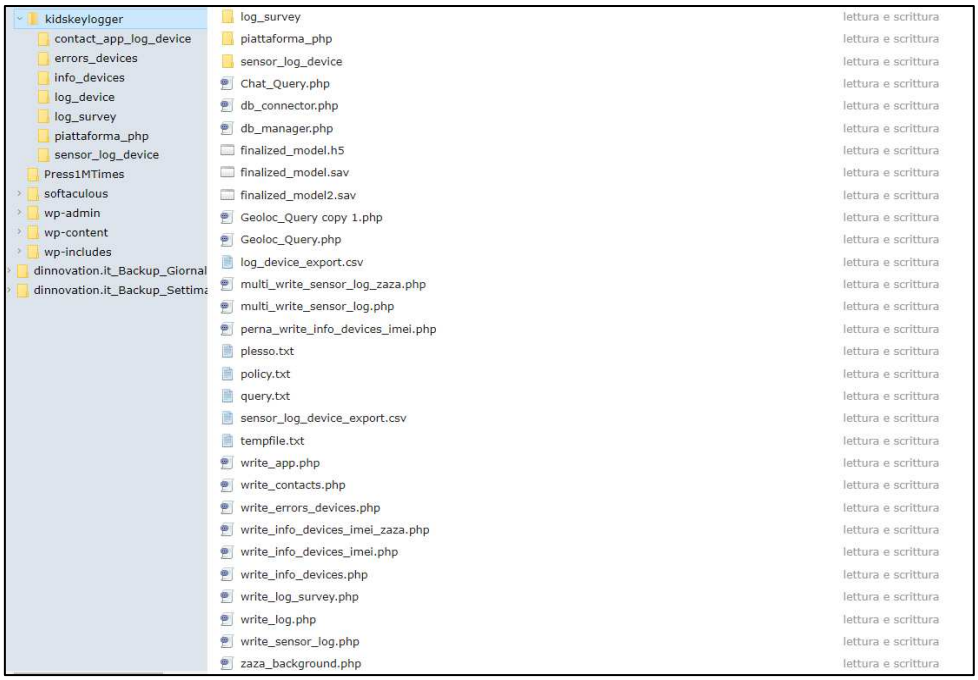


Figure 23 - Aruba server folders

The contents of the various .PHP files are similar to the one below. This PHP code receives data via POST, processes it, and writes some information to a log file. Here is a description in Italian of the main actions performed by the code:

1. *Retrieves the value sent via POST and saves it in the $data variable.*
2. *Extract the device ID and date value from $data_arr.*
3. *Transforms the date to a valid database format (with '-' character instead of '/') and converts it to a timestamp.*
4. *Formats the date and time in the format "YYYY-MM-DD HH:MM:SS" and saves it in the variable $date_val.*
5. *Remove any newline characters from the $question variable.*
6. *Extract the questionnaire response from the variable $data_arr and save it in the variable $answer.*
7. *Construct a text string with the date, question, and answer.*
8. *Opens a device-specific log file and adds the text string.*

Saving the data to the appropriate folders is done in these eight steps. In summary, the code receives data from a form, processes it, registers it in the database via the DatabaseManager object, and creates a device-specific log file with some information about the survey taken.

```
<?php
$data = $_POST["value"];
$data = str_replace(\", ' ', $data);
$data_arr = explode("**!!**", $data);
$device_id = $data_arr[0];
$data = $data_arr[1];
$dataStr = str_replace('/', '-', $data_arr[1]);
$timestamp = strtotime(str_replace('/', '-', $data));
$date_val = date("Y-m-d H:i:s", $timestamp);
$question = str_replace("\n", '', $data_arr[2]);
$answer = $data_arr[3];

$testo = "\"$dataStr\" -\"$question\" - \"$answer\" \n";
$my_file = "log_survey/file_" . $device_id . ".txt";
$handle = fopen($my_file, 'a') or die('Cannot open file:  ' . $my_file);
fwrite($handle, $testo);
```

3.2.2 Questionnaire Web Platform Implementation

This subchapter addresses the implementation of the Web Questionnaire Platform, referred to as "BB Questionnaire." Here, various critical aspects pertinent to the development of this application are examined, focusing on three principal elements: the different versions of the application, the libraries utilized, and the overall development process of the platform.

Notably, there is no discernible difference compared to the application described in Section 3.2.1 Android Smartphone Application Implementation. This web version thoroughly mirrors the design and implementation choices made for the Android application.

The first section centers on the libraries employed during development. These libraries proved indispensable, as they provided predefined functionalities that simplified and enhanced the development process, ultimately reducing the need for extensive manual coding. An overview is given of the libraries chosen, their core functionalities, and their contribution to the application's implementation. The subsequent section delves into the development process of the BB Questionnaire Web application, describing the methods, techniques, and tools applied by the development team to bring the platform to fruition. This section presents an overview of the challenges encountered, the decisions reached, and the best practices adopted throughout the development journey.

3.2.2.1 Libraries

The *Python libraries* you listed seem to be those commonly used to develop web applications using the Flask framework. Here is a brief explanation of each library:

1. **Flask**: Flask is a lightweight framework for web applications in Python. It is trendy because of its simplicity and ease of use. Flask allows you to create web applications with clean, well-structured code. It is flexible and offers many extensions to add additional functionality.
2. **render_template**: This function is part of the Flask module and is used to render an HTML template to generate the response sent to the client. With this function, you can incorporate dynamic data into the HTML template to make it custom.
3. **url_for**: This function is also part of Flask. It is used to build URLs dynamically within the application. This is particularly useful for avoiding the creation of rigid URLs, allowing easier management of URL changes.
4. **redirect**: This function allows the client to redirect to another URL. It helps handle redirects after processing a form or redirecting to other site pages.
5. **request**: This module is part of Flask and allows you to access data sent from the client to the server. You can use requests to get the parameters of an HTTP request, whether in a form, in the URL, or elsewhere.
6. **datetime**: This standard Python library provides functionality for working with dates and times. It can manipulate time data, convert formats, and calculate date differences.
7. **os**: This is another standard Python library that provides functions for interacting with the operating system. It helps handle file operations, such as read/write, directory management, etc.
8. **pandas**: Pandas is a viral data manipulation and analysis library. It is used to work with tabular data efficiently, offering data structures such as DataFrame to facilitate data management.
9. **time**: This standard Python library offers functions for working with time. It can generate pauses, compute runtimes, or for timestamp operations.
10. **requests**: The "requests" library sends HTTP requests to other servers. It is often used to make API calls, obtain data from Web sites, or send data to remote servers.

3.2.2.2 Development BB Questionnaire Web Platform

I have thoroughly examined the context in which the app specializes, analyzed the state-of-the-art techniques involving typical applications that address the problem, and focused on implementing the application to record sensor values, concentrating on accelerometer and touch acquisition. In the

implementation phase, the architectural scheme shown in Figure 24 was outlined. The following scheme is very similar to the one implemented for the Android application.

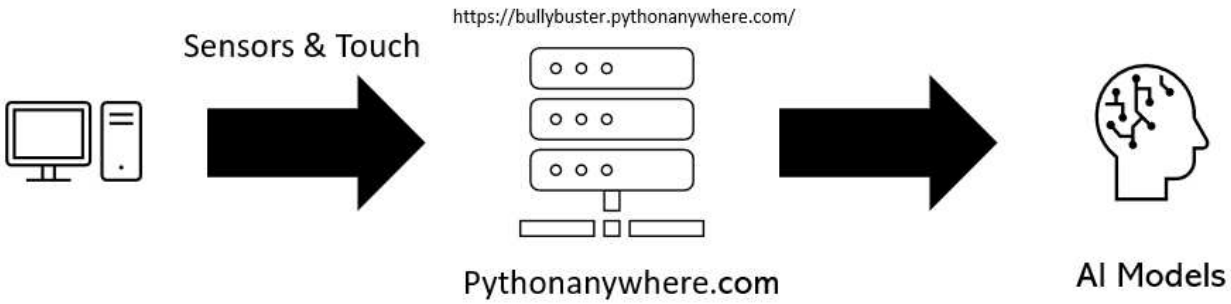


Figure 24 - Design architecture of the web system

A typical design architecture enshrines the connection points between the various Hardware and Software architectures in the schematic. The pivot of the design architecture was the implementation of the WEB platform, which carried out the questionnaire, sent the detected sensor data, and touch coordinates to the prepared server. Data saving was securely done on Pythonanywhere.com servers. Finally, this data was downloadable in an anonymous format, useful for the implementation phase of the AI algorithms in offline mode. The application was viewable at the following link: https://bullybuster.pythonanywhere.com/.

The web application was accessible from smartphones with Safari and Chrome browsers. It was also accessible from a PC, and the responses were saved, but the sensor listeners were not. The first step was accessing the application via a link. After that, the main screen displayed a button labeled, “Click me to start the quiz.” (Figure 25).

Figure 25 - Main Screen Web

After this phase, the permission issue was granted. Similar to that performed for the Android application, this step was needed to detect and save the sensors and touch (Figure 26).

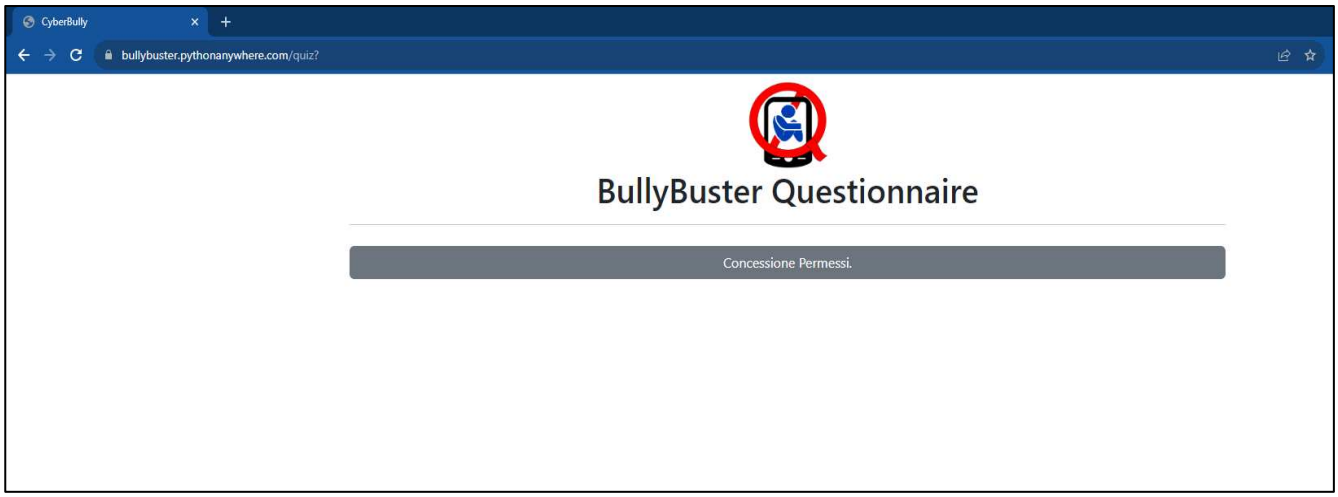


Figure 26 - Permission Screen

Much like the process for the Android application, video screens, emotionally resonant questions, and survey items appeared initially. Access to the Telegram group was excluded from the web version, as the project team deemed it outdated. The web version, in turn, demonstrated promising potential for surpassing the Android application in terms of innovation.

Here, the questions are divided by textual input and 5-likert scale questions. First, the text input questions that chronologically come first. Then, the 5-Likert scale questions allow us to do the future categorization of the personality index. In (Figure 27-Figure 28-Figure 29) the graphic part of the screens is shown.

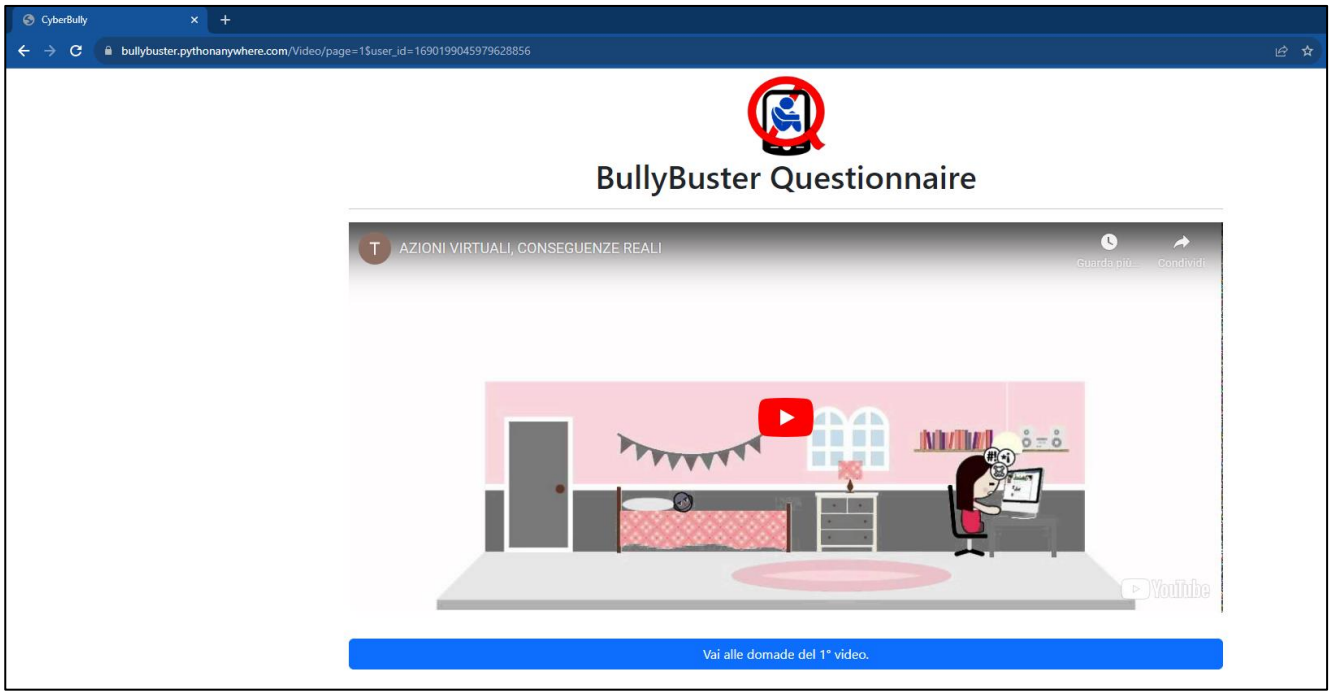


Figure 27 - Video Screen

Figure 28 - Emotion Question Screen

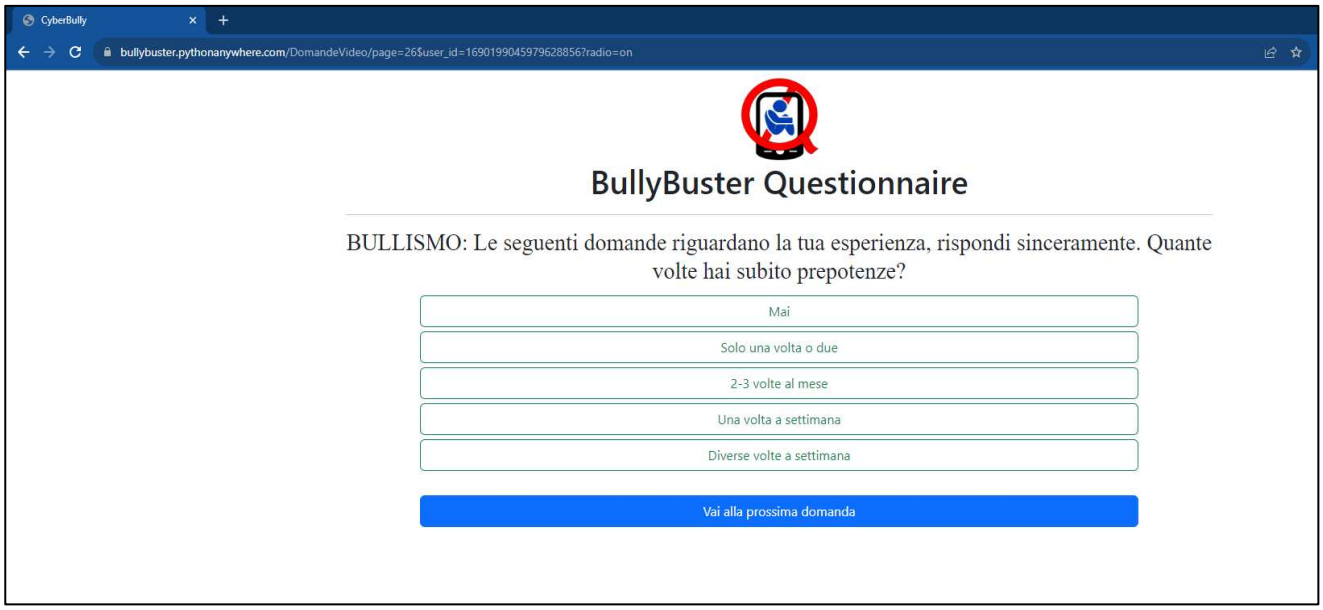


Figure 29 - 5 Likert scale questions screen

The code structure is straightforward: created several HTML pages displaying the frontend questionnair. *QuestionsVideo.html, Home.html, Quiz.html, Video.html, base.html, questionnaire.html, test.html*. For example, the file "*Home.html*".

A CSS style is defined for the class *"center"* that positions the element in the center horizontally. A div with the class *"row align-items-center"* is created to align its contents vertically. Within this div, another div with class "col h-100 w-100" occupies the entire available height and width. Inside this last div, there is a link "a" with attribute *"href"* pointing to *"/quiz."* Inside the link is an image from a static resource called "FirstPage.jpg." The image is horizontally centered, thanks to the "*center*" class. Next, another div with the class "*row align-items-center*" is created to align its contents vertically. Inside is another div with class "col h-100 w-100," occupying the available height and width. Within the latter div, a horizontally centered H5-level title prompts the user to click on the image to start the quiz, or they can click on the button below for the same purpose. Below the title is a "*form*" form with an "*action*" attribute pointing to "/quiz," and within the form is a button with class "*btn btn-*

secondary" that displays the text "*Click me to start the quiz.*" These. HTML files are contained in the "*templates/*" folder.

```
Home.html
{% block content %}
<style>
  .center {
    display: block;
    margin-left: auto;
    margin-right: auto;
    width: 100%;
  }
</style>

<div class="row align-items-center">
  <div class="col h-100 w-100">
    <a href="/quiz" align="center">
      <img src="{{url_for('static', filename='PrimaPagina.jpg')}}" alt="Clicca quì per iniziare il quiz" class="center">
    </a>
  </div>
</div>
<div class="row align-items-center">
  <div class="col h-100 w-100">
    <h5 align="center">Clicca l'immagine per iniziare il quiz, oppure clicca il bottone qui sotto.</h5>
    <form action="/quiz" align="center">
      <button class="btn btn-secondary">Cliccami per avviare il quiz.</button>
    </form>
  </div>
</div>
{% endblock %}
```

The platform used https://www.pythonanywhere.com/, and these folders were created. In the main folder, two critical files were found. The file flask_app.py created the Flask architecture, and the file QUESTIONS.csv contained all the questions in the questionnaire (Figure 30).

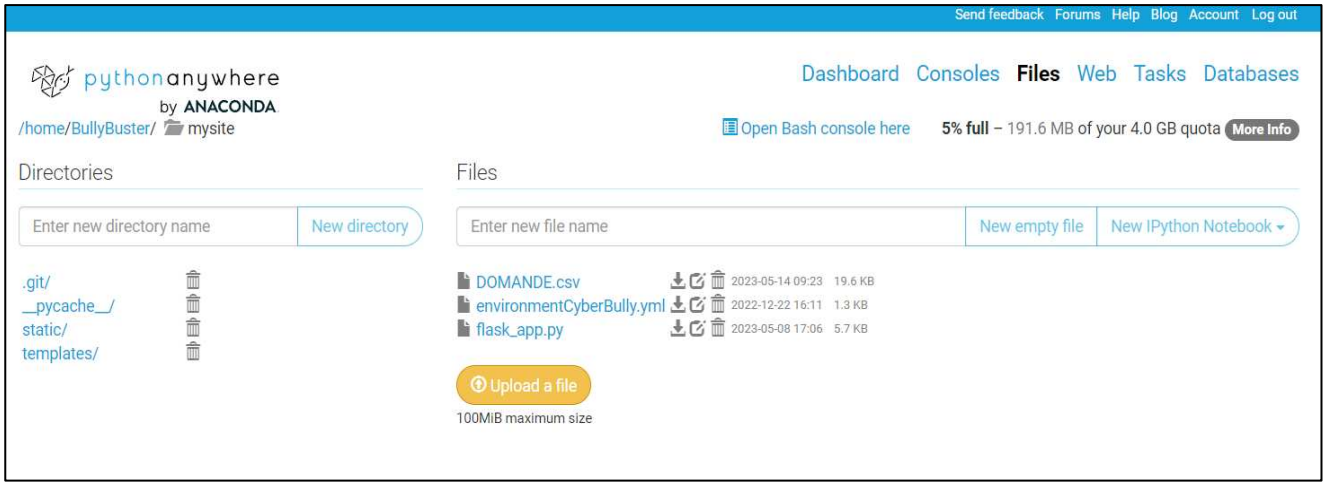


Figure 30 - Pythonanywhere folders

While sensor acquisition is done thanks to the file *"/home/BullyBuster/mysite/static/sensors.js."* Sensors.js, written in JavaScript, deals with data from motion sensors (accelerometer and gyroscope) and user interactions (touch coordinates).

1. *handleMotion(event):* This function is called when new data is available from the motion sensors (accelerometer and gyroscope). The function prints a log message *("Motion Added")* and updates the accelerometer and gyroscope fields by calling the *"updateFieldIfNotNull"* function.
2. *showCoordinates(event):* This function is called when a user input event (*click, mouse move, or touch)* occurs. The function gets the coordinates of the touch or click and updates the fields related to the coordinates by calling the *"updateFieldIfNotNull"* function.

Both functions use an *event* as a parameter, an object containing the event data that triggered the function call. For example, *event. Acceleration* and *event.rotationRate* contains the acceleration and angular velocity information detected by the motion sensors.

```
function handleMotion(event) {
    console.log("Motion Added");
    // updateFieldIfNotNull('Accelerometer_g', event.accelerationIncludingGravity.x + '*' + event.accelerationIncludingGravity.y + '*' + event.accel-
erationIncludingGravity.z);

    updateFieldIfNotNull('Accelerometer', event.acceleration.x + '*' + event.acceleration.y + '*' + event.acceleration.z + '*' + event.interval);

    updateFieldIfNotNull('Gyroscope', event.rotationRate.alpha + '*' + event.rotationRate.beta + '*' + event.rotationRate.gamma);
    event.stopPropagation();
    return;
}

function showCoordinates(event) {
    var nome_evento = event.type
    var x = 0
    var y = 0

    if (nome_evento === 'click' || nome_evento === "mousemove"){
        x = event.clientX;
        y = event.clientY;
    }else{

        x = event.touches[0].clientX;
        y = event.touches[0].clientY;
    }

    if (!e) var e = window.event;
    e.cancelBubble = true;
    if (e.stopPropagation) e.stopPropagation();

    var activity_name = nome_evento

    updateFieldIfNotNull(activity_name, x + '*' + y);
    event.stopPropagation();
    return;
```

Unlike the previous application, which saves everything on the Aruba server, this web application saves everything in the folder *"/home/BullyBuster/to_send/sensor_log_device,"* without splitting it into folders. In fact, in the (Figure 31) you notice an identifier created by the timestamp names all the files saved during the test.

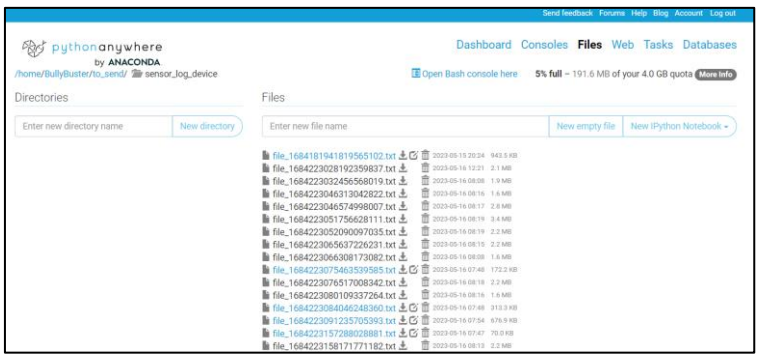


Figure 31 - sensor_log_device folder

An example of saving user data with ID 168418... etc., is shown in (Figure 32). Each raw has the field id, time, name, id, x, y, and z for sensors. The same is true for touch data.

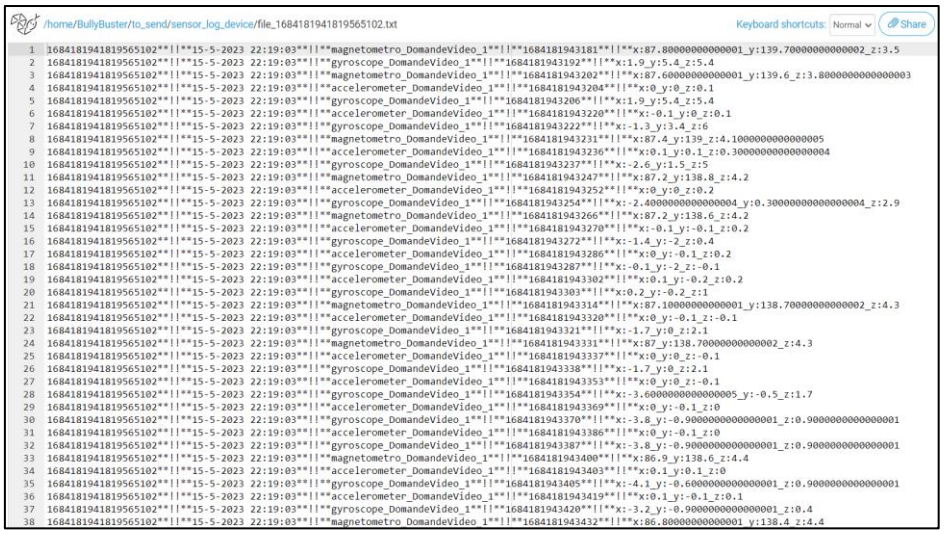


Figure 32 - Example of saving user data in the sensor_log_device folder

3.3 Experimental Strategy

The subchapter presents the methodologies adopted for data collection and analyzes the results of experiments conducted with both University Students and School Students. Initially, experiments were carried out with the University dataset, and subsequently, the same experiments were replicated using the School Students dataset. The research questions outlined below are common to both datasets to facilitate a comparative analysis.

Section 3.3.1 Real-Context Data, describes the data collection methods, including the test settings conducted in various Italian cities. A real-world context was deemed essential to ensure a detailed and contextualized analysis, guaranteeing that the data accurately reflected the conditions of the project. Section 3.3.2 Experiment Strategies – University Student, details the testing strategies applied and the associated publications that support the experimental approach adopted. The focus was placed on the dataset relating to students from the University of Bari Aldo Moro (UNIBA), aiming to better understand the specific behavioral dynamics of this population. Section 3.3.3 Experiment Strategies – Comparison of University Student-School Student, provides a comparative analysis between university students (UNIBA) and high school students (from Cagliari and Avellino), highlighting similarities and differences in the responses and behaviors of the two populations.

In the initial phase of the experimental process, data collection is crucial for obtaining the necessary information for analysis and validating results. The experiment's effectiveness heavily relies on the quality and precision of the data gathered. This section delves into the methods and rationales underlying data collection, with a focus on testing and interpreting the data itself. Testing strategies vary depending on the context and objectives of the test, defining the analytical and evaluative approach adopted to ensure the overall effectiveness of the experimental process. The chapter also addresses various testing-related issues and key aspects related to the analysis of questionnaires and the integration of sensors to assess behavioral activities.

These research questions illustrate the core considerations:

1. **First Aspect**: Questionnaire completion method. It was essential to verify whether each user had completed the questionnaire in person or if instances of remote or proxy completion were present to assess the reliability of responses.
2. **Second Aspect**: Identification of "Outlier" questions, specifically those with responses deviating from the normal statistical pattern, relevant for detecting discriminative elements based on personality indices.

3. **Third Aspect**: Examining whether the personality index obtained from questionnaires correlates with individuals' behavioral activity. This allowed for an understanding of whether personality traits mirrored individuals' actual daily behaviors.
4. **Fourth Aspect**: Evaluating the feasibility of replacing questionnaire-based calculations exclusively with sensor data. The integration of human activity recognition technologies may offer an innovative perspective for assessing behaviors with greater accuracy.

This chapter introduction provides a comprehensive overview of the experimental strategies and research questions addressed. Each research question led to the writing of scientific articles.

3.3.1 Real-Context Data

Prior to testing, comprehensive heuristic evaluations were meticulously conducted by domain experts actively engaging with the smartphone application and web platform. These evaluations primarily aimed to ascertain, with exceptional accuracy, the reliability of the saved data and the efficiency with which recorded values were promptly transmitted and stored on the servers.

To ensure a thorough and representative analysis, three schools attended by students (located in Avellino, Cagliari, and Marigliano) and a university first-year class in Bari participated in the testing process. This diverse participant group facilitated a well-rounded assessment of the application's effectiveness across various educational and academic settings.

3.3.1.1 Test Bari

During the test, the application was used by a total of 147 users. Among them, 86 completed the test, while 18 completed the questionnaire, a total of 60%. Uniquely, 28 users answered 20% of the questionnaire questions.

With the data collected, it was possible to assess the presence of Internet addiction using the CIUS-7 test. The CIUS-7 questionnaire was completed in full by 100 users. Analysis of the results revealed that three users, identified by the codes *"f0-bD5jOQ_2xuFB8SpBCZx," "c0KzsXDaSNWe6d1GNah-Wdy,"* and *"enTFBVMXSl6B5LRtgwKWdZ,"* were at risk of internet addiction. Inherent to the categorization of Bullying, the situation is described in Table 14. Where 0*: RANGE OF NORMALITY; 1: RANGE OF RISK; 2: RANGE OF PATHOLOGY*. Table 14 shows the frequencies of users who showed profiling by the questionnaire.

	Bullying _Victimization	**Bullying_Bully**
Range 1	15	12
Range 2	18	3
Range 0	67	85

Table 14 - Classification Bullying Test

Inherent to the Cyberbullying categorization, the situation is described in Table 8. Where 0: RANGE OF NORMALITY; 1: RANGE OF RISK; 2: RANGE OF PATHOLOGY. Table 15 shows the frequencies of users who showed profiling by the questionnaire.

	Cyberbullying _Victimization	**Cyberbullying _Bully**
Range 1	11	2
Range 2	1	1
Range 0	88	97

Table 15 - Classification Cyberbullying Test

The preceding tables played a significant role in categorizing into four classes, which made it possible to summarize the importance of the data collection and the questionnaire used. Table 16 provides a summary of the categorization assigned to each user.

Bullying _Victimization	Bullying_Bully	Cyberbullying _Victimization	Cyberbullying _ Cyberbully
c7pU0GkHRpudprE3y58jfo	c7pU0GkHRpudprE3y58jfo		
c30zVB50S02kI-ECEJaADq		c30zVB50S02kI-ECEJaADq	
c784pxfqRjKSsHVhHg-SEr			
c7pU0GkHRpudprE3y58jfo	c7pU0GkHRpudprE3y58jfo		
cd8QI45wT4-SkB6QYfakOe	cd8QI45wT4-SkB6QYfakOe	cd8QI45wT4-SkB6QYfakOe	
cTgjumLiRqmRscOM1YmYcZ	cTgjumLiRqmRscOM1YmYcZ	cTgjumLiRqmRscOM1YmYcZ	cTgjumLiRqmRscOM1YmYcZ
d20mrG7wQaOOyDcZFBywDY	d20mrG7wQaOOyDcZFBywDY	d20mrG7wQaOOyDcZFBywDY	
d2ElosSyQfaWcxQMfkEnDP			
		d8O1QFEKRjK3mznv00Bn8V	
da8h-ATwR7ORaTYNnQuFuN	da8h-ATwR7ORaTYNnQuFuN	da8h-ATwR7ORaTYNnQuFuN	
			dfj0xvXoTtW85FLanLAuy1
	dLgup2z4QmKioMKO-id7xR		
dQWWrr5USX2twTxqAtw7Px	dQWWrr5USX2twTxqAtw7Px		
eg9I0HgwQvuyU3isHsxOth			
eGysCKyzRy-579VzjPalKw			
ejn3SuECTWKa0qmlyDFllC	ejn3SuECTWKa0qmlyDFllC		
em1enZs4S1qnfUwta9vI6w			
	eRMmfV0TR7azD5v6PIrVnI		
	eT5JQaimThSU2VxzHe8W4W		
f0-bD5jOQ_2xuFB8SpBCZx			
f8NIBGV2SeCR48QgAXutgw		f8NIBGV2SeCR48QgAXutgw	
ffqakhWJRmSCv5lZcerpOF		ffqakhWJRmSCv5lZcerpOF	
fj9zb-KxQROpkPyl0lahWn			
fjAT0CMoRT6wjb2sV-k2Jo			
ftbfbvjLQ_aKtM1iltVWp-			
fymjgb9bRMyH-4QcvCe82z			
	c0KzsXDaSNWe6d1GNahWdy		
cF9s66XBSd2Omilp5dBIYU			
cMF5-g3QQnufXzbwv3Ont2			
cmwnwmBuTPKjbqUTJxRcYM			
d3iJOKLrRJuRrM9YnH_bpc			
		dFm7dUHLQM2_nFS3tE9oJ0	
	dK3kHozmTZiYuwdzEK0Iny		
dnSbCmwaRhiNkzYAHaMQ4t			
dWNbEiDFRIOS8Dt7gWnr5a			
		dxn40pKtQNm7ohqghq0g0D	
eeFWI3b4RwKPb5Q_4LF_h3			
			enTFBVMXSl6B5LRtgwKWdZ
	eoi1FCXSTbut2UViavoJKb		
eOqLVJCxTu2E7ZGTkmYt5O			
	eRFf9k3kSl2gDcQt10rXf1		
esPL90L2QqqL57IbtNe9dE		esPL90L2QqqL57IbtNe9dE	
eYEv4PIzSDS7lnbZnCd9t4		eYEv4PIzSDS7lnbZnCd9t4	
f-OAoY4CRpGTNanxe0h-Sg			
fjcDWvlKRm6zUbFrs4hfLN			
33	15	12	3

Table 16 - Test Categorization

3.3.1.1.2 Test Avellino

In the Test, the application was used by 110 users. Of them, 101 completed the entire test, and 3 completed 60% of the questionnaire. Only 6 performed 20% of the questionnaire. Inherent to the categorization of Bullying, the situation is described in Table 17. Where *0: RANGE OF NORMALITY; 1: RANGE OF RISK; 2: RANGE OF PATHOLOGY*. Table 17 shows the frequencies of users who showed profiling by the questionnaire.

	Bullying_Victimization	Bullying_Bully
Range 1	20	15
Range 2	16	19
Range 0	70	72

Table 17 - Classification Bullying Test

Inherent to the Cyberbullying categorization, the situation is described in Table 18. Where 0: RANGE OF NORMALITY; 1: RANGE OF RISK; 2: RANGE OF PATHOLOGY. Table 18 shows the frequencies of users who showed profiling by the questionnaire.

	Cyberbullying _Victimization	**Cyberbullying _ Cyberbully**
Range 1	7	7
Range 2	8	5
Range 0	91	94

Table 18 - Classification Cyberbullying Test

The previous tables were helpful with the purpose of drawing up a four-class categorization that summarizes the importance of the data collection and the applied questionnaire. Table 19 provides a summary of the categorization assigned to each user.

Bullying _Victimization	**Bullying_Bully**	**Cyberbullying _Victimization**	**Cyberbullying_Cyberbully**
	1674635247833711449		
			1674635276277846466
1674635317377321026	1674635317377321026		1674635317377321026
1674635325717651366	1674635325717651366		
1674635343560031444			
1674635498706984956		1674635498706984956	1674635498706984956
	1674635528576688915		
	1674635772763220701		
1674635774015188551			
	1674635786408981670		
	1674635786504718530		
1674635804517319192	1674635804517319192		1674635804517319192
1674635829986506061	1674635829986506061	1674635829986506061	1674635829986506061
1674635901132525174			
	1674642579544541333		
	1674642583089132740		
	1674642660174615455		
1674642663551046031			
1674642665659445201			
	1674642666865282140		
1674642671444960160		1674642671444960160	
1674642671490395141	1674642671490395141		
1674642673998323179			
1674642674530173340		1674642674530173340	
1674642675164041086			
1674642684238087391			
1674642702603623449		1674642702603623449	
1674642702808803699	1674642702808803699		1674642702808803699
1674642722760466701	1674642722760466701	1674642722760466701	1674642722760466701
1674642771566742377			
1674642773315709499	1674642773315709499	1674642773315709499	1674642773315709499
	1674642837885747062	1674642837885747062	
	cOIY6xpUSISPNeUjkc9hZo		
1674549756673341250	1674549756673341250		
1674549962411363771			
1674550007826776840	1674550007826776840	1674550007826776840	1674550007826776840
	1674550042014647786		
1674550082150372256	1674550082150372256	1674550082150372256	1674550082150372256
1674550082343854678	1674550082343854678	1674550082343854678	
1674550086847052518			
		1674550092505941503	
1674550111080523390			
1674555882761638634	1674555882761638634	1674555882761638634	1674555882761638634
	1674555933384001637		
	1674555933996364386	1674555933996364386	
	1674555934437381849		
1674555957247599543			
1674556010761941516	1674556010761941516		
	1674556017445407277		
	1674556043250710372		
1674556044624313054	1674556044624313054		1674556044624313054

1674556078592612404			
1674556098659072154		1674556098659072154	
1674556102686728090			
1674556104127511427	1674556104127511427		
	dbDupYycR_augTLkmvGnnI		
36	34	15	12

Table 19 - Test Categorization

3.3.1.1.3 Test Cagliari

During the test, the application was used by 124 users. Among them, 113 completed the test, while 2 completed the questionnaire, 60% of the total. Only 9 users answered 20% of the questionnaire questions.

The results of the categorization of bullying are shown in Table 20. In the table, 0: RANGE OF NORMALITY; 1: RANGE OF RISK; 2: RANGE OF PATHOLOGY. Table 20 presents the frequencies of profiled users according to the questionnaire.

	Bullying _Victimization	**Bullying_Bully**
Range 1	13	14
Range 2	12	3
Range 0	97	105

Table 20 - Classification Bullying Test

Inherent to the Cyberbullying categorization, the situation is described in Table 21. Where *0: RANGE OF NORMALITY; 1: RANGE OF RISK; 2: RANGE OF PATHOLOGY*. Table 21 shows the frequencies of users who showed profiling by the questionnaire.

	Cyberbullying _Victimization	**Cyberbullying _ Cyberbully**
Range 1	11	5
Range 2	5	3
Range 0	106	114

Table 21 - Classification Cyberbullying Test

The previous tables were helpful with the purpose of drawing up a four-class categorization that summarizes the importance of the data collection and the applied questionnaire.

Table 22 summarizes the categorization given for each user.

Bullying _Victimization	**Bullying_Bully**	**Cyberbullying _Victimization**	**Cyberbullying _ Cyberbully**
1676620017425216196	1676620017425216196		
1676620042634779389			

1676620049465375119		1676620049465375119	
	1676620072373727560		
	1676620092377040559		
	1676620094844484546	1676620094844484546	1676620094844484546
1676620095596613100		1676620095596613100	
1676620095848200935			
	1676620101901194421		
	1676620107332795302		
1676620977756108043			
1676621527810557199	1676621527810557199	1676621527810557199	1676621527810557199
1676621537784187794			
1676621547984800481			
	1676621592349895583		
	1676621592660817041		
1676621612260385684		1676621612260385684	
1676621617498670012			
	1676621619172859751		
1676621633112049967			
1676621633116150592			
1676621633539062429			
1676621645156300514		1676621645156300514	
		1676630838217969951	
		1676630848094489204	1676630848094489204
			1676630854509622963
	1676630855816334862		1676630855816334862
	1676630863361956618		1676630863361956618
1676630888905200214			
1676631352494818058			
1676631364719308264	1676631364719308264	1676631364719308264	1676631364719308264
1676631376296518128			
1676631385065305920	1676631385065305920	1676631385065305920	
		1676631418937906448	
1676631422906039145	1676631422906039145	1676631422906039145	1676631422906039145
1676631454981421283			
	1676631471156839911		
	1676631471469575020		
		1676631487192086212	
		1676631487430570009	
1676631668647558593		1676631668647558593	
1676632063806416443		1676632063806416443	
dZ0R9ehjQmW9AvDn1hfsB8			
25	17	16	8

Table 22 – Test Categorization

3.3.1.1.4 Test Marigliano-Napoli

During the test, the application was used by 101 users. Among them, 75 fully completed the test, while 18 completed the questionnaire to 60% of the total. The remaining 27 users answered 20 percent of the questionnaire questions. With the data collected, it was possible to assess the presence of Internet addiction using the CIUS-7 test. The CIUS-7 questionnaire was completed in full by 75 users. Analysis of the results revealed that three users, identified by the codes *"1684223180611345225," "1684228773172922300,"* and *"1684228862761389097,"* were at risk for Internet addiction. Inherent to the categorization of Bullying, the situation is described in Table 23. Where *0: RANGE OF NORMALITY; 1: RANGE OF RISK; 2: RANGE OF PATHOLOGY.* Table 23 shows the frequencies of users who showed profiling by the questionnaire.

	Bullying _Victimization	**Bullying_Bully**
Range 1	3	12
Range 2	4	6
Range 0	68	57

Table 23 - Classification Bullying Test

Inherent to the Cyberbullying categorization, the situation is described in Table 23. Where *0: RANGE OF NORMALITY; 1: RANGE OF RISK; 2: RANGE OF PATHOLOGY.*
Table *24* shows the frequencies of users who showed profiling by the questionnaire.

	Cyberbullying _Victimization	Cyberbullying_Cyberbully
Range 1	4	4
Range 2	4	3
Range 0	67	68

Table 24 - Classification Cyberbullying Test

The previous tables were helpful with the purpose of drawing up a four-class categorization that summarizes the importance of the data collection and the applied questionnaire. Table 25 summarizes the categorization given for each user.

Bullying _Victimization	Bullying_Bully	Cyberbullying _Victimization	Cyberbullying _ Cyberbully
	1684228835734740694		
	1684223185198898435		
1684223231035114647			
1684223743408297353			
		1684228798407020084	
	1684228830799046788		1684228830799046788
1684228839029160379	1684228839029160379		
1684228839277401354			
	1684228842744428529		
	1684228843067565378		
		1684228847141219014	
	1684228851070127801	1684228851070127801	
	1684228862761389097	1684228862761389097	1684228862761389097
	1684228863635556957		
	1684228867604395747		
		1684228908127131899	1684228908127131899
1684228926691664938	1684228926691664938		1684228926691664938
1684228935765900431	1684228935765900431	1684228935765900431	1684228935765900431
	1684228963550531215		
	1684228973432274955		
1684229007768004013	1684229007768004013		
	1684229063877721508	1684229063877721508	1684229063877721508
	1684229280491031464	1684229280491031464	1684229280491031464
	1684229291573273499		
7	**18**	**8**	**7**

Table 25 - Test Categorization

3.3.2 Experiment Strategies – University Student

This subchapter has addressed several testing problems and key issues concerning questionnaire analysis and sensor integration in evaluating human behavioral activities.

In the chapter introduction they were mentioned, here they are simply rewritten with the attached chapter response:

1. The **First Aspect** concerns the method of completing the questionnaires. It was crucial to understand whether each user completed the questionnaire in person or if instances of remote completion or completion through intermediaries occurred. This helped assess the validity and reliability of the responses obtained. This chapter contains the answer (3.3.2.1 Fixed Tasks for Continuous Authentication via Smartphone);
2. The **Second Aspect** concerns identifying "Outliers" questions, i.e., those that produced answers outside the normal statistical pattern. Identified such questions allowed the detection of discriminatory elements based on the personality index. This chapter reported the answer (3.3.2.2 Anomaly Detection Using Smartphone Sensors for a Bullying Detection app);
3. The **Third aspect** of our study is to examine whether the personality index obtained from the questionnaires correlates with human behavioral activity. "This allowed an understanding of whether the personality traits identified through the questionnaire indicated people's actual daily

behaviors. This chapter reported the answer." (3.3.2.3 Human Activity Recognition for the Identification of Bullying and Cyberbullying Using Smartphone Sensors);

5. The **fourth aspect** is to replace the calculation of questionnaire questions using sensors only. Integrating human activity sensing technologies could offer an innovative perspective to assess behaviors accurately. This chapter reports the answer (3.3.2.4 Classification Bullying/Cyberbullying through Smartphone Sensor and a Questionnaire Application).

In the remainder of the chapter, each of these problems was examined in detail, presenting methodological and analytical approaches to solving them and improving the quality and effectiveness of the research. The analyses conducted in this chapter focused exclusively on the "Bari Test" case study regarding the adult population. This case study was the first to be extracted from the Application-Web questionnaire. In 3.3.3 Experiment Strategies – Comparison of University Student-School Student, a comparison was made with other data collected from schools concerning the underage population. All results presented in this chapter had been published as scientific articles and were presented as such.

3.3.2.1 Fixed Tasks for Continuous Authentication via Smartphone

This work retrieved Touch and smartphone sensor data from two Android smartphone applications. The two experiments were placed in the Fixed Task category. The first experimentation verified through EER and AUC curves whether the tasks designed in the first application were used to authenticate a user. The second experimentation adds a social side with a second Android application that extracts the same raw data as the previous one, and it is intended to try to understand whether the single questionnaire was completed by one person or by several people at the same time, given the issue adopted in the questionnaire namely bullying and cyberbullying [113].

3.3.2.1.1 State of the Art

Lamb et al. [114] explore the concept of observational attracts by implementing a noncontinuous, swipe-based authentication method performed within a banking application. They then experimented with Blind-Attackers and Shoulder-Surfers to see how Swipe Behaviour may or may not affect the outcome of these attacks. They use Bayesian models as classifiers, particularly Shrunk Covariance, Bayesian Multivariance Gaussian, and Infinite Gaussian Mixture, reporting ERRs ranging from 4.54% to 15.70%, depending on the classifier used.

Vaishnav et al. [115] developed a framework called *KDSmart (Keystroke Dynamics Smart)* for the Android system. It consists of three phases: Registration, Login, and Final Test Phase. Using this method, they achieved an FRR of 6.73% and a FAR of 1.66% for a resulting EER of 4.1%. *Ku et al.* [116] implemented an application allowing users to access mobile devices using a public unlock pattern. The concept is to make the pattern visible to anyone and authenticate the user using touch behaviors. The goal is to avoid observational attacks. The classifiers used are *Decision Tree, Support Vector Machine, k-nearest Neighbor, Gaussian Naive Bayes, Random Forest,* and *Logistic Regression*. An EER of 2.66% was reported for tasks performed while sitting, 3.53% on tasks performed while walking, and 5.83% on a combination of the above. *Frank et al.* [117] introduced 30 different features that could be used in continuous authentication. They monitored only simple movements such as up-down and left-right swipes. Their results were an average EER of 0% for intra-session authentications, between 2% and 3% for inter-session authentication, and 4% for all sessions after the enrollment phase. *Levi et al.* [118] identified a framework capable of creating, through behavioral feature extraction, global models capable of identifying each user in a system but avoiding sharing other users' data, as in the case of binary classifiers. Their method achieved an AUC of 91.8% and an EER of 15.6%. *Incel et al.* (Incel et al., 2021) developed a system called DAKOTA, capable of recording user behavior within a banking application. Using a binary SVM classifier with an RBF kernel, they achieved a minimum EER of 3.5% and a TPR of 90%. *Estrela et al.* [120] created a

continuous authentication system based on bio-touch for a banking application, where observational attacks are more frequent. They proposed a framework capable of achieving an EER between 9.85% and 1.88% for static verifications such as login and post-login.

- ***Reichinger et al.*** [121]. Used by [122], [123]
- ***Decision Tree (DT):*** The classifier under consideration generates a prediction tree. In the tree, some nodes evaluate a class's features. On the other hand, the tree leaves represent the decision made, i.e., to which of the two classes a given test class instance belongs [121]. Used by
- ***Random Forest (RF):*** This classifier is always based on decision trees. It contains *nDTs*. During the training phase, multiple DTs are randomly generated, while in the testing phase, the class to which a test instance belongs is the one that is returned by the various *DTs* [121]. Used by [124][125]

Generally, the performance of a biometric system is measured in terms of *FAR (False Acceptance Rate, also called FPR, False Positive Rate) and FRR (False Rejection Rate, also called FNR, False Negative Rate*). The FAR allows one to understand the percentage of test samples misplaced as positive, while the FRR shows the percentage of samples mistakenly recognized as false. These values, however, reflect the behavior of the system only for a specific acceptance threshold value, so that for generalization aims, the ROC curve (Receiver Operating Characters) has been considered, which shows the *TPR (True Positive Rate, where TPR = 1 - FNR)* about the FPR for each possible acceptance threshold value. Through this curve, another metric arises, the *AUC (Area Under the Curve),* which is the area below the ROC curve. The closer this area is to 1, the better the model's performance. This is because if the curve is projected upward to the left, it means that with a low threshold, it could have low FPR and high TPR. Another helpful metric is EER (Equal Error Rate), the value where FAR and FRR are equal. This is a widely used metric to compare results with other studies.

3.3.2.1.2 Experimental Setup

The workflow is structured as follows (Figure 33):

- *Android Application*: Android Application 1 is helpful for the first experiment. In the first Android application, the three tasks, namely, tap, swipe, and zoom, are implemented; Android application 2 is helpful for the second experimentation. In the second Android application, the questionnaire performed by the 89 users is implemented;
- *Data Extraction and Feature Extraction*: The Data Extraction phase transforms the raw data obtained from the mobile device into a table format-oriented data manipulation. In addition, different features were extracted for the two datasets, described later;
- *Classification*: The most popular state-of-the-art classifiers are used in this phase.

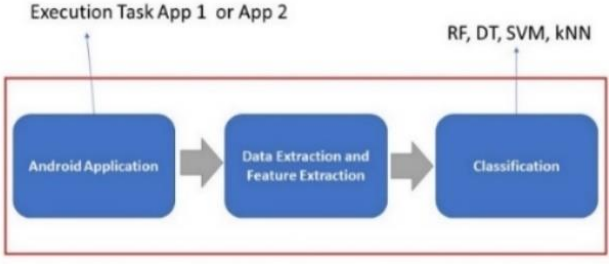


Figure 33 - Workflow

Android Application

The following raw data have been acquired for the two Datasets:

- *ACTION_DOWN_<TASK_NAME>:* indicates the start of an interaction with the screen and, in its values, returns the x and y coordinates, the press, the task number, and in the case of the tap, the coordinate of the center of the button and its size in pixels;
- *ACTION_UP_<TASK_NAME >:* task number and x and y coordinates of end of movement;

- *ACTION_MOVE_<NAME- TASK >:* task number, finger ID, speed of movement on as-the x and y, coordinates of the instant of movement, pressure, and surface;
- *ACTION_POINTER_DOWN_<NAME- TASK >:* same as ACTION_DOWN but with finger id;
- *ACTION_POINTER_UP_<NAME- TASK >:* like ACTION_UP but with the finger id;
- ACCELEROMETER: value read from the accelerometer in terms of x, y, and z coordinates;
- GYROSCOPE: value read from the gyroscope in terms of x, y, and z coordinates;
- MAGNETOMETER: value read from the magnetometer regarding x, y, z coordinates.

The first Dataset Experiment is created with this data capture. The data capture was done through the use of an application that uses a background service called *KeyloggerService (which is responsible for keeping track of all open apps, any text string keystrokes, selected menu items, the raw data from embedded sensors, etc.)* and an *AccessibilityService (created to assist users with disabilities, this service receives a call when AccessibilityEvents are created, which in turn allow the user to capture any interaction with the interface: key press, text entry, etc.)*.

When the app opens, the user can give the necessary permissions to allow the background services to function correctly. Once permissions have been obtained, the test can be started by clicking the Next button, and the user is immediately informed about how the experiment is being carried out. Then, the training phase begins, in which the user tries out all the tasks as they must perform them in the experiment. As mentioned above, the sequence of tasks is 15 taps, 4 slides, and 4 zooms. When the actual phase begins, the user is carefully notified. Once the test is finished, the user can click on the "Quit" button, which terminates the test. The second dataset experiment was created with this similar data acquisition but with a different Android application. This Android application implements a questionnaire designed to understand people's attitudes toward bullying and cyberbullying and the touch and sensor functions of the smartphone were extracted.

3.3.2.1.3 Feature Extraction

In the first Experiment Dataset, the features were computed starting from the raw data previously described. Each task has its own engineered features, as described below.

Tap:

- *Precision (px):* the distance between the tap and the center of the button clicked;
- *Pressure:* the pressure of the single tap;
- *Duration (ms):* duration of the tap;
- *Acceleration (m/s^2):* acceleration read at the first available instant following the tap;
- *Rotation (rad/s):* rotation read at the first available instant following touch;
- *Magnetic Field (Asp/m): magnetic field strength is* read at the first instant successive to the touch.

Swipe:

- *Precision (px):* the average distance from the center line of the slide concerning the swipe affected by the user;
- *AvgXSpeed (px/ms):* average speed on the x-axis at which the swipe is performed;
- *AvgYSpeed (px/ms):* average speed on the y-axis with which the swipe is affected;
- *AvgPressure:* average pressure with which the swipe is performed;
- *xMedianSpeedOfLast5Points (px/ms)* [117]*:* average speed of the last 5 points of the swipe on the x-axis. This feature and the next one are very significant as they give insight into how a user finishes such a gesture, whether quickly and coarsely or slowly and accurately;
- *yMedianSpeedOfLast5Points (px/ms)*[117]*:* average speed of the last 5 points of the swipe on the y-axis;
- *Duration (ms):* duration of the swipe;

- *AvgAcceleration (m/s^2):* average acceleration with which the swipe is performed;
- *AvgRotation (rad/s):* average rotation with which the swipe is performed;
- *AvgMagneticField (Asp/m):* average magnetic field strength with which the swipe is affected.

Zoom:
- *CenterDistanceXf0 (px):* distance from the center of the finger before finishing the zoom;
- *AvgXSpeedf0 (px/ms):* average speed on the x-axis with which the zoom is effected for the finger;
- *AvgYSpeedf0 (px/ms):* average speed on the y-axis with which zooming is performed for the finger;
- *AvgPressuref0 (px/ms):* average pressure with which zooming is performed for the finger;
- *xMedianSpeedOfLast5Pointsf0 (px/ms)* [117]*:* average speed of the last 5 zoom points on the x-axis for the finger in question;
- *yMedianSpeedOfLast5Pointsf0 (px/ms)* [117]*:* average speed of the last 5 points of the zoom on the y-axis for the finger;
- *Durationf0:* duration of motion for the finger;
- *AvgAccelerationf0 (m/s2):* average acceleration with which the zoom is performed for the finger in question;
- *AvgRotationf0 (rad/s):* average rotation with which zooming is performed for the finger;
- *AvgMagneticFieldf0 (Asp/m):* average magnetic field strength with which zooming is affected for the finger;
- Where f_0 refers to the first finger that touched the screen, this is repeated for the second finger, labeled f_1.
- *AllTask*: In this case, all tasks are unionized. For each user, all instances of the various tasks should be entered, i.e., all instances of the slide (24 in total), plus all instances of the zoom (also 24), plus 24 instances of the tap, to be chosen randomly from the 90 available.

After the experimentation:
- *The data for 15 taps × 6 users × 6 intakes = 540 taps;*
- *The data related to 4 slides × 6 users × 6 intakes = 140 slides;*
- *The data for 4 zoom-ins × 6 users × 6 intakes = 140 zoom-ins.*

In the second Dataset, Experiments are extracted this Feature Extraction is:
- *Coordinates (x, y) of tap:* tap start coordinates;
- *Pressure*: the pressure of the tap:
- *Surface (mm^2):* surface area of the finger;
- *Duration (ms):* duration of the tap;
- *Acceleration (m/s^2):* acceleration recorded now immediately following the tap;
- *Rotation (rad/s):* rotation recorded now immediately following the tap;
- *MagneticField (Asp/m):* magnetic field recorded now immediately following the tap;

More features have not been calculated because data are called generic gestures, which still need to be discovered. In addition, information such as the initial coordinates is also used [126]. The following detail is like the two experiments: Being features in different ranges, data normalization has been performed by adopting the *Min Max scaler* [2], [13]. This scaler brings all the data into the range of [0,1], and to do so, it sets the feature with the most significant value equal to 1 and the feature with the smallest value equal to 0.

All these have been tested by machine learning algorithms defined in the chapter "Methods."

3.3.2.1.4 Results

This section reports results obtained in the different experiments and the graphs inherent in the experiments performed, each illustration has the following abbreviations (*lx: far left-graph, clx: center-left graph, crx: center-right graph, rx: far-right graph*).

First Experiment

The results were computed considering different gestures*: tap, swipe, zoom-in*, and finally, a combination of the three tasks. Note that each curve represents a user. For each task, a distinction is drawn based on the trained model, and, in the observations, the best performance is considered, thus the highest AUC and the lowest EER.

Tap

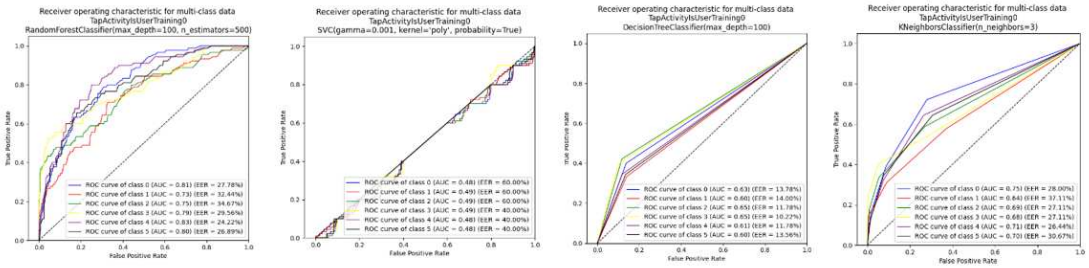


Figure 34 - Tap task (lx RF, clx SVM, crx DT, rx Knn)

Figure 34 (lx) shows a curve just above the diagonal with a maximum AUC of 83% and an EER of 24.22%. In Figure 34 (clx), the SVM is the worst-performing classifier, with an ROC curve that medially lies below the diagonal, an AUC of 49%, and an EER of 40%. Figure 34 (crx) shows fair but not reasonable results, with relatively low EER (10.22%) but an AUC of 60%. In Figure 34 (rx), on the other hand, there are similar results to DT, with a maximum AUC of 75% and an EER of 26.44%. As was expected, the Tap task is the one that performed worst (see successive results for more comparison), as the gesture in question has little discrimination. The classifier that performed the highest was Random Forest, with a maximum AUC of 83% and a minimum EER of 24.22%. The poor performance is also evident from the curve trend, which is much shifted toward the diagonal of the quadrant.

Swipe

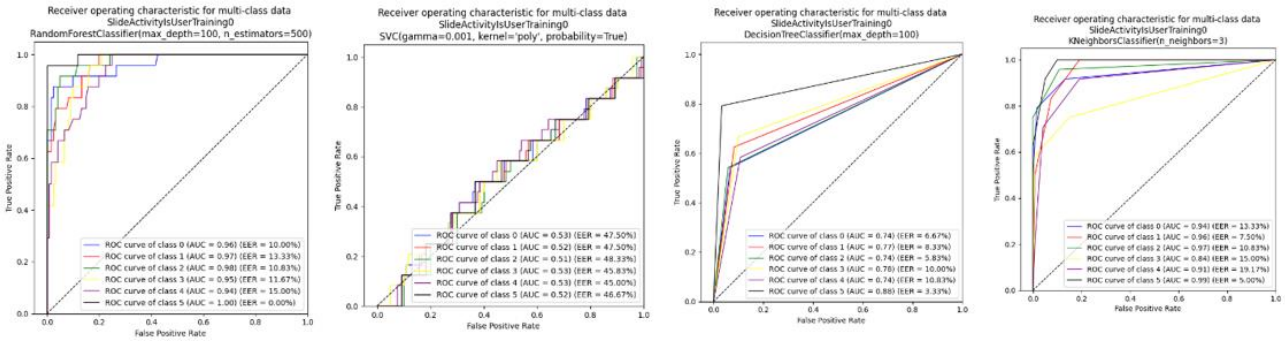


Figure 35 - Swipe task (lx RF, clx SVM, crx DT, rx Knn)

Figure 35 (lx) reports a maximum AUC of 100% and a minimum EER of 0%. Also, in Figure 35 (clx), the SVM performed worse, obtaining an AUC of 53% and an EER of 45%. Figure 35 (crx) shows clear improvements over the tap, with an AUC of 88% and EER of 3.33%. In Figure 35 (rx), the situation is also better than the tapping task, with an AUC of 99% and an EER of 5%. Once again, RF is the best, and in this case, a projected curve was seen in the upper left corner, which was just the

expected result. The swipe had higher performance than the tap because, as a movement, it allows using more characteristics that allow for better discrimination. With RF, an AUC of 100% and an EER of 0% were obtained. K-NN also performed very well, with an AUC of 99% and an EER of 5%.

Zoom-in

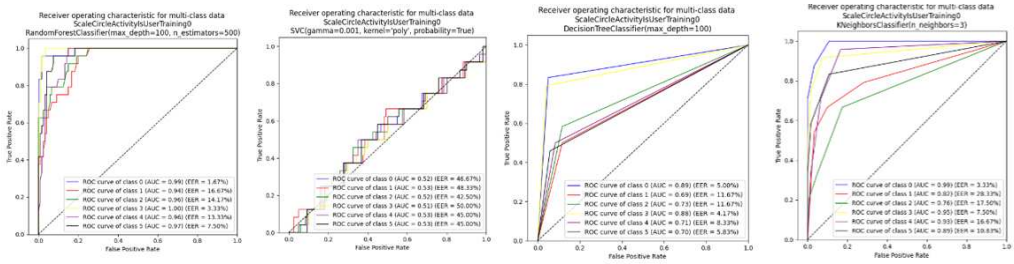


Figure 36 - Zoom-in task (lx RF, clx SVM, crx DT, rx Knn)

In Figure 36 (lx), excellent performance was obtained with an AUC of 100% and an EER of 3.33%. Also, in Figure 36 (clx), the SVM found an AUC of 53% and EER of 42.5%. In Figure 36 (crx), other discrete results were noted for this DT with an AUC of 89% and an EER of 4.17%. In Figure 36 (rx), the kNN again performs very well, with an AUC of 99% and an EER of 3.33%. The Zoom-in also performed very well, with the RF achieving an AUC of 100% and an EER of 3.33% and the kNN achieving an AUC of 99% and an EER of 3.33%.

All-tasks

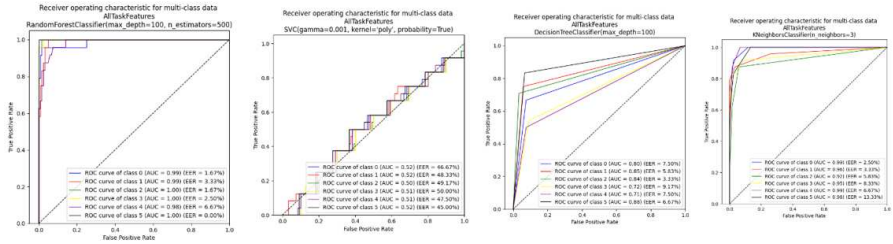


Figure 37 - All tasks (lx RF, clx SVM, crx DT, rx Knn)

In Figure 37 (lx), the RF performs to its all-time maximum with an AUC of 100% and an EER of 0%. In Figure 37 (clx), the SVM still shows problems consistent with the previous ones, with an AUC of 52% and an EER of 45%. In Figure 37 (crx), discrete results are observed with an AUC of 88% and an EER of 3.33%. In Figure 37 (rx), excellent results are also noted, with an AUC of 99% and an EER of 2.5%. The combination of the features of the various tasks led to a very high performance, which is evident in the RF with an AUC of 100% and an EER of 0%. However, the k-NN also performed very well, with an AUC of 99% and an EER of 2.5%.

<u>Second Experiment</u>

For the following experiment, the ROC curve is calculated on the two feature files obtained for each Test, each curve representing one user. For each test, a distinction was made according to the model trained. Session 1 and Session 2 cover the same questionnaire but with different users at different times. The sum of the users in Session 1 and Session 2 is 89 users.

Session1

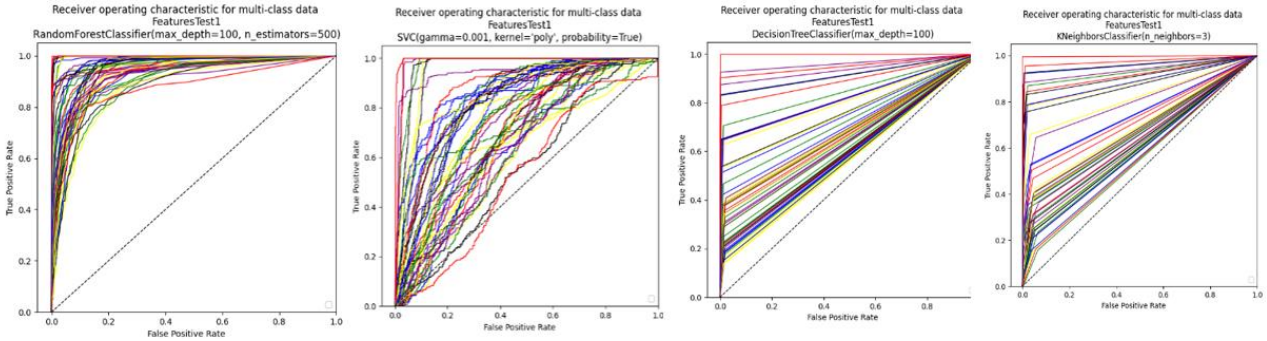


Figure 38 - Session1 (lx RF, clx SVM, crx DT, rx Knn)

In Figure 38 (lx), the RF is confirmed to be the best performing, with a maximum AUC of 100% and minimal EER of 0%. In Figure 38 (clx), the SVC performs slightly better here, but only with some users, for others, it still gets low results. In Figure 38 (crx), compared to the first experiment, the DT has lost a little performance but only in terms of AUC, as the EERs still turn out to be very low. Also, in Figure 38 (rx), the performance could be better for some users, but only in terms of AUC.

Session2

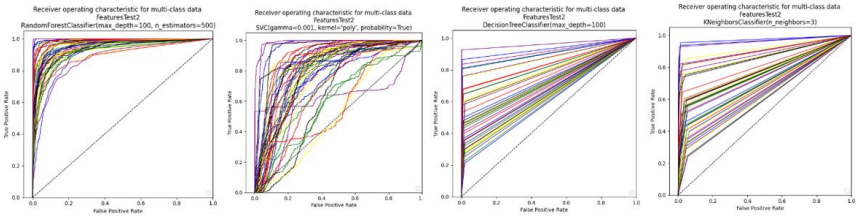


Figure 39 - Session2 (lx RF, clx SVM, crx DT, rx Knn)

In Figure 39 (lx), the same applies to RF, which is the one that performs best here as well. In Figure 39 (clx), the SVC performed worse than Session 1, which is evident from the trend of the various curves. In Figure 39 (crx), the DT remains consistent with what it accredited in Session 1. In Figure 39 (rx), the kNN is consistent with Session 1, with low average performance. With these two tests, helpful confirmations were sought for the first experiment. As usual, the RF had very high performance, while the one with lower performance was the SVM, also found in the first experiment. In addition, in the second experimentation, it could be considered that the data collected were from free tasks, and therefore, obtaining these kinds of results (with the RF a maximum AUC of 100% and a minimum EER of 0%) is more than excellent.

Model	**AUC (%)**	**EER (%)**
RandomForest	78	29.6
DecisionTree	62	12.52
K-nearest neighbors	69	29.41
SupportVectorMachine	48	50

Table 26 - First Experiment Tap task

Model	**AUC (%)**	**EER (%)**
RandomForest	***97***	***10.13***
DecisionTree	77	7.49
K-nearest neighbors	93	11.8
SupportVectorMachine	52	46.8

Table 27 - First Experiment Swipe task

Model	**AUC (%)**	**EER (%)**
RandomForest	***97***	***9.44***

DecisionTree	***77***	***7.78***
K-nearest neighbors	89	14.01
SupportVectorMachine	52	46.25

Table 28 - First Experiment Zoom-in task

Model	AUC (%)	EER (%)
RandomForest	***99***	***2.64***
DecisionTree	80	6.67
K-nearest neighbors	96	6.66
SupportVectorMachine	51	47.78

Table 29 - First Experiment AllTask

Model	AUC (%)	EER (%)
RandomForest	***95***	***10.54***
DecisionTree	***71***	***1.19***
K-nearest neighbors	72	3.81
SupportVectorMachine	78	27.46

Table 30 - Second Experiment Session1

Model	AUC (%)	EER (%)
RandomForest	***96***	***9.77***
DecisionTree	***74***	***1.21***
K-nearest neighbors	76	3.83
SupportVectorMachine	79	25.2

Table 31 - Second Experiment Session2

This chapter dealt with the problem of continuous authentication. Two Datasets were extracted from two Android applications, and two experiments were created. The goal was to find the best task for the first experiment and observe whether the uniqueness of filling out the bullying questionnaire could be inferred in the second experiment. For the first experiment, Table 26, Table 27, Table 28 and Table 29 were analyzed, summarizing the results for the tasks of tap, scroll, zoom, and their combination.

RF was always the classifier with the best performance. Furthermore, it can be concluded that the best-performing task is the combined task (*AllTask*), which sees the features of all tasks combined and achieves an average AUC of 99% and an EER of 2.64% with RF. When considered alone, the tapping task is the least performing task. For the second experiment, given the excellent performance of DecisionTree in terms of EER, it can be inferred that the users who filled out the questionnaires were always the same.

Therefore, there was most likely no device switching during the experiment (Table 30 - Table 31). In addition, the average best-performing model was RF and DT, as shown in Table 30 - Table 31. It would be appropriate to create an application allowing the models tested here to be used for user identification. Since these are binary-type models, creating a system allowing other users' data to be shared anonymously among the various authentication devices to train the various models in use would be necessary. After that, one can decide whether to have this application work continuously, whether to have the user authenticate once or continuously and entirely invisibly in the background (the second case is the most interesting). Another thing that could be done is to verify that this type of authentication is effective against observational attacks, such as those mentioned at the beginning. It would be helpful, for example, to create groups of three individuals: one is the victim, one is an attacker carrying out an observational attack, and the other is an attacker attempting to access/use the device(s) without having previously observed the user. Another exciting aspect is testing other models and seeing if they have lower performance than those already obtained.

3.3.2.2 Anomaly Detection Using Smartphone Sensors for a Bullying Detection app

This chapter is contextualized in bullying/cyberbullying, where the figure of the bully/cyberbully and that of the victim/cyber victim or the individual outside the incident manifest. In this project, data obtained from the sensors of smartphones (*after using the appropriate app*) to apply Anomaly Detection techniques (*more generically classified as Machine Learning*) helpful in detecting any abnormal behaviors adopted during the use of the app and thus the questionnaire. In other words, this work aims to analyze and detect any latent patterns within the dataset under investigation to understand any polarizing content proposed during the use of the app and identify users who exhibit strange behaviors, possibly familiar to class users. Anomaly has been defined as the sudden minor change in frequency detected by smartphone sensors. They are defined as micro-behaviors that can be intensified in one of four classes (*Bullying_Victimization, Bullying_Bullying, Cyberbullying_Victimization, Cyberbullying_Cybully, External*).

3.3.2.2.1 State of the Art
Before defining and classifying the most widely used algorithms in the field of Anomaly Detection, it is also necessary to differentiate between the anomalies they are found. Thus, one can mainly find three types of anomalies:

- *Contextual Anomalies*: where the anomaly is detected based on a particular context;
- *Collective Anomalies*: where an entire subset of the data deviates from the pattern of the global data;
- *Point or Global Anomalies*: where the value deviates entirely outside the entire data set.

Concerning the type of algorithm used, one can obtain either a label (label) to indicate whether the instance is an anomaly or a value relative to the score (score) that stands for the degree of confidence about the abnormality of the anomaly. Based on the algorithms mentioned below, a label is often used for supervised ones. In contrast, the score is usually binary for semi-supervised and unsupervised anomaly detection algorithms (-1, 1). In practice, algorithms with unsupervised approaches are of more common use, having data sets where anomalous instances are not known as a priori, and one wants to find the data that differ significantly from the norm. It should also be specified that these anomalies must be associated with their context, as the same data might be weird for some subjects and standard for others (they might change by location for environmental measurements, time of day, or age for some subjects).

Supervised Algorithms: These types of algorithms need a labeled data set. Such algorithms can detect new instances of an already known phenomenon by learning a model to classify cases of an unknown and unused dataset during training due to its features. In doing so, it can distinguish regular instances from abnormal ones. Standard algorithms include Decision Trees, K-nearest neighbors, Random Forests, and Logistic Regression. Although it is challenging to have labeled data available, they are necessary because unsupervised approaches often do not achieve the desired performance. Thus, there are semi-supervised algorithms that provide a middle ground.

Bayesian Network-based Algorithms: These algorithms are suitable for anomaly detection, as they can handle high-dimensional data that are humanly difficult to interpret. Although some anomalies are more easily detected visually, many are based on the interaction between multiple variables. Bayesian networks operate in a multi-class setting to estimate the posterior probability for instances as normal or abnormal. In the case of null possibilities, *"smoothing"* or a *"smoothing"* of the invalid probabilities for the class of anomalies by Laplace smoothing is required. Among other advantages, Bayesian networks allow continuous and discrete variables and partial missing data. They can contain data about time and outside the same model.

This algorithm arises from the assumption that regular instances thicken in a neighborhood while anomalous samples are located far away from the neighborhood closest to them. Distance and similarity can be computed in several ways; cosine similarity is common. There are several variations for detecting anomalies:

- Approach based on the calculation of the anomaly score and its definition, using relative density;
- Approach using the distance between the datum in question relative to a particular neighbor as the anomaly score;
- A hybrid approach calculates the average distance of n objects as the neighborhood distance and averages the number of neighbors within the average distance.

Other less common variants are currently being studied [127].

Clustering-based algorithms: This algorithm is used to create a model that clusters data to create a pattern later to detect anomalies in new data. Several clustering techniques can be categorized into Partitioning Clustering, Hierarchical Clustering, and Density-based Clustering. Each of them has its advantages and disadvantages. Indeed, you may have such anomalies because they do not belong to any cluster further from the centroid or directly belong to more minor, scattered clusters.

Statistical models: Mathematical models are beneficial and eventually adapted to various computational scenarios. A popular form of statistical modeling in Anomaly Detection is finding extremes of Univariate values [128]. However, according to a stochastic model, the basic idea is to exploit rules and properties of inferential statistics according to which regular instances are distributed in regions of higher probability. In comparison, anomalies are distributed in areas of lower probability. A statistical model is fitted to the data to determine the predicted behavior. This model is used on the new instances to identify those that do not belong to the model. Based on the applied statistics, instances with a low probability of being generated by the learned model are classified as outliers. While parametric techniques assume knowledge of the underlying distribution and estimate parameters from the data provided, nonparametric techniques generally do not take knowledge of the underlying distribution [129] [130]. Since the literature specifically in this proposed area is restricted, approaches related to tasks that appear to be like the ones under consideration in this paper have also been inspired. One of these is undoubtedly Intrusion Detection. Intrusion Detection [129] refers to the detection of potentially malicious activities (intrusions, penetrations, and other forms of abuse of computers) in a computer system. An intrusion consists of a discrepancy from normal system behavior; therefore, anomaly detection techniques are applicable in the intrusion detection domain. It refers to Intrusion Detection using an unsupervised approach, which is particularly interesting but at the same time challenging because a supervised approach requires inconsiderable maintenance of data labels in terms of how often they are updated. The key challenge for anomaly detection in this domain (as well as in the domain proposed in this paper) is the analysis of the huge volume of data. This aspect represents an affinity with the domain in which the work illustrated in this chapter is contextualized. The false alarms (false positives) rate is a further affinity between the two domains. Since the data amounts to millions of examples, detecting the outliers (a small fraction compared to the non-outlier data) is highly complex.

The work proposed in [131] is contextualized in the domain of Intrusion Detection in industrial control systems where, indeed, a supervised approach is very limiting as the methodologies behind intrusion attempts evolve with very high frequency; therefore, an unsupervised approach is undoubtedly helpful in detecting even minor variants of methodologies presented in the past. The algorithms used in [131].

3.3.2.2.2 Material & Methods

This section describes the dataset's structure and solves some related problems. The steps necessary to obtain the final dataset, which is later used with the various algorithms, are explained. Two main

stages, the Pre-adjustment and Post-adjustment, are performed. These two phases were necessary to clean up the dataset from the experiment with the users and detect the correct anomalies.

Pre-adjustment
Before carrying out the "pre-adjustment" phase, the test was conducted with users. The test consists of an Android application that incorporates a questionnaire re-done by psychologists. The graphical layout of the existing app is simple and essential. Every choice has been made to make the graphical interface simple and appealing to the end user and eliminate or at least drastically reduce the possibility that the user during the activities may make mistakes, have doubts about the actions to be performed, or other problems in general.

The application contemplates five basic steps:
1. Access to the app's functionality (3 screens);
2. Access to the four Cyberbullying videos (4 screens) for each video, an emotional question (4 screens);
3. Access to the Questionnaire (30+ screens) that is quick to fill out;
4. Access to the Telegram group (1 screen);
5. End of Test (1 screen).

The four Cyberbullying videos are [5]
1. *Video1Activity ("VIRTUAL ACTIONS, REAL CONSEQUENCES,"* https://www.youtube.com/watch?v=x2AxcllGLJg&t=4s&ab_channel=TabbyEUproject [5]
2. *Video2Activity ("ANYONE CAN BE ANYONE,"* https://www.youtube.com/watch?v=z3N24DpD64c [5]
3. *Video3Activity ("INTERNET = EVERYONE, FOREVER,"* https://www.youtube.com/watch?v=K31Kuc5pTXM&t=42s&ab_channel=TabbyEUproject [5]
4. *Video4Activity ("VIRTUAL VENDETTA (joke or* crime?)," https://www.youtube.com/watch?v=FpBVBwv6UQ4&ab_channel=TabbyEUproject [5]

The initial test dataset consists of 164 text files (.txt format) named "sensor_," to which the user ID was concatenated. The data for the Accelerometer, magnetometer, and gyroscope sensors are within a single file, sampled every 20 milliseconds. The sensor's name has been concatenated with the name of the task where it was sampled. Also specified at the beginning and end of each line is the date and time at which the sensor data was recorded. The data used are for four videos shown to users, where a question attached to the video is shown for each of them. The other data, however, are related to 83 general questions. These data were recorded using a client-server approach, where the client was the Android device sending its sampled data to the server. The format of the measurements is as follows: *"data" - "sensor_activity" - "x_value, y_value, z_value.".* The initial dataset contains files related to users who needed to complete the test correctly. These users, therefore, must be excluded because their data cannot be used for testing. Ninety-nine users completed the test correctly. A table duly compiled by the psychologist who processed the data is used as a ground truth for the users, where there are classifications of users in the five classes, namely *Bullying_Victimization, Bullying_Bullying, Cyberbullying_Victim, Cyberbullying_Cybully, and External.* Although most users recorded sampling consistent with expectations, the variety of smartphones used during the experiment led to unexpected sampling.

Four types of problems encountered can be listed:
1. *Duplicate sensors*: some users sampled one sensor twice as many times as others;
2. *Asynchronous sensors*: some users, due to the client-server architecture, did not send all three sensors at the same time, thus resulting in a file with a total number of sensor rows different from each other;

3. *Missing sensors*: 5/99 users registered a file with one or more missing sensors and consequently were removed from the dataset;
4. *Missing activities*: 18/99 users still need to answer the questions; for these users, the questions they could answer before the connection dropped were considered.

Post-adjustment

To overcome the problems mentioned in the previous section, two scripts were designed that could normalize the initial dataset to be compatible with the features related to the models used for experimentation. Specifically, two scripts were carried out, one related to the video tasks and attached questions and one associated with the generic question tasks, as the two tasks are different. The script related to the videos and attached questions was planned to iterate first on the tasks and then on the users, as users have yet to see all the videos. A file with all rows is initially extracted for each sensor. On the other hand, the script related to the generic questions was planned to iterate first on the users and then on the activities since it was not known a priori whether the user had answered all the questions. Again, a file with all rows is initially extracted for each sensor. The maximum number of rows among all files is calculated within the video-related script. Initially, the generated files contain raw data: that is, there are files containing dual and asynchronous sensors.

A check was made on the number of rows to:

- Identify files with several rows equal to 0 for one or more sensors related to users with missing sensors;
- Identify files with several rows for one sensor equal to approximately twice the number of rows for the other sensors (to identify files with duplicate sensors);
- Identify files with a different number of rows between them (to identify files with asynchronous sensors).
- Once the files containing "raw" data were identified, they were normalized according to the following logic:
- Files with several rows equal to 0 for one or more sensors were eliminated, and thus, the related users were excluded from experimentation;
- Files with several rows for one sensor equal to about twice the number of rows for the other sensors saw the number of rows for the doubled sensor halved by saving the second of the two instances (i.e., skipping those of odd index and saving those of even index);
- Files with different numbers of rows from each other underwent the extension of the last value for sensors with fewer than the maximum number of rows since this value is the one closest to reality.

The final output is, therefore, that of a file with a *".csv"* extension for each sensor, user, and activity, all having the same number of rows after trimming the empty rows. The three files related to the Accelerometer, magnetometer, and gyroscope sensors are merged within a single file, keeping the *".csv"* extension. Within the script related to the questions, on the other hand, given the significantly lower number of rows, it was decided to optimize the process and generate files having the correct number of rows directly, thus without empty rows. The process is like the video script, except it iterates first over the users and then over the tasks (so the files and folders are created only for the questions the user answered). All subsequent steps for sensor normalization were combined within a single script. The final output is the same. The last number of users for the post-adjustment dataset is 94 for video tasks and 93 for questions.

The models used in the experimental phase, named *"Anomaly Detection with ML models,"* are the following: *Elliptic Envelope, Isolation Forest, Local Outlier Factor, One-Class SVM,* and *Descriptive Statistics*. The first models are the standards, among the most widely used state-of-the-art models. Machine learning algorithms were run for each user for each task, and the returned predictions were

saved in a file having a ".csv" extension. The returned predictions were organized into *".csv"* files where *1 = Normals* and *-1 = Anomalies*.

This was done by sensor analysis (*Accelerometer, Magnetometer, Gyroscope*):

- *Elliptic Envelope*: contamination = 0,1
  - The "contamination" parameter indicates the portion of examples in the dataset averaged labeled as "Anomalous."
- *Isolation Forest*: contamination = 0,25, n_estimators = 100
  - The parameter "n_estimators" indicates the number of decision trees used to compose the ensemble
- *Local Outlier Factor*: contamination = 0,25, novelty = False
  - The parameter "novelty" sets up the algorithm to perform novelty detection, but this is not the case in this chapter
- *One Class SVM*: nu = 0,23
  - The "*nu*" parameter corresponds to the lower limit of the percentage of examples in the dataset used as support vectors to derive the hyperplane equation.

Given many users and activities, leading to a total number of predictions exceeding 8000, making inferences about the data or visualizing them is difficult. Therefore, descriptive statistics techniques were also used. A script was created ad hoc and used to generate tables (in the form of *.xlsx* files) containing the outputs of the algorithms' predictions in the form of the percentage of anomalies out of the total.

In these tables, it is possible to calculate the percentage of anomalies in users, activities, and user categories. Is it possible to see for which users and which tasks are funded a significantly higher than average number of anomalies (in this type of task, it is customary to find at least a minimum of anomalies; the total absence of anomalies would be equivalent to a motionless hand or a cell phone resting on a fixed surface). The designed script is created, for each model, a table with the results obtained, i.e., a number relating the anomalous rows to the total rows between 0 and 1 (where 0 is equivalent to the total absence of anomalous points and one is equivalent to the presence of only anomalous points). The users are on the table's rows, and the activities are on the columns. Once the tables for the individual models have been created with the results of the various predictions, a threshold value is set; a low, medium, and high threshold is chosen. Only the anomalies that deviate from normal are considered with a higher threshold.

The final Summary Table, on the other hand, was created manually; it is used to compare the various algorithms based on their performance and results on both users and activities; from the final table, graphs are generated that allow us to provide an overview of the behavior of the models. On the rows are the models, and on the columns are the activities and classes. Given many questions, it was decided to consider the first 20 and the last 20 based on the percentage of anomalies reported.

3.3.2.2.4 Experimentation setup

This chapter provides a perspective on how the experimentation was structured (see Figure 40):

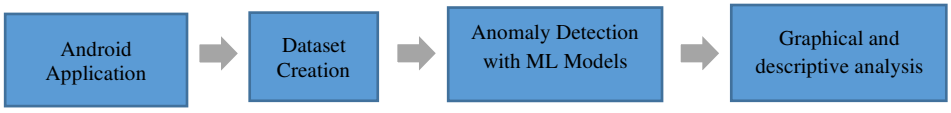


Figure 40 - The pipeline followed for the experimentation

- *Android Application:* The Android app contemplates five main and consecutive steps to be able to perform the entire test:

    - Grant the permissions that the app requires access to location, access the sensor, and keylogger services;
    - View four videos that depict situations of cyberbullying among young people through animated scenes or "cartoons." At the end of each video, an emotional question is asked to record the emotion at the end of viewing the video;
    - Fill in a questionnaire created with the collaboration of a psychologist to identify the user's attitude to the subject matter. Each question has a mandatory answer. Otherwise, it is not possible to continue;
    - Open a debate on a Telegram group created ad-hoc among the participants of the test;
    - Finish the test and uninstall the app. The uninstallation of the app also implies the cessation of all data acquisition. The app's architecture is client-server, capturing sensor values and returning them to a server.
- *Data Creation:* The phase of reacting and cleaning the dataset was divided into two phases: Pre-adjustment and Post-adjustment, as explained in the chapter *"3 Dataset";*
- *Anomaly Detection with ML models:* Standard ML models were implemented to identify anomalies by considering three different thresholds of experimental application, considering a wider to a narrower one;
- *Graphical and Descriptive Analysis*: Finally, it was possible to view the anomalies and give a visual interpretation of the psychological and behavioral situation of the actors of bullying and cyberbullying.

This section shows how to visualize the difference in the percentage of anomalies based on activities and classes of users. On the X-axis, it is possible to find the activities or classes of users. In contrast, on the Y axis, it is possible to find the percentage of anomalies, respectively, for each of the four models used. There are graphs for each threshold parameter used: low (Derived from the average value by considering everything as an anomaly), medium, and high (Captured the most experimentally relevant anomalies). A graph was also created for the first 20 questions and the last 20 questions for the high parameter. The total number of users is 94 for videos (*Bullying_Victimization=30,Bullying_Bully=13,Cyberbullying_Victimization=12,Cyberbullying_Cyberbully=3, External=53) and 93 users for quizzes (Bullying_Victimization = 30, Bullying_Bull = 13, Cyberbullying_Victimization = 12, Cyberbullying_Cyberbully = 3, External = 52*); A user can belong to more than one class.

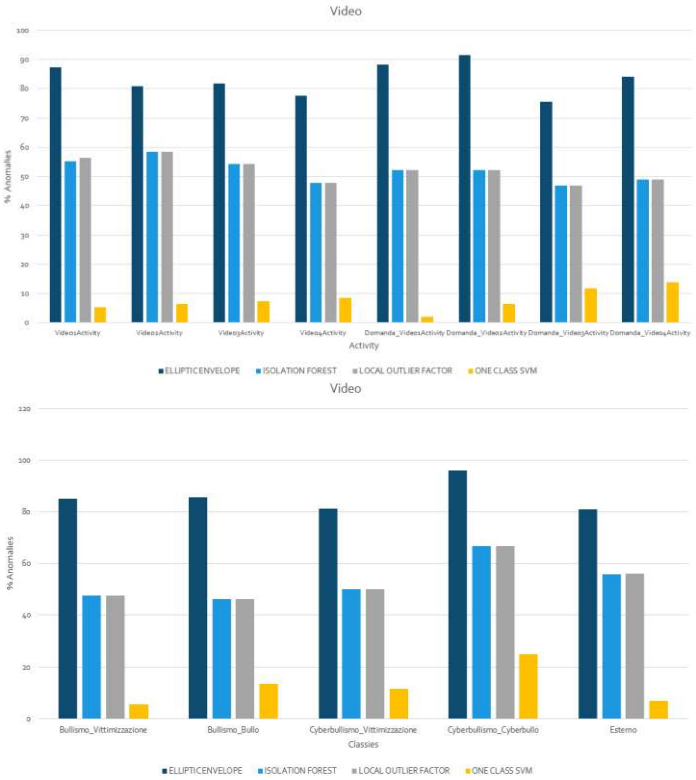


Figure 41 - Histograms for video activity (up) and video classes (down) for low threshold.

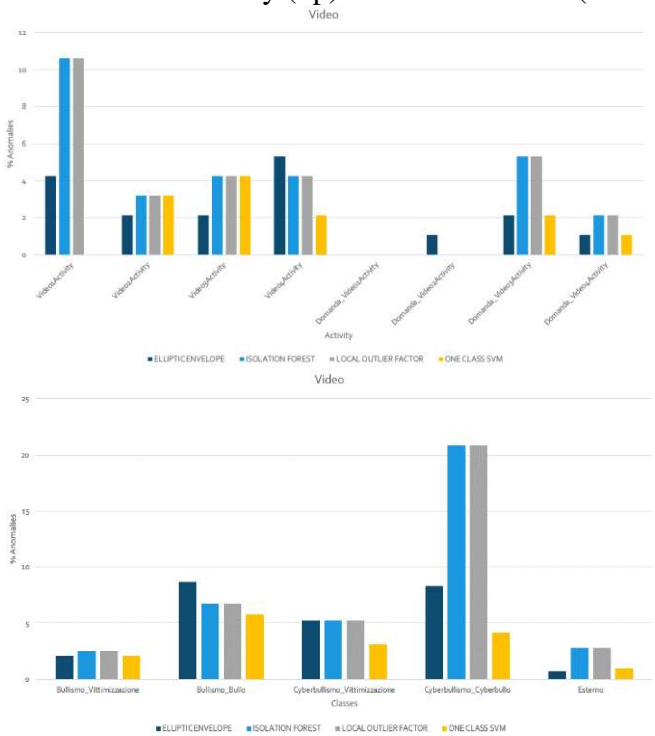


Figure 42 – Histograms for video activity (up) and video classes (down) for medium threshold

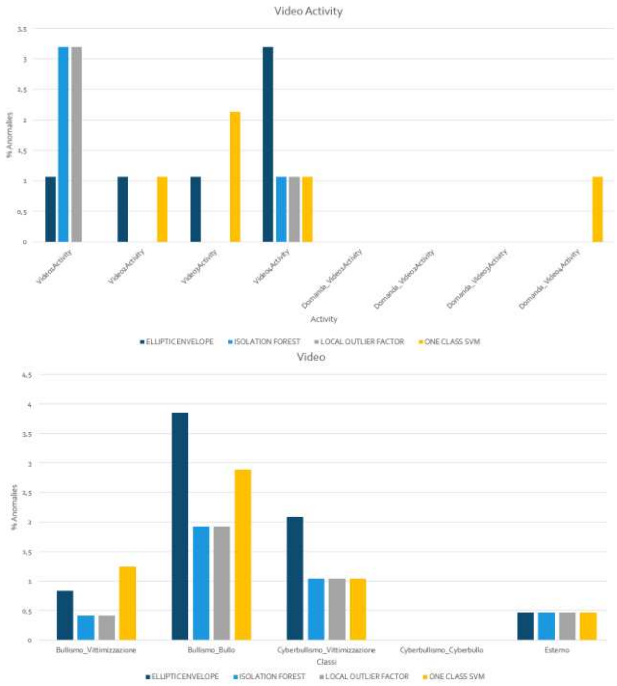


Figure 43 - Histograms for video activity (up) and video classes (down) for high threshold.

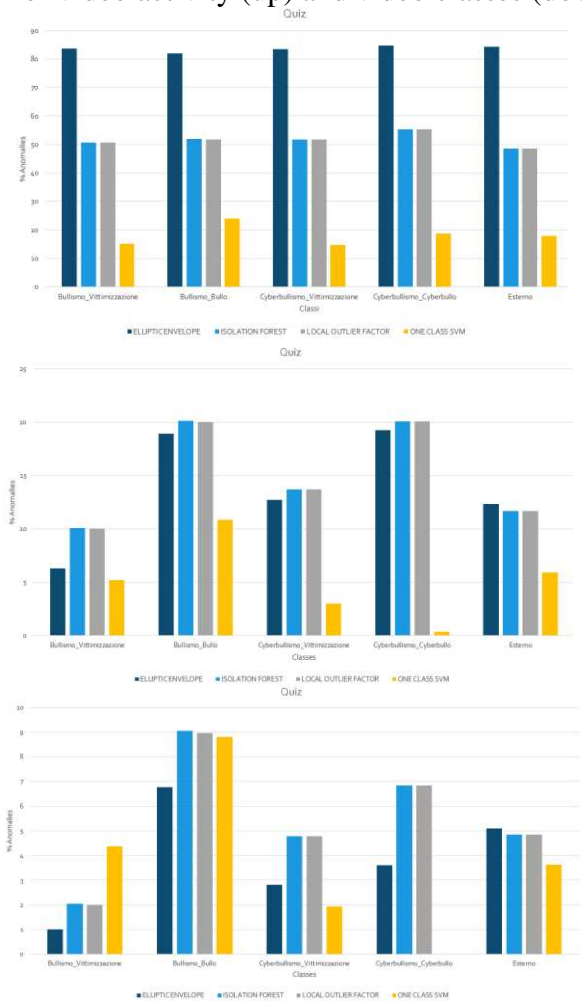


Figure 44 - Histograms for quiz classes: low threshold (up), medium threshold (center), high threshold (down).

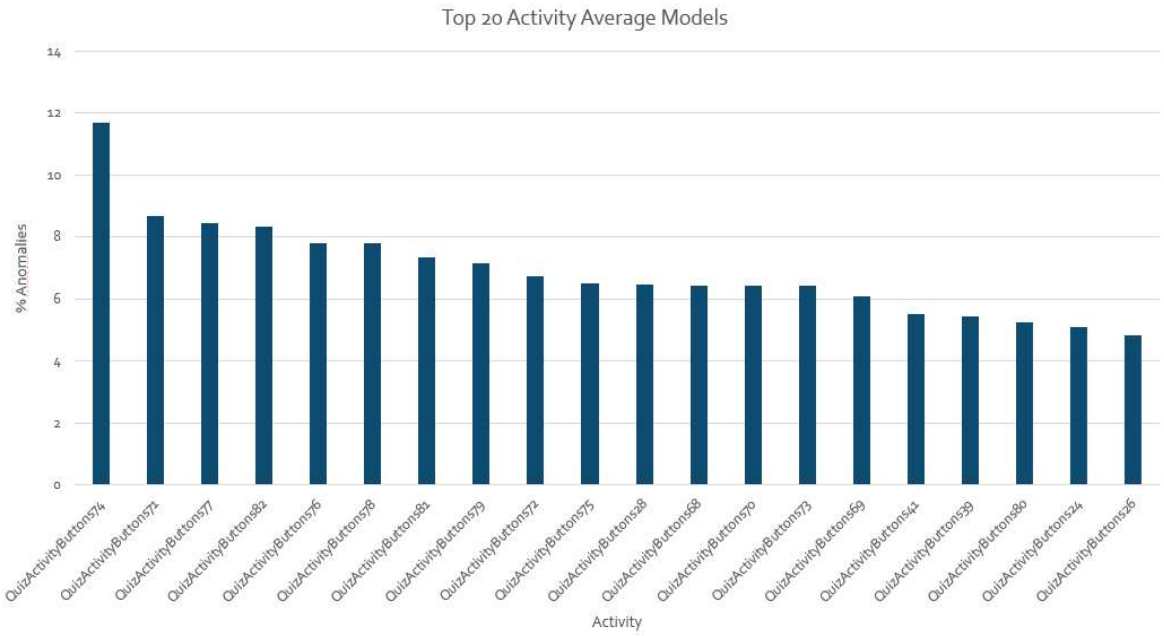


Figure 45 - Histograms for the first 20 questions with the most anomalies, averaging high threshold scores obtained between the 4 models.

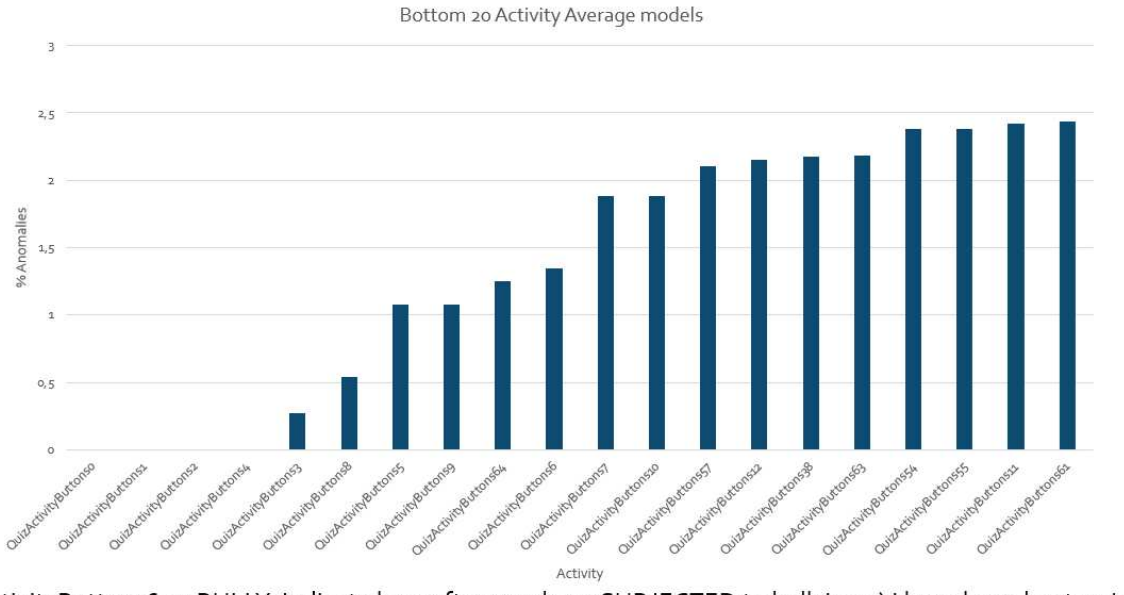


Figure 46 - Histograms for the last 20 questions with the most anomalies (therefore the first 20 with the most minor anomalies) by averaging the scores obtained between the 4 models, high threshold.

From an algorithmic point of view, the *Elliptic Envelope* is much more discriminating toward values that deviate from the expectation of the dataset's probabilistic distribution (Figure 41-Figure 44). The Local Outlier Factor focuses on outliers by looking locally at the data density relative to its neighborhood. The isolation forest cannot optimally identify anomalous points close to each other, tending to label them all anomalous or not (the total number of anomalies is like that of the LOF). The One-Class SVM differentiates abnormalities from normality more sharply, identifying fewer of them than the other models. From a social perspective, it is possible to trace computer data back to inferences from observing the data. The bully and cyberbully user classes encounter more anomalies on the

medium. Thus, a bully and a cyberbully may be active participants in the affair, making you more restless while performing this questionnaire. Even if the threshold is narrowed down, i.e., one considers a threshold other than that that best identifies the peaks of abnormality, the bully class encounters more abnormalities. The bully is an active part of the impact; it is more involved in the affair psychologically than actively. Composing a questionnaire could lead him to perform micro-behaviors identifying a belonging class. The questionnaire is divided into two macro-parts: The first phase is watching four videos about bullying and cyberbullying, and the second phase is composed of a 5-point Likert scale. The activities related to the videos encountered more anomalies than those related to the questions.

Observing Figure 45-Figure 46, the question that was part of the questionnaire, *"QuizActivityButtons74 = CYBERBULLY: Indicate how often you have DONE acts of cyberbullying 8. Send videos/photos/pictures of assaults and violence on the internet (Websites, YouTube, Facebook...) "*and *"QuizActivityButtons61 = BULLY: Indicate how often you have SUBJECTED to bullying a) I have been beaten"* were found to be the most anomalous among the other questions. The action *"I have been beaten"* elicits feelings to emphasize the fear of the situation among the victims but that among those who actively act. But sending or receiving media files also arouses a different feeling than usual and, therefore, abnormal feelings among these users.

In the chapter, data obtained from smartphone sensors were analyzed to apply Anomaly Detection techniques that help detect anomalous behaviors adopted during the use of the app. This work aims to analyze and detect any latent patterns within the data set under examination to understand any polarizing content proposed during app use and identify users who exhibit anomalous behaviors, possibly common to classes of users. Data from three smartphone sensors, namely the Accelerometer, magnetometer, and gyroscope, were analyzed for 94 users. The anomaly was the most minor abrupt change in frequency detected by smartphone sensors. They were defined as identifiable micro-behaviors in one of the four classes (*Bullying_Victimization, Bullying_Bullying, Cyberbullying_Victimization, Cyberbullying_Cyberbully, External*). The conclusions are diverse; however, comparing them with similar work is difficult because the literature in this proposed area is very young and modest.

The first type of conclusion possibly relates to the models used:

- *The Elliptic Envelope* is much more discriminating against values that deviate from the expected probabilistic distribution proper to the dataset;
- *The Local Outlier Factor* focuses on outliers by looking locally at the density of the data relative to its neighborhood: it is therefore not ideal for this application domain and specific task;
- *The Isolation Forest* cannot optimally identify anomalous points close to each other, tending to label them all anomalous or not (the total number of anomalies is like that of the LOF.);
- *The One-Class SVM* succeeds in differentiating abnormalities more sharply from normality, identifying fewer of them than the other models.

The second type of conclusion possibly relates to classes of users:

- The bully and cyberbully classes experience more anomalies on the medium than the other user classes;
- The bully class encounters more anomalies when using a higher threshold, resulting in the most discriminating;
- However, anomalies are also found among users, not in bully or victim classes; this could be due to user embarrassment in answering questions.

The third type of possible conclusion relates to the activities within the dataset:

- Activities related to videos encounter more anomalies than those related to questions;

- Few anomalies are found among the first questions (these questions are used to classify the category *Bullying_Victimization*);
- More anomalies are found among the last questions (these questions are used to classify the *Cyberbullying_Cybully* category);
- Some questions that with a high parameter do not find anomalies could be eliminated from the quiz: i.e., *Do you know all your friends you have in your Internet profiles? From the beginning of school to now, have you ever brought your smartphone* (*With Internet access using social networks such as Facebook, WhatsApp, etc...) to class to do a non-school activity? From the start of school to now, how many times have you used your smartphone to connect to the Internet while in class to do a non-school activity?*, *From the start of school to now, how many times have you used your smartphone to connect to the Internet while in class to do a non-school activity? Typically, how many hours a day do you actively spend online (browsing, chatting, attending social networks, writing emails, etc.) excluding DAD activities;*
- The questions that are part of the questionnaire, *"QuizActivityButtons74 = CYBERBULLY: Indicate how often you have DONE acts of cyberbullying 8. Send videos/photos/pictures of assaults and violence on the internet (Websites, YouTube, Facebook "*e *"QuizActivityButtons61 = BULLY: Indicate how often you have SUBJECTED to bullying a) I have been beaten"*, were found to have a higher percentage of anomalies.

Regarding future developments, it is helpful to conduct a study on hyperparameters (tuning), avoiding automated Machine Learning approaches, since this domain inherently exploits unsupervised methodologies. Use feature extraction techniques and do not necessarily use raw data. Use ensemble learning methods by combining the models used or integrating others, and finally, compare the results obtained having improved/refined the app's content.

3.3.2.3 Human Activity Recognition for the Identification of Bullying and Cyberbullying Using Smartphone Sensors

This work discusses a new smartphone methodology that combines the final label elicited from the cyberbullying/bullying questionnaire (*Bully, Cyberbully, Bullying Victim, and Cyberbullying Victim*) and the human activity performed (*Human Activity Recognition*). At the same time, the individual fills out the questionnaire. In addition, for the scientific aspect, a new Dataset named " *Dataset HAR Uniba* " inherent to Human Activity Recognition (containing the following activities, namely, walking, running, jumping, and sitting, in addition to other scientific papers the "fall") was introduced. The problem focuses on finding an existing behavioral correlation between the activity/behavior and the individual's personality index (label). At the same time, it fills out a simple questionnaire dealing with sensitive social issues. Therefore, HAR methodology was used differently from other papers, which only try to discriminate activities in the dataset they consider as best they can.

The chapter highlights these methodological aspects:

- In this chapter, activity recognition is performed, including walking, running, jumping, sitting, and falling. Action recognition in Human Activity Recognition is treated extensively in line with the state of the art. Additionally, fall detection is recognized, which is often not considered in many papers. Compared to the state of the art, recognizing the primary actions of walking, running, jumping, and sitting aligns with the standards mentioned in the chapter.
- Mentioning the first problem, a dataset "*Dataset HAR Uniba*" that detects actions (walking, running, jumping, sitting, falling) was created by our research team in the lab. This dataset was compared with another UniMiB SHAR state-of-the-art dataset via Deep Learning algorithms, among the best known in the literature.
- After identifying the best model that identifies Human Activity Recognition, a new smartphone methodology was proposed that combines the final label (personality index) triggered by the

cyberbullying/bullying questionnaire (Bully, Cyberbully, Bullying Victim, and Cyberbullying Victim) and the human activity performed (Human Activity Recognition). At the same time, the individual fills out the questionnaire. This smartphone methodology elicited accelerometer acquisition while 52 users for test 1 and 48 users for test 2 filled out a questionnaire drafted by psychologists.

The submitted questionnaire deals with questions about the active and passive actions of bullying and cyberbullying drafted by psychologists. Through the final labels of the questionnaire, i.e., "Bully, cyberbully, victim of Bullying, and victim of cyberbullying," conclusions were drawn by comparing the received label of the questionnaire with the activity performed by the user. Other than just sitting, restless behavior can characterize an identifying behavior for the category. As the individual filled out the questionnaire, the app extrapolated data from the sensors, specifically the accelerometer. A standard questionnaire could not capture users' movements to extract new behavioral knowledge.

*3.3.2.3.*1 State of the Art

This chapter reviewed the most exciting studies considered. The research team dealt with projects related to cyberbullying and machine learning. Specific papers addressing Human Activity Recognition with Cyberbullying were few. Papers strongly related to Human Activity Recognition but not inherent to Cyberbullying were mentioned. The Human Activity Recognition approach was used to identify possible activity-related anomalies while taking a psychological questionnaire, which could identify the individual's attitude inherent in bullying/cyberbullying.

Many studies do not adopt a Feature Extraction and Feature Selection phase; the following are examples of this type of study. The study by *Minarno et al.*[132]. In the study by *Cho et al.*[133] both approaches are tried, using only accelerometer data. Three public datasets are chosen, specifically *UMAFall* [134], *UniMiB SHAR* [41] and *SisFall*, a publicly available dataset not considered because it uses only Inertial Measurement Units to collect data. Of these three datasets, Cho et al. perform a Leave One Subject Out Cross Validation using a Convolutional Neural Network as the classifier, but with various features. They try various combinations between raw data, *Signal Vector Magnitude, Singular Value Decomposition, Kernel Principal Component Analysis, and Sparse Principal Component Analysis*. The results show that using Signal Vector Magnitude[135] alone leads to severe feature lowering. Still, accompanying it with the raw signal features increases its performance, at least in the case of UMAFall and UniMiB. The maximum achieved in these experiments is obtained by UniMiB, with 75.65% accuracy, UMAFall stands, on the other hand, at 64.69% in the best combination.

The study by *Concone et al.*[136] aims to recognize only four actions with data from the accelerometer and gyroscope. The actions are standing still, driving a vehicle, walking, and running. The sliding window used is three seconds without overlap. This study compares its results with those of most frameworks and Google API. The best-performing classifier is a K-Nearest Neighbor with an accuracy of 95.43%, precision of 91.98%, and recall of 92.87%. For the other classifiers, there is difficulty in distinguishing between standing still and staying in the vehicle due to the nature of the acceleration values. The Google API, according to the results of this study, would need help distinguishing between walking and running.

The approach proposed by Gupta [137] in his study is one without feature selection and extraction in that a Convolutional Neural Network and a Deep Convolutional Long Short-Term Memory are again chosen as classifiers, which, like all Deep Learning algorithms, can compute a set of features of interest by itself to allow for a smoother final classification step. In this study, Gupta prepares to distinguish among the activities available in the WISDM dataset [138].

These activities are divided, for classification convenience, into three categories:

- Ambulation Oriented Activities;
- Hand Oriented Activities (General);
- Hand Oriented Activities (Concerning Food).

The first activities include body movements, such as walking, stairs, standing, jogging, kicking, and sitting. These actions are recognized using WISDM data that predicted that the smartphone was positioned at the waist. Data is obtained through a smartwatch for the other two types of actions, which involve movements that affect only the subject's hands and not their entire body.
In the case of the first subtype of actions resulting from hand movements, general activities such as writing, typing, brushing teeth, clapping, folding clothes, playing with a tennis ball, etc., are dealt with; in the second subtype, on the other hand, the focus is on activities concerning eating such as drinking, eating sandwiches, pasta, chips and so on. Each classifier's data from smartwatches and smartphones is considered separately and divided into training, validation, and test sets according to a 60-20-20 scheme. The best results are achieved by CNN, which has good accuracy levels of 96.54%, precision of 96.35%, and recall of 96.61% for data from a smartwatch. These results drop to about 90% for those obtained from smartphones. Changing classifiers, with the DeepConv LSTM, results in 87 and 88 percent being obtained for data derived from smartwatches, dropping to 75 percent for those obtained from smartphones.

Continuing along the lines of studies not purely inherent to the recognition of bullying or violence more generally, the study by *Lee et al.*[57] also proposes a distinction between three activity classes with two classifiers, Random Forest and CNN, from raw data and vector magnitude. This is referred to as a priori Feature Extraction, probably based on previous studies or knowledge. This study, however, prepares a more straightforward purpose than others by distinguishing between simple discernment activities such as walking, running, and standing still. The real purpose of the study is to show that although the classification task is simple, Convolutional Neural Network manages to beat in performance (accuracy of 92.71%) a classical machine learning algorithm such as Random Forest (accuracy of 89.10%).

Jordao et al. [27] propose recognizing 12 everyday life actions through the sensors of a smartwatch placed at a sampling rate of 32 Hz. This low frequency led to low performance, and an increase in the above could lead to a much better result than the 79.31% accuracy obtained with the CNN classifier at the end of the experiment. This study is interesting because it asserts that a low frequency of data could reasonably lead to an inaccurate representation of activity, which would not be the case with high frequencies.

Ismail et al. [139] propose a classification with the same number of classes as in this thesis, namely five, which are walking, sitting, standing up, climbing, and descending stairs. The data are hand-balanced and split into training and test sets according to the 80-20% instance-based proportion. Two optimizers are tested in the study, Adam and RMSprop, and the latter is the one that performs best, coming in at 95.83% accuracy.

Many interesting studies have been analyzed regarding feature extraction, selection, and classification. It is good, however, to dwell on the type of pre-processing used to understand the State of the Art. In the related studies seen so far, all of them adopted the sliding window mechanism for data extraction, with the window size varying from a second to a maximum of nine seconds in the study by *Cho et al.*[133] . In these studies, the overlap, where implemented, ranged from 33 percent to 60 percent.

An interesting example from the perspective of sliding window overlap is that brought by the study of Dehkordi et al. [140] which uses a two-second sliding window with 75% overlap. The study is assumed to classify two types of actions: static actions, in which there is no real change in the global coordinates of the body, and dynamic actions, in which the whole body is in motion. Ten users were given a smartphone and instructed on what actions to perform with this smartphone held in their preferred hand. Data were collected at 50Hz from the triaxial accelerometer of the smartphones (a Samsung Galaxy or an LG Nexus). Many classifiers were implemented in the classification phase, including a Support Vector Machine, Decision Tree, and Naïve Bayes. The last two performed the best, reaching 98% average accuracy between static and dynamic task recognition.

Studies regarding the recognition of bullying or violence

Studies of bullying-centered, sensor-based activity recognition are few, and most of them can be traced to a small circle of authors [141] [142] [143] [144]. Given the small number of these studies, all the proposed approaches are reviewed below. In addition, those approaches involving fall recognition have been considered, as this is one of the two actions recognized in this study as an action resulting from probable bullying activity. As previously stated, these articles belong to the same authors; their approach has varied, starting and continuing chronologically.
In 2014, *Ye et al.*[141] proposed a model that distinguishes everyday actions, such as walking, running, and bending over, from bullying actions, such as pushing, hitting, and shaking, through a Fuzzy multiple-threshold classifier. The dataset is created with a roleplay phase with the help of some students who were made to wear inertial measurement sensors containing a triaxial accelerometer and triaxial gyroscope. Once the data is collected, the sliding window applied is eight seconds, with a 50 percent overlap; thus, the sliding window is four seconds. Although the classification results are promising (accuracy of 92 percent), the discernment results of the crucial tasks, i.e., bullying tasks, are the worst with a classification error in the worst case, i.e., falling, of 25 percent.

Turning to 2015, *Ye et al.* [142] use the same set of features already defined in 2014 for activity recognition but add some activities more appropriate to the school case, such as playing or getting up from a chair and removing others inappropriate, such as bending. Performance, however, drops to 80 percent, and cases of bullying actions interpreted the wrong way go up. Activities such as shaking and colliding are classified as bullying with 58 percent and 69 percent accuracy, respectively, while falling from pushing remains stable with an error rate of 23 percent.

After three years, Ye et al. proposed a new study [39] that, in addition to recognizing the movements of user subjects through sensor data, combines this decision with emotion recognition through analysis of an audio source. This study shows that combining these two methods works better than single methods; however, this study needs to give more insight into our work. Ye et al.'s 2020 study [144] has the same structure. Only the classifier, which has become a Radial Basis Function Neural Network, is changed. Also removed are two of the bullying tasks that gave the most problems, namely shaking and collision, which have been replaced with pushing and shoving on the ground.

Another study dealing with Bullying Detection is that of Zihan et al.[145] which, like the previously cited paper by Ye et al. [2], combines Sensor-Based Activity Recognition and Emotion-Based Recognition of Tone of Voice. Sensor-based recognition was implemented with smartphone and smartwatch sensors in this study. Some features were extracted a priori and then selected with the Principal Component Analysis algorithm. This dimensionality reduction algorithm takes feature spaces of size greater than or equal to four and recognizes which among them are the most discriminating for the classification task. Features were computed using the Mel Frequency Cepstral Coefficient (MFCC) for emotion recognition by voice. This is a representation of the short-term power spectrum of a sound. The classifier used is K-Nearest Neighbors. After cross-validation, the results were 77.8%

accurate for the sensor-based system and 81.4% for the audio-based system. Finally, those studies using publicly available datasets were considered to take cues regarding the creation of our dataset and to see how they are used in the literature.

An example of using *UMAFall* comes from the study by Amara et al. [146]. In this paper, in addition to *UMAFall, SisFall* is also used, a dataset created with the participation of 38 volunteer subjects divided into two groups based on age. The division of the data corresponds to 50% for training, 30% for validation, and 20% for testing. In pre-processing, however, a label is assigned to activities to identify whether they are Activities of Daily Living or falls. The classification is, therefore, binary, and the number of class instances is unbalanced. This class-unbalanced approach that then results in binary classification is the only way *UMAFall* is used in studies purely related to Fall Detection. A confirmation of this statement can be found in the study mentioned above by *Cho et al.* [3], who also brings up the work regarding *UniMiB SHAR*, saying that precisely because of the binary classification, there is no real guideline for analyzing the results of their experiment. Unfortunately, this problem has reappeared when analyzing our models' performance.

Many studies have been conducted on falls, including that of Nguyen et al. [147]. This paper presents a method that extracts 44 features on the domain: frequencies, time, and Hjorth parameters. The classifiers used were SVM, k-NN, ANN, J48, and RF. Following the F1-Score metric with the *MobileAct 2.0* dataset, the following values were 95.23% (falls) and 99.11% (non-falls). In contrast, following the F1-Score metric with the *UP-Fall* dataset, 96.16% (falls) and 99.90% (non-falls) were found.

In this work, a comparison with the state of the art was made only for the Human Activity Recognition methodology, comparing values with known algorithms and a dataset like the one created by our Team. The results of the HAR macro-activities align with the results of the current papers. The results inherent to cyberbullying/bullying are new and cannot be compared; therefore, the current findings serve as a good starting point for other researchers to extend the work.

*3.3.2.3.*2 Dataset

The datasets considered belong to two categories. The first category considers the two HAR datasets, namely "*Dataset HAR Uniba*" and "*UniMiB SHAR,*" which have the classical feature columns that identify a HAR Dataset with an accelerometer. So, the activity's raw x, y, z, and target features. The second category, "Questionnaire experiment dataset," identifies the dataset extracted from the user test with the questionnaire. This dataset has three accelerometer features (x, y, z). In addition, ideally, one more feature was considered for the final analysis, namely, the users' personality index (*Bully, Cyberbully, Bully Victim, Cyberbully Victim*) provided as output from the questionnaire.

Dataset HAR Uniba:

The creation of this dataset took place in a controlled environment. Nineteen participants, 13 males and 6 females participated in the construction. Each was made to perform eight actions divided into the two categories already encountered:

- *ADLs* (walking, running, jumping, sitting);
- *Falling* (forward, backward, right, and left).

A smartphone was placed in the right pocket with the screen toward the participants' bodies, and accelerometer values were collected at a sampling rate 200Hz via the smartphone application. The data were sent to a server in *.txt* format and reprocessed to arrive at the final format. Each task was performed between two and three times. Each trial was 15 seconds long. The falls were performed on a mattress placed on the floor. Within the .csv file, the first column contains the user ID, the second the activity performed, the third contains the timestamp in milliseconds from the start of the action, and next are the three triaxial accelerometer values x, y, and z.

UniMiB SHAR:
A Dataset consisting only of values obtained from the accelerometer. Thirty participants. Recognized activities are again divided into two categories:

- *Falling* (Falling forward, backward, right, left, hitting an obstacle, with protective strategies, without protective strategies, Syncope)
- *ADL* (Walking, running, climbing stairs, descending stairs, jumping, lying down from standing, sitting)

For each activity, there are 2 to 6 trials for each user. For the actions with two trials, the smartphone in the right pocket is used in the first and the left in the second. For actions with six trials, the first three have the smartphone in the right pocket and the others in the left pocket. Data are provided in windows of 51 or 151 samples around an original signal peak higher than 1.5g, with g being the acceleration of gravity [41].

Questionnaire experiment dataset:
The *Questionnaire experiment test* consists of an Android application that incorporates a questionnaire redone by psychologists. The graphical layout of the existing app is simple and essential. Every choice has been made to make the graphical interface simple and appealing to the end user and eliminate or at least drastically reduce the possibility that the user during the activities may make mistakes, have doubts about the actions to be performed, or other problems in general.

The application contemplates five basic steps:

1. Access to the app's functionality (3 screens);
2. Access to the four Cyberbullying videos (4 screens) for each video, an emotional question (4 screens);
3. Access to the Questionnaire (30+ screens) that is quick to fill out;
4. Access to the Telegram group (1 screen);
5. End of Test (1 screen).

This specific test returns the personality index: *Bully, Cyberbully, Bully Victim, Cyberbully Victim* for each user [5].

The four Cyberbullying videos specifically are [5]:

1. *Video1Activity ("VIRTUAL ACTIONS, REAL CONSEQUENCES,"* https://www.youtube.com/watch?v=x2AxcllGLJg&t=4s&ab_channel=TabbyEUproject [5])
2. *Video2Activity ("ANYONE CAN BE ANYONE,"* https://www.youtube.com/watch?v=z3N24DpD64c [5])
3. *Video3Activity ("INTERNET = EVERYONE, FOREVER,"* https://www.youtube.com/watch?v=K31Kuc5pTXM&t=42s&ab_channel=TabbyEUproject [5])
4. *Video4Activity ("VIRTUAL VENDETTA (joke or* crime?)," https://www.youtube.com/watch?v=FpBVBwv6UQ4&ab_channel=TabbyEUproject [5])

The dataset used in this study results from a *questionnaire experiment test using an Android application* created and used to acquire data. The tests were conducted with 99 students aged 18-24 (average 20 years old), all enrolled in their first year of college. Two test sessions were carried out on real users (referred to in the chapter as *Test1* and *Test2*). The users between the two tests are different. *Test 1* and *Test 2* were set up by allowing participants to sit at their desks. They were allowing them any freedom to move around. However, the results predicted sedentary activity from the accelerometer readings taken during the questionnaire. Sitting is defined as the baseline activity in the chapter. Sitting was then considered non-restless behavior, therefore quiet.

3.3.2.3.3 Experimental Setup

The Experimental Setup was structured into several parts:

- *Phase One*: Extraction of sensor coordinates;
- *Phase Two*: Model training;
- *Third Phase*: Recognition of the activity performed by the model above.

Phase One is devoted to extracting the *Raw Accelerometer Coordinates (X, Y, Z)* from the experiment performed in real-world contexts with the questionnaire application. The accelerometer was recorded for each screen of the application. Each screen included a single question from the questionnaire application. The resulting Dataset from the extraction, for each user of the 90 tested, is used in the final stage as a test with the HAR model. In Phase Two, the HAR model is trained with two HAR datasets, one state-of-the-art "*UniMiB SHAR* [41]" and the second created by our research group " *Dataset HAR Uniba*." In the Third Phase, through the HAR model, the activities are, for each user, predicted using the experiment of the first phase.

Data preprocessing

At this phase, text files containing data obtained from device sensors from the questionnaire app were retrieved, the accelerometer data only, discarding those from the magnetometer and gyroscope. Once I selected the row of interest (*with X, Y, Z coordinates*), I would insert a row for each triple of coordinates (*X, Y, Z*) within a data structure, which also specified the recorded questionnaire screen during that movement. CSV files were created for each user, which was helpful for predictions in later stages. The same procedure was addressed for both the users participating in Test 1 (52) and Test 2 (48); however, only some were considered because some still needed to complete the entire questionnaire.

Models

State-of-the-art models were considered for this test:

- *CNN*, a convolutional, feed-forward neural network, consisting in this case of 3 ReLU layers, each alternating with a pooling layer for simplifying the output obtained from the previous layer, whose practical goal is to reduce the number of parameters the network must learn. After these, a Flatten layer is for linearizing the output, and a *softmax* layer is for actual classification;
- *Bi-LSTM*: this model is a particular type of LSTM network that is practically trained to make predictions about the knowledge of the past and the future and then go backward with the predictions. Unlike the LSTM network, which can learn unidirectionally.

Specifically, Table 32 shows the network parameters for this work.

CNN	Bi-LSTM
Model Sequential	*Model Sequential*
Conv2D: filters 64, kernel_size 2, activation relu	*Layers LSTM: units 64*
Dropout 0.5	*Dropout 0.2*
Conv2D: filters 32, kernel_size 2, activation relu	*Layers LSTM: units 64*
Dropout 0.5	*Dropout 0.2*
Conv2D: filters 16, kernel_size 2, activation relu	-
Flatten layer	-
Dense layer 256, activation relu	*Dense layer 64, activation relu*
Dropout 0.5	*Dropout 0.5*
Optimizer Adam, loss categorical_crossentropy	*Optimizer Adam, loss categorical_crossentropy*

Table 32 - Parameters used for networks

They employed several Deep Learning models with different combinations for the HAR experimentation to find the best performance model. Table 33 shows the averages of accuracy and f1 score overall users. Specifically, they constitute the averages concerning each user in the Dataset trained using the *Leave One Out technique*.

Model	Dataset	Average Accuracy	Average F1_Score
CNN	*Dataset HAR Uniba*	*0.9199*	*0.9084*
CNN	*UnimibShar*	*0.9768*	*0.9473*
Bi-LSTM	Dataset HAR Uniba	0.8955	0.8581
Bi-LSTM	UnimibShar	0.7617	0.7885

Table 33 – Table that shows the best accuracy results and average F1 scores

In Table 33, it can be observed that the CNN model outperforms the Bi-LSTM models. This means the model could better discriminate all activities from the *Dataset HAR Uniba* and *UnimibShar datasets*. Especially similar activities such as falling and sitting. The best model, CNN, was thus used to perform activity prediction with the dataset "*Dataset questionnaire experiment*" to analyze activity predictions by class. The results by classes of interest are illustrated in Figure 47 - Figure 50.

In Figure 47 - Figure 50, the color legend in the graphs is as follows: Bullying (Gray), Victims of bullying (Blue), cyberbullying (Orange), and Victims of Cyberbullying (water green).

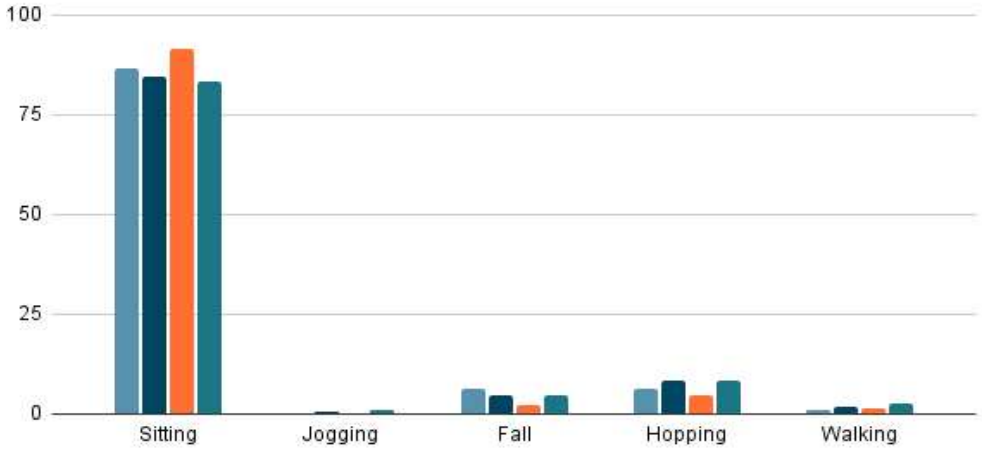


Figure 47 - Averages Result in Users Test 1. Bullying (Gray), Victims of bullying (Blue), cyberbullying (Orange), Victims of Cyberbullying (water green). The x-axis identifies activities, while the y-axis identifies the average percentage of activities of each test participant, divided by personality index (questionnaire class)

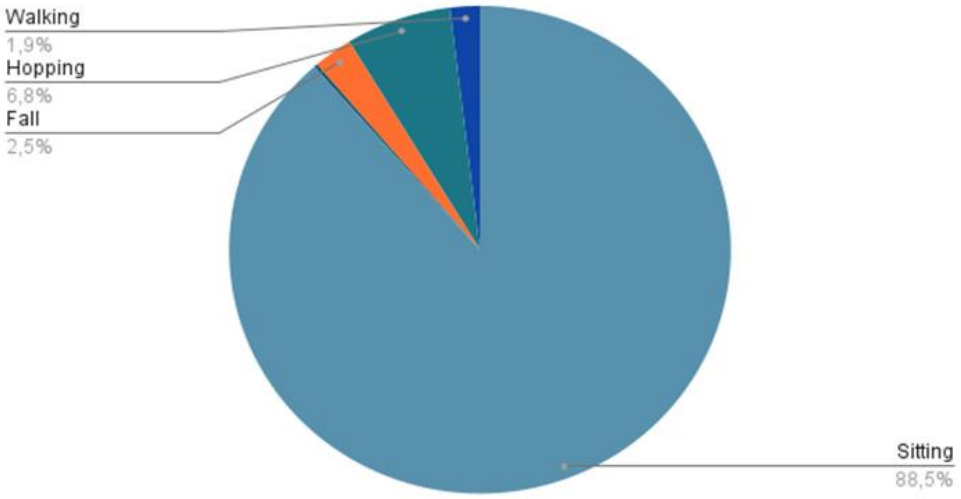


Figure 48 - Total-User Averages Test 1.: Bullying (Gray), Victims of bullying (Blue), Cyberbully (Orange), Victims of Cyberbullying (Water Green).

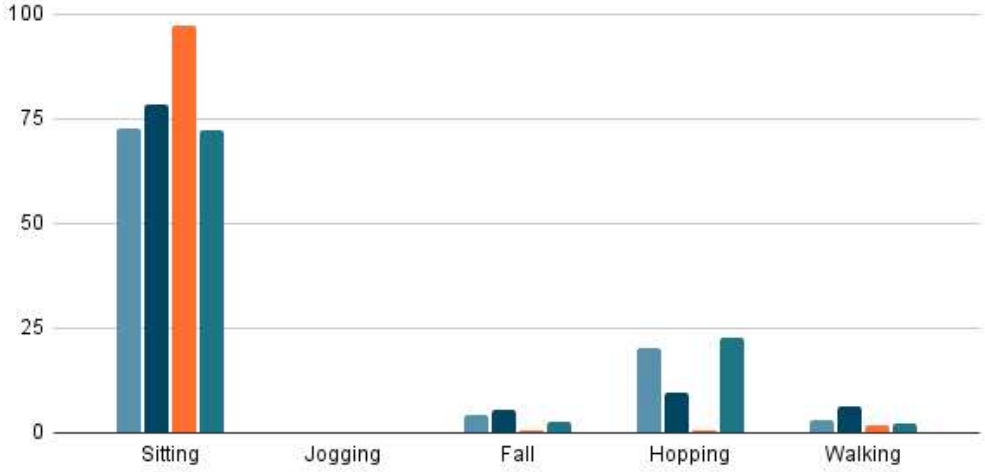


Figure 49 - Averages Result in Users Test 2. Bullying (Gray), Victims of bullying (Blue), cyber-bullying (Orange), Victims of Cyberbullying (water green). The x-axis identifies activities, while the y-axis identifies the average percentage of activities of each test participant, divided by personality index (questionnaire class)

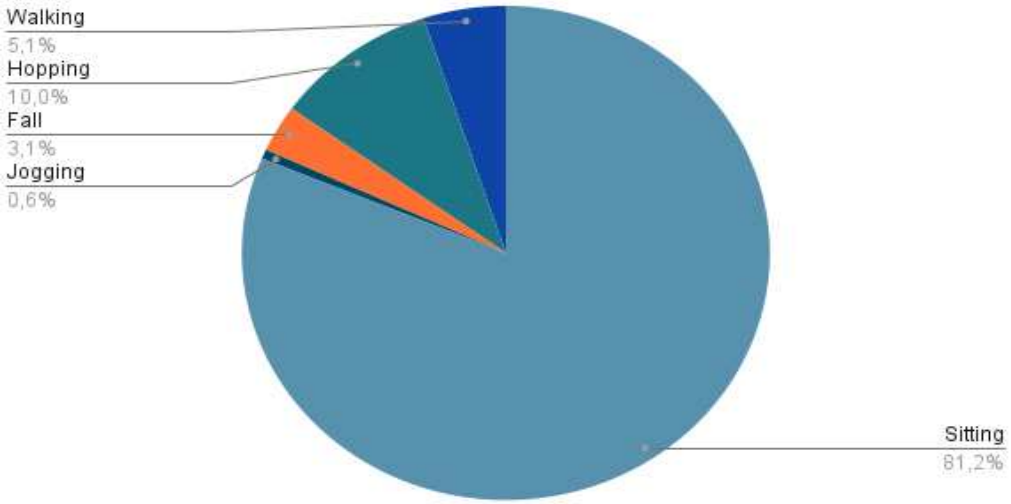


Figure 50 - Figure 51. Total-User Averages Test 2. Bullying (Gray), Victims of bullying (Blue), cyberbullying (Orange), Victims of Cyberbullying (water green).

Regarding **Figure 47** and **Figure 48**, the y-axis identifies the average percentage of activities of each individual participating in the individual test, divided by personality index (questionnaire class). The individual has their percentage of activity about filling out the questionnaire and, in turn, possesses a personality index; these percentages are averaged from each individual.

Looking at the results obtained from the experiment, it was seen that (see **Figure 47**-**Figure 50**), considering Test1 and Test2:

- *Cyberbullying* was found to be "quieter" because they were recognized as sitting most of the time and in higher percentages than the other categories of users;
- Bullying has jumped as their predominant abnormal activity, particularly during the questionnaire phases of *QuizActivityButtons*. The *QuizActivityButton* stage contemplates questions that ask about activities performed or experienced in bullying and Cyberbullying;
- Bully victims, as well as Bullying, showed more abnormal phases during the *QuizActivityButton* but with prevalent activities of jumping, walking, and falling. Falling is also defined as dropping the cell phone alone;
- The victims of *Cyberbullying* were the only users participating in the questionnaire, which showed several abnormalities in the initial part, in which they were subjected to some videos to watch with questions related to the feelings experienced by watching them.
- It is believed that the questionnaire could be reduced to only those questions about the category it is intended to identify:
- In the case of Bullying and victims of Bullying, *QuizActivityButtons* would suffice, thus removing the initial part of the video submission and related questions;

The latter, on the other hand, were helpful in the category of cyberbullying victims with additional questions that are reported below:

- *Video1Activity ("VIRTUAL ACTIONS, REAL CONSEQUENCES," https://www.youtube.com/watch?v=x2AxcllGLJg&t=4s&ab_channel=TabbyEUproject* [5]*)*
- *Question_Video1Activity (5-point Likert scale emotional question);*
- *Video2Activity ("ANYONE CAN BE ANYONE," https://www.youtube.com/watch?v=z3N24DpD64c* [5]*)*
- *Question_Video2Activity (5-point Likert scale emotional question);*
- *Video3Activity ("INTERNET = ALL, FOREVER," https://www.youtube.com/watch?v=K31Kuc5pTXM&t=42s&ab_channel=TabbyEUproject* [5]*)*
- *Question_Video3Activity (5-point Likert scale emotional question);*
- *Video4Activity ("VIRTUAL SALVAGE (joke or crime?)," https://www.youtube.com/watch?v=FpBVBwv6UQ4&ab_channel=TabbyEUproject* [5]*)*
- *Question_Video4Activity (5-point Likert scale emotional question);*
- *Activity_Curtain: Select the plexus to which you belong from the drop-down menu.*
- *QuizActivityText0: Enter the modified Telegram Nickname at the beginning of the app.*
- *QuizActivityText1: How old are you?*
- *QuizActivityText2: Are you a boy or a girl?*
- *QuizActivityText3: What school do you attend?*
- *QuizActivityText4: What grade do you attend?*
- *QuizActivityText5: What country were you born in?*
- *QuizActivityText6: Specify what part of Italy you live in.*
- *QuizActivityText7: Write in the box below, What social networks do you use?*
- *QuizActivityText8: If you have at least one profile, how many friends do you have on social networks?*
- *QuizActivityText9: Is one of your parents or another adult you trust among your friendship contacts? (YES / NO)*
- *QuizActivityText10: Do you have at least one profile on social networking sites (Example: Facebook, WhatsApp, Instagram, Ask.com, etc...)?*

Looking at the results and the questions, it is possible that the victims of Cyberbullying were strongly influenced and impressed by the videos containing images of bullying, especially Cyberbullying. It could be that some Cyberbullying victims may have been impressed by remembering some events that happened in the past. In addition, abnormal activities were recorded in the generic initial questions; victims always tend to hide and not be found out.

Regarding Cyberbullying, *QuizActivityButtons* 55 to 59 were found to be more discriminating, particularly with Falling activities:

- *Receiving threats and insults on the Internet (websites, chat rooms, blogs, SMS, Facebook, Twitter...),*
- *Receiving silent phone calls*
- *Receiving e-mails with threats and insults*
- *Receiving videos/photos/pictures of embarrassing or intimate situations via cell phone*
- *Receiving phone calls with threats and insults*

Looking at the results and the questions, it is possible that Cyberbullying is highly conditioned by their actions; therefore, contentment to think about the victim's receipt of content could be weighed positively and thus appear to be discriminated against.

In this study, through *Human Activity Recognition (HAR)* models, behavioral analysis was performed by analyzing users' behavior while filling out a questionnaire via smartphone useful for classifying users as *Bullying, Cyberbullying, victims of bullying, and victims of Cyberbullying*. The smartphone questionnaire extracted the accelerometer, which helps recognize activities and behaviors different from just sitting. Any activity other than sitting (considered non-abnormal) is classified as abnormal or representative. The question found at the right time of "Abnormality" was considered for analysis among the Target classes.

Observation and conclusions:

- Appropriate DL models could be used to perform HAR based on data obtained from sensors on a smartphone, concluding that the best-performing Deep Learning model was the convolutional neural network;
- Abnormal aspects were observed and highlighted during the prediction phase, anomalous in the sense of recognized activity different from what is expected (sitting). Cyberbullying was found to be "quieter" because they were recognized as sitting most of the time. Bullying is noted as an abnormal activity jumping during the steps of the *QuizActivityButton* questionnaire, which contemplates questions asking about activities performed or experienced in bullying and Cyberbullying. Bully victims and Bullying showed more abnormal during the *QuizActivityButtons* but with prevalent activities of jumping, walking, and falling. The victims of Cyberbullying were the only users participating in the questionnaire who demonstrated several anomalies concerning all activities in the HAR datasets.
- For each category, "*anomalous*" questions were selected and reported in the chapter, justified by appropriate theories inherent in the various classifications.
- There were many limitations; one problem was the small amount of data. Both the HAR dataset and the dataset resulting from the questionnaire test were small samples of users. In addition, a significant limitation was the proposed methodology that brought together HAR and cyberbullying/bullying via smartphone sensors, which had not yet been widely addressed in the state of the art. Moreover, it would have been interesting to try multiple other algorithms capable of detecting the micro-behaviors identified by the personality index.

The results inherent to HAR align with state-of-the-art, even considering a second dataset, namely the *" Dataset HAR Uniba."* At the same time, the results derived from the questionnaire study and the marriage with the HAR approach are embryonic and can be considered a pathfinder for future work. However, while embryonic, the conclusions made are still interesting because correlations were found between the psychology and behavioral attitude of the individual.

*3.3.2.*4 Classification Bullying/Cyberbullying through Smartphone Sensor and a Questionnaire Application

In this study, an Android application was developed to implement a questionnaire on bullying and cyberbullying, using smartphone sensors to predict the Personal Index. Sensor data were collected in the "*UNIBA HAR Dataset*" and analyzed using AI algorithms to find a correlation between the categorization class of the questionnaire (*Personality Index*) and the prediction of ML behavioral models. The results indicate that the Bayesian Bridge with *"Bullying bully vs. Victimization bullying"* and *"Total bullying vs. Total victimization"* performs better on average 0.94 accuracy, and the LSTM with the last categorization performs 0.89 accuracy. These results are crucial for future development in the same research area.

This chapter considered a questionnaire on the topic of *Bullying/Cyberbullying*. It detects the user's categorization class based on the answers given. The end-user categorization classes are *Bullying-Bully, Bullying-Victimization, Cyberbullying-Cyberbully, Cyberbullying-Victimization, and Non-risk users*. The categorization class is assigned with full respect for privacy.

In this work, the questionnaire was implemented in an Android application. As users fill out the questionnaire, smartphone haptic coordinates and sensors are extracted, i.e., accelerometer, magnetometer, gyroscope, etc. *Accelerometers or gyroscopes* are specialized to detect and highlight behavioral features, even minimal derivation [148], [149]. Sensor data are used in *Behavioral Biometrics* models. The chapter's challenge is correlating personality indices (questionnaire labels) and predictions of machine-learning behavioral models with sensors. The chapter discusses a methodology that currently needs more evidence in the literature. This is because no paper has considered behavioral biometrics in the context of bullying/cyberbullying. From simple video viewing and sensor extraction of behavior, one wants to predict the personality index. The chapter conclusion determines to consider only one video among the four in the questionnaire because it is considered relevant to a particular class of categorization. The work's end user is the user who fills out the questionnaire.

The objectives of the chapter are:

1. Develop a machine learning model using smartphone sensor data to classify users as active or passive participants in bullying or cyberbullying. The objective is to investigate the potential correlation between the questionnaire-based categorization (*Personality Index: Bullying-Bullying, Bullying-Victimization, Cyberbullying-Cyberbullying, Cyberbullying-Victimization, Non-risk Users* [5]) and the behavioral patterns predicted by machine learning models trained on accelerometer data.
2. Determine the most distinguishing *VideoActivity* among the four available Android applications, aiming to identify both victims and bullies accurately.
3. Evaluate and compare the performance of shallow learning and deep learning approaches to determine the most effective method for the given task. Additionally, identify the optimal machine learning model with the highest classification accuracy and predictive capability.
4. Create the *"UNIBA HAR"* dataset. Currently, no state-of-the-art datasets contain sensor information extracted from smartphone questionnaires. The dataset is a valuable resource for research, enabling comprehensive analysis of smartphone sensor data on bullying behaviors.

The innovation of this article lies in the fusion of psychology and computer science to implement this methodology through smartphones. The project's psychologist meticulously examined this phase, and through her expertise, the decision was made to develop this groundbreaking application. Since no dataset extracts the identified features, and no similar methodology exists, personality indices were predicted using basic, shallow, and deep machine learning models. The innovation in this approach

is intricately linked to the dataset, the novel methodology applied, and the features extracted from smartphones.

The proposed solution could encounter countless impacts naturally. It could affect several ways: Early *intervention, identification of risk factors, and improved effectiveness of prevention measures.* Using smartphone sensing freely, without constraints, and the implemented ML models can lead to accurate classification of users into bullies or cyberbullies. All this is useful for identifying those at risk or involved in such aggressive behaviors early. In addition, it could be helpful to identify risk factors associated with bullying and cyberbullying through the help of psychologists in the behavioral field. These procedures and in-depth studies would lead to adopting prevention policies/programs that address these factors. In addition, it is also the optimization or implementation of preventive measures. That is, employing the questionnaire app or tools based on sensor data that detect potentially problematic behaviors. The purpose of these apps could always be for prevention and notification to tutors or parents.

The impact in the fundamental mode assists in the *Real usage scenario*. The Real usage scenario could be the school context or online communities in which bullying and cyberbullying risks are present.

The project mentioned in the Acknowledgments, *"PRIN2017 - BullyBuster project - A framework for bullying and cyberbullying action detection by computer vision and artificial intelligence methods and algorithms,"* aims to prevent the phenomenon in schools. Other works related to the same project have been done [150]. The Android application mentioned in the chapter could be used in schools to collect data from students' sensors while completing the bullying and cyberbullying questionnaire. The app is helpful in these contexts because a teenager is unlikely to inform a guardian/parent of the events that happen to them. The app's anonymity helps create an intimate space by prompting truth-telling. This could enable school personnel to identify at-risk cases and provide appropriate support to those involved. Schools are not the only ones affected by these issues. So are online communities. The application could be used and extended to platforms or social media where there is a community of users, for example, a school forum or an online gaming platform. Again, user categorization could help the population.

3.3.2.4.2 Related Works

Approaches to cyberbullying/bullying related to smartphone sensor data should be covered better in the literature. Many articles dealing with violence in comments or emails are considered spam [151], [152]. Numerous articles advance this research through precise machine-learning algorithms that classify data into two or more categories [152].

Given the extensive research on the text, this section examines the state-of-the-art approaches, including the *Human Activity Recognition Approach*, the *Authentication Approach*, and the *Techno-Regulation Paradigm*. All smartphone and sensor approaches use secure authentication and *Human Activity Recognition (HAR)* [25], [153]. First, it was focused on *HAR activities* (using sensors and Machine Learning models) and authentication in the security world. In addition, the topic of *bullying/cyberbullying* falls under the *techno-regulation paradigm*, depending on the data type and the purpose of the research. The methods proposed in this article take cues from these works but still address a different topic.

Human Activity Recognition Approach

Some approaches that have combined sensors and hardware devices to predict human activities have been studied. Gattulli et al. [90] analyzed cyberbullying by analyzing individuals' Human Activity

Recognition while they similarly filled out a questionnaire. The work aimed to implement a new smartphone methodology that combines the final label obtained from the cyberbullying/bullying questionnaire (*Bully, Cyberbully, Bullying Victim, and Cyberbullying Victim*) and the human activity performed (*Human Activity Recognition*). While filling out the questionnaire, the work aimed to understand whether a question is more discriminating than another, considering the way of holding the phone and especially the activity performed. This work deals with behavioral biometrics by referring to the smartphone accelerometer. The algorithm detected small changes in the axes to predict activity. The previous work has the same structure but a different aim. The previous work aimed to understand if Human Activity correlates more with the personality index. This work has another goal (via smartphone sensor decree, the personality Index) outlined at the end of Section 1. Introduction.

Also, Luo et al. [11] recently chose to use activity-based people identification approaches, thus HAR and sensors. In fact, unlike the approach implemented in this work, Luo et al. [154] also use Apple Watch and Google Glass. These devices are optimal for continuously detecting and collecting information related to users' activity, and activity patterns can be extracted to differentiate different people. Other articles propose a method to recognize walking activity using accelerometers, improving the quantification of physical activity [155]. Wang et al. [156] propose a smartphone-based epidemic alert system to identify contacts with infected people. Both approaches are validated through experiments and offer innovative solutions to their respective problems.

Authentication Approach

Some approaches that have combined sensors and smartphones for authentication and, thus, Smartphone Security have been studied. Wearable devices equipped with various sensors facilitate the measurement of physiological and behavioral characteristics. These could be categorized and incorporated into the concept of device security and then, through smartphone sensors, understand user behavior and thus identify the user. It could mention some state-of-the-art approaches, namely:
Hu et al. [157] believe sensor-based continuous biometrics authentication is promising. Hu et al. [157] present a new *"AuthConFormer"* system of continuous authentication based on a convolutional smartphone transformer. The experimental results exhibited by Hu et al. [157] show that the proposed AuthConFormer can achieve good EER results, respectively [157]. The work of Rayani et al. [158] also traces the previously considered thesis that continuous user authentication on the smartphone with sensors is one of the best behavioral biometrics approaches. Specifically, Rayani et al. [158] propose a system that captures touch-based smartphone data through the touchscreen and inertial sensors of the device. Experimental results show that the approach proposed by Rayani et al. [158] achieved the best authentication score compared to other machine learning models and more advanced methods [158]. In the expository work, Alzahrani et al. [17] consider behavioral biometrics and then introduce the powerful sensing capabilities of IoT devices, such as smartphones and smartwatches. Due to the implemented hardware, they enable Machine learning algorithms to apply Continuous Authentication methods. Touch coordinates and smartphone sensors can better identify an individual's behavior. An article illustrated by Alzahrani et al. [159] presents a passive continuous authentication approach that can learn IoT user signatures using smartphone sensors, such as a gyroscope, magnetometer, and accelerometer, to recognize users by their physical activities [159].

The behavior can also be related to the mnemonic approach [108], [160]. Basic knowledge approaches in the touch paradigm are essential. Storing a simple PIN or password and concatenating it with sensor-based behavioral techniques could enrich its security. The problem of cyberbullying is related to this. After they are unlocked, many devices lose security approaches, such as these in combination, which can lock the smartphone in situations where it is held by another individual identified as such. A bully could possess an unlocked phone and impersonate the victim on social media. Consequently, the victim could be subjected to phishing, smear, and side-channel attacks [161].

The authentication method illustrated by Nerini et al. [162] increased the security of PIN authentication by precisely considering behavioral biometrics and, thus, the typical smartphone movements of each user. Along with this approach, a methodology based on anomaly detection has been proposed to understand whether the PIN entered is from the smartphone owner or is related to an attack by an outsider. Nerini et al. [162] believe the classifier's decision is based on smartphone movements recorded during everyday use. This procedure emphasizes the non-invasiveness of the user approach. Numerical results shown by Nerini et al. [162] emphasize that this new authentication method can achieve a low error rate [162]. Another approach simultaneously uses knowledge-based (password and pin storage) and biometrics-based approaches is Teh's. et al. [163]. Teh et al. [163] describe the design, implementation, and evaluation of an authentication system that combines a knowledge-based and a biometrics-based method of touch dynamics. In their experiment, raw, tactile data are acquired and then used to extract various features successively fed to one-class models such as OCKNN and SVDD and binary models such as KNN and SVM to verify user identity. The evaluation results of Teh et al. [163] show that, based on the Equal Error Rate (EER) values obtained by integrating touch dynamics biometrics into the PIN-based authentication method, the levels of protection against representation attacks are significantly improved [163]. This work has prompted more scientific papers about computer science, bullying, and cyberbullying. The following work inserted itself between psychology and computer science topics. Each expertise was essential when working concurrently. In fact, this work inevitably led to exploring the topics of Bullying and Cyberbullying, as they were of fundamental importance. Moreover, bullying and cyberbullying were strongly related to individuals' natural behavior and attitudes. These studies emphasized the importance of behavioral biometrics due to innovative handheld devices, as they brought significant discrimination in terms of EER.

In addition, they show us that smartphone sensor, especially accelerometers, gyroscopes, and just accelerometers, can easily discriminate an activity, a way of behaving, and thus biometric behavior. This chapter, therefore, tries to discriminate normal human behavior while filling out a questionnaire. As seen from the previous section, a smartphone is used, and data is extrapolated from its sensors, namely the accelerometer.

Techno-regulation Approach

The current work collects sensor data from smartphone devices to classify users by indexing the classification to bullying/cyberbullying. This phase is closely related to the techno-regulation paradigm in several areas: behavior analysis, child protection [164], online safety [165], and worker protection in the gig economy. These areas were later dissected by comparing the various issues with the work presented.

Regarding behavior analysis, touch coordinates and sensor values were extrapolated in the macrowork of data extraction from the questionnaire. In this work, only sensor data were used as a first experiment, namely, Accelerometer (X, Y, Z). Touch data could also be used in the future to augment the experimentation. This analysis is linked to behavioral analysis to identify meaningful patterns. The work aims to identify a micro-behavior that could fall under the categorization of the questionnaire. The work in this chapter also aims to identify a possible personality index of the user, that is, a personality attitude that can be inferred from the questionnaire categorization. The following work could detect a machine learning behavioral model (user profile) used on new users to identify their personality index. This leads us to talk about child protection, the second examination point. The child has difficulty discussing these situations with an adult; the questionnaire may help ensure the user's content's truthfulness.

The latter approach could help create a safer and more secure online environment for youth. This definition leads to the third point: Online Safety and Protection of Private Content. It is essential to consider the implications for privacy protection and unauthorized dissemination of private content. In fact, in the following work, with the help of a psychologist, data extraction was performed while fully respecting privacy principles. Everyone was identified by a Numerical ID that was never traced back to the tested individual. For privacy reasons, the questionnaire output (Ground Truth) is not disclosed to the user and cannot be traced back to the user. Regarding data storage, taking appropriate measures to ensure data security and comply with privacy regulations was essential.
Regarding Worker Protection in the gig economy, no workers outside the research group mentioned in the chapter were considered. The tests were performed in controlled settings. The classification was followed by the work and study of a psychologist in bullying/cyberbullying.

3.3.2.4.3 Materials and Methods
This Section illustrates the experiment pipeline with the methodology starting from the essential part of this experiment, namely, the innovative dataset created (4 video tests and a questionnaire). Secondly, the *"sensor data acquisition"* included in the dataset's treatment is illustrated. Immediately following is the essential pre-processing phase that helps clean the data to make it usable for Machine Learning models. Finally, Machine Learning models are illustrated by considering as input the datasets in the three categories shown at the line of the previous Section, namely*, Users at Risk, Bullying Bully,* and *Total Bullying*.

The illustrative pipeline is shown graphically in Figure 51. Figure 51 illustrates the work pipeline and describes the various design steps. First, the triaxial accelerometer was extracted while users viewed the four videos explained in the next Section. After accumulating the sensor data and the subsequent pre-processing, classification was performed using the selected models.

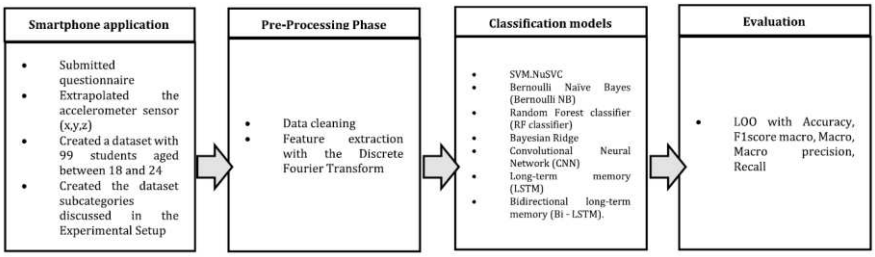


Figure 51 - The pipeline consists of four main phases: the questionnaire sensors extraction, the pre-processing phase, the feature extraction phase, the classification phase, and the model evaluation phase.

3.3.2.4.4 Dataset
The introductory section summarizes the experiment's design and the dataset's creation; more details are provided in this Section. The dataset used in this study was collected by involving first-year University Student s. The "*UNIBA HAR Dataset*" was conducted with the help of a questionnaire on the topic of bullying and cyberbullying. The questionnaire prompted by psychological researchers was incorporated into an Android application. *Why was an Android application introduced?* Through it and the subsequent completion of the questionnaire, it was possible to extrapolate basic features belonging to behavioral biometrics. Smartphone sensors, gyroscopes, accelerometers, proximity sensors, atmospheric pressures, magnetometers, ambient brightnesses, and step Detectors were extrapolated. In particular, this study considered the accelerometer one of the state-of-the-art discriminative sensors. For this specific reason, it was chosen promptly. In addition, features inherent to touch dynamics were also extrapolated about possible future work.

The application's architecture is like many questionnaires viewed on the Web [166]. The first part shows four videos depicting animated skits about bullying and cyberbullying. These animated skits were introduced to impersonate the person tested in their viewing scene. Each video manifests different difficulties that a victim might experience again while and after they suffer an attack. The first phase of the application is the focal point of this chapter because it only deals with sensor data as users view the videos and answer the four questions after each video. A question arises after each video intends to extrapolate the emotional degree of the individual after they view the video. The question asks, *"What emotion did you feel while viewing Video 1?"* it is assigned answer buttons based on the five basic emotions of the 5-Likert scale *(Sadness, Happiness, etc....)* [108].

The second part of the questionnaire includes ninety-nine questions to label the user in the primary classes (Personality Index), namely (*Bullying-Bully, Bullying-Victimization, Cyberbullying-Cyberbully, and Cyberbullying-Victimization* [109], [110], [111]) (*Table 34*).

	X	Y	Z
0	-0,65,563	4,830,152	6,139,504
1	-0,69,775	4,586,802	6,225,885
2	-0,77,432	4,491,089	6,346,484
3	-0,69,392	4,416,432	6,319,684
4	-0,67,478	4,353,262	6,206,742
5	-0,64,415	4,418,347	6,197,171
6	-0,63,075	4,535,117	6,342,655
7	-0,78,389	4,380,062	6,411,569
8	-0,90,855	4,176,911	6,626,205
9	-0,97,364	4,146,283	6,788,917
10	-0,88,176	4,127,141	6,689,375
...	...	...	...

Table 34 – A sample of raw accelerometer data extracted from the smartphone.

Table 34 is a sample of raw accelerometer data extracted from the smartphone. The X-axis accelerometer range is around min/max (-0.9736395/0.2349752), Y is around min/max (3.7651067/4.8301516), Z is min/max (6.0016775/7.216036). Of these ninety-nine, only seventy-three are essential for Personality Index categorization. The following scientific papers illustrate the mathematical calculations that enable us to decree categorization [109], [110], [111]. Only the sensor values inherent in the four videos were considered in this chapter. Users run the application but then analyze it in macro by psychologists. The results are not shown to the users because they are anonymous. Psychologists used the results to estimate the whole class problem statistically.

Specifically, the asset explained above is as follows:

- *Video1Activity ("VIRTUAL ACTIONS, REAL CONSEQUENCES," https://www.youtube.com/watch?v=x2AxcllGLJg&t=4s&ab_channel=TabbyEUproject* [5]*)*
- *Question_Video1Activity (5-point Likert scale emotional question);*
- *Video2Activity ("ANYONE CAN BE ANYONE," https://www.youtube.com/watch?v=z3N24DpD64c* [5]*)*
- *Question_Video2Activity (5-point Likert scale emotional question);*
- *Video3Activity ("INTERNET = EVERYONE, FOREVER," https://www.youtube.com/watch?v=K31Kuc5pTXM&t=42s&ab_channel=TabbyEUproject* [5]*)*
- *Question_Video3Activity (5-point Likert scale emotional question);*
- *Video4Activity ("VIRTUAL VENDETTA (joke or crime?)," https://www.youtube.com/watch?v=FpBVBwv6UQ4&ab_channel=TabbyEUproject* [5]*)*

- *Question_Video4Activity (5-point Likert scale emotional question);*

Testing via Android applications was conducted with 99 students aged 18-24 (average age 20), all enrolled in their first year of college. The test was conducted to avoid anticipating anything from the individuals who then took the test. This made the questionnaire data much more truthful. As soon as the test was concluded, everyone was asked to comment on the experiment through sincere commentary on the experience, especially on the relevance of the topic approved by all participants. Finally, everyone then uninstalled the application. After the comprehensive test, separately, each user was assigned a class (Personality Index) among these four (Bullying-Bully, Bullying-Victimization, Cyberbullying- Cyberbully, and Cyberbullying-Victimization [5]). The proposed dataset incorporates the sensors (accelerometer) from the four videos for each user. This database helps us detect the micro-behaviors that occurred while watching the videos.

TEST1 and TEST2 are two different sessions, done on two days with different Users. For this reason, they have been referred to in the chapter (as Test1 and Test2). The users who performed the two tests (Test1 and Test2) are different from each other, and using their smartphones, the data acquired affects different smartphone models [167]. Based on the numerosity of the extracted sample and according to the label results (personality index) that were attributed, three types of overall classifications were created for the tests with Machine Learning classifiers. As a new sample, datasets and sub-datasets were created to conduct a test that completes training. Machine Learning models must be trained correctly, especially considering strongly balanced classes. Since this is a new approach with a new dataset and new classes, needing to know the actual discriminant between personality indices, it was chosen to consider three different categories, also unbalanced among them, to test them.

The experiments used in this chapter considered three different categories defined as follows [167]:
- At-Risk Users (Users who have actively or passively participated in bullying incidents) and Non-Risk Users (Users who have never actively or passively participated in bullying incidents);
- Bully Bullying (Users who have actively participated in bullying but not cyberbullying) and Victim Bullying (Users who have passively participated in bullying but not cyberbullying);
- Total Bullying (Users who actively participated in bullying and cyberbullying) and Total Victimization (Users who passively participated in bullying and cyberbullying).

The pertinent findings and conclusions are based on the tasks and categories listed immediately above for the reasons given above.

Pre-Processing and Feature Extraction

The preprocessing phase allows the generated files to be *"cleaned up"* and extract the *X, Y,* and *Z* coordinates related to the accelerometer sensor. During the testing phase, the brand of smartphones utilized was recorded. As all the phones used were Android devices, the formation of two distinct clusters based on the brands was observed. Fortunately, no issues were observed with the accelerometer sensor in this scenario, as it exhibited similar hardware specifications and sampling rates across the brands. Preprocessing procedures tailored to our study's requirements were employed to ensure accuracy. These extracted data are raw. In the proposed model, the technique used is the Discrete Fourier Transform [168]. This step was used to reduce and standardize the number of samples of each coordinate (X, Y, and Z). The number of transformations applied to each coordinate is 1000, for a total of 3000 samples. This was done to solve the problem of data interoperability. That is, to make the sampling range of sensor data from different smartphones similar.

3.3.2.4.4.2 Classification and Evaluation

The previously extracted data are inputted into the chosen classifiers in the classification stage. When executed, results are returned to be appropriately evaluated in the fourth stage, the evaluation stage. Accuracy, Precision, Recall, and F1-score are considered for evaluating the results. Several classifiers made available by Machine Learning, supervised and unsupervised learning, were used for this project [169].

The models used are *SVM.NuSVC, Bernoulli Naïve Bayes (Bernoulli NB), Random Forest Classifier (RF Classifier)* [170]*, Bayesian Ridge, Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (Bi - LSTM).* The models considered are the result of careful, state-of-the-art analysis. This analysis allowed consideration of these models for the study. Not having full state-of-the-art feedback regarding the topic, basic shallow and deep learning models were considered. The aim was to validate and verify the proposed methodology and validate the dataset. In this case, the information regarding hyperparameters is not included because shallow and deep learning networks and approaches with basic hyperparameters were considered. The goal was to validate the proposed methodology.

3.3.2.4.5 Experimental Setup

The following Section discusses the subdivisions made in detail. The experiments are TEST1 (first phase of testing with the first half of the individuals), TEST 2 (second phase of testing with the second half of the individuals), UNION TEST (Union of TEST 1 and TEST 2), and finally the UNION TEST + BALANCING (Same dataset as previous UNION TEST but carrying out the balancing of classes, i.e., personality indices).

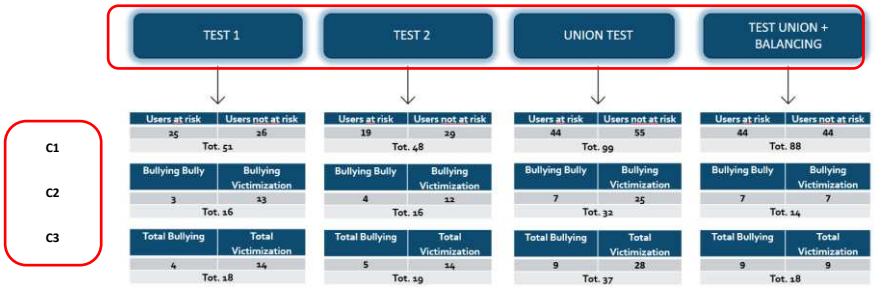


Figure 52 - Design of the four macro-experiments (TEST1, TEST2, UNION TEST, UNION TEST + BALANCING) with three categorizations C1 *"At-risk users vs. non-risk users,"* C2 *"Bullying bully vs. Victimization bullying"* and C3 *"Total bullying vs. Total victimization."*

The design of the four macro-experiments is shown in Figure 52. Thus, ad hoc datasets were created for each test based on the data obtained in the pre-processing phase. *Test1* and *Test2* were derived from the two experiments performed with different users on two days (TEST BASE, i.e., how they were extracted from the users). The experiments UNION TEST and UNION TEST + BALANCING were derived from this baseline. The UNION TEST is a normal consequence that combines both tests. The UNION TEST + BALANCING results from the need to balance machine learning models. As for category divisions, these three were considered: **C1** *"At-risk users vs. Non-risk users,"* **C2** *"Bullying bully vs. Victimization bullying,"* **C3** *"Total bullying vs. Total victimization."*

This choice is dictated by the categorization found by the questionnaire (Personality Index). The failure to balance the categorizations (Personality Index) *Bullying-Bully, Bullying-Victimization, Cyberbullying-Cyberbully, Cyberbullying-Victimization,* and *Non-risk users*, led to the consequent creation of the three derived categorizations C1, C2, and C3. For example, considering the C3 categorization, *"Total bullying"* is the union of Bullying-Bullying and Cyberbullying-Cyberbully personality index users. Victims of bullying and cyberbullying are united in the *"Total victimization"* category. C1 categorization lumps all four classes (*Bullying-Bully, Bullying-Victimization, Cyberbullying-*

Cyberbully, Cyberbullying-Victimization) together in the categorization *"At-risk users"* versus the remaining *"Non-risk users."*

Each classification task was evaluated on the accelerometer sensor data belonging to the following screens of the app in chronological order [5]:

1. **Video1 Activity** ("VIRTUAL ACTIONS, REAL CONSEQUENCES," https://www.youtube.com/watch?v=x2AxcllGLJg&t=4s&ab_channel=TabbyEUproject [5])
2. **Question_Video 1Activity** (5-point Likert scale emotional question);
3. **Video2Activity** ("ANYONE CAN BE ANYONE," https://www.youtube.com/watch?v=z3N24DpD64c [5])
4. **Question_Video 2Activity** (5-point Likert scale emotional question);
5. **Video3Activity** ("INTERNET = EVERYONE, FOREVER," https://www.youtube.com/watch?v=K31Kuc5pTXM&t=42s&ab_channel=TabbyEUproject [5])
6. **Question_Video 3Activity** (5-point Likert scale emotional question);
7. Video4Activity ("VIRTUAL VENDETTA (joke or crime?)," https://www.youtube.com/watch?v=FpBVBwv6UQ4&ab_channel=TabbyEUproject [5])
8. **Question_Video 4Activity** (5-point Likert scale emotional question);

Table 35 defines the semantics of the various tests with the division of users according to each of the three classifications.

C	Test1	Test2	Union Test	Union Test + Balancing
C1	Dataset with 25 "Users at risk" labeled 1 and 26 "Users not at risk" labeled 0, for a total of 51 participating students in the age range of 18 to 24 years	Dataset with 19 "Users at risk" labeled 1 and 29 "Users not at risk " labeled 0, for a total of 48 participating students in the age range of 18 to 24 years.	The dataset had 44 "Users at risk" labeled with label 1 and 55 "Users not at risk" labeled with label 0 for 99 participating students aged 18 to 24.	Dataset with 44 "Users at risk" labeled 1 and 44 "Users not at risk" labeled 0, for a total of 88 participating students in an age range of 18 to 24 years.
C2	Dataset with 3 "Bullying Bully" labeled with label 1 and 13 "Bullying Victimization" labeled with label 0, for a total of 16 participating students in an age range of 18 to 24 years.	Dataset with 4 "Bullying Bully" labeled with labels 1 and 12 "Bullying Victimization" labeled with label 0, for a total of 16 participating students in an age range of 18 to 24 years.	Dataset with 7 "Bullying Bully" labeled with label 1 and 25 "Bullying Victimization" labeled with label 0, for 32 participating students aged 18 to 24.	Dataset with 7 "Bullying Bully," labeled with label 1, and 7 "Bullying Victimization," labeled with label 0, for a total of 14 participating students in an age range of 18 to 24 years old.
C3	Dataset with 4 "Total Bullying," labeled with label 1, and 14 "Total Victimization," labeled with label 0, for a total of 18 participating students in an age range of 18 to 24 years.	Dataset with 5 "Total Bullying," labeled with label 1, and 14 "Total Victimization," labeled with label 0, for a total of 18 participating students in an age range of 18 to 24 years.	Dataset with 9 *"Total Bullying,"* labeled with label 1, and 28 *"Total Victimization,"* labeled with label 0, for a total of 37 participating students in an age range of 18 to 24 years.	Dataset with 9 "Total Bullying," labeled with label 1, and 9 "Total Victimization," labeled with label 0, for a total of 18 participating students in an age range of 18 to 24 years.

Table 35 - Test Configuration Experimentation Classification (C)

This Section reports the results regarding the experimentation carried out in the previous Section. All possible combinations with all different classifiers and all different tasks were considered. Only the best results (in terms of Accuracy and F1-Score) are reported in Table 27 for the sake of readability. For each Macro-experiment (*TEST1, TEST2, UNION TEST, UNION TEST + BALANCING*), tests were conducted for each of the three categorizations (*C1, C2, C3*) for each of the seven Machine Learning models. The test was performed using the *Leave One Out (LOO)* for each user. Only the

best results on a single model are reported in Table 36, where *F1-Score, Accuracy, Precision, and Recall* perform well. C1-categorization is absent because it has bad results for all Experiments.

Screen	Classification	Model	Accu-racy	F1score macro	Macro Pre-cision	Macro Recall	Test
Video1Activity	C3	NuSVC	0.83	0.73	0.77	0.71	Test1
Video3Activity	C2	Bayesian Bridge	0.81	0.64	0.63	0.68	Test1
Video3Activity	C3	Bayesian Bridge	0.82	0.65	0.63	0.70	Test1
Question_ Video1Activity	C3	NuSVC	0.78	0.68	0.70	0.70	Test1
Question_ Video1Activity	**C2**	**Bayesian Bridge**	**0.99**	**0.99**	**0.99**	**0.99**	**Test1**
Question_ Video1Activity	**C3**	**Bayesian Bridge**	**0.99**	**0.99**	**0.99**	**0.99**	**Test1**
Video3Activity	C2	LSTM	0.81	0.64	0.90	0.71	Test2
Video3Activity	**C3**	**LSTM**	**0.89**	**0.80**	**0.94**	**0.75**	**Test2**
Video3Activity	C3	BiLSTM	0.84	0.74	0.77	0.72	Test2
Question_ Video2Activity	C2	LSTM	0.81	0.73	0.76	0.71	Test2
Question Video3Activity	C2	LSTM	0.75	0.70	0.66	0.70	UNION TEST
Video4Activity	C3	NuSVC	0.72	0.71	0.75	0.72	UNION TEST + BALANCING
Question _Video2Activity	**C3**	**Bayesian Ridge**	**0.83**	**0.84**	**0.84**	**0.85**	**UNION TEST + BALANCING**
Question _Video3Activity	C3	NuSVC	0.72	0.71	0.75	0.72	UNION TEST + BALANCING
Question _Video3Activity	C2	BiLSTM	0.71	0.71	0.71	0.71	UNION TEST + BALANCING

Table 36 - The best results were obtained in the four experiments with accelerometer sensors.

Table 36 shows that **C3**, with the categories *"Total Bullying - Total Victimization,"* was found to be more discriminating. The algorithms that returned better results in the tests are the two models, LSTM and Bayesian Bridge. *Question_Video1Activity* with categorization **C2** and *Question_Video1Activity*, *Video3Activity*, and *Question_Video2 Activity* with **C3** achieve the best results with Bayesian Bridge and LSTM models. The implemented RNN models, LSTM and Bi-LSTM, are found to have been efficient on TEST2. Interestingly, the worst classification task was ***C1****: "User at risk vs. Users not at risk."* All the classifiers exhibited values of the adopted metrics between 0.40 and 0.50. Still analyzing the experiments in general, the *Video_Activity* that brought the most discriminant results was the *Video3Activity*. This shows that video #3 of the four proposed in the app guaranteed the most variance in the data. The reverse result regarding *Video2Activity* was the least discriminating (see Figure 53).

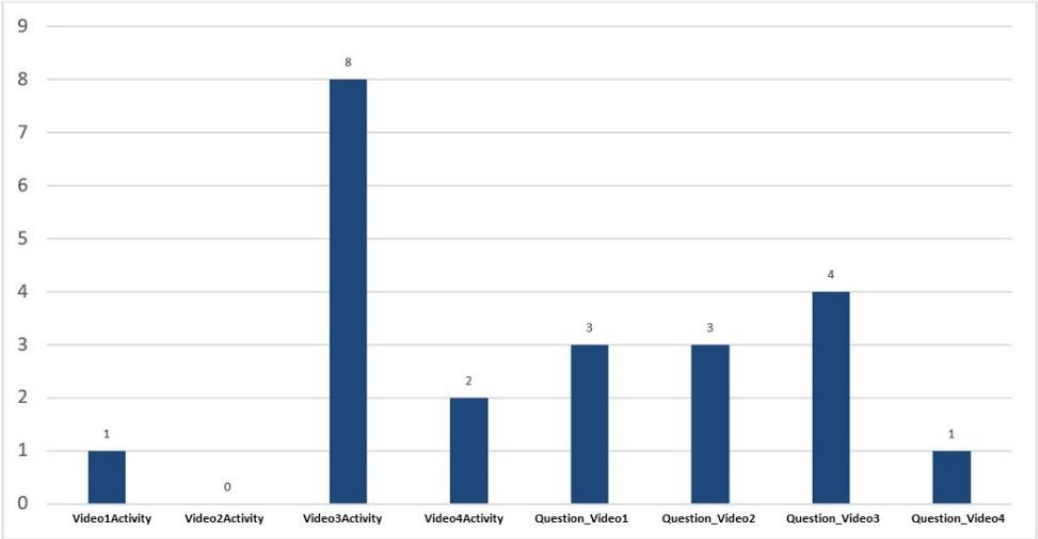


Figure 53 - Best Video Activity Graph

Going into detail, the model that could best distinguish the bullied class from the bully class was the *Bayesian Ridge*. Experimentation, classification, and activities involved are *UNION + BALANCE test, C3 (Total Bullying - Total Victimization), and Question_Video2Activity.* The *Bayesian Ridge* scored 0.84 accuracies. That is, it effectively classified 8 out of 9 victims of bullying and 7 out of 9 bullies. The Union Test was considered less discriminating than the other tests. The difficulty found for the unit test is the balancing solved, then in the *UNION TEST + BALANCING* finding, the best result is with *Bayesian Ridge*.

The study of this chapter focused on several macro-topics: *Behavioral Biometrics, Cyberbullying, Bullying,* and *Machine Learning*. In this work, an Android application consisting of two essential parts was implemented: the first features the watching of four videos on *bullying* and *cyberbullying* (*cartoon scenes*), and the second presents ninety-nine questions on bullying and cyberbullying. The objective of this work is to determine the correlation between the categorization class of the questionnaire (*Personality Index: Bullying-Bullying, Bullying-Victimization, Cyberbullying-Cyberbullying, Cyberbullying-Victimization, Non-risk Users*) and the prediction of Machine Learning (Accelerometer) behavioral models. This experiment focuses only on sensors detected during users' viewing of videos. To test the semantic correlation, three experiments (*TEST1, TEST2, UNION TEST, UNION TEST + BALANCING*) were adopted, considering the three different categorizations: ***C1*** *"At-risk users vs. Non-risk users,"* ***C2*** *"Bullying bully vs. Victimization bullying" and* ***C3*** *"Total bullying vs. Total victimization."*

The results show that the most fruitful automatic categorization is C3; moreover, one specific video (number 3, titled *"INTERNET = ALL, FOREVER"*) was the one that ensured the most significant variance in data among the four videos offered in the app. In contrast, video number 2, titled *"ANYONE CAN BE ANYONE*," was found to be less significant. *Question_Video1Activity* with categorization C2 and *Question_Video1Activity, Video3Activity, and Question_Video2 Activity* with C3 achieve the best results concerning all metrics. Moreover, the results suggest that the Bayesian Ridge model and LSTM are the best machine-learning approaches. The implemented RNN models, LSTM and Bi-LSTM, are found to have been efficient on TEST2.

This work helps parents/guardians or schoolteachers understand a child's behavior with this Smartphone Android Questionnaire. The results obtained from this work form the basis for possible future developments in the same research area. The models used are not recent. The purpose of the work is to validate the proposed methodology. There needs to be feedback on what kind of model to use at the state of the art. Therefore, the basic shallow and deep models were used appropriately. Future work developments could be manifold. One attractive option would be to use overlooked algorithms (multi-modal learning, Multiview learning, etc.) and expand the dataset to include underage students. In addition, increasing the dataset could provide an opportunity to conduct statistical analysis to give more significance to the experiment. Regarding sensitive data, techniques such as Elliptic Curve Cryptography (ECC) and AES [69] could be implemented.

This work could engage researchers with expertise in data analysis to address the problem of bullying and cyberbullying more effectively and raise awareness of this issue. Our work checks the correlation through Machine Learning approaches by reading the psychologists' literature in Chapter Dataset. Indeed, as a result, managed to get a finding that should be further investigated.

3.3.3 Experiment Strategies – Comparison of University Student-School Student

This subchapter compares the work done in previous chapters with the results of tests conducted on minor students. Specifically, data from the Avellino and Cagliari datasets were combined and considered *'School Student,'* while the Bari dataset was classified as *'University Student.'* Results from the chapter (3.3.2 Experiment Strategies) conducted with Mayan databases, were considered and compared with minor datasets."

The Sub-Chapter *3.3.3.1 Human Activity Recognition for Identifying Bullying and Cyberbullying: A Comparative Analysis Between School and University Student*, compares the detected human activities of the University Student and School Student during the questionnaire test.

The Sub-Chapter *3.3.3.2 Anomaly Detection using smartphone Sensors & Fixed Tasks for Continuous Authentication*, compares the detected anomalies of the University Student /School Student and the touch values by questionnaire test.

3.3.3.1 Human Activity Recognition for Identifying Bullying and Cyberbullying: A Comparative Analysis Between School and University Student

The smartphone is an excellent source of data. Sensor values can be extrapolated from smartphones. This work exploits *Human Activity Recognition (HAR)* models and techniques to identify human activity performed while filling out a questionnaire that aims to classify users as Bullies, Cyberbullies, Victims of Bullying, and Victims of Cyberbullying. The work aims to identify activities related to the questionnaire class other than just sitting. The work starts with a state-of-the-art analysis of HAR to arrive at the design of a model that could recognize everyday life actions and discriminate them from actions resulting from alleged bullying activities (Questionnaire Personality Index). Five activities were considered for recognition: *Walking, Jumping, Sitting, Running,* and *Falling*. The best HAR activity identification model was applied to the dataset obtained from the *"Smartphone Questionnaire Application"* experiment to perform the analysis. The best model for HAR identification is CNN.

Identifying and recognizing human activities is called *Human Activity Recognition (HAR)*. These actions can vary, such as walking, sitting, falling, etc.[171]. Each requires *Artificial Intelligence (AI)* algorithms to analyze and classify raw data collected from devices like smartphones or smartwatches. According to [172], the latter has sensors that can capture data during activities. This data can classify and recognize activities [172].

These devices can monitor users' mental and physical states with their sensors. Within an *IMU (Inertial Measurement Unit)*, an electronic device consists of accelerometers, gyroscopes, and occasionally magnetometers; these sensors are combined. To determine the number of axes on which the measurement of each sensor used is made, an n-axis IMU can be created, where the number of axes is the sum of the axes on which the measurement of each sensor is made. For example, when a three-axis accelerometer and a three-axis gyroscope are integrated, this is called a six-axis IMU. If the magnetometer is included, speak of a 9-axis IMU [173], [174].

One of the fundamental sensors is the triaxial accelerometer, but other sensors, such as the triaxial gyroscope and magnetometer, are also used.

Specifically:

1. The *Accelerometer* detects linear motion and gravitational force by measuring acceleration along the three axes: X, Y, and Z;
2. The *Gyroscope* measures the rate of rotation of a body about the X, Y, and Z axes;

3. The *Magnetometer* is used to detect and measure geomagnetic fields, which are only sometimes useful for HAR purposes; therefore, it is only sometimes included among the sensors used [155].

Research on HAR is advanced, but few studies have recognized bullying actions over other activities. Most studies have focused on recognizing actions of physical violence, such as punching or pushing [175]. For this reason, datasets for recognizing bullying activities are less common than those for recognizing more general human activities. A fall can indicate a bullying action experienced [175]. A real-world example is a boy falling due to direct pushes or hits.

This paper focused on the recognition of bullying activities such as falling, an activity little considered in other studies, and other activities in line with other studies in the field. This paper aims to identify activities performed during the completion of a questionnaire by an experimental group of School Student s School Student and about 16 years old and an experimental group of University Student s University Student and about 19 years old in Italy.

The study aims to achieve several objectives.

1. To understand and study the most widely used techniques in the state of the art of Human Activity Recognition and then use them for Bullying Detection systems, which deal with identifying and recognizing cases of Bullying.
2. To compare the results obtained on this experimental group of under-18-year-olds and analyze any differences from the University Student group [176].

To achieve these goals, the literature was studied to understand the typical architecture of a Human Activity Recognition system, commonly used sensors, and the most relevant activities to be recognized. Next, dataset creation techniques were studied, and a public dataset was considered. Once this information was understood, a dataset containing data generated by the triaxial accelerometer for five different activities (*Walking, Running, Jumping, Sitting, and Falling*) was performed by nineteen participants. This dataset, named *"Uniba HAR Dataset,"* was later compared with one of the best-known public datasets in the literature, *UniMiB SHAR*. The aim is to distinguish activities resulting from bullying from *Activities of Daily Living*. Next, SHAR studied smartphone-based methods to combine the activities in the datasets to the final label given by the questionnaire (*Bully, Cyberbully, Victim of Bullying, and Victim of Cyberbullying*).

3.3.3.1.1 State of Art

Several studies on Human Activity Identification are viewed and analyzed in this section. A new type of human activity detection approach is considered that combines potential abnormal activities performed while filling out a questionnaire to identify attitudes associated with bullying or cyberbullying by the individual.

Many studies still need to include the feature extraction and selection phase; some examples of this type of study are given below. The investigation [177] uses a triaxial gyroscope and accelerometer to recognize the daily actions of thirty volunteers. The researchers [177] examined daily life actions, such as *Lying Down, Standing, Sitting, Walking,* and *Going Down* or *Upstairs*, including *Decision trees, Random forests, and K-nearest neighbors*. Some classifiers have excellent performance with up to 98 percent accuracy without implementing a feature selection and extraction step.

Testing is done using three public datasets: the *UMAFall* [178][179] and *SisFall* [180]. The latter is a publicly available dataset that is only sometimes considered because it uses only Inertial Measurement Units to collect data.

The study presents An interesting example [181] in which a 75% overlap is applied to a two-second sliding window. According to the study, there are two types of actions: static actions, in which the global coordinates of the body do not change, and dynamic actions, in which the whole body is in motion. Ten users were considered. They were each given a smartphone to hold in their dominant hand and instructions on what to do. The smartphone's triaxial accelerometer (either a *Samsung Galaxy* or an *LG Nexus*) collected data at 50 Hz.

The classification phase took place with the implementation of many classifiers, including *Support Vector Machine, Decision Tree,* and *Naïve Bayes*. The latter two performed the best by achieving 98% average accuracy between static and dynamic task recognition.

Studies concerning the recognition of Bullying or Violence

The vision-based approach can identify bullying or violent actions in human activity detection. According to the study [182], sensor-generated data are more difficult to obtain. There are two categories of data: *RGB* and *RGB-D*. The second type of data offers higher accuracy than RGB data but is less used because it is more complex and expensive. This study also emphasizes how vital the pre-processing and feature extraction stages are. Classifiers are trained on datasets consisting of movies or frames to determine whether there are violent situations in the described scene. After a thorough analysis by the authors regarding several classifiers, the most efficient one turns out to be CNN, with an accuracy ranging from 93.32% to 97.62%.

Researchers have combined sensor-based activity recognition and voice tone emotions to recognize activities [183]. In this study [184], sensors used in smartphones and smartwatches were used for sensor-based activity recognition. To reduce the size of the feature matrix, some features were extracted a priori and then selected using the principal component analysis algorithm. *Mel Frequency Cepstral Coefficients (MFCC)*, a representation of the short-term power spectrum of a sound, were used to calculate features for emotion recognition by voice. The classifier used is a K-Nearest Neighbors. Cross-validation results showed 77.8 percent accuracy for sensor-based systems and 81.4 percent for audio-based systems.

Also considered were studies in which public datasets were used. The first example considered is the study [185] that used the *UMAFall* and *SisFall Datasets* created with 38 volunteer participants. The data created were then divided: 20% is devoted to examination, 30% to validation, and 50% to training. In pre-processing, activities are assigned a label to determine whether they are daily activities or falls. As a result, the classification is binary, and the number of instances in the classes needs to be more balanced. *UMAFall* is used only with these class-unbalanced methods with binary classification in the fall detection domain.

Some studies, also conducted by the authors of this paper, have used smartphone-based sensor methods to deal with cyberbullying. Indeed, the smartphone is an excellent source of information. An anomaly detection analysis characterized by human behavior can be performed using smartphone sensor values using machine learning techniques.

The researchers [176] use *Human Activity Recognition (HAR)* models and techniques to identify human activity performed while filling out a questionnaire using a smartphone application that aims to classify users as *Bullies, Cyberbullies, Bullying Victims,* and *Cyberbullying Victims*. The work aims to discuss an innovative smartphone method that integrates the results of the cyberbullying and bullying questionnaire (*Bully, Cyberbully, Bullying Victim,* and *Cyberbullying Victi*m) and the human activity performed. At the same time, the individual fills out the questionnaire. The work begins with state-of-the-art analysis of HAR to arrive at the design of a model capable of recognizing actions of

daily life and distinguishing them from those that might result from alleged bullying. Five activities were considered for recognition: *Walking, Jumping, Sitting, Running, and Falling* [186]. The best HAR activity identification model is applied to the dataset derived from the *"Smartphone Questionnaire Application"* experiment to perform the previously described analysis. The work presented in the following paper is the precursor to the one presented [176].

Another work analyzed is that of [187] which analyzes a method of *Detection Anomaly,* an essential process for identifying a situation different from the ordinary. The following study analyzes anomalies in the human behavioral domain observed when filling out a questionnaire on bullying and cyberbullying. This work analyzes smartphone sensor data (*Accelerometer, Magnetometer,* and *Gyroscope*) to use anomaly detection techniques to identify anomalous behaviors used during questionnaire completion in an Android application. To understand any polarizing content suggested during the application and identify users who exhibit abnormal behaviors, which could be expected of classes of users, psychology and computer science work together to analyze and detect any latent patterns within the dataset under consideration [187].

3.3.3.1.2 Dataset
The datasets used during this experiment are as follows.

Dataset HAR Uniba: This dataset was created in a controlled environment with nineteen participants, thirteen males and six females [176]. Each participant performed eight different actions divided into the two categories previously described:

ADL:
- *Walking;*
- *Running;*
- *Hopping;*
- *Sitting;*
- *Falling (forward, backward, right, and left).*

Only accelerometer-generated data were collected using a smartphone placed in each participant's right pocket, with the screen facing the body. This was done at a sampling rate of 200 Hz using the Android application. The collected raw data were then sent to a server in *TXT format* to be reprocessed and converted to *CSV format*. Each task was performed several times (two or three). Each trial was 15 seconds long. The falls were performed on a mattress placed on the floor.

The CSV file is divided into three columns: the first column contains the user ID, the second contains the activity performed, the third contains the Timestamp in milliseconds from the start of the action, and the next three contain the three triaxial accelerometer values X, Y, and Z.

Dataset HAR Uniba Resampled: A resampled version of the Dataset HAR Uniba.

UniMiB SHAR: Dataset consisting only of values obtained from the accelerometer. Recognized activities are again divided into two categories:

Falling:
- Falling forward;
- Falling backward;
- Falling to the right;
- Falling to the left;

- Falling by hitting an obstacle;
- Falling with protective strategies;
- Falling backward without protective strategies;
- Syncope.

ADL:
- Walking;
- Running;
- Climbing stairs;
- Descending stairs;
- Jumping;
- Lying down from a standing position;
- Sitting.

For each activity, there are 2 to 6 trials for each user. For the actions with two trials, the smartphone in the right pocket is used in the first and the left in the second. For actions with six trials, the first three have the smartphone in the right pocket and the others in the left pocket. Data are provided in windows of 51 or 151 samples around an original signal peak higher than 1.5g, with g being the acceleration of gravity. The best results obtained in the experimentation performed from this dataset are with a K-NN in the ADL-only category. In all datasets, all activities are considered anomalous except the sitting activity.

3.3.3.1.3 Methods

The machine learning models used are:
- **CNN**, a convolutional, feed-forward neural network, consisting in this case of 3 ReLU layers, each alternating with a pooling layer for simplifying the output obtained from the previous layer, whose practical goal is to reduce the number of parameters the network must learn. After these, a flattened layer is used for linearizing the output, and a SoftMax layer is used for the classification.
- **LSTM**, a type of RNN, differs from CNN networks because of the addition of *feedback* layers, whose peculiarity is the ability to learn from long-time sequences and then maintain memory. This network is structured by 2 LSTM layers, alternating with a Dropout layer. Then, the actual layers are devoted to the prediction of relu and SoftMax.
- **Bi-LSTM** is a particular type of LSTM network that is practically trained to make predictions not only on past knowledge but also on future knowledge and then go backward with the predictions. Unlike the LSTM network, which can learn unidirectionally.

Each of these models was used in different combinations with the various datasets. Starting with CNN, it was noted that remarkable performance was achieved compared with LSTM and Bi-LSTM.

3.3.3.1.4 Experimentation

The experiment is organized into several stages:
- Data pre-processing;
- Extraction of data generated by the sensors;
- Classifier Training;
- Activity recognition;
- Comparing results with data from users School Student with results from users University Student.

The pre-processing phase was carried out to prepare the data for further processing to simplify the classification of activities. Specifically:

- Activities that were not relevant (in the case of *UniMiBSHAR*) were removed;
- Activities were renamed and grouped into a more manageable and meaningful set. For example, the various types of falls (right, left, front, and back) were aggregated into a single activity representing each type of fall;
- The data were appropriately adapted into a three-dimensional form for input into a CNN.

The second phase of the experiment involved a data feature extraction phase by which only the accelerometer values were extracted from the text files from the sensors of the devices used by the users when filling out the questionnaire. Values generated by the magnetometer and gyroscope were then discarded. The row of interest (X, Y, and Z coordinates) was placed within a data structure offered by the Python Pandas library, called a data frame. Each coordinate triple (X, Y, Z) also specified the questionnaire screen in which that movement was recorded. This procedure obtained a series of CSV files used in the following steps to make predictions. The entire process was repeated for all users participating in all test phases (Test1, Test2, Test3) except a few who needed to be considered because they still needed to complete the entire questionnaire.

In phase three, the chosen HAR classifiers are trained with three HAR datasets: *UniMiB SHAR* [179], *Dataset HAR Uniba*, and *Dataset HAR Uniba Resampled*, a sampled version of Dataset HAR Uniba. In the fourth phase, through the HAR model, the activities performed using the results of the first phase are predicted for each user. The fifth phase compares the results obtained on University Student and those obtained on School Student s under 18. This way, the differences between the two groups of participants are analyzed.

Our research methodology focused on evaluating a variety of Deep Learning datasets and model combinations to determine the most effective configuration. Various model and dataset combinations were used to evaluate their performance to complete the offline training phase. Table 37 shows the results, including the overall users' accuracy averages and F1-scores. It is critical to note that these values represent the averages for each user in the dataset, which was trained using the Leave One Out technique. The latter technique ensures a reliable training process in which each user is isolated during model training. This allows evaluation of the system's performance under more realistic and generalizable conditions. In this way, a comprehensive assessment of the predictive capabilities of the model for the dataset in question is provided.

Model	Dataset	AverageAccuracy	Average F1-Score
CNN	Dataset HAR Uniba	0,915	0,901
CNN	UniMiBShar	0,998	0,996
CNN	Dataset HAR Uniba Resampled	0,659	0,459
LSTM	Dataset HAR Uniba	0,865	0,819
Bi-LSTM	Dataset HAR Uniba	0,891	0,855

Table 37 - Accuracy results and F1-Score averages

Below is a comparison of the results obtained by School Student and University Student (Figure 55-58). Legend (Figure 54 - Figure 55): *Bullying (blue), Victim of Bullying (orange), Cyberbully (gray), Victim of Cyberbullying (yellow).*

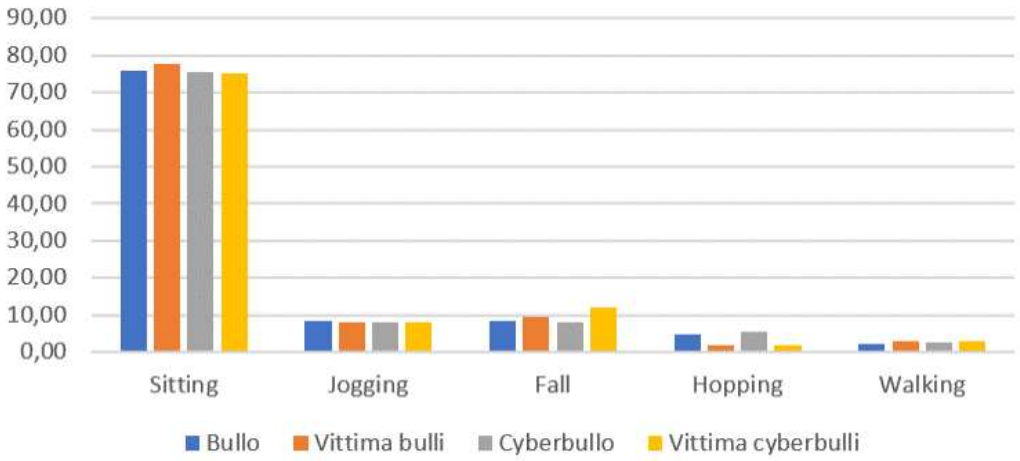


Figure 54 - Recognized activities for School Student .

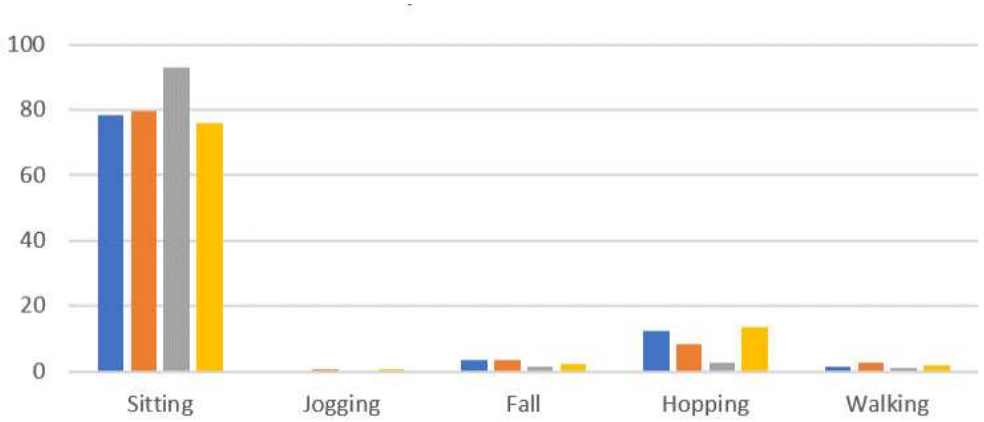


Figure 55 - Recognized activities of University Student years by category

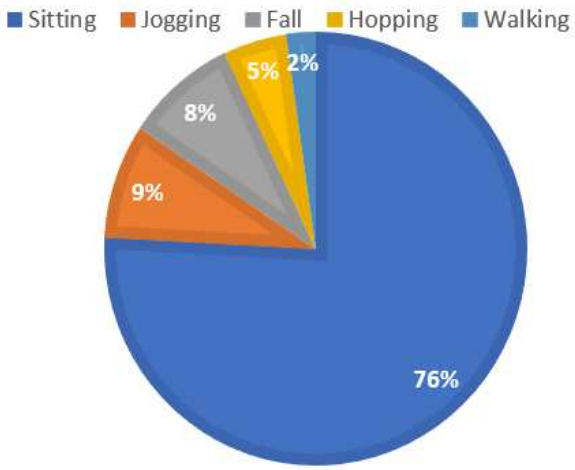


Figure 56 - Activity percentages of School Student

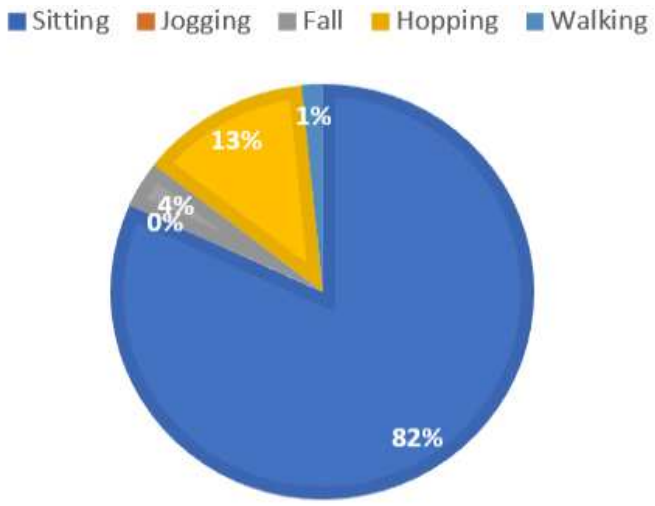


Figure 57 - Percentages of University Student .

Looking at the results obtained, some psychological and scientific observations could be made:

- **Cyberbullies** School Student showed abnormalities in the initial part of the questionnaire, with jumping, falling, and running activities while watching videos related to the feelings they experienced. **Cyberbullies** University Student , on the other hand, were considered "quieter" as they spent most of their time sitting in higher percentages than the other categories;
- **Bullies**, among high schoolers School Student , showed running and falling as prevalent abnormal activities. Among those University Student , on the other hand, the prevalent abnormal activity shown was jumping. These abnormal activities were recorded, particularly during the questionnaire phases composed of the *QuizActivityButtons/DomestionsVideos*;
- **Bully victims School Student showed more remarkable abnormal phases** during *QuizActivityButtons* but with prevalent falling and running activities. For those University Student , the abnormal phase is mainly characterized by jumping activities;
- The **Victims of CyberBullies** School Student , as well as the bullies, recorded falling and running as their main abnormal activity, particularly during the *QuizActivityButton/DomestionsVideo* phase. Victims University Student , on the other hand, were the only users who showed abnormalities in the early part, as well as **Cyberbullies** in our experiment.

In general, frequent abnormal running and falling activities were observed in the sample of School Student , while hopping activities were less frequent and walking activities were almost nonexistent. Among University Student years, on the other hand, the predominant abnormal activity recorded was hopping, while the other activities, except sitting, were infrequent.

In addition, the questionnaire could be reduced to only those questions about the category of **bully** to be identified. In both experiments, in the case of **bullies and victims** of bullies, the *QuizActivityButton* activities would suffice, thus removing the initial part of submitting videos and related questions.

For the **Bullies School Student** , questions regarding the running activity were more discriminating:

- *Bullying*: the frequency with which bullying was done (stealing items, discriminating against someone because of disability/skin color/sexual orientation, pushing);
- *Cyberbullying*: in what environment it occurs, how many incidents have been suffered, and how many text messages containing insults have been received.
- For the fall activity, questions regarding.
- *Habits*: rules imposed by parents and control internet use, whether cyberbullying has been discussed in the classroom.

- *Bullying*: If you intervene if you see a case of bullying in action, how often you were bullied and the frequency you were bullied (Physical, Verbal, Behavioral, and Threats).

Victims of bullying School Student recorded abnormal fall activities in responding to questions concerning:

- *Bullying*: whether one has been subjected to threats, exclusion, physical violence, false rumors about oneself, discrimination because of one's skin/culture, theft, or damage to one's belongings.
- *Cyberbullying*: the frequency with which you have been bullied through messages, media, websites, and e-mails; the frequency with which you have been ignored online; whether someone has impersonated you, received false news about you, impersonated you on social media or with your address book contacts.

Regarding Cyberbullies School Student , questions regarding the following were found to be discriminating for the fall activity.

- *Bullying*: whether one has been subjected to threats, exclusion, physical violence, false rumors about oneself, discrimination because of one's skin/culture, theft, or damage to one's belongings.

For the running activity, the questions are covered.

- *Cyberbullying*: the frequency with which acts of cyberbullying such as threatening and abusive texting/emails, sending violent, embarrassing, or intimate media, threatening online, making silent or intimidating phone calls, and spreading false rumors about other people.

For the skipping activity, the questions covered:

- Emotions were felt when watching some videos and asking general questions.

For the **Cyberbullies University Student** , questions regarding:

- *Cyberbullying*: specifically, the frequency with which acts of bullying were carried out through messages, media, phone calls, and e-mails.

Victims of Cyberbullying School Student recorded abnormal running activities for the following questions:

- *Cyberbullying*: the frequency with which acts of cyberbullying such as threatening and abusive texting/emails, sending violent, embarrassing, or intimate media, threatening online, making silent or intimidating phone calls, and spreading false rumors about other people.

In addition, abnormal fall activities have been recorded:

- *Bullying*: threats, exclusion, physical violence, false rumors about oneself, discrimination because of one's skin/culture, theft of or damage to one's belongings were experienced.
- *Cyberbullying*: how often you have been bullied via messages, media, websites, and e-mail, how often you have been ignored online, whether someone has impersonated you, received false rumors about you, impersonated you on social

Victims of cyberbullying University Student , on the other hand, recorded abnormal activities during the initial part concerning The emotions felt when watching some videos and general questions.

In this work, *Human Activity Recognition (HAR)* models are used to examine the behavior exhibited by School Students School Student and University Student s University Student while performing a questionnaire. The distinguishing feature of this research is the use of accelerometer sensors built into smartphones, which allow us to record users' behaviors in detail.

This technique allowed us to analyze the actions of individuals, distinguishing between different movements and actions. They are considered *"abnormal,"* all those behaviors beyond simply sitting, focusing on more dynamic or unusual activities or movements that could provide significant information about students' emotional state or active participation while filling out the questionnaire.
By using HAR models in this context, it was possible to better understand the behavioral patterns formed during the interaction with the questionnaire and find attractive cues for evaluating the responses.

In this research, accelerometer technology was used innovatively, allowing for improved analysis capabilities and offering new perspectives for interpreting and monitoring student behavior in scientific research contexts.

The experiments can be concluded by saying that the goals were achieved, namely:

- It was possible to use appropriate DL models to perform HAR on data generated by sensors on a smartphone, all after comparing different models and datasets to achieve optimal accuracy. The best performance was obtained with a *Convolutional Neural Network (CNN) with Dataset HAR Uniba* and *UniMiBSHAR datasets*;
- The results obtained in this experiment on an experimental group of School Student were compared with those obtained from University Student years.

Participants School Student did a lot more running and falling, but they did less jumping than their University Student counterparts, who said jumping was one of the most frequent unusual things they did.

Both groups spent most of their time sitting, with 76 percent of participants School Student and 82 percent University Student . The two groups spent the same amount of time on other activities, except jumping, an unusual behavior commonly seen by the School Student participants but never seen by the University Student participants.

This view allows us to observe how the under-18-year-olds are particularly active and active compared to their University Student peers. This difference in behavior could be because young people are more sensitive and concerned about bullying. Their greater involvement in the complex social dynamics that foster bullying could be the reason for this interest.
Implementing more advanced cybersecurity measures, such as data encryption, could be expected. More excellent protection of sensitive information can be ensured through modern techniques, helping preserve privacy and prevent harmful phenomena such as online bullying. In this context, cybersecurity becomes essential to ensure a safer and more secure digital environment, particularly considering how actively young people are involved in online activities [188].

3.3.3.2 Anomaly Detection using smartphone Sensors & Fixed Tasks for Continuous Authentication

In this subchapter, data obtained from the sensors and touch of smartphones (*after using the appropriate questionnaire app*) were analyzed to apply Machine Learning techniques helpful in detecting any abnormal behaviors adopted while using the app.

3.3.3.2.1 Experimental Summary Chapter 5.2.2

The objective of the work carried out in the chapter (*3.3.2.2 Anomaly Detection Using Smartphone Sensors for a Bullying Detection*) was to analyze and detect any latent patterns within the dataset under consideration (specifically, data obtained from smartphones' sensors via a unique app), to understand any polarizing content proposed during the use of the app and to identify users who exhibit anomalous behaviors, possibly common to classes of users. The data collected by the sensors are

inherent to four videos regarding bullying/cyberbullying issues shown to users (users from the University of Bari). For each video, the user answered a question inherent to the specific video, after which the user was asked 83 general questions.

Dataset

The dataset used for this experiment included 164 .txt files with the nomenclature *sensor_userid*, where user-id represents the user's ID. In the single file, the data for the accelerometer, magnetometer, and gyroscope sensors are found, each sampled every 20 milliseconds. The sensor's name has been concatenated with the name of the task where it was sampled. Also specified at the beginning and end of each line is the date and time the sensor data was recorded. This data was recorded using a client-server approach, where the client was the Android device that sent its sampled data to the server.

However, this dataset had some anomalies, including:

- *Duplicate sensors*: some users sampled one sensor twice as many times as others;
- *Asynchronous sensors*: some users, due to the client/server architecture, did not send all 3 sensors at the same time, thus resulting in a file with a total number of sensor rows that differed from each other;
- *Missing sensors*: 5/99 users registered a file with one or more missing sensors and consequently were removed from the dataset;
- *Missing activities*: 18/99 users did not answer the questions; for these users, the questions they were able to answer before the connection broke were considered.

Therefore, special Python scripts were designed first to convert the raw data collected from the sensors into a .csv format (pre-adjustment phase) and then clean the data by averaging the issues revealed in the above list (post-adjustment phase).

Scripts

Among the scripts used by the students previously mentioned in the source project work, in the scripts folder:

- *script_from_txt_to_csv_final(Videos)*: allows to generate, from a .txt file containing raw data, 7 .csv files (Figure 58):

```
for file in os.listdir(path_dataset):
  user = file.split("sensor_")[1].replace('.txt', '')
  print("Current User: ", user)

  df = pd.read_csv(file, delimiter = "-", skipinitialspace = True, names = ["DATE\TIME","ACTIVITY","DATA"]).drop(["DATE\TIME"], a

  df1 = pd.DataFrame(columns = ['X_{sens}'.format(sens = sensor) , 'Y_{sens}'.format(sens = sensor), 'Z_{sens}'.format(sens = se

  df['mask_df1'] = np.where(df['ACTIVITY'].str.contains(word, na=False), True, False)
  acc_activities = df[df['mask_df1'] == True]
  acc_activities.reset_index(inplace=True, drop=False)

  i = 0
  for item in acc_activities['ACTIVITY']:
    f1, f2, f3, f4 = acc_activities['DATA'][i].split(' ')
    f1, f2, f3 = f1.split(',')
    df1.loc[i] = [f1, f2, f3]
    i += 1

  x = np.array(df1['X_{sens}'.format(sens = sensor)])
  x1 = np.pad(x, (0, max-len(x)), 'constant')
  y = np.array(df1['Y_{sens}'.format(sens = sensor)])
  y1 = np.pad(y, (0, max-len(y)), 'constant')
  z = np.array(df1['Z_{sens}'.format(sens = sensor)])
  z1 = np.pad(z, (0, max-len(z)), 'constant')

  df2 = pd.DataFrame(columns = ['X_{sens}'.format(sens = sensor) , 'Y_{sens}'.format(sens = sensor), 'Z_{sens}'.format(sens = se

  df2['X_{sens}'.format(sens = sensor)] = x1.tolist()
  df2['Y_{sens}'.format(sens = sensor)] = y1.tolist()
  df2['Z_{sens}'.format(sens = sensor)] = z1.tolist()

  w1, w2 = str(file).split(".")
  #Sta salvando i primi files ".csv" pre_adjustment, cambiare se cambia l'albero delle cartelle
  df2.to_csv('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/exp/{user}/{ac}/{word}_{ac}_{sen}_pre_adjustment
```

Figure 58 - Portion of code to generate pre-adjustment .csv files (video)

This script was explicitly used for only the rows of the file containing the data for the videos.

- *script_from_txt_to_csv_final(Quiz)*: a separate script was used for quizzes (Figure 59):

```
if not rows_acc == 0:
  #Da cambiare se è cambiata la cartella di output dello script
  os.makedirs ('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/exp/'+ user + "/" + activity2 + "/", exi
  word = '{sens}_{act}'.format(act= activity2, sens= sensor)

  #Da cambiare se è cambiata la cartella dove stanno i files sensors degli utenti
  df = pd.read_csv('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/sensors/sensor_{us}.txt'.format(us =
  df1 = pd.DataFrame(columns = ['X_{sens}'.format(sens = sensor) , 'Y_{sens}'.format(sens = sensor), 'Z_{sens}'.format(sens

  df['mask_df1'] = np.where(df['ACTIVITY'].str.contains(word, na=False), True, False)
  acc_activities = df[df['mask_df1'] == True]
  acc_activities.reset_index(inplace=True, drop=False)

  i = 0
  for item in acc_activities['ACTIVITY']:
    f1, f2, f3, f4 = acc_activities['DATA'][i].split(' ')
    f1, f2, f3 = f1.split(',')
    df1.loc[i] = [f1, f2, f3]
    i += 1

  x = np.array(df1['X_{sens}'.format(sens = sensor)])
  #x1 = np.pad(x, (0, max-len(x)), 'constant')
  y = np.array(df1['Y_{sens}'.format(sens = sensor)])
  #y1 = np.pad(y, (0, max-len(y)), 'constant')
  z = np.array(df1['Z_{sens}'.format(sens = sensor)])
  #z1 = np.pad(z, (0, max-len(z)), 'constant')

  df2 = pd.DataFrame(columns = ['X_{sens}'.format(sens = sensor) , 'Y_{sens}'.format(sens = sensor), 'Z_{sens}'.format(sens

  df2['X_{sens}'.format(sens = sensor)] = x.tolist()
  df2['Y_{sens}'.format(sens = sensor)] = y.tolist()
  df2['Z_{sens}'.format(sens = sensor)] = z.tolist()

  file_aux = "sensor_" + user + ".txt"
  w1, w2 = str(file_aux).split(".")
  #Da cambaire se cambia l'albero delle cartelle di output, sta salvandi i files pre_adjustment
  df2.to_csv('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/exp/{user}/{ac}/{word}_{ac}_{sen}_pre_adju
```

Figure 59 - Portion of code that allows you to generate pre-adjustment .csv files (quizzes)

- *script_models_videos*: This script applied machine learning models to the data generated by the script script_from_txt_to_csv_final(Videos). Precisely, one is placed in the folder related to the .csv file of interest (the one obtained by merging the post-adjustment .csv files), and a results_videos folder is created, within which a subfolder is created for each Machine Learning algorithm used. Specifically, the following were used: Isolation Forest, Local Outlier Factor, Elliptic Envelope, and One Class SVM (Figure 60).

```
outliers_fraction = 0.1
random_state = 0
support_fraction = 0.7



model_EllipticEnvelope = EllipticEnvelope(contamination=outliers_fraction, random_state=random_state, support_fraction=support_fr

c_users=0
for user in list_users:
  print("Current User: ", user)
  os.makedirs('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/results_videos/Elliptic_Envelope/' + user + "/
  for activity in activities:
    print("Current Activity: ", activity)
    os.makedirs('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/results_videos/Elliptic_Envelope/' + user +
    #Percorso da cui si prende il file "finale" per ogni utente per ogni attività video, da cambiare se il nome della cartella d
    path = 'C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/exp/' + user + '/' + activity + '/sensors_' + use
    df = pd.read_csv(path, delimiter = ",", skipinitialspace = True)
    df1 = df.loc[:, ['X_ACCELEROMETER', 'Y_ACCELEROMETER', 'Z_ACCELEROMETER','X_MAGNETOMETER', 'Y_MAGNETOMETER', 'Z_MAGNETOMETER

    preds = pd.DataFrame(model_EllipticEnvelope.fit_predict(df1), columns = ['PREDICTION'])
    preds.to_csv('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/results_videos/Elliptic_Envelope/{user}/{ac
  c_users=c_users + 1

print("Total Users: ", c_users)
```

Figure 60 - Portion of code where the Elliptic Envelope algorithm is applied

- *script_models_quiz*: This script was used to apply machine learning models to the data generated by the script script_from_txt_to_csv_final(Quiz) (Figure 61).

```
outliers_fraction = 0.25


#list_preds = list()
model_IsolationForest = IsolationForest(contamination=outliers_fraction)

c_users=0
for user in list_users:
  print("Current User: ", user)
  os.makedirs('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/results_quiz/Isolation_Forest/' + user + "/",
  for activity in activities[c_users]:
    print("Current Activity: ", activity)
    os.makedirs('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/results_quiz/Isolation_Forest/' + user + "/"
    #Percorso da cui si prende il file "finale" per ogni utente per ogni attività quiz, da cambiare se il nome della cartella da
    path = 'C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/exp/' + user + '/' + activity + '/sensors_' + use
    df = pd.read_csv(path, delimiter = ",", skipinitialspace = True)
    df1 = df.loc[:, ['X_ACCELEROMETER', 'Y_ACCELEROMETER', 'Z_ACCELEROMETER','X_MAGNETOMETER', 'Y_MAGNETOMETER', 'Z_MAGNETOMETER

    preds = pd.DataFrame(model_IsolationForest.fit_predict(df1), columns = ['PREDICTION'])
    preds.to_csv('C:/Users/guido/Desktop/SistemiBiometrici/Esperimenti/Esperimento_1/results_quiz/Isolation_Forest/{user}/{ac}/p
  c_users=c_users + 1

print("Total Users: ", c_users)
```

Figure 61 - Portion of code where the Isolation Forest algorithm is applied

3.3.3.2.2 Experimentation Summary Chapter 5.2.1

Inherent to the work examined (3.3.2.1 Fixed Tasks for Continuous Authentication via Smartphone), the following objectives were specified:

- To test the efficiency of a set of ML algorithms in authenticating several users through biometric traits (e.g., dynamics of touch screen use) and to understand which tasks are best.
- The second objective is the classification of bullying and cyberbullying. Data collected from School Student s were used for this experiment. These data were then compared with those from the University of Bari students. During the experiment, users had to fill out questionnaires for the classification of bullying and cyberbullying. For this very reason, the raw data contains values related to generic actions that can be associated with the execution of free tasks.

Features

As mentioned earlier, the features in this case are minor because the raw data contains values obtained from free tasks during questionnaire completion.

- Coordinates (x, y) of tap: tap start coordinates;
- Pressure: pressure of the tap;
- Surface (mm2): the surface of the finger;
- Duration (ms): duration of the tap;
- Acceleration (m/s2): acceleration recorded at the moment immediately following the tap;
- Rotation (rad/s): rotation recorded at the moment immediately following the tap;
- MagneticField (Asp/m): magnetic field recorded at the moment immediately following the tap;

Treating the data as generic gestures became necessary since it was essential to know their type.

Scripts

It contains the following Scripts in the project:

- *DataExtraction*: Script that transforms raw data obtained from the mobile device into a table format-oriented data manipulation.
- *FeatureExtraction*: Feature extraction is extracting relevant features from raw or unstructured data to create a more compact and meaningful representation of the data. In other words, feature extraction involves identifying the data's most important or distinctive features and converting them into a form suitable for processing by a machine learning or artificial intelligence algorithm. Such a script was used to obtain only the features of interest for our case study, i.e., the x, y, and z coordinate values detected by the sensors.
- *roceauc*: In statistics, ROC (Receiver Operating Characteristic) and AUC (Area Under the Curve) curves are tools used to evaluate the performance of a binary classification model. The ROC curve represents the relationship between the sensitivity (true positive) and specificity (false positive)

of the model on a two-dimensional graph. In practice, the ROC curve shows the model's ability to correctly distinguish between the two classes based on the different threshold values that can be set for classification. The AUC curve, on the other hand, is a numerical index representing the area subtended by the ROC curve. The higher the AUC, the better the model can distinguish between the two classes. Generally, a binary classification model with an area subtended by the ROC curve close to 1 and a high AUC is considered very accurate and reliable. However, the choice of the optimal threshold value depends on the specific context in which the model is used and the user's preferences.

Metrics

Generally, papers use metrics such as FAR (False Acceptance Rate, also called FPR, False Positive Rate) or FRR (False Rejection Rate, also called FNR, False Negative Rate). The FAR gives insight into the percentage of test samples misplaced as positive, while the FRR shows the percentage of samples mistakenly recognized as false. These values, however, must be calculated for each acceptance threshold value set, so to avoid this, the ROC curve (Receiver Operating Characteristics) is used, which shows the TPR (True Positive Rate, where TPR = 1 - FNR) in relation to the FPR for each possible acceptance threshold value.

Through this curve, another metric arises, the AUC (Area Under the Curve), the area below the ROC curve. The closer this area is to 1, the better the model performed. This is because if the curve was projected to the upper left, it meant that with a low threshold, there was a low FPR and a high TPR. Another handy metric is the EER (Equal Error Rate), which would be nothing more than the value where FAR and FRR are equal; this is more for comparing results with other studies as it is a widely used metric.

Normalization

When features have different ranges, it is a good idea to go for normalization. Therefore, it becomes necessary to use the *MinMax scaler*. This scaler brings all the data into the range of [0,1], and to do this, it sets the feature with the most significant value equal to 1 and the feature with the smallest value equal to 0. All others are equal to:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

5.3.2.2.5 Design Experiments

This section repeats the experimentation already carried out with the new "School Student" dataset. In this case, a dataset containing values calculated from the sensors of a BullyBuster Questionnaire application was used.

Dataset

The dataset used in this phase includes data on School Student s. Before starting the actual experiments, a selection of the data contained in the dataset was made. A folder named Statistics was received with the dataset, within which Excel files were found:

- *Avellino_24.xlsx*
- *Avellino_25.xlsx*
- *Cagliari.xlsx*

Expressly, regarding the Avellino school, two datasets were provided:

- The first dataset consisted of 49 detections, of which 45 were taken on the web platform (run on PCs and iOS devices), while 5 were taken via the Android app. Of these surveys, 3 users completed less than 20% of the questionnaire, another 3 completed less than 60%;
- The second dataset consists of 59 surveys, of which 56 were taken on the web platform (executed on PC and mobile devices), while 2 were executed via the Android app. Of these detections, 3 users completed less than 20% of the questionnaire;

Finally, the last dataset, consisting of the surveys conducted in a school in Cagliari, consists of 124 surveys, of which 111 were conducted on the web platform (run on PC and mobile devices). In contrast, 13 were conducted via the Android app. Of these surveys, 9 users completed less than 20% of the questionnaire, and 2 others completed less than 60%.

These files contain, for each user, the answers they gave to the various questions. The first 4 questions are related to the videos shown to the users (e.g., questions about the emotions they felt while watching the video). In contrast, the others are general questions (e.g., questions about age and any bullying they experienced).

All those users who did not answer most of the questions were excluded. The data (in .txt format) processed are presented with two different formatting: a first format where each line of the file is in the form (Figure 62):
"DD-MM-YYYY HH:MM:SS" ; "sensor_Activity_n" ; "x:valx_y:valy_z:valz" ; "timestamp"

```
"24-1-2023 09:42:41" ; "magnetometro_Video_1" ; "x:357.9432411263988_y:30.248744589158918_z:6.584127368817154" ; "1674549761845";
"24-1-2023 09:42:41" ; "accelerometer_Video_1" ; "x:-0.016364932410791514_y:0.007795768442749976_z:-0.17512743432521818" ; "1674549761843";
"24-1-2023 09:42:41" ; "magnetometro_Video_1" ; "x:357.85886409538284_y:30.430290088575827_z:6.586721014963248" ; "1674549761861";
"24-1-2023 09:42:41" ; "gyroscope_Video_1" ; "x:16.090774797674893_y:-0.41850511379713345_z:-2.5569400127324533" ; "1674549761873";
"24-1-2023 09:42:41" ; "magnetometro_Video_1" ; "x:357.74940193768066_y:30.740753163901918_z:6.626105755746244" ; "1674549761874";
"24-1-2023 09:42:41" ; "accelerometer_Video_1" ; "x:0.004098009903356433_y:0.21832418649494648_z:-0.3234849351465702" ; "1674549761872";
"24-1-2023 09:42:41" ; "accelerometer_Video_1" ; "x:0.10938888774067163_y:0.19160218378603458_z:-0.22289164054095745" ; "1674549761889";
"24-1-2023 09:42:41" ; "gyroscope_Video_1" ; "x:12.766351827674084_y:1.0071101124579107_z:0.07084448297951765" ; "1674549761889";
```

Figure 62 - Example of data in the first format

And a second format, where each line of the file is in the form (Figure 63):

"DD-MM-YYYY HH:MM:SSS" - "SENSOR_Activity" - "valx,valy,valz – DD-MM-YYYY HH:MM:SSS"

```
"2023-01-24 09:47:38.456" - "GYROSCOPE_Secondary_Main" - "-0.0672374963760376,0.00976249948143959,-0.0037124999798834324 - 2023-01-24 09:47:38.456"
"2023-01-24 09:47:38.465" - "MAGNETOMETER_Secondary_Main" - "21.487501,-29.5125,-18.712502 - 2023-01-24 09:47:38.465"
"2023-01-24 09:47:38.465" - "ACCELEROMETER_Secondary_Main" - "-0.68796,6.0792007,4.8696003 - 2023-01-24 09:47:38.465"
"2023-01-24 09:47:38.469" - "LIGHT_Secondary_Main" - "40.39235 - 2023-01-24 09:47:38.469"
"2023-01-24 09:47:38.476" - "GYROSCOPE_Secondary_Main" - "-0.09075000137090683,0.031075000762939453,-0.014437500387430191 - 2023-01-24 09:47:38.476"
"2023-01-24 09:47:38.485" - "MAGNETOMETER_Secondary_Main" - "21.4125,-29.5125,-18.4125 - 2023-01-24 09:47:38.485"
"2023-01-24 09:47:38.485" - "ACCELEROMETER_Secondary_Main" - "-0.8184,6.0608406,4.9752 - 2023-01-24 09:47:38.485"
"2023-01-24 09:47:38.496" - "GYROSCOPE Secondary Main" - "-0.05664999783039093,0.029012499377131462,-0.009074999950826168 - 2023-01-24 09:47:38.496"
```

Figure 63 - Example of data in the second format

Selection and cleaning operations (*Data Cleaning*) were performed on the data provided. In particular, these operations were necessary because much sensor data was incomplete (e.g., no data for the accelerometer, gyroscope, and magnetometer), and the scripts could not run. In addition, some sensor data were corrupted, thus blocking the entire execution of the algorithms. In particular, by manually analyzing the corrupted files, it was noticed that, in some files, numerous lines needed to be completed (e.g., containing only part of the timestamp and nothing else).

In less severe cases, the corrupted rows were removed automatically; in more severe cases, the entire file was discarded. Following this step, the two file formats were also adapted (as far as possible) to

make as few changes as possible in the scripts provided for this work. In particular, the primary operation was to change the format of the "*DATE/TIME*" column for the files in the second format, bringing them into line with the first.

Experiment 1

For each set of users, the scripts mentioned above (*Script_from_txt_to_csv(Videos) and Script_from_txt_to_csv(Quiz)*) were executed, through which the data were cleaned (adjustment phase) and converted into a single final file that was then read by the scripts for applying the Machine Learning algorithms.

The above two scripts were used to process the data in the first file format, while the two new versions of the code, namely *Script_from_txt_to_csv(Videos)_2* and *Script_from_txt_to_csv(Quiz)_2*, were used for the second format. Specifically, for each dataset, three executions were performed:

- The first execution was performed with the script Script_from_txt_to_csv(Videos), obtaining for each user 6 CSV files.

Questi file si presentano nella forma:
- *sensor_userid_Activity_SENSORNAME_pre_adjustment.csv*
- dove userid è l'ID dell'utente, Activity è la specifica attività svolta dall'utente al momento del campionamento, SENSORNAME è il sensore specifico preso in considerazione (accelerometro, giroscopio o magnetometro).

In this case, only the rows with the following Activity values were extracted from the .txt file: Video_1, Video_2, Video_3, Video_4 (containing the coordinate values recorded by the SENSOR-NAME sensor at the time of viewing the specific video), QuestionsVideo_1, QuestionsVideo_2, QuestionsVideo_3, and QuestionsVideo_4 (similar to the previous case, but containing the coordinate values recorded at the time the user answers the questions about the specific video). These are CSV files containing data cleaned in the post-adjustment phase.

The last 3, however, are in the form:
- sensor_userid_Activity_SENSORNAME_post_adjustment.csv
- containing the values in the pre-adjustment file following the data cleaning operation.

- The second execution was performed with the script Script_from_txt_to_csv(Quiz), obtaining for each user 6 CSV files having the same structures as those described in the previous point. The difference with the previous case is that, in this case, all rows of the file containing Activity values equal to QuestionsVideo_i, for i ranging from 5 to 93, were extracted since the users were asked 88 general questions not related to videos.

At the end of each of the two runs, a function in the code was used to merge, for each user and each Activity, all 3 post-adjustment type files, which were then read by the scripts for applying the Machine Learning models.

Each CSV file given by merging the above files is in the form: sensor_userid_Activity_final.csv. After that, the two scripts, script_models_videos, and script_models_quiz, were executed to apply the Machine Learning models on the videos and quizzes dataset, respectively. Following execution, two folders, results_videos and results_quiz, were created. Within each, a subfolder was created for each applied model. Within each, a subfolder was created per user, each containing the predictions computed by that model for each Activity.

Experiment 2

After modifying the DataExtraction script to accommodate the second file format (not covered in the original code), proceeded with executing that script first on the two Avellino datasets and finally on the Cagliari dataset.

Specifically, for each dataset, two executions were carried out:
- The first execution was carried out with the original DataExtraction script, processing only the data in the old format;
- The second execution was done with the DataExtraction_2 script, processing only the data in the new format.

Once the data extraction was done, any malformed sensor data that caused errors during the parsing/processing phase was removed. This was done in two test sessions (Test1 and Test2) to maintain uniformity with previous experimentation. Each test had the same number of users to balance the two Tests.

Following the DataExtraction, the FeatureExtraction script was run, generating CSVs containing the extracted features. These features were calculated on the touch data and from sensors such as the accelerometer, magnetometer, and gyroscope. Finally, the roc-auc script was run to calculate the ROC and AUC curves.

3.3.3.2.3 Comparison of Results

In this section, the results obtained on School Students are compared with those obtained for University Students for both experiments.

Experiment 1

Temporal graphs

In the case of undergraduate students, the trend of application use during individual activities (for each user) was graphically illustrated through time graphs. On the x-axis, time instants were represented on the y-axis, and each sensor's X, Y, and Z values were represented. As a result, three graphs were generated (one for each sensor involved) for each activity and each user. Among the results obtained on University Students, the following temporal graphs were found (Figure 64):

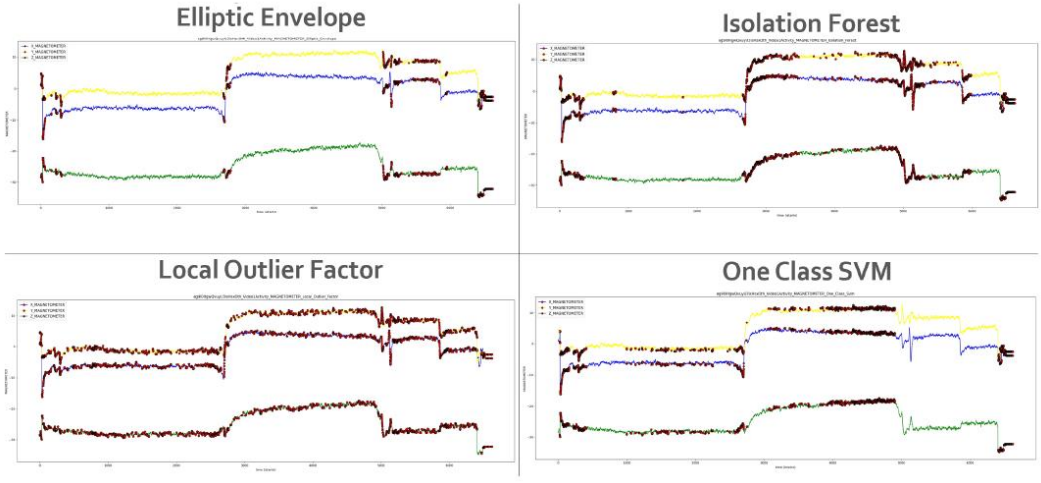


Figure 64 - Results obtained by students University Student , divided by the Anomaly Detection algorithm used for the Video3Activity activity related to the accelerometer sensor

In this case, it can be seen that each algorithm detected a moderate number of anomalies. For each algorithm, the trend in the graph appears to be approximately similar. In contrast, below are the time graphs for the same activity for a different user (belonging to the dataset of School Students) (Figure 65).

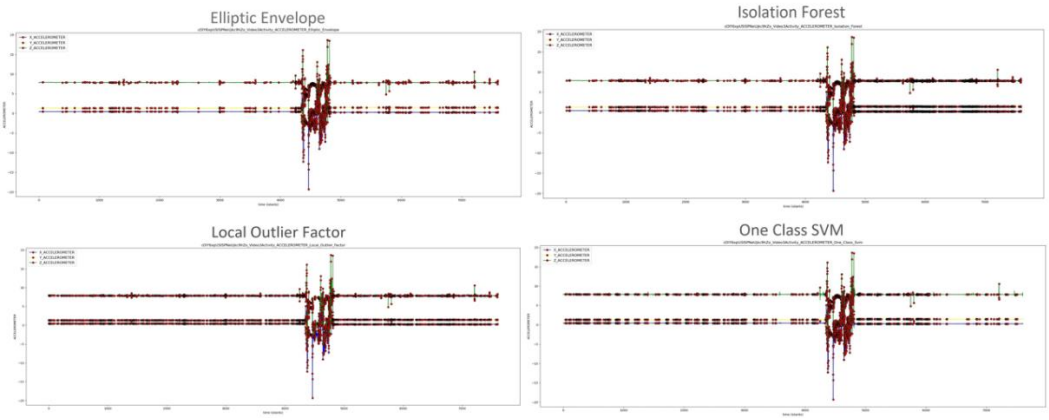


Figure 65 - Time graph related to the Video3Activity related to the accelerometer sensor.

As can be seen, the same conclusion can be drawn here as in the previous case.

Histograms

This section analyzes the differences and similarities between high school seniors and University Student s. In the latter, histograms are presented in which the X-axis shows the activities or classes of users, while the Y-axis shows the percentage of anomalies. This analysis was done for each algorithm used. A threshold value was applied for each algorithm: if a particular value calculated by the Anomaly Detection algorithm exceeds this threshold, it is considered an anomaly. Three different thresholds were used for each algorithm: one low, one medium, and one high.

A histogram showed the first 20 quiz questions with the highest number of anomalies and the same for the last 20 questions. As for high school seniors, the following results were obtained:
Histograms for activities/videos and classes/videos (low parameter) (Figure 66 - Figure 67):

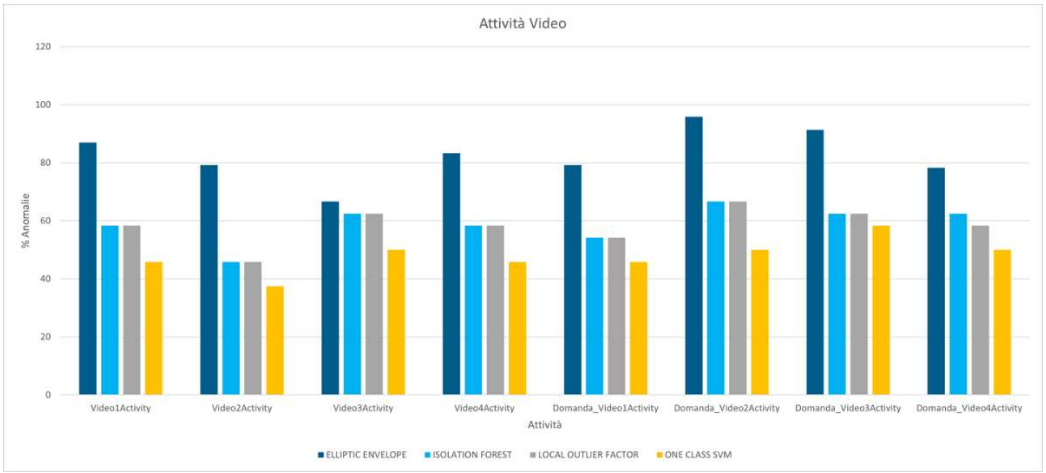


Figure 66 – Histogram by video activity (School Student)

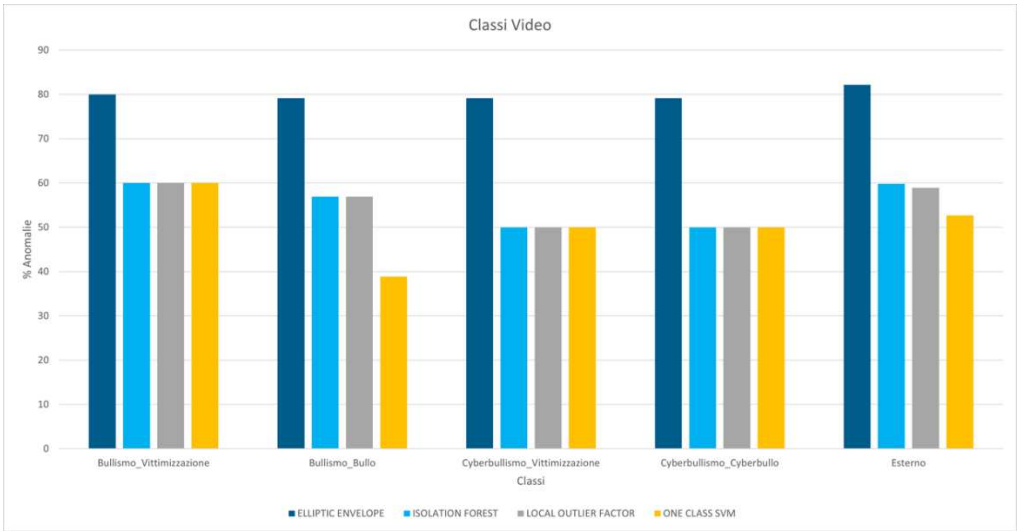


Figure 67 - Histogram by classes (School Student)

While in the case of University Students:

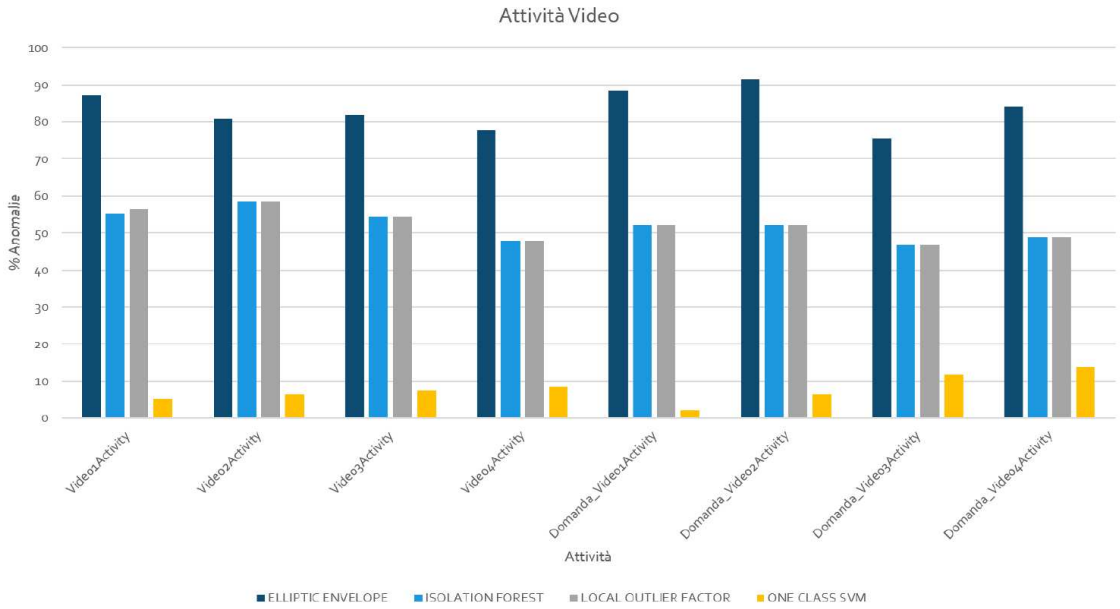


Figure 68 - Histogram by video activity (University Student)

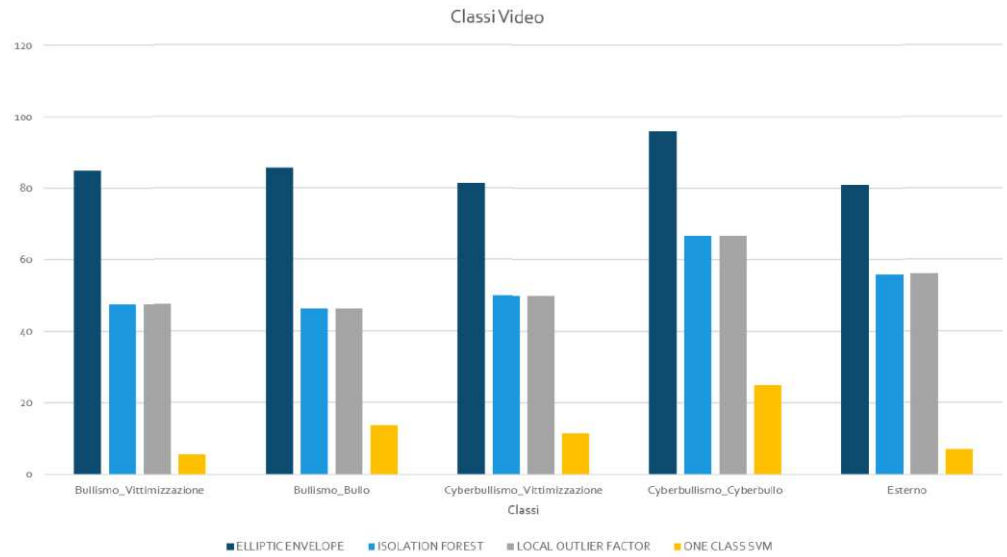


Figure 69 - Histogram by classes (University Student)

In the case of video-related activities, using a low parameter, it can be seen that there are Anomaly Detection algorithms such as, for example, Elliptic Envelope, Isolation Forest, and Local Outlier Factor, which on the dataset of School Student s recognize a similar percentage of anomalies as those obtained on the dataset of University Student s. In contrast, the One-Class SVM algorithm was able to detect a more significant number of anomalies through the dataset used for this case study (on average, counting all activities around 50 percent), while in the case of the results obtained in the case of University Student s, the One-Class SVM detected a significantly lower percentage of anomalies (on average, less than 10 percent, always counting all activities). [Figure 66 and Figure 68]

As far as classes are concerned, there are significant differences in almost all the algorithms used. The only algorithm that calculated a similar outlier value between the two datasets is Isolation Forest: around 50 percent on the dataset of School Student s and around 42 percent on the dataset of University Student s (Figure 67 and Figure 69).

Histograms for activities/videos and classes/videos (average parameter):

As for high school seniors, the following results were obtained :

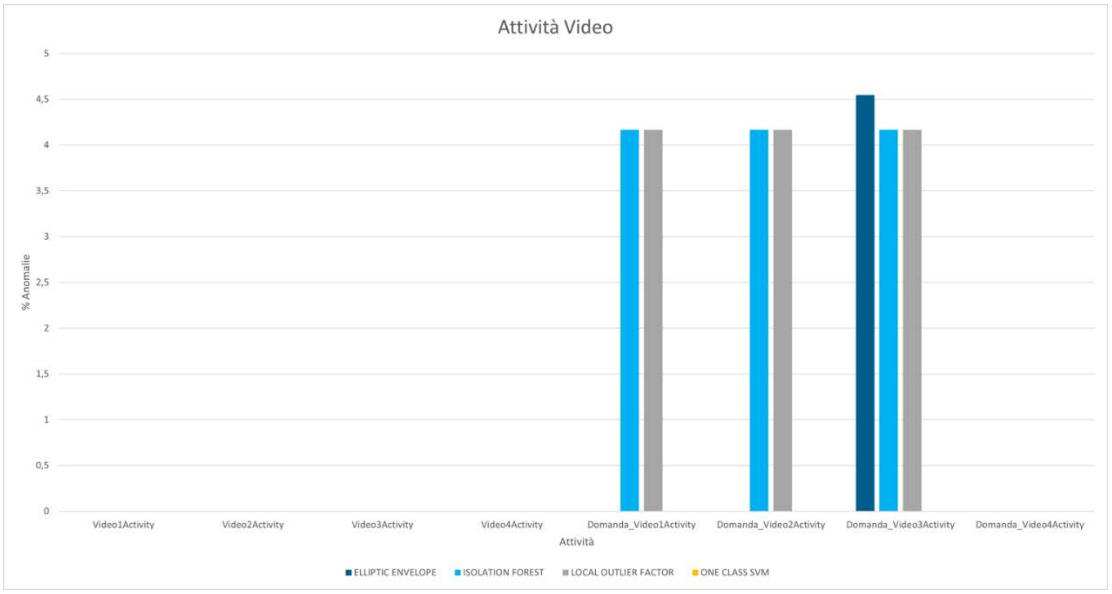


Figure 70 - Histogram by video activity - Average parameter (School Student)

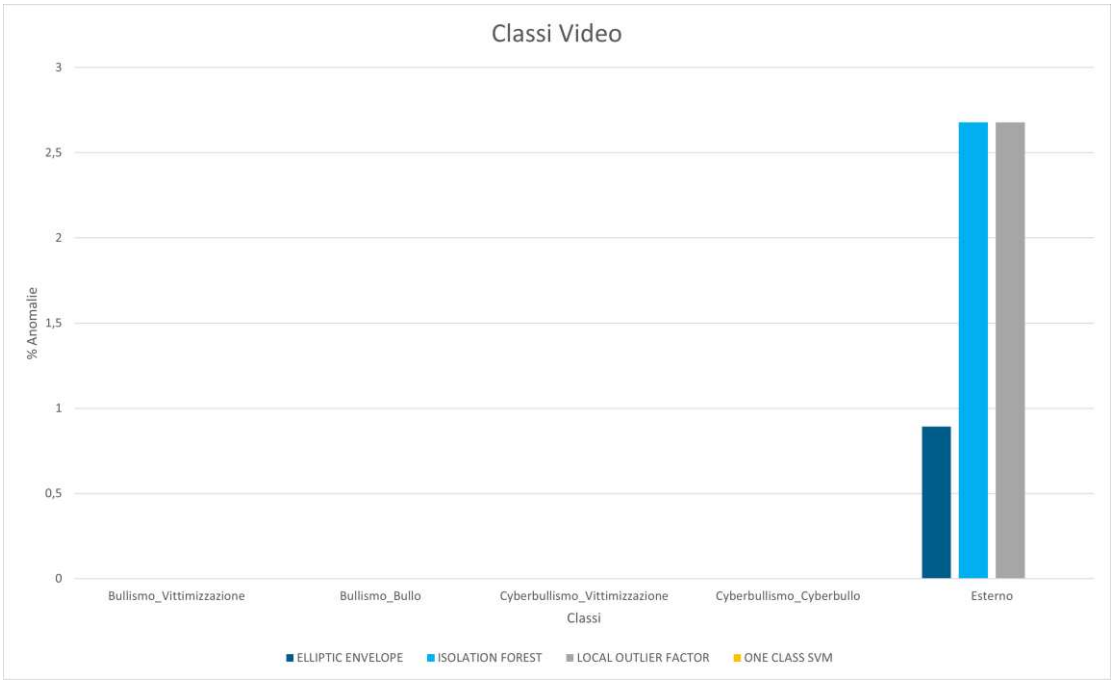


Figure 71 - Histogram by Classes - Average Parameter (School Student)

In the case of University Students, it is obtained:

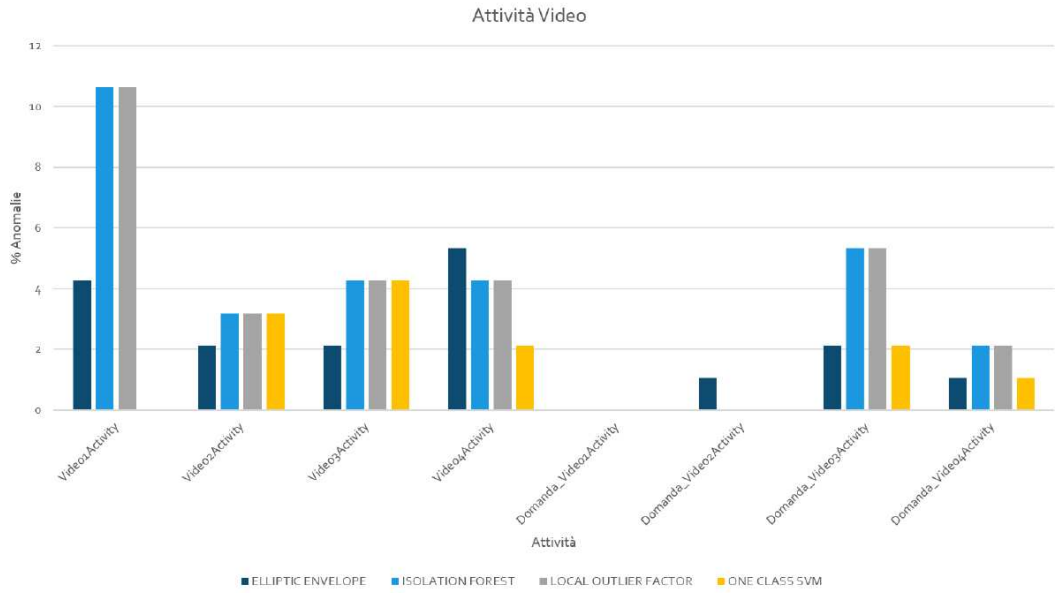


Figure 72 - Histogram by video activity - Average parameter (University Student)

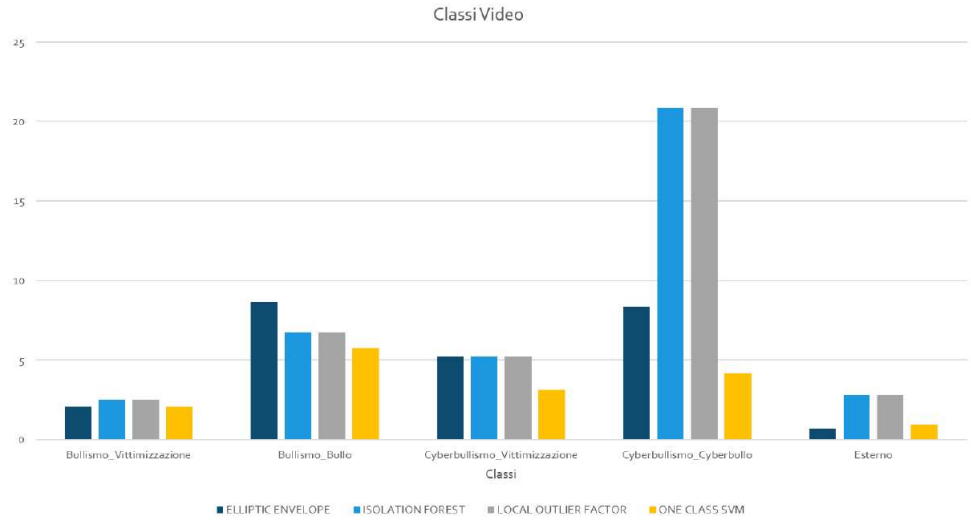


Figure 73 - Histogram by Classes - Average Parameter (University Student)

Using an average parameter for video-related tasks shows significantly lower numerical results than in the previous case. In our case, the One-Class SVM algorithm detects a percentage of outliers of 0, while the Elliptic Envelope algorithm detects, on average, around 0.5 percent. On the other hand, the Isolation Forest and Local Outlier Factor algorithms record an average value of outliers around 2 percent. In the case of the results obtained on University Student s, the values are significantly lower than in the case of the low parameter. However, all the algorithms used record an average percentage of outliers around 2 percent (Figure 70 and Figure 72). About classes, it can be seen that in our case, almost all applied algorithms record average values of percentage of outliers around 0, and most users are identified in the *"External"* class (Figure 71 and Figure 73).

Histograms for activities/videos and classes/videos (high parameter):

Achievements for high school seniors:

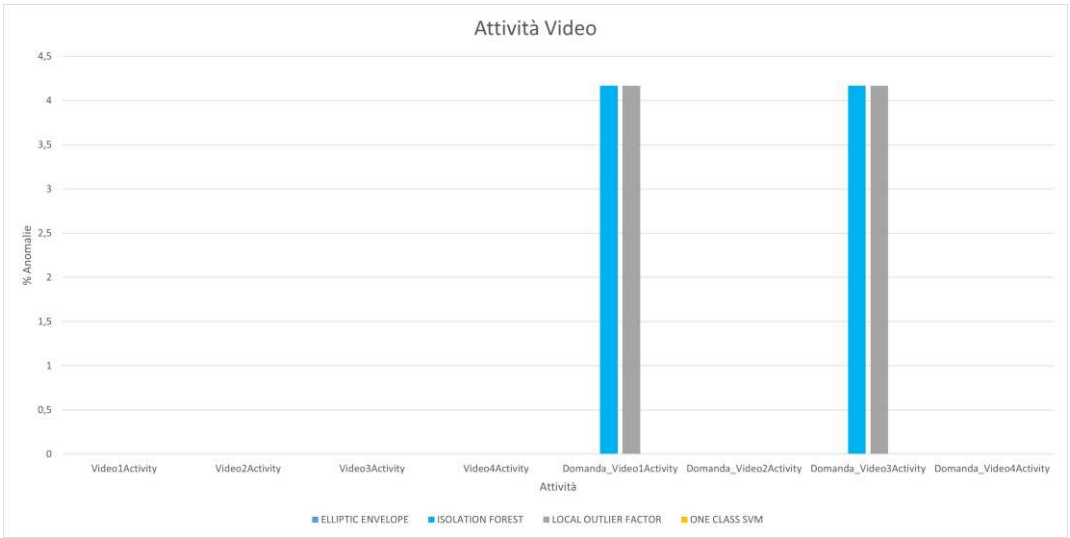


Figure 74 - Histogram by video activity - High parameter (School Student)

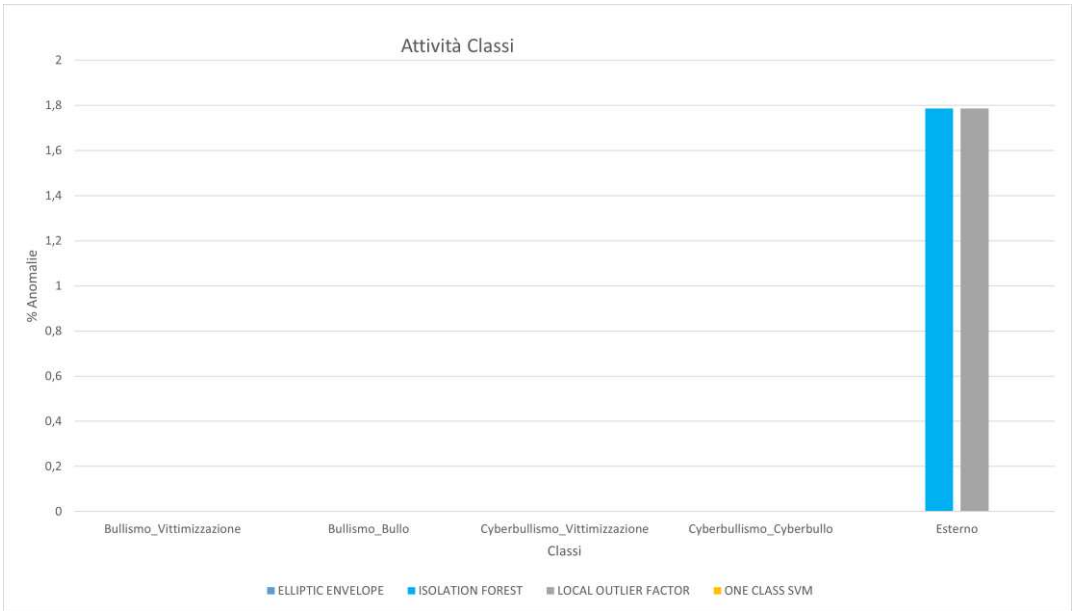


Figure 75 - Histogram for classes - High parameter (School Student)

In the case of University Students, it is obtained:

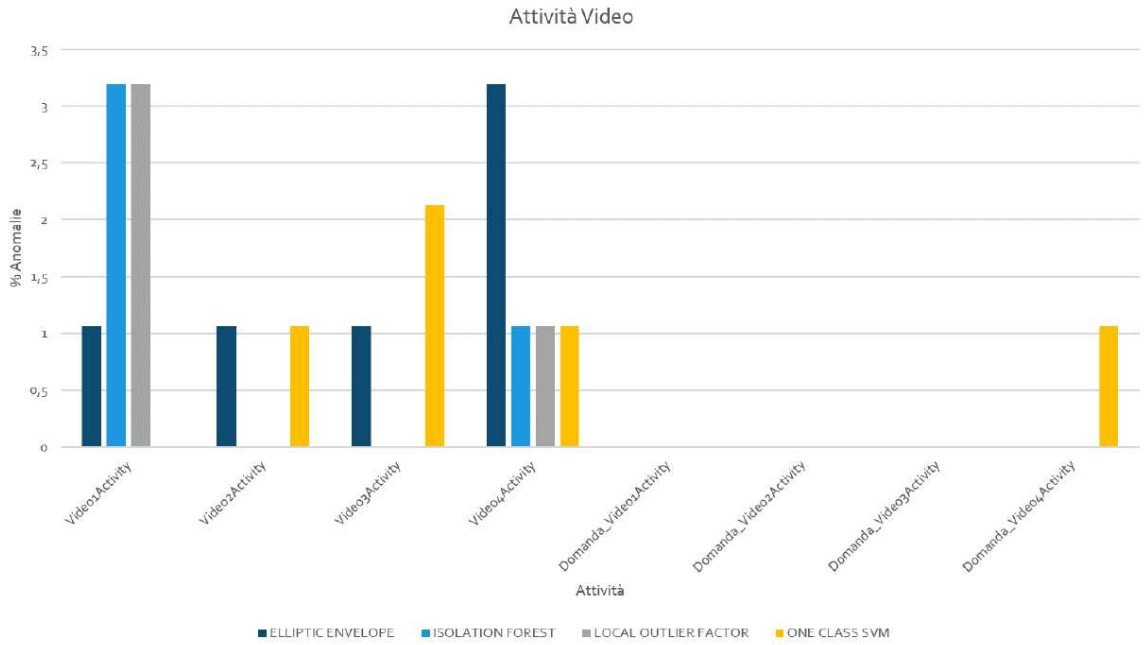


Figure 76 - Histogram by video activity - High parameter (University Student)

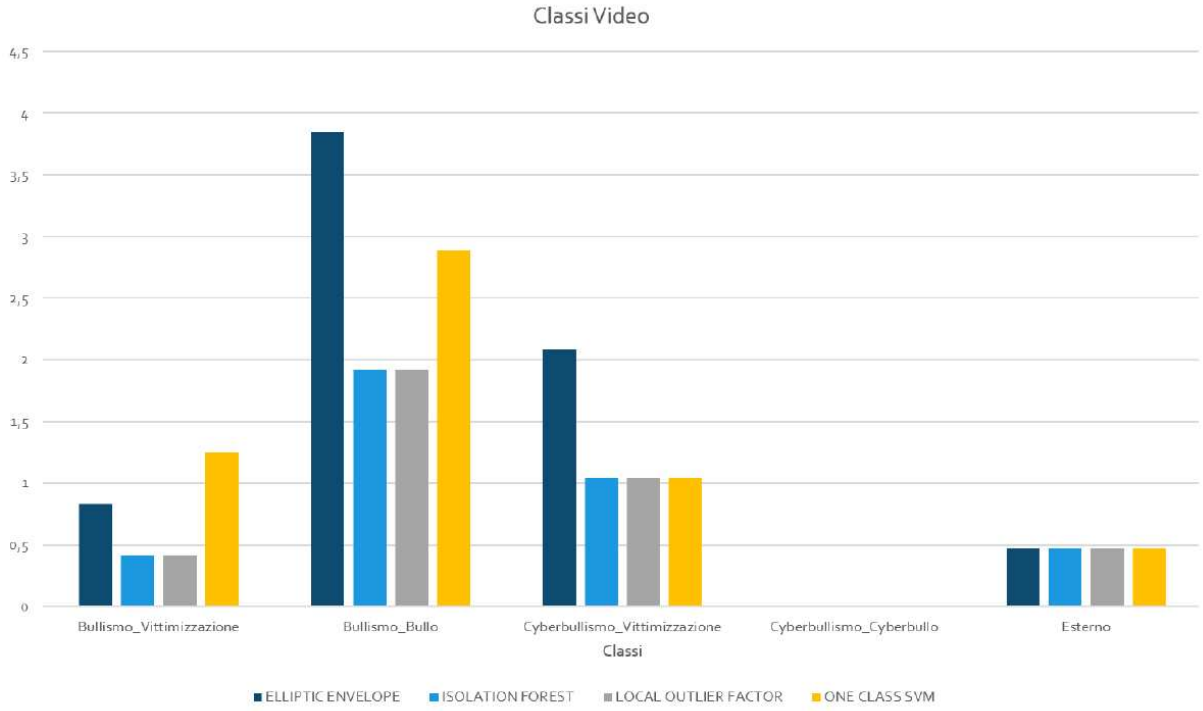


Figure 77 - Histogram by Classes - High Parameter (University Student)

Regarding video-related activities, using a high parameter shows a slight difference between the results obtained with the two datasets. In our case, the Elliptic Envelope and One-Class SVM algorithms detect a percentage of anomalies equal to 0. In contrast, the Isolation Forest and Local Outlier Factor algorithms detect, on average, a percentage of anomalies around 4%. As for University Student s, each algorithm detects outliers, but average values around 0.5 percent are obtained (Figure 74 and Figure 76). As for classes, in the case of School Student s, the two algorithms, Elliptic Envelope, and One-Class SVM detect a percentage of anomalies around 0, while the two algorithms, Isolation Forest

and Local Outlier Factor, detect, on average, a percentage of anomalies around 4%. In the case of University Student s, there is a slightly higher average percentage of outliers in the case of Isolation Forest and Local Outlier Factors. Still, the two algorithms, Elliptic Envelope and One-Class SVM, detect a small average percentage of outliers, approximately 1% (Figure 75 and Figure 77).

Histograms by quiz classes (low parameter)

Results obtained for high school seniors (Figure 78):

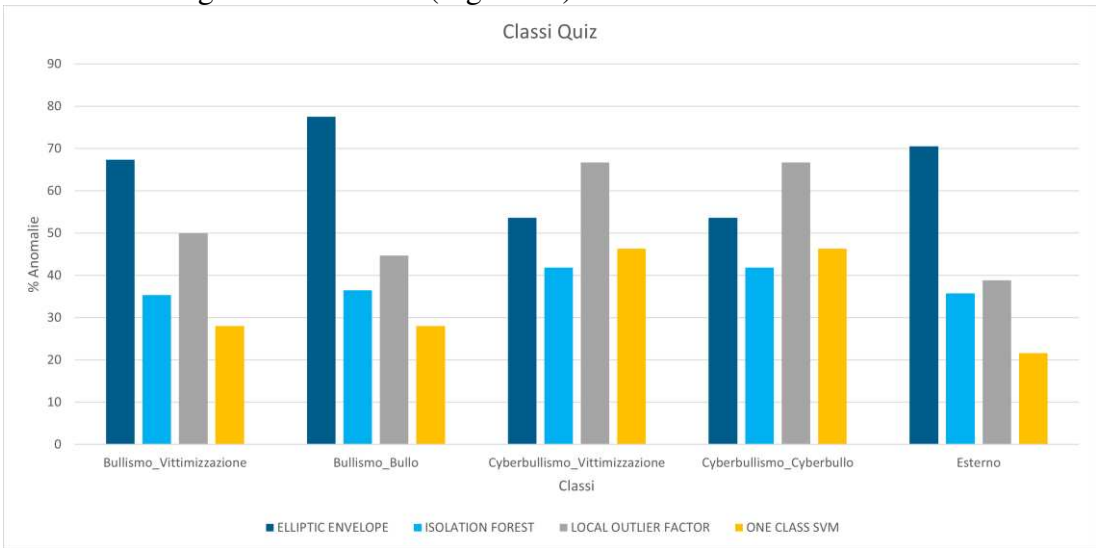


Figure 78 - Histogram by quiz classes - Low parameter (School Student)

In the case of University Student s, it is obtained (*Figure 79*):

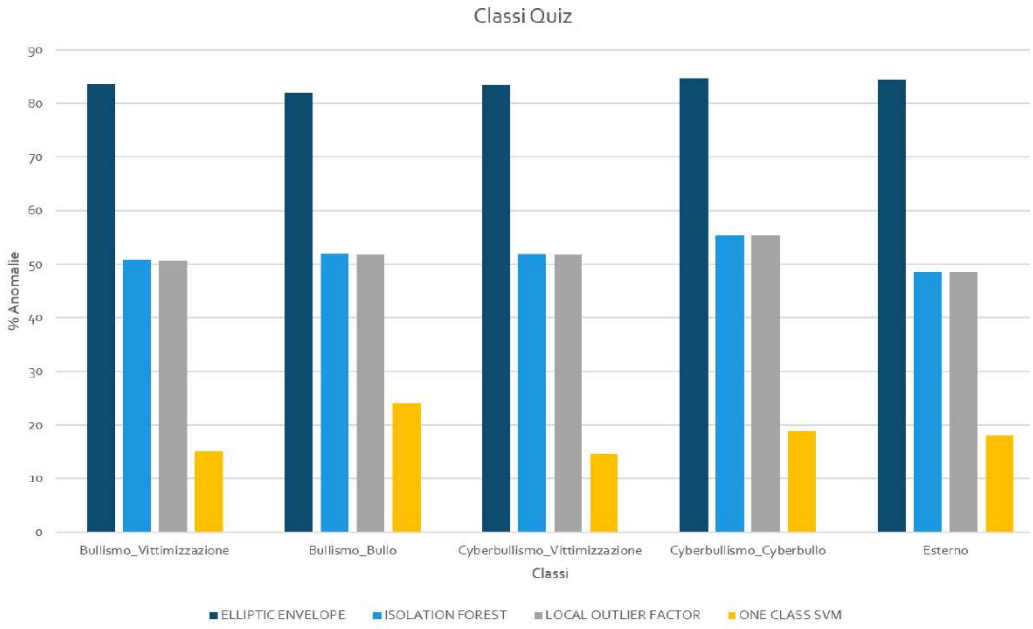


Figure 79 - Histogram by quiz classes - Low parameter (University Student)

Regarding the percentages of anomalies found during the quiz, the only fundamental similarity identifiable between the two datasets lies in the average percentage anomaly values detected by the Local Outlier Factor algorithm. This algorithm found an average percentage of outliers of around 50 percent for high school and University Student datasets. There are more pronounced differences between the other algorithms.

Histograms by quiz classes (average parameter)

Achievements for high school seniors (Figure 80):

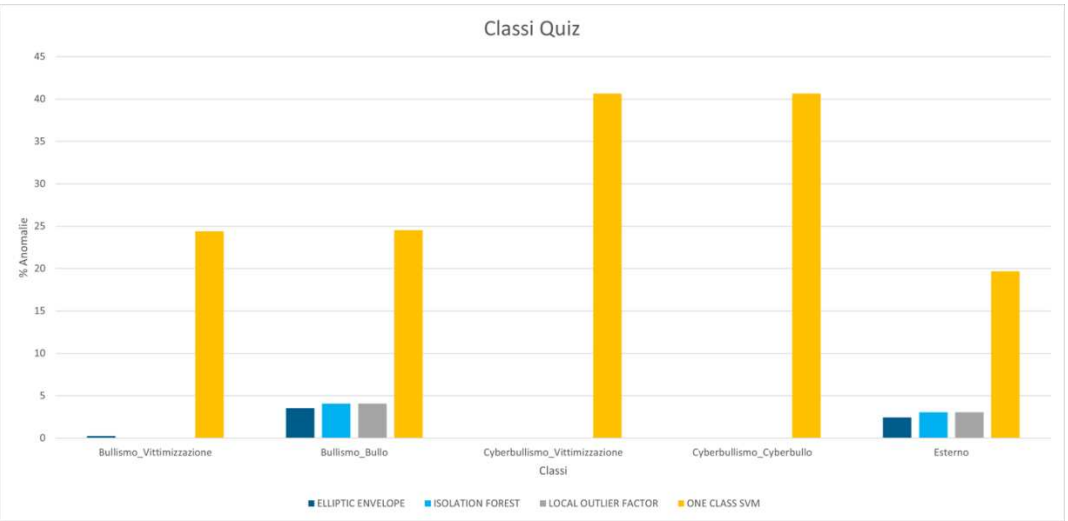


Figure 80 - Histogram by quiz classes - Average parameter (School Student)

In the case of University Student s, it is obtained (Figure 81):

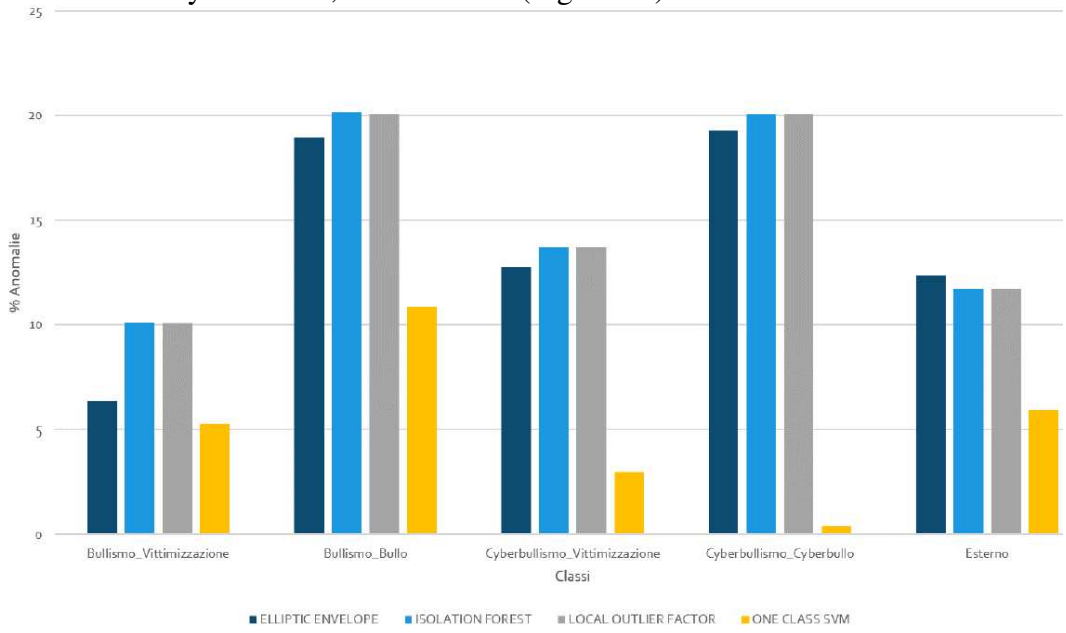


Figure 81 - Histogram by quiz classes - Average parameter (University Student)

Here, it is possible to see several differences between the two datasets about all the algorithms used. In the case of School Student s, the One-Class SVM algorithm detected an average percentage of outliers of around 30 percent, unlike University Student s, where an average percentage of outliers of only 5 percent is detected. As for the Elliptic Envelope, Isolation Forest, and Local Outlier Factor algorithms, average outlier percentage values around 1 percent are detected in our case, while values around 14 percent are recorded for University Students.

Histograms for quiz classes (high parameter)

Achievements for high school seniors (Figure 82):

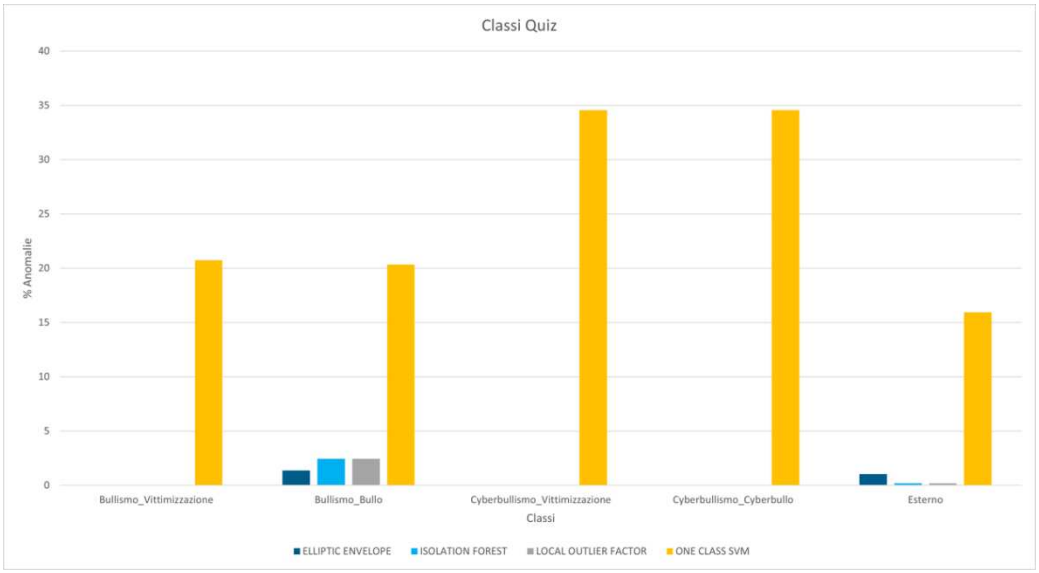


Figure 82 - Histogram by quiz classes - High parameter (School Student)

In the case of University Student s, it is obtained (Figure 83):

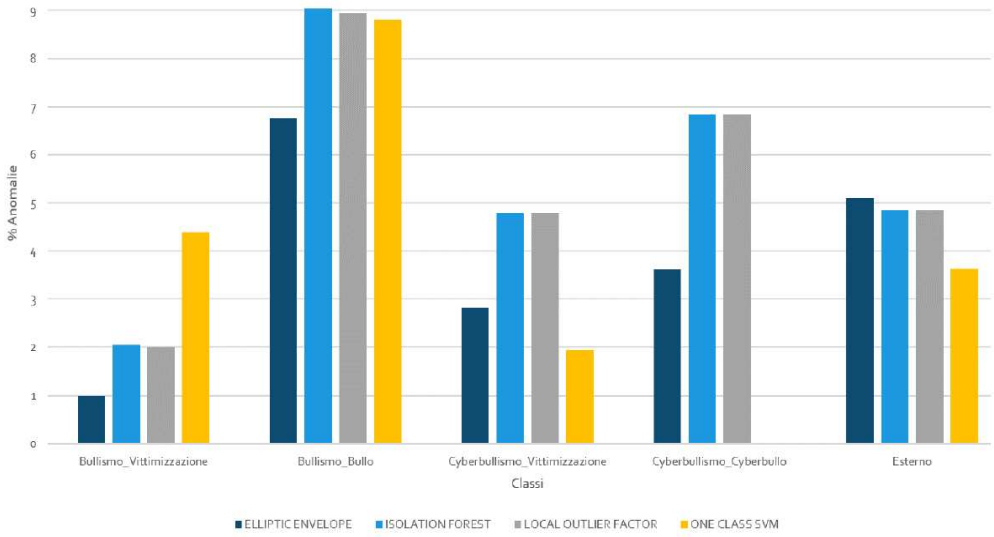


Figure 83 - Histogram by quiz classes - High parameter (University Student)

In this case, comparing the two datasets shows a significant difference in the case of the One Class SVM algorithm, which records a mean value of outlier percentage of around 25% for School Student s but around 4% for University Student s. In the case of the other algorithms, an average outlier percentage value of around 0.5 percent was found in the case of School Student s, while values of around 5 percent were found in the case of University Students.

Histograms for the top 20 questions with the most anomalies by averaging the scores obtained among the 4 models (high parameter)

In the case of School Students, it is obtained (Figure 84):

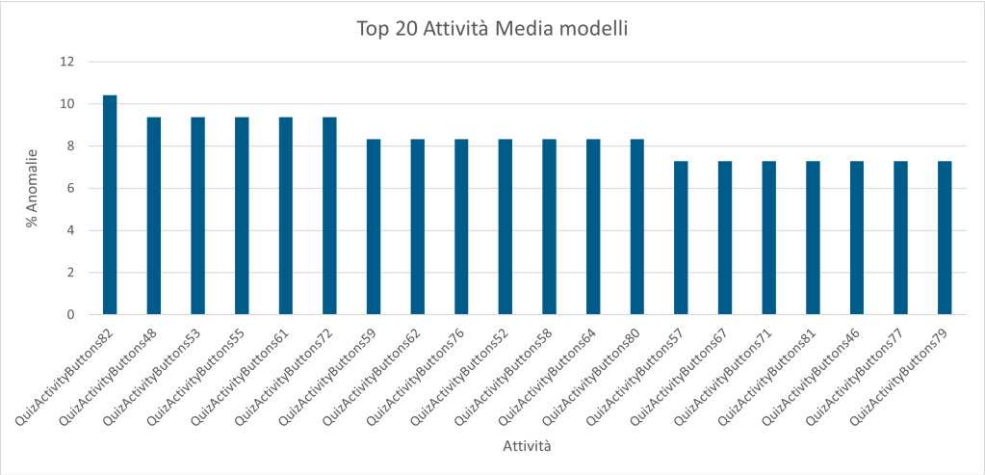


Figure 84 - Top 20 Activity Average Models (School Student)

In the case of University Students, it is obtained (Figure 85):

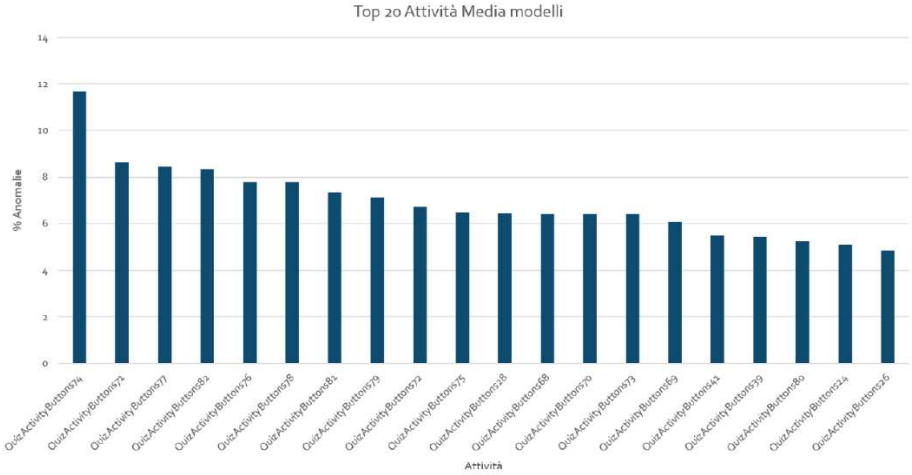


Figure 85 - Top 20 Activity Average Models (University Student)

Regarding the top 20 questions with the most anomalies, the main similarity between the case of high school and University Student s lies in the fact that in both cases, the following activities fall in the top 20: QuizActivityButtons72, QuizActivityButtons76, QuizActivityButtons80, QuizActivityButtons77, and QuizActivityButtons79. There is a similarity (in terms of average probability percentages detected among the various models) on the QuizActivityButtons76 activity: in the case of School Student s, there is an average value of outlier percentage among the various models, around 8%, in the case of University Student s. As for the QuizActivityButtons79 activity, in the case of School Student s, there is an average value of around 7%, like that of University Student s.

Histograms for the last 20 questions with the most anomalies by averaging the scores obtained among the 4 models (high parameter)

In the case of School Students, it is obtained (Figure 86):

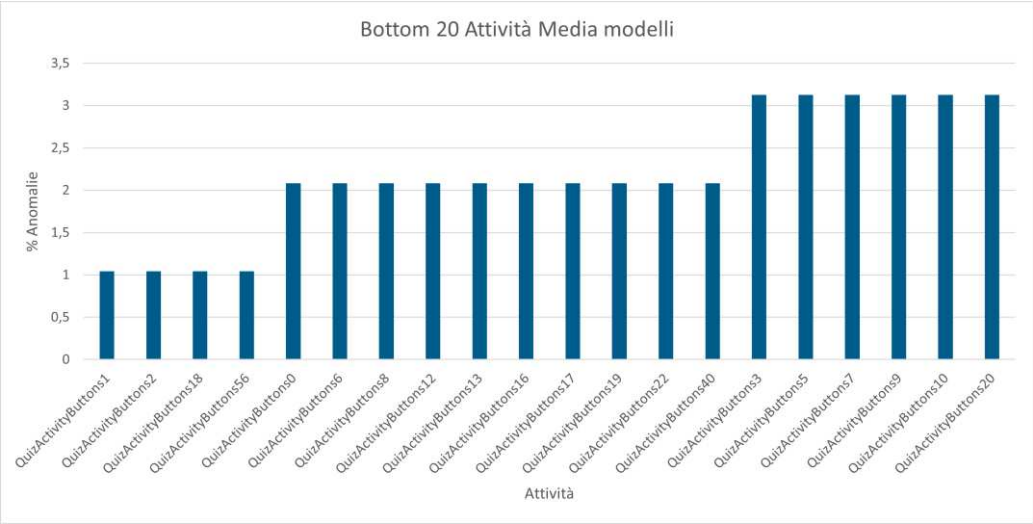


Figure 86 - Bottom 20 Activity Average models (School Student)

In the case of University Student s, it is obtained (Figure 87):

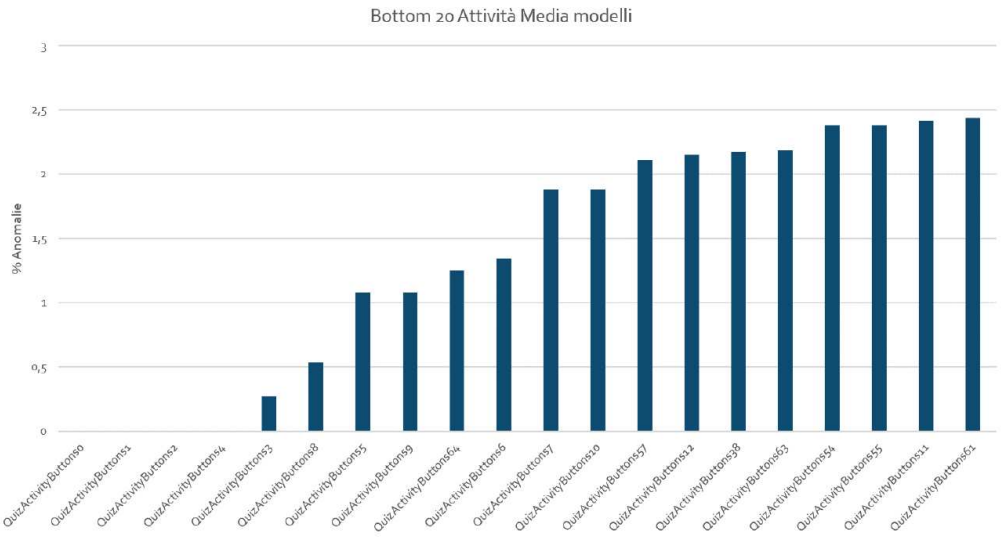


Figure 87 - Bottom 20 Activity Average models (University Student)

Regarding the top 20 questions with the most anomalies, the main similarity between the case of high school and University Student s lies in the fact that the following activities fall in the top 20 in both: *QuizActivityButtons1, QuizActivityButtons2, QuizActivityButtons0, QuizActivityButtons6, QuizActivityButtons8, QuizActivityButtons3, QuizActivityButtons5, QuizActivityButtons7, QuizActivityButtons9, QuizActivityButtons10.*

The first difference between the two datasets lies in the fact that in the case of university students for QuizActivityButtons0, QuizActivityButtons1, and QuizActivityButtons2 activities, averaging the results obtained across models yields a percentage of anomalies of 0. In contrast, our case study has average values of 1%. On the other hand, a similarity was found in the QuizActivityButtons9 and QuizActivityButtons10 activities, with an average percentage of outliers among the various models of around 3% on both datasets.

Table

This section show the final tables from which the histograms presented in the previous section were extrapolated. In the case of School Students, the following final tables are obtained (*Figure 88* - *Figure 89*):

PARAMETRO BASSO

VIDEO ACTIVITY	Video1Activity	Video2Activity	Video3Activity	Video4Activity	Domanda_Video1Activity	Domanda_Video2Activity	Domanda_Video3Activity	Domanda_Video4Activity	MEDIA
ELLIPTIC ENVELOPE	86,95652174	79,16666667	66,66666667	83,33333333	79,16666667	95,83333333	91,30434783	78,26086957	82,58605072
ISOLATION FOREST	58,33333333	45,83333333	62,5	58,33333333	54,16666667	66,66666667	62,5	62,5	58,85416667
LOCAL OUTLIER FACTOR	58,33333333	45,83333333	62,5	58,33333333	54,16666667	66,66666667	62,5	58,33333333	58,33333333
ONE CLASS SVM	45,83333333	37,5	50	45,83333333	45,83333333	50	58,33333333	50	47,91666667
MEDIA ANOMALIE VIDEO	62,36413043	52,08333333	60,41666667	61,45833333					

VIDEO CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	80	79,16666667	79,16666667	79,16666667	82,14285714	79,92857143
ISOLATION FOREST	60	56,94444444	50	50	59,82142857	55,3531746
LOCAL OUTLIER FACTOR	60	56,94444444	50	50	58,92857143	55,17460317
ONE CLASS SVM	60	38,88888889	50	50	52,67857143	50,31349206
MEDIA ANOMALIE CLASSI	65	57,98611111	57,29166667	57,29166667	63,39285714	

PARAMETRO MEDIO

VIDEO ACTIVITY	Video1Activity	Video2Activity	Video3Activity	Video4Activity	Domanda_Video1Activity	Domanda_Video2Activity	Domanda_Video3Activity	Domanda_Video4Activity	MEDIA
ELLIPTIC ENVELOPE	0	0	0	0	0	0	4,545454545	0	0,568181818
ISOLATION FOREST	0	0	0	0	4,166666667	4,166666667	4,166666667	0	1,5625
LOCAL OUTLIER FACTOR	0	0	0	0	4,166666667	4,166666667	4,166666667	0	1,5625
ONE CLASS SVM	0	0	0	0	0	0	0	0	0
MEDIA ANOMALIE VIDEO	0	0	0	0					

VIDEO CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	0	0	0	0	0,892857143	0,178571429
ISOLATION FOREST	0	0	0	0	2,678571429	0,535714286
LOCAL OUTLIER FACTOR	0	0	0	0	2,678571429	0,535714286
ONE CLASS SVM	0	0	0	0	0	0
MEDIA ANOMALIE CLASSI	0	0	0	0	1,5625	

PARAMETRO ALTO

VIDEO ACTIVITY	Video1Activity	Video2Activity	Video3Activity	Video4Activity	Domanda_Video1Activity	Domanda_Video2Activity	Domanda_Video3Activity	Domanda_Video4Activity	MEDIA
ELLIPTIC ENVELOPE	0	0	0	0	0	0	0	0	0
ISOLATION FOREST	0	0	0	0	4,166666667	0	4,166666667	0	1,041666667
LOCAL OUTLIER FACTOR	0	0	0	0	4,166666667	0	4,166666667	0	1,041666667
ONE CLASS SVM	0	0	0	0	0	0	0	0	0
MEDIA ANOMALIE VIDEO	0	0	0	0					

VIDEO CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	0	0	0	0	0	0
ISOLATION FOREST	0	0	0	0	1,785714286	0,357142857
LOCAL OUTLIER FACTOR	0	0	0	0	1,785714286	0,357142857
ONE CLASS SVM	0	0	0	0	0	0
MEDIA ANOMALIE CLASSI	0	0	0	0	0,892857143	

Figure 88 - Final Table - Video quiz (School Student)

PARAMETRO BASSO

QUIZ ACTIVITY	QuizActivityButtons0	uizActivityButton	QuizActivityButtons2	QuizActivityButtons3	QuizActivityButtons4	QuizActivityButtons5	QuizActivityButtons6	QuizActivityButtons7	QuizActivityButtons8
ELLIPTIC ENVELOPE	75	91,66666667	70,83333333	87,5	79,16666667	70,83333333	79,16666667	79,16666667	75
ISOLATION FOREST	54,16666667	50	37,5	41,66666667	50	29,16666667	29,16666667	25	25
LOCAL OUTLIER FACTOR	54,16666667	50	41,66666667	45,83333333	54,16666667	33,33333333	33,33333333	29,16666667	33,33333333
ONE CLASS SVM	16,66666667	8,333333333	8,333333333	12,5	20,83333333	12,5	8,333333333	20,83333333	20,83333333

QUIZ CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	67,31707317	77,50677507	53,65853659	53,65853659	70,55749129	64,53968254
ISOLATION FOREST	35,36585366	36,4498645	41,8699187	41,8699187	35,71428571	38,25396825
LOCAL OUTLIER FACTOR	50	44,71544715	66,66666667	66,66666667	38,85017422	53,37979094
ONE CLASS SVM	28,04878049	28,04878049	46,34146341	46,34146341	21,60278746	34,07665505
MEDIA ANOMALIE CLASSI	45,18292683	46,6802168	52,13414634	52,13414634	41,68118467	47,5625242

PARAMETRO MEDIO

QUIZ ACTIVITY	QuizActivityButtons0	uizActivityButton	QuizActivityButtons2	QuizActivityButtons3	QuizActivityButtons4	QuizActivityButtons5	QuizActivityButtons6	QuizActivityButtons7	QuizActivityButtons8
ELLIPTIC ENVELOPE	0	0	0	0	0	0	0	0	0
ISOLATION FOREST	0	0	0	0	0	0	0	0	0
LOCAL OUTLIER FACTOR	0	0	0	0	0	0	0	0	0
ONE CLASS SVM	8,333333333	4,166666667	8,333333333	12,5	16,66666667	12,5	8,333333333	12,5	20,83333333

QUIZ CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	0,243902439	3,52303523	0	0	2,43902439	1,241192412
ISOLATION FOREST	0	4,06504065	0	0	3,048780488	1,422764228
LOCAL OUTLIER FACTOR	0	4,06504065	0	0	3,048780488	1,422764228
ONE CLASS SVM	24,3902439	24,52574526	40,6504065	40,6504065	19,68641115	29,98064266
MEDIA ANOMALIE CLASSI	6,158536585	9,044715447	10,16260163	10,16260163	7,055749129	8,516840883

PARAMETRO ALTO

QUIZ ACTIVITY	QuizActivityButtons0	uizActivityButton	QuizActivityButtons2	QuizActivityButtons3	QuizActivityButtons4	QuizActivityButtons5	QuizActivityButtons6	QuizActivityButtons7	QuizActivityButtons8
ELLIPTIC ENVELOPE	0	0	0	0	0	0	0	0	0
ISOLATION FOREST	0	0	0	0	0	0	0	0	0
LOCAL OUTLIER FACTOR	0	0	0	0	0	0	0	0	0
ONE CLASS SVM	8,333333333	4,166666667	4,166666667	12,5	16,66666667	12,5	8,333333333	12,5	8,333333333

QUIZ CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	0	1,35501355	0	0	1,045296167	0,480061943
ISOLATION FOREST	0	2,43902439	0	0	0,174216028	0,522648084
LOCAL OUTLIER FACTOR	0	2,43902439	0	0	0,174216028	0,522648084
ONE CLASS SVM	20,73170732	20,32520325	34,55284553	34,55284553	15,94076655	25,22067364
MEDIA ANOMALIE CLASSI	5,182926829	6,639566396	8,638211382	8,638211382	4,333623693	6,686507937

Figure 89 - Final Table - Quiz (School Student)

In the case of University Students, it is obtained (Figure 90-Figure 91):

PARAMETRO BASSO

VIDEO ACTIVITY	Video1Activity	Video2Activity	Video3Activity	Video4Activity	Domanda_Video1Activity	Domanda_Video2Activity	Domanda_Video3Activity	Domanda_Video4Activity	MEDIA
ELLIPTIC ENVELOPE	87,23404255	80,85106383	81,91489362	77,65957447	88,29787234	91,4893617	75,53191489	84,04255319	83,37765957
ISOLATION FOREST	55,31914894	58,5106383	54,25531915	47,87234043	52,12765957	52,12765957	46,80851064	48,93617021	51,99468085
LOCAL OUTLIER FACTOR	56,38297872	58,5106383	54,25531915	47,87234043	52,12765957	52,12765957	46,80851064	48,93617021	52,12765957
ONE CLASS SVM	5,319148936	6,382978723	7,446808511	8,510638298	2,127659574	6,382978723	11,70212766	13,82978723	7,712765957
MEDIA ANOMALIE VIDEO	51,06382979	51,06382979	49,46808511	45,4787234					

VIDEO CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	85	85,57692308	81,25	95,83333333	80,89622642	49,94103774
ISOLATION FOREST	47,5	46,15384615	50	66,66666667	55,89622642	42,20224057
LOCAL OUTLIER FACTOR	47,5	46,15384615	50	66,66666667	56,13207547	37,63757862
ONE CLASS SVM	5,416666667	13,46153846	11,45833333	25	6,839622642	28,39033019
MEDIA ANOMALIE CLASSI	46,35416667	47,83653846	48,17708333	63,54166667	49,94103774	

PARAMETRO MEDIO

VIDEO ACTIVITY	Video1Activity	Video2Activity	Video3Activity	Video4Activity	Domanda_Video1Activity	Domanda_Video2Activity	Domanda_Video3Activity	Domanda_Video4Activity	MEDIA
ELLIPTIC ENVELOPE	4,255319149	2,127659574	2,127659574	5,319148936	0	1,063829787	2,127659574	1,063829787	2,260638298
ISOLATION FOREST	10,63829787	3,191489362	4,255319149	4,255319149	0	0	5,319148936	2,127659574	3,723404255
LOCAL OUTLIER FACTOR	10,63829787	3,191489362	4,255319149	4,255319149	0		5,319148936	2,127659574	3,723404255
ONE CLASS SVM	0	3,191489362	4,255319149	2,127659574	0	0	2,127659574	1,063829787	1,595744681
MEDIA ANOMALIE VIDEO	6,382978723	2,925531915	3,723404255	3,989361702					

VIDEO CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	2,083333333	8,653846154	5,208333333	8,333333333	0,70754717	4,997278665
ISOLATION FOREST	2,5	6,730769231	5,208333333	20,83333333	2,830188679	7,620524915
LOCAL OUTLIER FACTOR	2,5	6,730769231	5,208333333	20,83333333	2,830188679	7,620524915
ONE CLASS SVM	2,083333333	5,769230769	3,125	4,166666667	0,943396226	3,217525399
MEDIA ANOMALIE CLASSI	2,291666667	6,971153846	4,6875	13,54166667	1,827830189	

PARAMETRO ALTO

VIDEO ACTIVITY	Video1Activity	Video2Activity	Video3Activity	Video4Activity	Domanda_Video1Activity	Domanda_Video2Activity	Domanda_Video3Activity	Domanda_Video4Activity	MEDIA
ELLIPTIC ENVELOPE	1,063829787	1,063829787	1,063829787	3,191489362	0	0	0	0	0,79787234
ISOLATION FOREST	3,191489362	0	0	1,063829787	0	0	0	0	0,531914894
LOCAL OUTLIER FACTOR	3,191489362	0	0	1,063829787	0	0	0	0	0,531914894
ONE CLASS SVM	0	1,063829787	2,127659574	1,063829787	0	0	0	1,063829787	0,664893617
MEDIA ANOMALIE VIDEO	1,861702128	0,531914894	0,79787234	1,595744681					

VIDEO CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	0,833333333	3,846153846	2,083333333	0	0,471698113	1,446903725
ISOLATION FOREST	0,416666667	1,923076923	1,041666667	0	0,471698113	0,770621674
LOCAL OUTLIER FACTOR	0,416666667	1,923076923	1,041666667	0	0,471698113	0,770621674
ONE CLASS SVM	1,25	2,884615385	1,041666667	0	0,471698113	1,129596033
MEDIA ANOMALIE CLASSI	0,729166667	2,644230769	1,302083333	0	0,471698113	

Figure 90 - Final Table - Video (University Student)

PARAMETRO BASSO

QUIZ ACTIVITY	QuizActivityButtons0	QuizActivityButtons1	QuizActivityButtons2	QuizActivityButtons3	QuizActivityButtons4	QuizActivityButtons5	QuizActivityButtons6	QuizActivityButtons7	QuizActivityButtons8
ELLIPTIC ENVELOPE	80,64516129	82,79569892	88,17204301	86,02150538	84,94623656	75,2688172	87,09677419	80,64516129	82,79569892
ISOLATION FOREST	37,6344086	48,38709677	45,16129032	49,46236559	49,46236559	45,16129032	50,53763441	52,68817204	52,68817204
LOCAL OUTLIER FACTOR	37,6344086	48,38709677	45,16129032	49,46236559	49,46236559	45,16129032	50,53763441	52,68817204	52,68817204
ONE CLASS SVM	3,225806452	1,075268817	1,075268817	2,150537634	2,150537634	2,150537634	4,301075269	4,301075269	2,150537634

QUIZ CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	83,62491926	82,00537397	83,43567484	84,73895582	84,33691099	83,62836698
ISOLATION FOREST	50,77745567	52,0002933	51,82215609	55,42168675	48,59526361	51,72337108
LOCAL OUTLIER FACTOR	50,64738314	51,70012593	51,69777798	55,42168675	48,59526361	51,61244748
ONE CLASS SVM	15,08636266	23,98565857	14,70262846	18,87550201	17,99443464	18,12891727
MEDIA ANOMALIE CLASSI	50,03403018	52,42286294	50,41455934	53,61445783	49,88046821	51,2732757

PARAMETRO MEDIO

QUIZ ACTIVITY	QuizActivityButtons0	QuizActivityButtons1	QuizActivityButtons2	QuizActivityButtons3	QuizActivityButtons4	QuizActivityButtons5	QuizActivityButtons6	QuizActivityButtons7	QuizActivityButtons8
ELLIPTIC ENVELOPE	0	0	0	0	0	1,075268817	4,301075269	4,301075269	0
ISOLATION FOREST	0	0	0	0	0	1,075268817	2,150537634	3,225806452	1,075268817
LOCAL OUTLIER FACTOR	0	0	0	0	0	1,075268817	2,150537634	3,225806452	1,075268817
ONE CLASS SVM	0	0	1,075268817	1,075268817	1,075268817	2,150537634	2,150537634	3,225806452	2,150537634

QUIZ CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	6,329851178	18,95077465	12,74733301	19,27710843	12,3358437	13,92818219
ISOLATION FOREST	10,09669158	20,1500144	13,70137953	20,08032129	11,67578384	15,14083813
LOCAL OUTLIER FACTOR	10,05653094	20,05733599	13,70137953	20,08032129	11,67578384	15,11427032
ONE CLASS SVM	5,22852985	10,86101884	2,999370141	0,401606426	5,927620714	5,083629194
MEDIA ANOMALIE CLASSI	7,927900888	17,50478597	10,78736555	14,95983936	10,40375802	12,31672996

PARAMETRO ALTO

QUIZ ACTIVITY	QuizActivityButtons0	QuizActivityButtons1	QuizActivityButtons2	QuizActivityButtons3	QuizActivityButtons4	QuizActivityButtons5	QuizActivityButtons6	QuizActivityButtons7	QuizActivityButtons8
ELLIPTIC ENVELOPE	0	0	0	0	0	0	1,075268817	1,075268817	0
ISOLATION FOREST	0	0	0	0	0	1,075268817	1,075268817	2,150537634	0
LOCAL OUTLIER FACTOR	0	0	0	0	0	1,075268817	1,075268817	2,150537634	0
ONE CLASS SVM	0	0	0	1,075268817	0	2,150537634	2,150537634	2,150537634	2,150537634

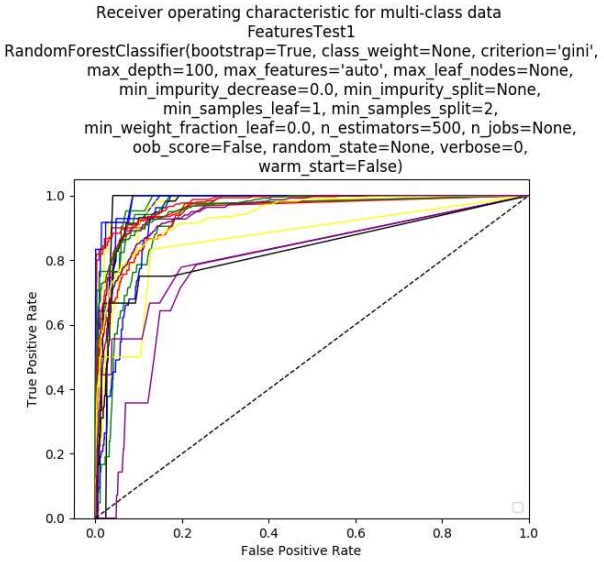
QUIZ CLASSI	Bullismo_Vittimizzazione	Bullismo_Bullo	Cyberbullismo_Vittimizzazione	Cyberbullismo_Cyberbullo	Esterno	MEDIA
ELLIPTIC ENVELOPE	1,000549296	6,759817202	2,8294486	3,614457831	5,106358684	3,862605575
ISOLATION FOREST	2,045490328	9,04358228	4,792826839	6,827309237	4,842503228	5,510342382
LOCAL OUTLIER FACTOR	2,005329685	8,950903874	4,792826839	6,827309237	4,842503228	5,483774573
ONE CLASS SVM	4,381557978	8,813789958	1,939504048	0	3,621741219	3,751318641
MEDIA ANOMALIE CLASSI	2,359731822	8,392023328	3,587775647	4,317269076	4,60327659	4,652015293

Figure 91 - Final Table - Quiz (University Student)

As can be seen, the activity related to Video 1 (Video1Activity) is the one that, on average, registers the highest value of percentage of anomalies (equal to 62 percent) among the various algorithms. This suggests that a particular video may have aroused more emotions in students at the time of its viewing.

Experiment 2

Test 1 – Random Forest

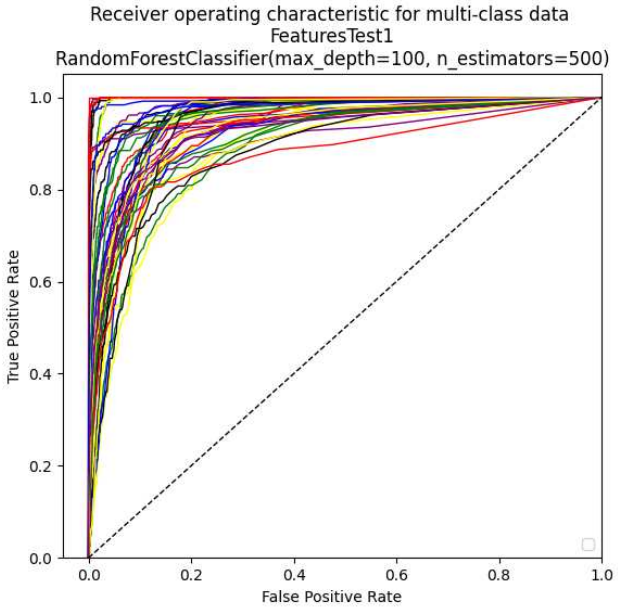


UT	AUC	EER
0	0.99	7.33%
1	0.97	9.38%
2	0.98	7.03%
3	0.99	7.67%
4	0.97	10.49%
5	0.97	4.01%
6	0.98	7.11%
7	0.97	8.89%
8	0.92	13.20%
9	0.96	9.87%
10	0.78	23.25%
11	0.96	8.67%
12	0.94	12.20%
13	0.95	13.34%
14	0.98	9.09%
15	0.86	12.83%
16	0.83	19.78%
17	0.84	17.57%
18	0.99	7.67%
19	0.97	9.27%
20	0.97	10.47%
21	0.94	13.32%
22	0.95	12.36%

Figure 92 - Test 1_Random Forest_Final Table_School Students

In this test, the ROC resulting from the tests performed on School Student s differs from those performed on University Student s. However, there is little difference in the AUC value, which, in the coded experiment, reaches the maximum value of 99% versus 100% obtained in the tests conducted on University Student s. However, regarding EER, there is a noticeable difference between the results obtained in this work and those obtained for those University Student .

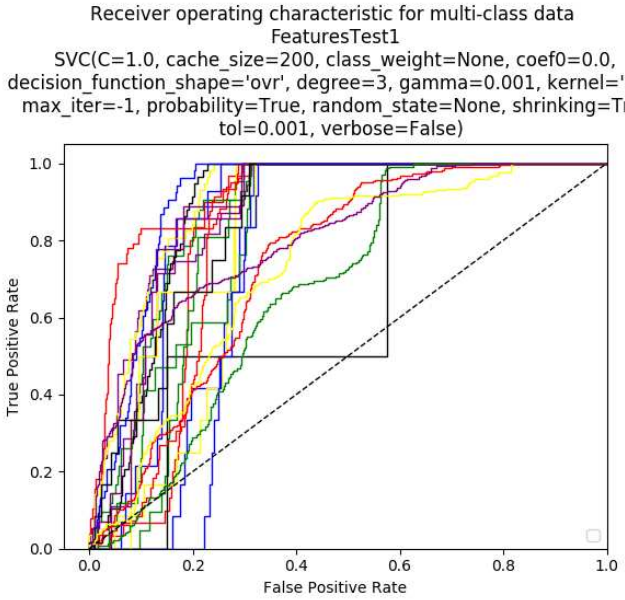
Specifically, in this work, the minimum EER is 4% against 0% obtained in the tests conducted on University Students (Figure 91):



UT	AUC	EER
0	0.94	13.44%
1	0.97	10.66%
2	0.99	2.97%
3	0.9	20.11%
4	0.94	13.92%
5	1	1.87%
6	0.99	2.45%
7	0.95	11.83%
8	0.89	19.37%
9	0.93	14.02%
10	0.9	15.49%
11	1	1.15%
12	0.96	10.57%
13	0.92	14.74%
14	0.95	11.54%
15	0.97	5.86%
16	1	0.82%
17	0.98	7.42%
18	0.94	11.01%
19	0.94	13.34%
20	0.95	10.96%
21	0.92	14.47%
22	0.96	8.76%
23	0.95	10.76%
24	0.93	15.65%
25	0.96	7.15%
26	0.92	15.17%
27	0.9	17.63%
28	0.98	6.03%
29	0.93	13.63%
30	0.97	8.01%
31	1	0.00%
32	0.97	9.41%
33	0.99	2.97%
34	0.94	12.51%
35	0.98	7.41%
36	0.94	11.91%
37	1	1.40%
38	0.97	7.87%
39	0.94	12.04%
40	0.93	13.53%
41	0.89	18.68%
42	0.98	6.34%
43	0.87	17.63%
44	0.91	16.10%
45	0.93	13.46%
46	0.93	13.51%

Figure 93 - Test 1_Random Forest_Final Table_University Student

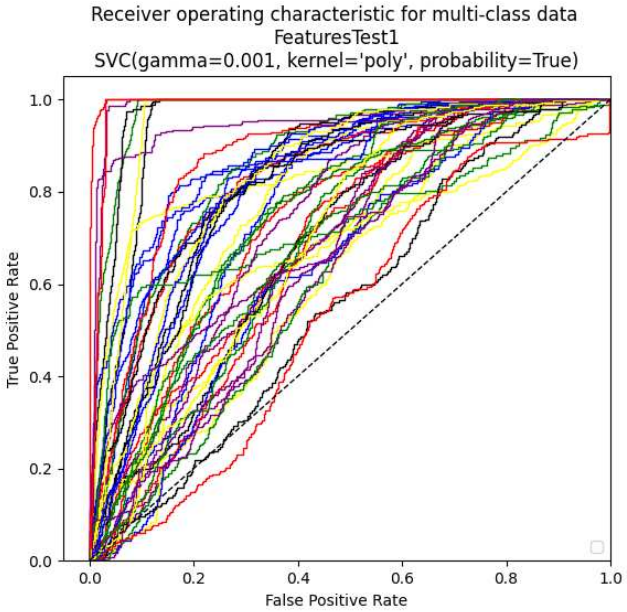
Test 1 – Support Vector Machine



UT	AUC	EER
0	0.76	28.93%
1	0.82	20.12%
2	0.81	21.03%
3	0.78	28.09%
4	0.89	21.85%
5	0.64	57.54%
6	0.86	16.92%
7	0.93	9.87%
8	0.87	19.98%
9	0.91	18.54%
10	0.87	16.64%
11	0.88	18.15%
12	0.91	13.80%
13	0.81	22.66%
14	0.82	29.07%
15	0.84	27.89%
16	0.87	18.27%
17	0.85	26.13%
18	0.73	29.88%
19	0.74	31.47%
20	0.68	35.97%
21	0.73	32.01%
22	0.81	28.74%

Figure 94 - Test 1 Support Vector Machine Final Table School Student

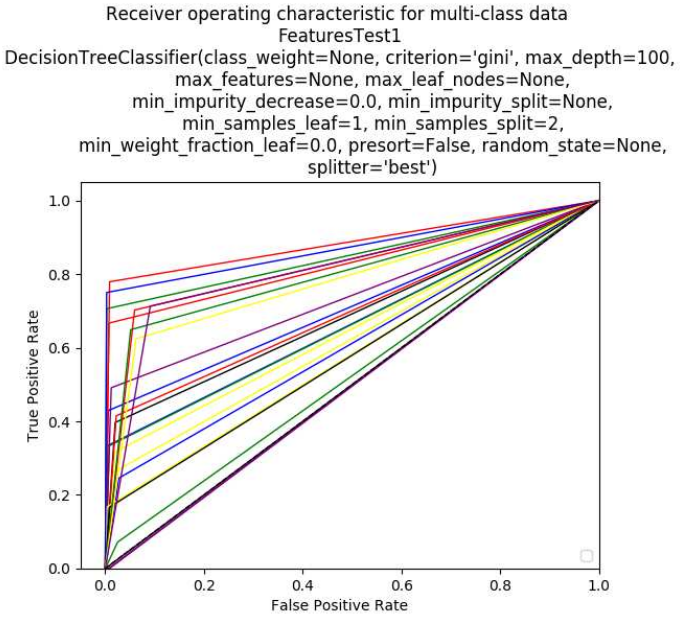
Regarding the test conducted on the SVM model, there is a significant discrepancy between the results obtained on high school seniors and those obtained for University Student s. Specifically, a maximum AUC of 93% and a minimum EER of 9.87% was obtained in this test. This differs from the results obtained on University Student s, which (as reported below) obtained a maximum AUC of 100% and a minimum EER of 2.20%. However, in both cases, the SVM model performs less than the RF model (Figure 94 - Figure 95).



UT	AUC	EER
0	0.83	21.90%
1	0.74	30.77%
2	0.97	7.94%
3	0.63	42.58%
4	0.65	41.71%
5	0.94	10.70%
6	0.87	18.92%
7	0.71	36.60%
8	0.62	41.15%
9	0.66	37.17%
10	0.7	38.08%
11	0.97	6.55%
12	0.76	29.71%
13	0.66	38.11%
14	0.85	22.82%
15	0.84	21.31%
16	0.98	3.30%
17	0.69	36.78%
18	0.88	20.70%
19	0.83	24.08%
20	0.74	30.27%
21	0.78	28.49%
22	0.95	11.94%
23	0.82	24.31%
24	0.64	37.55%
25	0.87	16.99%
26	0.74	30.55%
27	0.67	35.42%
28	0.73	36.84%
29	0.83	24.71%
30	0.85	23.09%
31	1	2.20%
32	0.73	33.71%
33	0.94	9.94%
34	0.73	32.09%
35	0.8	24.92%
36	0.81	26.13%
37	0.98	3.04%
38	0.69	35.92%
39	0.83	24.55%
40	0.64	39.58%
41	0.56	45.38%
42	0.81	23.94%
43	0.53	45.89%
44	0.69	37.05%
45	0.69	38.02%
46	0.64	37.20%

Figure 95 - Test 1_ Support Vector Machine_FinalTable_University Student

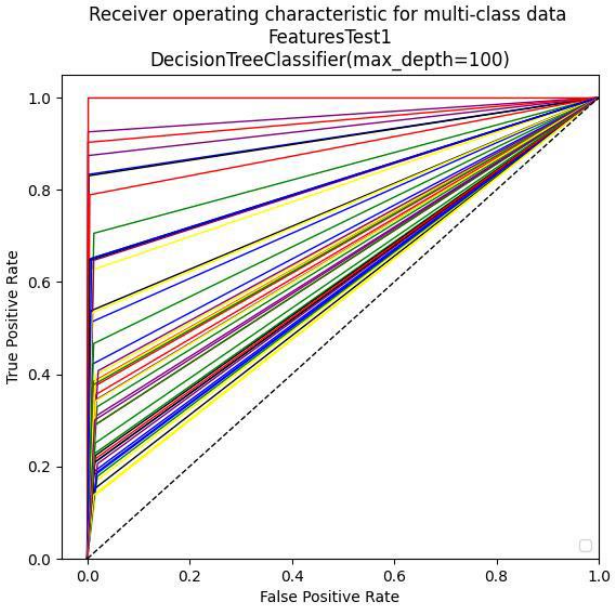
Test 1 – Decision Tree



UT	AUC	EER
0	0.66	0.45%
1	0.83	0.73%
2	0.66	0.84%
3	0.62	0.17%
4	0.74	1.20%
5	0.50	0.06%
6	0.71	0.56%
7	0.89	0.87%
8	0.52	2.50%
9	0.65	1.89%
10	0.50	0.67%
11	0.69	2.01%
12	0.61	2.75%
13	0.70	2.21%
14	0.85	0.28%
15	0.58	0.33%
16	0.50	0.89%
17	0.58	0.90%
18	0.87	0.28%
19	0.82	5.93%
20	0.80	5.14%
21	0.78	6.23%
22	0.81	9.11%

Figure 96 - Test 1_ Decision Tree_FinalTable_School Student

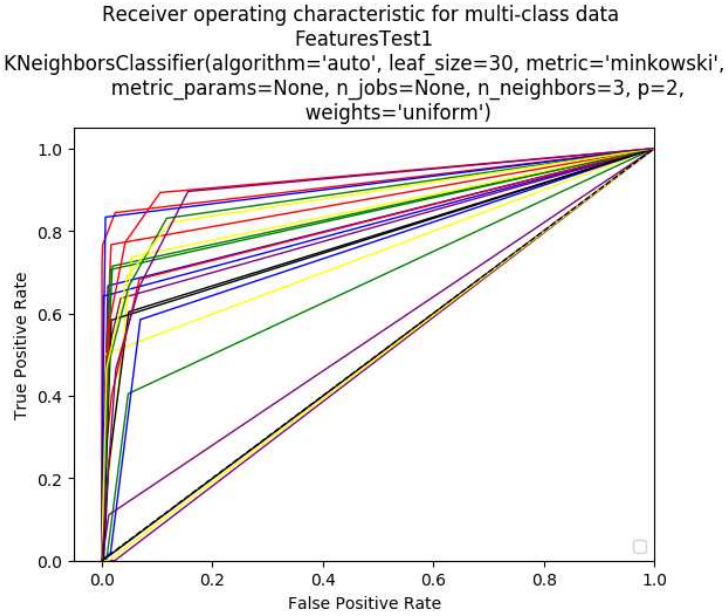
As for the DT model, there is a decline in performance, but only in terms of AUC, as the EERs are all still found to be very low. This observation validates the results obtained on high school seniors and University Student s. Specifically, in the tests conducted in this work with DT, a maximum AUC of 89% and a very low minimum EER of 0.06% are obtained (Figure 96-Figure 97).



UT	AUC	EER
0	0.75	1.03%
1	0.69	1.27%
2	0.85	1.31%
3	0.58	1.76%
4	0.65	1.59%
5	0.82	0.45%
6	0.92	0.27%
7	0.68	1.58%
8	0.58	1.99%
9	0.69	1.66%
10	0.66	1.32%
11	0.91	0.28%
12	0.6	1.42%
13	0.64	1.56%
14	0.61	1.43%
15	0.81	0.99%
16	0.96	0.18%
17	0.77	0.73%
18	0.59	0.74%
19	0.67	1.72%
20	0.66	1.95%
21	0.56	1.64%
22	0.94	0.21%
23	0.6	1.60%
24	0.59	0.98%
25	0.89	0.50%
26	0.64	1.47%
27	0.68	1.52%
28	0.82	1.28%
29	0.57	1.22%
30	0.71	1.04%
31	1	0.00%
32	0.68	0.64%
33	0.76	0.93%
34	0.59	2.02%
35	0.82	0.48%
36	0.58	1.89%
37	0.95	0.16%
38	0.73	1.28%
39	0.56	1.30%
40	0.69	2.16%
41	0.6	1.50%
42	0.82	0.84%
43	0.6	1.45%
44	0.62	1.58%
45	0.66	1.73%
46	0.64	1.48%

Figure 97 - Test 1 Decision Tree_FinalTable_University Student

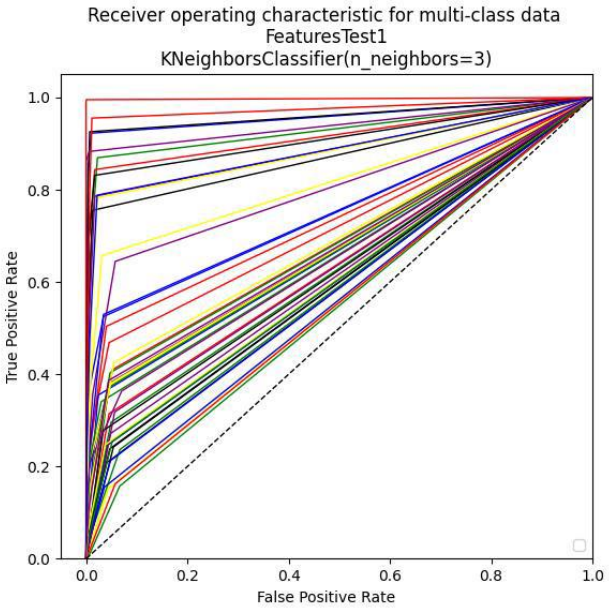
Test 1 – k-Nearest Neighbors



UT	AUC	EER
0	0.83	1.12%
1	0.88	1.70%
2	0.85	1.91%
3	0.75	0.50%
4	0.81	3.44%
5	0.50	0.17%
6	0.82	0.67%
7	0.92	2.44%
8	0.68	4.78%
9	0.84	5.45%
10	0.49	2.35%
11	0.78	4.88%
12	0.75	6.99%
13	0.82	6.81%
14	0.85	1.57%
15	0.49	1.23%
16	0.55	1.28%
17	0.78	1.73%
18	0.91	0.67%
19	0.92	10.60%
20	0.88	11.75%
21	0.88	12.52%
22	0.90	15.61%

Figure 98 - Test 1_k-Nearest Neighbors_FinalTable_ School Student

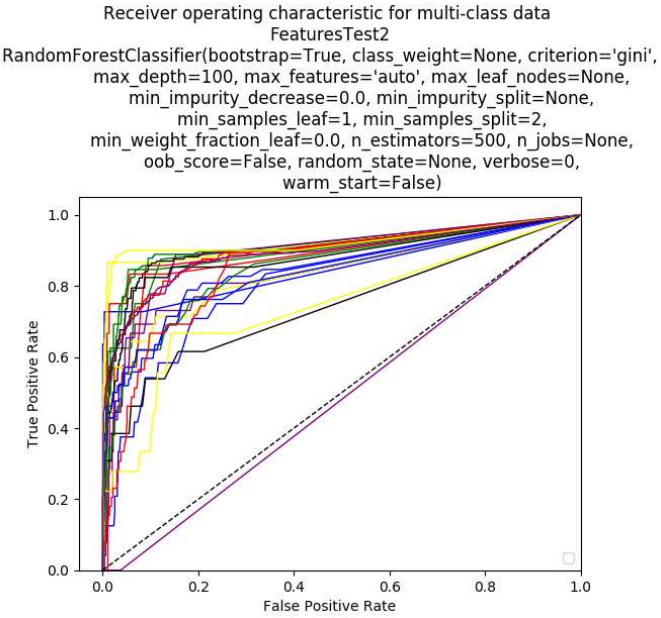
Also, in terms of the results obtained with the kNN algorithm, similarities are noted between the performance obtained on high school seniors and the performance obtained on University Student s. In particular, in the results of University Student s and high school seniors, performance could be better regarding AUC (Figure 98-Figure 99).



UT	AUC	EER
0	0.75	3.45%
1	0.73	4.00%
2	0.93	2.25%
3	0.56	4.62%
4	0.61	4.80%
5	0.87	1.28%
6	0.96	1.03%
7	0.72	4.55%
8	0.55	6.65%
9	0.67	4.60%
10	0.67	4.53%
11	0.96	0.73%
12	0.6	4.06%
13	0.64	4.54%
14	0.62	4.93%
15	0.82	3.13%
16	0.89	1.74%
17	0.58	4.00%
18	0.67	2.38%
19	0.68	5.22%
20	0.59	6.68%
21	0.55	4.43%
22	0.94	0.64%
23	0.59	5.41%
24	0.56	3.60%
25	0.92	1.73%
26	0.66	4.43%
27	0.67	4.93%
28	0.8	5.77%
29	0.62	3.37%
30	0.75	3.44%
31	1	0.06%
32	0.66	2.99%
33	0.88	2.21%
34	0.63	5.65%
35	0.91	1.69%
36	0.58	4.83%
37	0.97	1.15%
38	0.68	4.74%
39	0.61	3.84%
40	0.65	7.03%
41	0.6	4.69%
42	0.89	2.18%
43	0.55	5.76%
44	0.6	5.17%
45	0.69	5.53%
46	0.67	4.46%

Figure 99 - Test 1_k-Nearest Neighbors_FinalTable_University Student

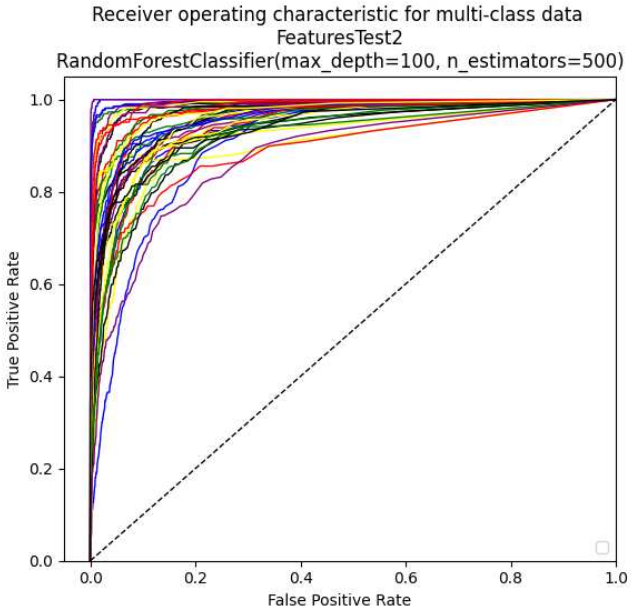
Test 2 – Random Forest



UT	AUC	EER
0	0.80	23.74%
1	0.89	12.88%
2	0.90	12.17%
3	0.92	11.64%
4	0.84	21.99%
5	0.90	13.02%
6	0.85	8.39%
7	0.90	13.45%
8	0.90	11.23%
9	0.84	23.46%
10	0.89	15.32%
11	0.74	21.27%
12	0.84	19.81%
13	0.90	16.41%
14	0.85	22.03%
15	0.74	27.54%
16	0.48	3.71%
17	0.88	14.31%
18	0.83	25.08%
19	0.85	23.05%
20	0.91	10.98%
21	0.93	10.03%
22	0.89	16.30%

Figure 100 - Test 2_Random Forest_Final Table School Student

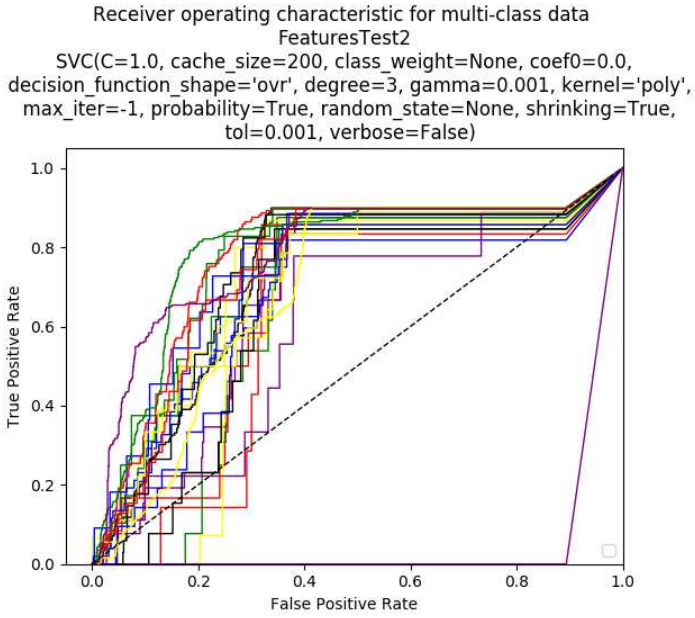
In the second group of tests, the RF algorithm also performs better, consistent with the observations made on University Student s. The only difference is that while the RF algorithm performs better on University Students on Test 2, on high school seniors, the RF algorithm performs better on Test 1. Specifically, a maximum AUC of 93% and a minimum EER of 3.71% are achieved (Figure 100-Figure 101).



UT	AUC	EER
0	0.94	12.69%
1	0.94	13.83%
2	0.96	9.85%
3	0.92	13.32%
4	0.99	3.60%
5	0.94	13.29%
6	1	1.34%
7	0.98	7.34%
8	0.99	3.01%
9	0.93	13.88%
10	1	2.31%
11	0.99	4.93%
12	0.9	18.38%
13	0.95	12.26%
14	0.95	11.71%
15	0.94	11.58%
16	0.95	11.19%
17	0.94	11.31%
18	0.96	10.34%
19	0.89	17.47%
20	0.97	8.40%
21	0.99	4.23%
22	1	0.41%
23	0.92	15.47%
24	0.96	9.73%
25	0.97	8.02%
26	0.92	13.97%
27	0.94	11.52%
28	0.96	11.20%
29	0.96	11.12%
30	1	2.35%
31	0.99	5.41%
32	0.97	7.43%
33	0.95	11.61%
34	0.98	5.17%
35	0.96	9.52%
36	0.96	8.90%
37	0.98	5.38%
38	0.92	14.78%
39	0.97	7.92%
40	0.88	20.27%
41	0.94	13.79%

Figure 101 - Test 2_Random Forest_Final Table_University Student

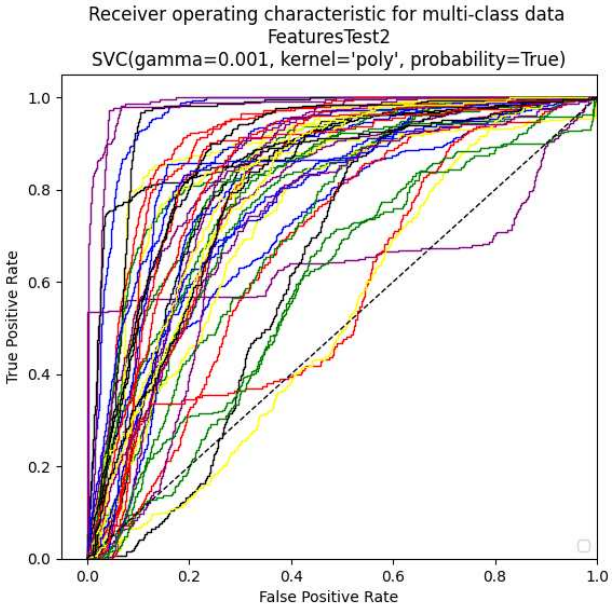
Test 2 – Support Vector Machine



UT	AUC	EER
0	0.71	31.48%
1	0.63	31.92%
2	0.73	28.27%
3	0.73	26.96%
4	0.68	32.25%
5	0.73	30.21%
6	0.71	22.75%
7	0.64	33.20%
8	0.62	34.01%
9	0.62	34.10%
10	0.60	37.92%
11	0.64	31.25%
12	0.72	32.14%
13	0.78	23.55%
14	0.76	23.89%
15	0.72	29.38%
16	0.05	89.38%
17	0.73	26.17%
18	0.68	28.51%
19	0.75	27.99%
20	0.79	19.65%
21	0.69	36.00%
22	0.79	29.48%

Figure 102 - Test 2_ Support Vector Machine_Final Table_School Student

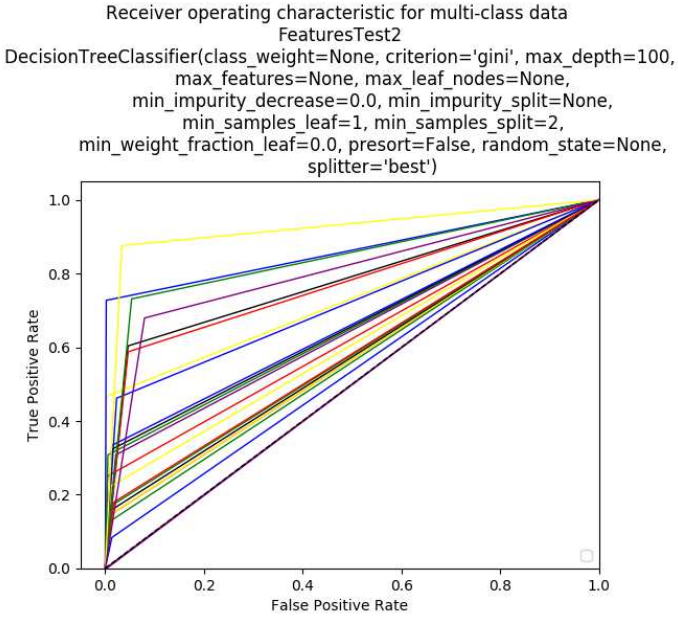
As for the SVM, it was noted that it performed worse than Test 1, as is evident from the trend of the various curves. This observation holds both for the results obtained on University Student s and for the results obtained on high school seniors, where in Test1, a maximum AUC of 93% was obtained and a minimum EER of 9.87%, while in Test2 a maximum AUC of 79% was obtained, and a minimum EER of 19.65% (Figure 102 - Figure 103).



UT	AUC	EER
0	0.72	34.40%
1	0.68	35.80%
2	0.84	23.12%
3	0.84	15.51%
4	0.66	37.39%
5	0.81	23.83%
6	0.96	9.18%
7	0.9	14.50%
8	0.76	28.46%
9	0.8	26.64%
10	0.97	4.27%
11	0.93	9.22%
12	0.77	28.91%
13	0.8	26.02%
14	0.71	34.95%
15	0.74	31.01%
16	0.74	26.37%
17	0.9	18.39%
18	0.8	24.40%
19	0.58	51.63%
20	0.86	19.13%
21	0.85	19.74%
22	0.98	5.71%
23	0.63	39.68%
24	0.77	28.82%
25	0.88	16.64%
26	0.79	25.90%
27	0.53	50.19%
28	0.82	23.79%
29	0.85	18.92%
30	0.83	15.56%
31	0.84	18.44%
32	0.6	41.28%
33	0.81	27.40%
34	0.87	22.07%
35	0.87	20.04%
36	0.88	18.27%
37	0.84	21.59%
38	0.59	40.93%
39	0.82	24.69%
40	0.75	29.51%
41	0.79	26.02%

Figure 103 - Test 2_ Support Vector Machine Final Table_University Student

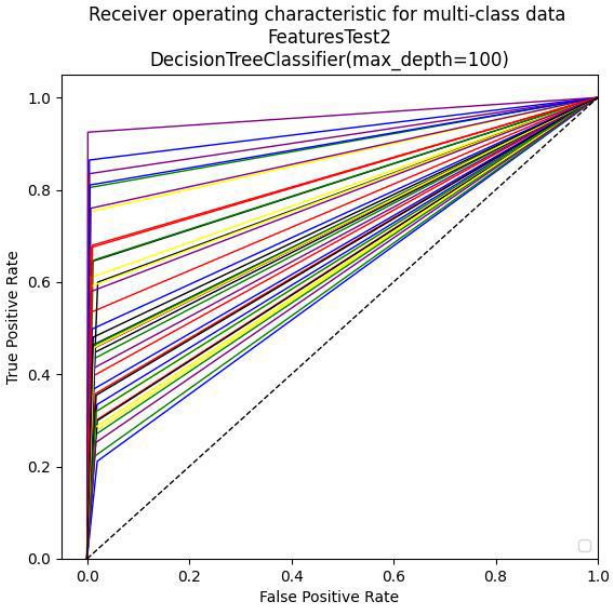
Test 2 – Decision Tree



UT	AUC	EER
0	0.54	1.32%
1	0.57	0.78%
2	0.56	0.65%
3	0.73	0.26%
4	0.64	2.12%
5	0.78	4.61%
6	0.86	0.20%
7	0.62	0.46%
8	0.65	0.53%
9	0.57	0.72%
10	0.50	0.33%
11	0.57	0.72%
12	0.72	2.29%
13	0.77	4.64%
14	0.58	1.53%
15	0.61	1.05%
16	0.50	0.07%
17	0.65	1.40%
18	0.66	1.32%
19	0.58	1.74%
20	0.84	5.33%
21	0.92	3.34%
22	0.80	7.94%

Figure 104 - Test 2 Decision Tree_Final Table_School Student

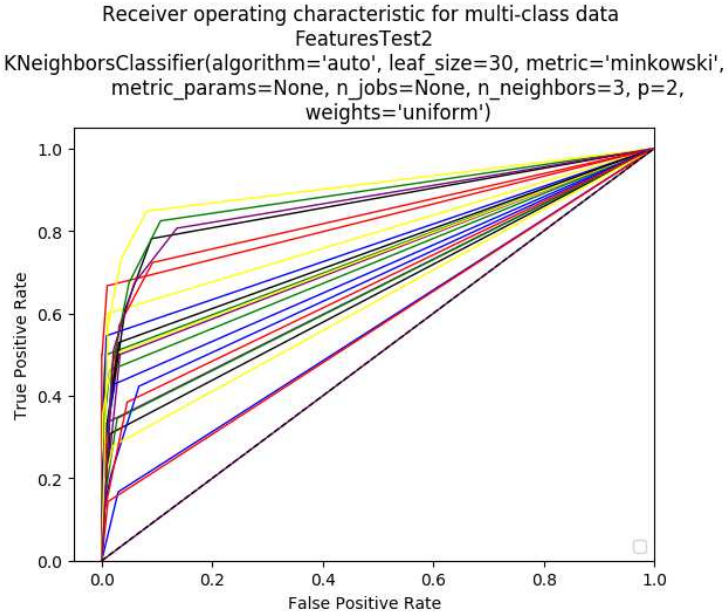
For both works, the DT remains consistent with what happened in Test 1 (Figure 104 - Figure 105).



UT	AUC	EER
0	0.66	1.94%
1	0.67	1.60%
2	0.71	1.48%
3	0.87	0.78%
4	0.88	0.74%
5	0.72	1.23%
6	0.93	0.46%
7	0.76	0.87%
8	0.9	0.38%
9	0.63	1.40%
10	0.92	0.45%
11	0.82	0.83%
12	0.6	2.00%
13	0.69	1.21%
14	0.63	1.15%
15	0.8	0.69%
16	0.72	1.52%
17	0.79	2.10%
18	0.72	1.62%
19	0.64	1.47%
20	0.82	1.21%
21	0.79	0.82%
22	0.96	0.16%
23	0.64	2.02%
24	0.68	1.10%
25	0.72	0.85%
26	0.6	1.41%
27	0.65	1.82%
28	0.7	1.25%
29	0.73	1.21%
30	0.9	0.59%
31	0.83	0.76%
32	0.72	1.35%
33	0.63	1.57%
34	0.79	0.81%
35	0.67	1.78%
36	0.74	0.97%
37	0.83	1.06%
38	0.65	1.81%
39	0.72	1.40%
40	0.62	1.74%
41	0.73	1.20%

Figure 105 - Test 2_ Decision Tree_Final Table_University Student

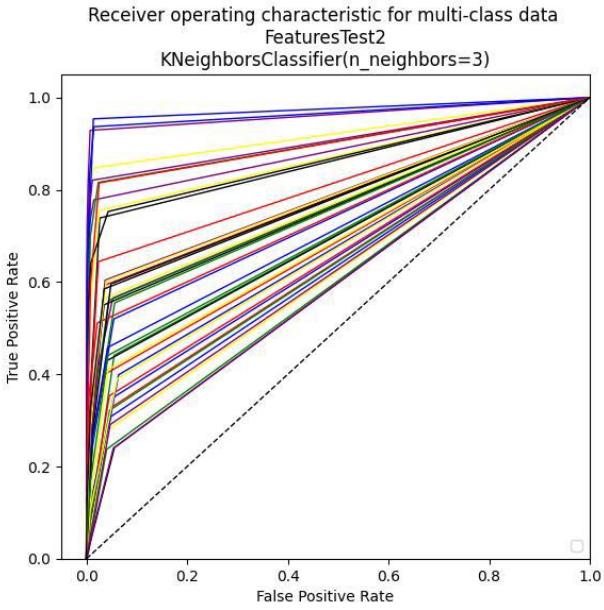
Test 2 – k-Nearest Neighbors



UT	AUC	EER
0	0.57	2.98%
1	0.57	1.11%
2	0.75	0.92%
3	0.80	1.18%
4	0.73	3.31%
5	0.86	9.00%
6	0.77	0.85%
7	0.83	0.98%
8	0.73	1.44%
9	0.74	2.04%
10	0.66	0.72%
11	0.65	1.51%
12	0.68	6.74%
13	0.83	9.20%
14	0.66	2.79%
15	0.63	1.98%
16	0.50	0.07%
17	0.75	3.13%
18	0.71	2.38%
19	0.67	4.61%
20	0.88	10.66%
21	0.90	8.16%
22	0.87	13.60%

Figure 106 - Test 2_k-Nearest Neighbors_Final Table_School Student

The k-NN also remains consistent with Test 1, with low average performance for college and high school seniors (Figure 106 - Figure 107).



UT	AUC	EER
0	0.67	6.40%
1	0.64	5.08%
2	0.69	5.50%
3	0.92	1.61%
4	0.89	1.47%
5	0.7	4.11%
6	0.97	1.38%
7	0.78	3.64%
8	0.9	2.36%
9	0.62	4.33%
10	0.91	1.19%
11	0.86	2.64%
12	0.65	5.35%
13	0.65	4.55%
14	0.6	4.01%
15	0.87	2.60%
16	0.76	5.25%
17	0.87	4.32%
18	0.74	5.44%
19	0.69	3.50%
20	0.76	4.40%
21	0.77	3.31%
22	0.96	0.74%
23	0.59	5.57%
24	0.63	4.86%
25	0.75	2.17%
26	0.64	4.64%
27	0.67	6.14%
28	0.62	4.73%
29	0.76	3.52%
30	0.97	1.49%
31	0.81	2.36%
32	0.75	5.62%
33	0.68	4.30%
34	0.79	3.70%
35	0.78	4.86%
36	0.71	4.24%
37	0.9	2.55%
38	0.7	4.30%
39	0.79	3.93%
40	0.59	5.34%
41	0.78	3.54%

Figure 107 - Test 2_k-Nearest Neighbors_Final Table_University Student

3.3.3.2.4 Observations

Experiment 1

Regarding the video-related activities, it was observed that, in the present case study, on the dataset of School Student s using a low threshold for *Anomaly Detection algorithms*, the Elliptic Envelope algorithm is the one that succeeds in detecting, averaged over the various activities, the highest percentage of anomalies. Relative to the anomalies detected in the 5 classes considered (*Bullying_Victimization, Bullying_Bully, Cyberbullying_Victim, Cyberbullying_Cyberbully, External*), again, Elliptic Envelope was the algorithm that was able to detect the highest number of anomalies in the various classes. The exact proportionality as in the activity case is maintained: again, following this case, Isolation Forest, Local Outlier Factor, and One-Class SVM were found (in order). Similar behavior of the algorithms used on the University Student dataset was observed.

As the threshold mentioned above (average parameter) increases, for both University Student s and high school seniors, there is the following behavior of the various algorithms: the Elliptic Envelope algorithm averaged over the activities (or classes, respectively), detects a lower average percentage of outliers than the two algorithms Isolation Forest and Local Outlier Factor.

Using a high parameter, however, in the case of School Students, the two algorithms, Elliptic Envelope and One-Class SVM, could not detect any outliers. The opposite behavior was, on the other hand, detected in the dataset of University Student s, where (both in the case of the activities and in the case of the 5 classes), the values computed by Elliptic Envelope remain constant above the values computed by Isolation Forest and Local Outlier Factor.

It is also possible to infer the specific activities performed by users. In the case of School Student s, it is possible to note that the activity related to the first video (Video1Activity) is the one with the highest number of anomalies detected at the time of viewing it (62.36%) and with the highest number of anomalies detected in the *Bullying_Victimization* class (low parameter). A similar situation was found in the case of University Student s, where for the first video, there is an average percentage of anomalies of 51.06%, while as far as the categories are concerned, the highest number of anomalies was found in the class *Bullying_Bully* (also increasing the threshold value).

Therefore, it is hypothesized that among high school seniors (analyzed in this case study), there is a more significant number of bullied pupils and that the first video may have content that is particularly susceptible to younger users.

Regarding the quiz and 5-class activities, it was noted that however one chooses the level of the parameter used (low, medium, or high), the most significant number of anomalies were found in the *Cyberbullying_Victimization* and *Cyberbullying_Cyberbully* classes (School Student s). In contrast, the most significant number of anomalies were found in the Bully class in the case of University Students.

Therefore, it is hypothesized that among high school seniors, there is a higher frequency of users resorting to digital tools for bullying, probably also due to the young age of such users.

In summary, three different types of conclusions can be drawn, as well as in [4]: Regarding the algorithms used, it is possible to state that:

- The **Elliptic Envelope algorithm** is the one that was able to detect, averaged over the various tasks, the highest percentage of anomalies, both in the experimentation carried out on University Student s and in the experimentation carried out on high school seniors;

- Just as observed in the experimentation carried out on University Student s, in the case of high school seniors, there is an intense closeness between the values calculated by Isolation Forest and those calculated by Local Outlier Factor;
- The **One-Class SVM** proved to be the algorithm that can differentiate outliers, identifying fewer of them, just as in the experimentation on University Student s.

Regarding classes, the following differences were identified:
- In the experiment on University Student s, the **Bully** and **Cyberbully** classes encounter more anomalies on average. However, you set a level of the considered threshold. In the case of School Student s, increasing the threshold means that the **Bully** and **Cyberbully** classes encounter zero anomalies;
- In the experiment on University Student s, the **Bully** class experiences more anomalies when a higher threshold is used. In the case of School Student s, the opposite occurs even by increasing the threshold;
- As for the 5 classes, the most anomalies were found in the **Bullying_Bully** class (even by increasing the threshold value). Therefore, it is hypothesized that among School Student s (analyzed in this case study), there is a more significant number of bullied pupils and that the first video may have content that is particularly susceptible to younger users.

Regarding activities:
- On both the University Student dataset and the School Student dataset, the **videos encounter more anomalies than the questions;**
- In the case of University Student s, no anomalies were found in the QuizActivityButtons0, QuizActivityButtons1, QuizActivityButtons2, and QuizActivityButtons4 (Category: Bullying) activities. That is, "Let's say a boy/girl gets bullied when another boy/girl or a group of boys/girls: *," "How many times have you been bullied? *", "*BULLISM: Indicate how often you have SUFFERED bullying a) I have been beaten up **," "*BULLISM: Indicate how often you have SUFFERED bullying b) I have been called bad names **."

In the case of **School Student s**, the *QuizActivityButtons0*, *QuizActivityButtons1*, and *QuizActivity-Buttons2* (Category: Bullying) questions found anomalies, namely, "Let's say a boy/girl gets bullied when another boy/girl or a group of boys/girls: *", "How many times have you been bullied? *", "BULLISM: Indicate how often you have SUBJECTED to bullying a) I have been beaten up *." Therefore, keeping these questions in the quiz is deemed necessary rather than removing them, as the presence or absence of anomalies depends on the specific user.

- In the case of University Student s, the most significant number of anomalies was found in the last questions (**Cyberbullying_Cybully**). The opposite situation was observed in the case of School Student s. The same can be said for the first question (**Bullying_Victimization**)

Experiment 2

Tables are now shown with the average AUC and EER obtained first on Tests taken on high school seniors and later those taken on University Student s (Table 38-Table 39):

Classifier	*AUC (%)*	*EER (%)*
RandomForest	94	11.08
DecisionTree	69	2.00
K-nearest neighbors	78	4.36
SupportVectorMachine	82	24.94

Table 38 - Test1 School Student

Classifier	*AUC (%)*	*EER (%)*
RandomForest	85	16.43
DecisionTree	66	1.88
K-nearest neighbors	73	3.89
SupportVectorMachine	67	32.19

Table 39 - Test2 School Student

For each Test conducted on University Student s, the average AUC and EER values are as follows (Table 40-Table 41):

Classifier	*AUC (%)*	*EER (%)*
RandomForest	95	10.54
DecisionTree	71	1.19
K-nearest neighbors	72	3.81
SupportVectorMachine	78	27.46

Table 40 - Test1 University Student

Classifier	*AUC (%)*	*EER (%)*
RandomForest	96	9.77
DecisionTree	74	1.21
K-nearest neighbors	76	3.83
SupportVectorMachine	79	25.2

Table 41 - Test2 University Student

Looking at the average values obtained on both datasets, the best-performing algorithm turns out to be the RF (as also observed in the previous work for the University Student students, the best-performing algorithm on average was the RF on the second test, in which a maximum AUC of 100% and a minimum EER of 0.41% is noted). In addition, there is some uniformity in the performance of the other algorithms between the two datasets. In fact, regarding Test1, SVM is the second algorithm with the highest AUC value, followed by k-NN and DT. The exact proportionality can be seen on the EERs. The only discrepancy is in Test 2, where, about high school seniors, k-NN performs better than SVM, while in Test 2 on University Student s, the opposite occurs. Last observation, not least, concerns that the RF and SVM algorithms present higher EER values, while the DT presents lower EER values on the dataset of high school seniors than University Student s. As for the University Student s, the users who filled out the questionnaires were always the same, and, therefore, most likely, there was no device exchange during the experiment. This is due to the excellent performance obtained with such experimentation. However, regarding high school seniors, the decrease in AUC values leads one to think that there was more influence caused by external factors (e.g., device swapping or influence from any classmates) during the completion of the questionnaires by high school seniors.

The last question on the quiz, namely Question No. 82 (Indicate how often you have DONE acts of cyberbullying 18. Blocking someone in chat or on Facebook to exclude them from the group *"), from the Cyberbullying category, is among the questions with the most anomalies, both for college and School Student s. The second question of the quiz, i.e., Question No. 1 ("*Let's say a boy/girl gets bullied when another boy/girl or a group of boys/girls:" *" - "hits, kicks, pushes or threatens him/her*"), from Bullying category, was the question with the slightest anomalies. The video with the most minor anomalies was Video 2 (Video2Activity), with an average percentage of anomalies of 52.08%. Videos had approximately more anomalies than quizzes. Among the first questions, in the Bullying category, in the case of School Student s, more anomalies are found than in the case of University Students.

In the case of School Students, it is possible to note that the activity related to the first video (Video1Activity) is the one with the highest number of anomalies detected at the time of viewing it (62.36%) and with the highest number of anomalies detected in the Bullying_Victimization class (low parameter). Looking at the average values obtained for Experiment 2, it can be seen that the Random Forest is the best-performing algorithm for both School Student s and University Student s. While in the case of the University Students, it is emphasized that the users who filled out the questionnaires were always the same and, therefore, most likely, there was no device swapping during the experiment, in the case of the School Student s, the decrease in AUC values leads one to think that external factors caused more influence during the filling out of the questionnaires.

In general, both similarities and differences can be seen in both experiments. There are Machine Learning algorithms that maintain their performance even on different datasets, as they are still predefined procedures. However, it has been noted that some results are mainly attributable to the category of users analyzed since bullying and cyberbullying are two phenomena that are increasingly prevalent among young people. Therefore, it is believed that the methodology applied is sufficiently valid in that, on the one hand, similarities are found if the specific algorithms that are used are considered, and on the other hand, differences are found whose causes may be multiple: the young age of the users, the lived experiences of the users, their emotional sensitivity regarding particularly susceptible multimedia content, and whatnot. One possible way to validate this methodology even more would be to use behavioral characteristics that are even more discriminating, given the issue at hand.

4. Human Activity Recognition for Security and Safety

Mobile device security, particularly for smartphones, represents one of the most critical challenges within the realm of cybersecurity. With the rise in daily use of these technologies, the necessity of ensuring robust protection for sensitive data has become paramount. In recent years, research has focused on innovative approaches to continuous authentication, combining touch events and human activity to enable constant monitoring of user interactions with the device, all without compromising user experience. This evolution has been facilitated by advancements in machine learning and data analysis, allowing for the extraction of significant insights from users' daily behaviors.

In this chapter, I present three papers that explore various dimensions of this critical theme.

The first paper, *"4.1 Touch Events and Human Activities for Continuous Authentication via Smartphone,"* develops a method for continuous authentication by leveraging touch events and smartphone sensors. This study highlights how characterizing user behavior through data collected during device interactions can markedly enhance the ability to identify legitimate users and detect unauthorized access. Employing movement analysis and touch-based interactions allows for a background security system that operates without requiring active input from the user, making authentication smoother and more intuitive.

The second paper, *"4.2 Two-factor authentication by combining PIN and biometrics Touch Dynamics"* proposes a user verification system that integrates touch dynamics during PIN entry. This innovative approach not only addresses the vulnerabilities of traditional authentication methods, which rely solely on mnemonic knowledge (such as passwords and PINs), but also introduces an additional security layer through biometric analysis. The combination of these two methods reduces the risk of cyber-attacks, as a potential intruder could not simply replicate a PIN without also possessing the unique touch characteristics of the legitimate user. This study underscores the need to adopt multi-factor approaches to ensure robust protection of sensitive data.

Finally, the third paper, *"4.3 Human Activity Recognition using Smartphone Sensors: Focusing on Fall Detection with the UNIBA HAR Dataset,"* examines how smartphone sensors can be employed to recognize human activities, with a particular focus on fall detection. This research zeroes in on high-risk activities like falls, which not only pose a significant public health concern but may also indicate situations of bullying or physical aggression. Through the collection and analysis of data from a purposefully designed dataset, this work demonstrates how machine learning models can distinguish between routine activities and dangerous situations, thereby contributing to the development of timely monitoring and alert systems in critical contexts.

Through an in-depth analysis of these studies, this chapter aims to highlight the potential of continuous authentication methods and human activity recognition. It underscores how the integration of advanced technologies can bring substantial improvements to mobile device security, not only protecting personal data but also promoting the well-being and safety of users in vulnerable situations. This chapter aspires to provide a comprehensive view of the challenges and opportunities within the cybersecurity domain, emphasizing the need for innovative, multidisciplinary solutions to address emerging threats.

4.1 Touch Events and Human Activities for Continuous Authentication via Smartphone

The security of modern smartphones is related to the combination of Continuous Authentication approaches, Touch events, and Human Activities. The approaches of Continuous Authentication, Touch Events, and Human Activities are silent to the user but are a great source of data for Machine Learning Algorithms. This work aims to develop a method for continuous authentication while the user is sitting and scrolling documents on the smartphone. Touch Events and Smartphone Sensor Features (from the well-known *H-MOG Dataset*) were used with the addition, for each sensor, of the feature called Signal Vector Magnitude. Several Machine Learning Models have been considered with different experiment setups, 1-class, and 2-class, for evaluation. The results show that the 1-class SVM achieves an accuracy of 98.9% and an F1-score of 99.4%, considering the selected features and the feature Signal Vector Magnitude very significant [189].

Protecting smartphones is one of the main challenges in cybersecurity. *Knowledge-based approaches* are authentication methods that verify user identity based on secret mnemonic knowledge. Human tends to want to memorize simple information and passwords, being simple and short can easily be guessed and stolen. In addition, the peculiarity of *Knowledge-Based approaches* is that they are one-shot: authentication is performed only once and is no longer required while using the smartphone. In fact, after performing the first and unique authentication, an attacker, since that moment, impersonates the victim. These actions can then fall into personal violence or cyberbullying [190].

Scientific research is moving toward implementing approaches called Continuous Authentication and approaches that combine *Touch Events and Human Activities* [176]. Continuous Authentication tends to perform more security check while the user uses the device. Touch and Human Activities approach, as with Continuous Authentication approaches, depend on characterizing the user's behavior while using, in this case, the smartphone. The behavior identifies the user because they naturally touch the screen like no other user and walk or run like no other user. Touch Events (*moments when the user touches the screen*) and Human Activities (*Walking, Running, Jumping, Fall, etc....)* [191] are intrinsic to user behavior and can, together, identify them. In addition, continuous authentication approaches have the advantage of working in the background (*silently to the user*). This latter aspect increases the usability of the approach because it becomes universal without adding additional hardware or requiring specific actions to be performed by the user.

This work is focused on a specific approach that ties into Continuous Authentication and Touch and Human Activities events from smartphones. A method is developed for continuous user

authentication while using a smartphone by identifying possible illegitimate users during sitting and reading activity (*scrolling a document in the background on the smartphone screen*). Touch event-related features and sensor-related features, including accelerometer, magnetometer, and gyroscope, were considered in this experiment. For each sensor (X, Y, Z), the Signal Vector Magnitude was considered, thus producing four features (X, Y, Z, M). In addition, experiments are performed on a portion of the HMOG dataset that, as mentioned earlier, characterizes the action of sitting while strolling through a chat/document. Machine learning models trained with GENUINE and IMPOSTOR features with two different setups (1-class and 2-class) are used to evaluate the experiment. The result is valid as a comparison between the selected machine learning models to determine and test the selected features if they perform for the document reading and sitting task.

To the best of the authors' knowledge, this is the very first experiment in this direction, that is, to consider this inherent combination of Touch Events and Human Activities with a portion of the HMOG dataset inherent in the activity of reading a document and being seated via smartphone.

4.1.1 Related Works

Biometrics is grouped into two categories: behavioral biometrics and physiological biometrics. Physiological biometrics is based on a person's physical attributes such as fingerprints, finger or palm veins, face shape, DNA, handprint, hand geometry, iris, or eye retina recognition. On the other hand, behavioral biometrics is closely related to a person's habits, such as typing rhythm, gait, and voice. Behavioral biometrics enables continuous and passive authentication. This means behavioral characteristics are continuously captured and compared with the user's profile throughout the session, not just log-in. Behavior profiling is considered in many studies.

Numerous studies deal with the problem of continuous authentication using the accelerometer, magnetometer, and gyroscope as sensors. In Zhu et al.[192], a framework, SenSec, is presented that constantly collects sensory data from accelerometers, gyroscopes, and magnetometers and builds the gesture model of how a user uses the device. *SenSec* calculates the confidence that the mobile device is being used by its owner. The authors show that this framework can achieve 75% accuracy in identifying users and 71.3% accuracy in identifying non-owners, with only 13.1% false alarms. In Lee et al. [193], researchers design a system based on multiple sensors that continuously learn the owner's behavior patterns and the characteristics of the environment and then authenticate the current user without interrupting user-smartphone interactions. This method can adaptively update the user's model by considering the temporal change of the user's patterns. Experimental results show that the method provides more than 90% accuracy. The method also shows that the combination of multiple sensors provides better accuracy. In Amini et al. [194], motion sensors embedded in available smartphones are utilized to learn users' behavioral characteristics during interaction with the mobile device and provide an implicit re-authentication mechanism. This approach uses time and frequency domain features extracted from motion sensors and a short-term memory model (LSTM) with negative sampling to build a re-authentication framework. The framework can re-authenticate a user with 96.70% accuracy in 20. In Ehatisham-ul-Haq et al. [195] authentication framework is proposed that provides a platform for multi-class user authentication using twelve extracted features. It is reported that the Bayes Net classifier provides the best performance for activity recognition on the device regarding EER accuracy and computation time required for activity classification. In Abuhamad et al. [196], AUToSen, a deep-learning-based active authentication approach, is proposed, demonstrating that AUToSen works accurately using readings from only the three sensors. The use of one-second sensor data allows an F1 authentication score of approximately 98%, a false acceptance rate (FAR) of 0.95%, a false rejection rate (FRR) of 6.67%, and an equal error rate (EER) of 0.41%. In Mekruksavanich et al. [197], a new continuous authentication framework called DeepAuthen is introduced. It identifies smartphone users based on their smartphones' physical activity patterns measured by the accelerometer, gyroscope, and magnetometer sensors. Scientists conduct a series of tests on user authentication using different deep learning classifiers and a new deep learning network called

DeepConvLSTM.

Some studies need to consider the use of the three sensors. Some researchers use only accelerometer data as a sensor. In particular, Kwapisz et al.[198] researchers collect accelerometer data from thirty-six users while performing normal daily activities such as walking, jogging, and climbing stairs. They then aggregate these time series data and apply classification algorithms to the resulting data to generate predictive models. In Centeno et al. [199], an approach based on a deep learning autoencoder is studied that achieves an EER of 2.2% in real-world scenarios. The system uses accelerometer data. In addition, the sensing process is carried out in the cloud to reduce the computational load of the smartphone.

Another approach is proposed by Li et al. [200], in which only two sensors are used. Researchers present SCANet, a continuous authentication system based on two-stream convolutional neural networks that use the accelerometer and gyroscope of smartphones to monitor users' behavioral patterns. The system uses the two-stream CNN to learn and extract representative features. With the features extracted from the CNN, SCANet uses the class support vector machine to train the classifier in the enrolment phase. The experimental results show that the CNN achieves 90.04% accuracy, and SCANet achieves an average of 5.14% equal errors.

However, not all studies on passive and continuous authentication are based on using sensors in smartphones. Some research in the literature studies the problem of using the touchscreen to detect a legitimate user. For instance, Frank et al. [122] propose 30 haptic data features obtained from users interacting with a smartphone by performing basic navigation operations such as up-down and left-right scrolling. The trained classifier obtained an EER of 0% for intra-session authentication, 2%-3% for between-session authentication, and less than 4% when the authentication test was performed one week after the registration phase. Garbuz et al. [201] present a continuous user authentication system based on user interaction with the touchscreen in combination with micro-movements performed simultaneously by smartphones. Two of the users' most common gestures (vertical swipes up and down and taps) are considered. The researchers use the One-Class Support Vector Machine algorithm to obtain a model of a legitimate user. The results show that the legitimate user is blocked on average after 115-116 gestures (*a combination of swipes and taps*), and an imposter is detected in 2-3 gestures. Shen et al. [202] considered four common types of touch operations, features are extracted to characterize users' touch behavior, and one-class classification algorithms are used. The results are a FAR of 4.68% and FRR of 1.17%.

Other studies combine several approaches to study the problem. For example, some researchers use sensors and touchscreen data. Volaka et al. [203] examines the impact of using the touchscreen and sensor-based features in an authentication model using deep learning methods. A three-level deep neural network is constructed on the combined feature sets. The results achieved 88% accuracy and EER values of 15%. In Incel et al. [123], researchers examine whether it can continuously authenticate users via behavioral biometrics on a mobile banking application. A continuous authentication scheme, called DAKOTA is developed that records data from the phone's touch screen and motion sensors to monitor and model the user's behavioral patterns. The results reveal that binary-SVM has an EER of 3.5 percent. Another approach is considered in Smith-Creasey et al. [204], where facial and haptic modalities are combined, demonstrating that a stacked classifier can improve continuous authentication on mobile devices. An EER of 3.77% for a single sample is achieved.

Data security in the way of smartphones is critical. Defining a secure device goes through standard or continuous authentication and *general security issues, Data Analysis, Energy Efficiency, and Anomalous Behavior*. In security, issues such as edge computing are highly relevant in smartphone authentication. Edge computing could provide greater security and reduce latency while performing authentication [205]. Device security also passes through device immunity it is essential to protect it

from possible data poisoning performed by third parties [206]. Smartphone protection also involves data analysis to optimize smartphone processes and identify likely suspicious patterns [207]. In addition to processes, energy efficiency is essential in smartphones, so optimizing energy would lead to improved security in authentication and user experience with the same [208]. Finally, the anomaly can also be identified by people's abnormal behavior (Smart City scope), a concept emphasized in this paper, through analysis of smartphone data generated [209].

4.1.2 Material

In order to extract features considering Touch Events and Human Activities from smartphones, the H-MOG dataset (A Multimodal Data Set for Evaluating Continuous Authentication Performance)[210] has been used. The dataset has three user usage scenarios or activities: Reading Documents, Text Writing, and Navigating a Map to locate a Destination.

The dataset has been built adopting an Android smartphone to record the data stream related to the Touch and Hardware Sensors installed in the device in real time. This was performed to capture user behavior. One hundred users were recruited to experiment. Users are randomly assigned a session to read, write or navigate the map. Each session lasts about 5-15 minutes, and each user has 24 sessions (eight reading sessions, eight writing sessions, and eight map navigation sessions). Each user contributes about 2-6 hours of behavioral traits.

The collected data are stored in CSV files. Data acquisition from the sensors has a sampling rate of 100 Hz. Nine categories of data are collected [210]:

1. *Accelerometer*: Timestamp, Acceleration along X/Y/Z-Axis.
2. *Gyroscope*: Timestamp, Rotation Rate along X/Y/Z-axis.
3. *Magnetometer*: Timestamp, Ambient Magnetic Field along X/Y/Z-axis.
4. *Raw touch event*: timestamp, finger count, finger ID, raw touch type, X/Y coordinate, contact size, screen orientation.
5. *Tap gesture*: timestamp, tap type, raw touch type, X/Y coordinate, contact size, screen orientation.
6. *Scale gesture*: timestamp, pinch type, time delta, X/Y focus, X/Y span, scale factor, screen orientation.
7. *Scroll gesture*: starting and current timestamp, X/Y coordinate, and contact size; speed along X/Y coordinate; screen orientation.
8. *Fling gesture*: starting and ending timestamp, X/Y coordinate, and contact size; speed along X/Y coordinate; screen orientation.
9. *Keypress on virtual keyboard*: timestamp, press type, key ID, screen orientation

4.1.3 Method and Experiment setup

This chapter describes the adopted approach using the following Pipeline: Dataset and pre-processing, features extraction, models, and evaluation.

In this case, the design pipeline is more concerned with the *Dataset* and the *Features Extracted* for model evaluation. The first phase (A) contemplates the information about the sample extracted from the H-MOG Dataset and the importance of cleaning the dataset from incorrect detections, which affects the preprocessing phase more. The second phase (B) involves the extraction of features from the raw data of the dataset. Phase (B) prepares the data for the machine learning models that are mentioned later and the methods for evaluating them (Phase (C)).

4.1.3.1 Dataset and Pre-processing

Twenty users were considered in this experiment, including those who performed the "Reading Documents" usage scenario. The activity is reading a document from a smartphone while strolling through

it with the finger (*Touch Event*) and while sitting (*Human Activity*). The hardware sensor of the device picks up the accelerometer, gyroscope, and magnetometer triaxial.

From a preliminary analysis of the dataset, repeated activities are identified. Some activities have the same code, and the same activity starts time but at a different end time. Since it is impossible to understand why this situation occurs, removing the records related to these activities is preferred. An example is shown in Table 42:

ActivityID	SubjectID	Session_number	Start_time	End_time	Relative_Start_time	Relative_End_time	Gesture_scenario	TaskID	ContentID
100669011000001	100669	1	1396226213027	1396226407573	6792617	6987163	1	7	1
100669012000001	100669	1	1396226421894	1396226578198	7001484	7157788	1	7	2
100669012000002	100669	1	1396226600720	1396226650781	7180310	7230371	1	7	2
100669012000002	100669	1	1396226600720	1396226653876	7180310	7233466	1	7	2
100669012000003	100669	1	1396226672419	1396226737648	7252008	7317238	1	7	2
100669013000001	100669	1	1396226745978	1396226937462	7325568	7517052	1	7	3
100669013000002	100669	1	1396226942237	1396226953506	7521827	7533096	1	7	3
100669013000003	100669	1	1396226964010	1396226974209	7543600	7553798	1	7	3

Table 42 - For activity 100669012000002 two records have the same activity start time and a different activity end time

The dataset is preprocessed to identify reading session 1 among the 20 selected users. The session activities are different for each. For example, in the first session, one user might have performed a writing activity, while another might have performed a map browsing activity.

4.1.3.2 Feature extraction

In this work, the following sensors are considered for each user: Event Touch for sensors accelerometer, magnetometer, and gyroscope. Each sensor contains, among other data, the X, Y, and Z coordinates. These data are augmented with the *"Signal Vector Magnitude"* calculated on each sensor as follows:

$$M = \sqrt{X^2 + Y^2 + Z^2}$$

Concerning the *Event Touch*, each of these events has a system that indicates the moment when the user makes a touch on the screen. For each *Event Touch*, the following time interval was considered:

[SYSTIME - 100ms, SYSTIME + 100ms]

This time interval is used to extract features from the sensor. For each sensor coordinate, the sensor data's maximum, minimum, mean, and standard deviation lie between SYSTIME-100ms and SYSTIME, between SYSTIME and SYSTIME+100ms, and the difference of the values 100ms before and between the values 100ms after are calculated.

An explanatory image of the time points is shown in Figure 108.

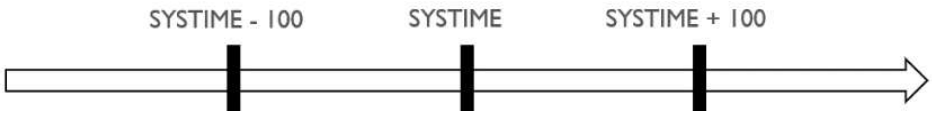


Figure 108 - Time points

For example, is considered the X direction of the accelerometer:

- The *MAXIMUM, MINIMUM, AVERAGE*, and *DVST* of the X data in the interval between [*systime-100, systime*];
- The *MAXIMUM, MINIMUM, AVERAGE*, and *DVST* of the X data in the interval between [*systime, systime+100*];
- The differences between the values 100ms before and 100ms after.

Performing a calculation inherent to the feature calculated in this study: four sensors axis (X, Y, Z, M), three sensors (*Accelerometer, Gyroscope, Magnetometer*), and 12 features extracted from each sensor, there are 144 features (*4*3*12*). Moreover, seven following additional features are added:

1. *gesture_scenario*
2. *task_id*
3. *pointer_count*
4. *pointer_id*
5. *action_id*
6. *content_id*
7. *phone_orientation*

In total, for each user, there are 151 features.

4.1.3.3 Model and Evaluation

The classification problem has been considered in two different approaches. In the first case, a binary classification has been performed between the target (authorized) users' class and the non-authorized (impostor) class. The following Machine Learning Models are adopted: Decision Tree, Random Forest, and Multi-layer Perceptron. Regarding the second case, the 1-class SVM classifier was considered. These machine learning models were chosen because they are most used in the context of continuous authentication. Evaluation of the models is done by accuracy and f1-score, these metrics are among the most widely used for supervised approaches and are very significant metrics for data observation and analysis.

The first experiment considers 2 classes for each user: GENUINE and IMPOSTOR. The records of the GENUINE class identify the genuine user, and the records of the IMPOSTOR class are considered malicious users. More specifically, the IMPOSTOR class for each genuine user includes the entire set of features of the remaining 19 other users. The dataset is divided into Training and Test, respectively, 70-30% randomly. Finally, the average accuracy and F1-score of the twenty users are shown in the results table of the first experiment. An example of the first experiment is shown in Table 43.

User	Number of records	CLASS
Authorized user U1	1000	GENUINE
U3	995	IMPOSTOR
U4	993	IMPOSTOR
U5	1010	IMPOSTOR
U6	1015	IMPOSTOR
U7	1017	IMPOSTOR
U8	2300	IMPOSTOR
U9	2700	IMPOSTOR
…	…	…
U20	2200	IMPOSTOR

Table 43 - Table depicting GENUINE and IMPOSTOR users for modeling User 1

The second experiment considers 1-class to consider real cases in which impostors are not available in advance at training time. In fact, in the previous case, the assumption that the impostors are known at training time is very unreal. In this experiment, for each genuine user, a model is trained considering only its own 151 features. At testing time, the model is tested on the features of each of the 19 other users considered as never-seen impostors. This process is carried out for each of the twenty selected. Finally, the averages inherent in model accuracy and F1-score across all twenty users are extracted.

Table 44 shows results related to the first experiment. This Experimental setup confirms that both classes were correctly recognized in the test. The model that performed best was the Random Forest.

	Accuracy	*F1-score*
Decision Tree	0,95	0,93
Random Forest	**0,96**	**0,95**
MLP	0,92	0,91

Table 44 - Average Results 20 users in 2-class.

The result of the second experiment is considered a more compliant and balanced model, in which each model is trained on the individual user and tested on each of the others. The average accuracy obtained is 98.9% (Table 45).

#User	Accuracy	F1 score macro
Average	0,98	0,99

Table 45 - Average Results 20 users in 1-class

The purpose of this work has been to develop a method for continuous user verification while using a smartphone and to identify illegitimate users during a reading activity (an activity that an illegitimate user, after stealing the smartphone device, could perform by reading and scrolling through a chat while comfortably sitting in a chair).

The set of raw features acquired by the sensors has been augmented by calculating the "Signal Vector Magnitude" feature. The classification problem has been considered a two-class problem and a one-class one. In the former case, the hypothesis is that impostor trials are available at training time, in the latter, the (real) hypothesis is that impostors are not known at training time. Regarding the models considered, the setup of the *first experiment (2-class)* decreed the *Random Forest* as the best model, while in the second setup test, the 1-class SVM performed well. Even if the results are encouraging, these conclusions cannot be generalized due to the limited number of users within the dataset.

As a future development, other bullying-related activities could be identified, and their authentication verified using smartphones. It is also essential to build an extended dataset so that more complex methods can be applied: Multi-speed transformer network [211] e AUCO Resnet [212]. Another relevant point could be developing these solutions on a smartphone device.

4.2 Two-factor authentication by combining PIN and biometrics Touch Dynamics

Mobile devices are becoming increasingly popular. The widespread use exposes individuals to unintentionally sharing sensitive information that could allow direct access to the mobile device. This paper implements and tests a user verification system based on touch dynamics biometrics while typing a Personal Identification Number (PIN). The proposed framework includes several features and feature selection. The classification problem has been considered in terms of one-class and binary classifiers. The proposed approach can outperform the current State of the art on the binary classification problem on the considered dataset, achieving an Equal Error Rate (EER) of 0.01%.

Mobile devices, such as smartphones and tablets, have become more and more prevalent in the population due to the continuous evolution of technology. The wide use of these devices for daily activities such as work, online payments, taking photographs, etc., makes these devices huge containers of sensitive information. Therefore, it is necessary to design techniques to protect access and ensure the security of their data. User authentication is the security measure that intervenes as the

first line of defense. Usually, user authentication uses knowledge-based methods such as entering a code based on numbers, characters, or patterns or biometrics-based methods such as fingerprints and facial recognition [122], [192], [202].

However, these approaches are subject to vulnerabilities[213]. The Knowledge-Based ones are prone to brute force attacks [214], shoulder surfing attacks [215], and smudge attacks [216]; on the other hand, the latter are subject to coercive attacks, impersonation attacks, and replay attacks and require the implementation of specialized hardware (such as sensors and cameras). Moreover, their acceptability is lower than knowledge-based systems [195].

One solution to secure system access is to create two-factor authentication by combining knowledge-based and biometric-based methods. Touch dynamics is a behavioral biometric that studies and analyzes how a person interacts with a touchscreen [217]. This approach takes advantage of the sensors already on the device and does not require specialized hardware.

Therefore, the idea inspected in this work is to integrate into a traditional unlocking method, such as entering a Personal Identification Number (PIN), the biometrics of touch dynamics to make access to the device more secure [218]. In this way, a potentially m

alicious user who somehow learns the PIN of a device, thanks to touch dynamics biometrics, cannot authenticate because the interaction with the device is different from that of the genuine user.

The approach used is static; it comes into play only at user authentication within the system and is not dynamic, which works throughout device use. The static approach is used in turn in verification mode, that is, to figure out whether the user attempting to access the device is who they say they are, and not in identification mode, that is, to determine which user is using the device. This work deals with so-called "Data Analysis." Countless examples have been taken in the behavioral field, i.e., analyzing human movement data to diagnose neurodegenerative diseases [219]. Time series analysis is carried out to predict traffic flow (traffic behavior) [220].

4.2.1 Related Works

Teh et al. [221] describe the design, implementation, and evaluation of an authentication system that combines a knowledge-based and a biometrics-based method of touch dynamics. In their experiment, raw, tactile data are acquired, which are then used to extract various features successively fed to one-class models such as OCKNN and *Support Vector Data Description (SVDD)* and binary models such as *K-Nearest Neighbors Algorithm (KNN) and Support Vector Machine (SVM)* to verify user identity. The evaluation results show that, based on the *Equal Error Rate (EER)* values obtained by integrating touch dynamics biometrics into the PIN-based authentication method, the levels of protection against representation attacks are significantly improved. Specifically, they claim that if a PIN is compromised, the success rate of a representation attempt is drastically reduced from 100 percent (if only a 4-digit PIN is used) to 9.9 percent (if the PIN is used in combination with touch dynamics biometrics).

In contrast to a strict touchscreen technology approach such as the one mentioned above, Vàzquez et al. [222] propose an authentication system by combining the typing of a four-digit PIN with the biometrics of touch dynamics, also making use of sensors such as accelerometers, gyroscopes, pressure, and touch size. Principal Component Analysis (PCA) was applied to consider only the most essential information and the MLP neural network as a classifier based on a database acquired in a controlled and uncontrolled environment. A 90% correct identification rate is reported. However, a sensor-based approach might be more energy-intensive for the device.

Similarly, Sen et al. [223] propose an authentication method based on a four-digit PIN and touch dynamics, considering the pressure exerted by the user on the screen and the duration of touches as

features. Several classifiers are used to acquire a sample of one hundred users, including Decision Trees, Naive Bayes, a K* classifier, and an MLP classifier. False Rejection Rate (FRR) is then analyzed to determine how often the system rejects the legitimate user. Their system records an FRR of 14%. This analysis is essential since the system should always recognize the legitimate user.

Trojahn [224] analyzes the performance of a touch dynamics-based authentication system by varying the temporal feature extraction. Specifically, different time intervals are tested based on which to extract and acquire features on a 17-digit PIN. For example, with an interval equal to 2, the differences between the features of one touch and those of the next are considered. In contrast, with an interval of 1, only the features of a single touch are analyzed independently. Using K-Means as a model and comparing different time intervals, they observe that a gap of 1 is already sufficient to return excellent performance in terms of False Acceptance Rate (FAR) (8.03%) and False Rejection Rate (FRR) (12.3%). However, with an interval of at least two, the best performance can be achieved, even though the number of features increases. Based on their studies, they consider increasing the time interval beyond what is proposed unnecessary.

Concerning the features that can be extracted from the raw data, Shen et al. [17] propose calculating statistics such as mean, median, minimum, maximum, and standard deviation to obtain valuable features for classification. They work on 48 subjects using different models, including one-class SVM, Neural Network, and KNN. With the one-class SVM classifier, they obtained a FAR of 5.01% and an FRR of 6.85% on a dataset obtained during their experiments.

4.2.2 Dataset

The first step was to process the dataset consisting only of raw features (The datasets generated and analyzed during the current study are available in the Teh et al. [221] repository, at https://goo.gl/sNACU8). The data belong to 150 subjects who participated in the experiment, and specifically, 1,500 samples and 16,500 tactile actions were acquired during a single session. Use the same smartphone device for their experiments to avoid differences between the sensors of different devices. The dataset initially consists of 150 files, one for each user, where ten samples related to the attempt to enter a prefixed PIN are collected for each.

Among the raw features collected are the PIN typed by the user, the time when the user touched the screen to type the number (Touch Action Press or TAP), and the time when the user lifted their finger from the screen (Touch Action Release or TAR), and the pressure exerted by the user on the screen. The TAP and TAR are expressed in nanoseconds using Android's nanoTime() function to associate timestamps. Instead, the pressure exerted by the user on the screen is obtained with Android's getSize() method, which calculates the screen area covered by the user's finger.

4.2.3 Experiments and Methods

The work can be described according to the different phases of raw data pre-processing (during which feature selection and normalization of the data were applied); selection, training, and testing of the machine learning models; and finally, the evaluation of results obtained using different metrics (Figure 109).

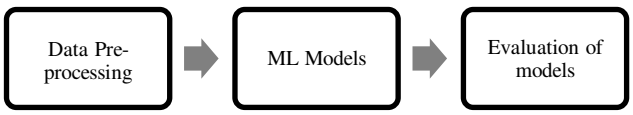


Figure 109 - Pipeline Experimental Design

The enrolment and verification phases are then distinguished. Raw data of a subject's Touch dynamics are captured in the recording stage, processed, and transformed into a model. In the verification phase, the touch dynamics data of a test subject (*i.e., an applicant*) are compared with the archived

authentication model to verify whether the applicant is indeed who they claim to be (*i.e., the mobile device owner*).

4.2.3.1 Data and Features Extraction

Raw data described have been used to extract features of two categories: first-order and second-order features. The former are features extracted directly from the raw data, such as screen time, flight time, and input time; the latter features are extracted from the former by calculating statistics.

4.2.3.1.1 First-order Features

Raw data from each sample have been directly used to extract first-order features. The following features have been evaluated:

- *PS (Pressure Size)* refers to the pressure the user exerts on the screen to type each PIN value.
- *IT (Input Time)* refers to the time the user takes to type the entire PIN and thus are given by the difference between the final and initial timestamps.
- *DT (Dwell Time)* refers to the dwell time on the screen, given by the difference between the *TAR* and *TAP* timestamps related to a single tap on the net.
- *FT (Flight Time)* is the time elapsed from when the user released the finger from the device (TAR) until the next *TAP*. This feature can be computed in several ways depending on the time interval. A time interval of 2 has been adopted here. With a time interval of 2, the timestamps of two consecutive touches are compared. With the time interval fixed, the FT feature is calculated in four ways: *FT1, FT2, FT3,* and *FT4*. Each performs the time-of-flight calculation with a different combination of *TAP* and *TAR*.

Table 46 reports details about features calculation. In Table 46, *r* and *p* denote the timestamp in nanoseconds of the *TAR* and *TAP*, respectively, *m* represents the PIN length, *n* indicates the chosen time interval, and in this case, *n=2*. In brief, the change in the time interval would lead to considering more TAP and TAR sequences in the feature calculation. For example, if n were equal to three, has been considered the difference between the timestamp of the TAP associated with the third PIN value and the timestamp of the TAR associated with the first PIN value, and no longer the timestamps of two consecutive numbers in the FT time-of-flight calculation.

Features	Descriptions	Equations
DT	The interval between the TAP and TAR of a key	$dt_i = r_i - p_i$
FT1	The interval between the TAR of a key and the TAP of the next key	$ft1_i = pi+(n-1) - r_i$
FT2	The interval between the TAR of a key and the TAR of the next key	$ft2_i = r_i+(n-1) - r_i$
FT3	The interval between the TAP of a key and the TAP of the next key	$ft3_i = p_i+(n-1) - p_i$
FT4	The interval between the TAP of a key and the TAR of the next key	$ft4_i = r_i+(n-1) - p_i$
IT	The interval between the TAP of the first key and the TAR of the last key	$it = r_m - p_1$

Table 46 - Feature

Finally, all features extracted from a PIN entry attempt were collected into a first-order feature vector. This process is repeated for each user and each PIN entry attempt. Thus, ten vectors containing 27 first-order features are obtained for each user.

4.2.3.1.2 Second-order Features

Second-order features are extracted from the first ones. These features consist of statistics such as minimum, maximum, mean, variance, standard deviation, and others [225]. Specifically, statistics are extracted for each type of first-order feature. For example, screen dwell time can be analyzed for each user interaction with the screen. Thus, with a PIN length of 4, it had four exchanges plus a fifth to confirm the PIN typed in. Statistics are calculated on the same set of these five interactions.

Below is a list of all 18 statistical metrics that have been calculated: *Minimum (mn), Maximum (mx), Arithmetic Mean (am), Quadratic Mean (qm), Harmonic Mean (hm), Geometric Mean (gm), Median (md), Variance (VR), Standard Deviation (sd), Skewness (sk), Kurtosis (ku), First quartile (fq), Third quartile (tq), Interquartile range (ir), Mean Absolute Deviation (ma), Median Absolute Deviation (mi), Coefficient of Variance (cv) and Standard Error of the Mean (se).*

Like the previously extracted features, a vector containing the list of associated statistics for each first-order feature is created. Thus, a total of 7×18 = 126 second-order features is obtained.

First-order and second-order features have been lumped within a single vector (i.e., a vector of 153 features that describe each sample).

4.2.3.2 Data Pre-processing and Models

Features have been normalized, and feature selection has been applied.

4.2.3.2.1 Normalization

The normalization process helps bring feature values back into a particular range. For this reason, the Min-Max Scaler algorithm was applied to get the values back into the range [0,1]. The formula underlying the algorithm is as follows:

$$scaled\_x = \frac{x - min(x)}{max(x) - min(x)} \times (new\_max - new\_min\ ) + new\_min \qquad (1)$$

In this case, *new_min* and *new_max* are 0 and 1, respectively. Applying this formula, all the data in the dataset are returned to the same range of values.

4.2.3.2.2 Feature Selection

The Feature Selection process is typically used to select the most relevant features. Often, not all features are useful in discriminating different samples. For the reasons above, an analysis of the values of each feature has been conducted, thus finding that nine features assumed constant values and, therefore, are useless for classification purposes. The removal of irrelevant features from the dataset has been performed according to *Teh et al.* [221]. Conversely, the feature selection algorithm has been the *minimum redundancy-maximum-relevance (mRMR)*[226]. The *mRMR* algorithm has been applied to the normalized dataset with a feature selection size of 20 to use the most relevant ones. However, experiments were also conducted by increasing the feature selection size. Still, it has been observed in the rank that the scores associated with the variables decrease dramatically after feature number 20. Therefore, it is useless to consider a higher number of features.

4.2.3.2.3 ML Training

This work has created a model for each user by dividing the dataset into legitimate and illegitimate examples, identified by a value of 1 and 0 of the target variables, respectively. In detail, one user has been considered a legitimate user, and all others have been considered illegitimate users. This has

been repeated for all users. This way, an attempt is made to train the models to correctly classify the genuine user and distinguish it from the malicious one.

4.2.3.2.4 Unbalanced Data

One problem encountered in the training phase, and often found during this type of study, is the unbalancing of the dataset. Legitimate user data are always significantly lower than those of malicious users. However, it must be argued that this problem needs to be addressed. In our case, ten samples for each user are processed against 1490 illegitimate examples, making the classifiers' task arduous. In such an unbalanced setting, it is very complicated for some classifiers to distinguish the legitimate user from the illegitimate one. Therefore, to cope with this problem, a bootstrap technique was applied to generate duplicates of the examples in the dataset to balance the data better. It is usually used in the examples of the minority class so that the algorithms take more account of it during the training phase. Several experiments have been conducted about the number of duplicates: empirically, it was found that ten bootstrap applications, taking the number of legitimate examples from 10 to 100, increased the classifiers' performance. Of course, this result is empirical and for further research. The whole process was critical not only to get good performance from the classifiers but also to try to get the *False Rejection Rate (FRR),* which refers to the number of times the system rejects the genuine user tends to zero. FRR is a critical parameter in system security, as a high rejection rate of a genuine user would impact the system's usability. However, the problem of data imbalance is mainly relevant when binary classifiers are considered. With one-class classifiers, the model is trained only on the examples of the legitimate user.

4.2.3.2.5 ML Models

A one-class class classifier is trained only on the legitimate user's data and, based on them, tries to classify with an Anomaly Detection approach all users who deviate from a normality profile given by the legitimate user's examples. On the other hand, two-class class classifiers consider both legitimate and illegitimate examples. However, this second approach does not reflect reality since one needs to possess the data of potential malicious users. ML models include the K-Nearest Neighbors Algorithm (KNN), Support Vector Machine (SVM), Random Forest model, and Logistic Regression as binary models. The Support Vector Data Description (SVDD) model was chosen as a one-class and the *One-Class SVM model*. Each model was initialized differently, optimizing its parameters.

The K-Nearest Neighbors Algorithm (KNN) model was the one for which *k=3*. For the SVM-based model, tuning operations were conducted to optimize its parameters. An RBF (Radial Basis Function) kernel was chosen. Then, the algorithm-modified class weights assigned by default were modified, setting a weight of 7 for the legitimate user class and 4 for the illegitimate user class. This combination of weights obtained the best performance against the various experiments. For the Random Forest model, it was sufficient to set gini as the criterion for evaluating the quality of the splits and to set a number on the random state parameter (42) to avoid getting different results at each run.

The binary model, which also required the most attention on parameter optimization, was Logistic Regression. This model was parameterized with a liblinear solver, random state fixed as the Random Forest, a penalty in case of "l1" errors, the maximum number of iterations set to 200 because of problems in convergence, and regularization factor *C=100*. Finally, in this case, the weight of the classes was changed, finding the best combination of weights 93 for the legitimate user class and 12 for the illegitimate user class.

Turning to the single-class models, the One-Class SVM was set as a kernel *RBF (Radial Basis Function)*, *nu=0.5*, and *gamma=" scale"* as parameters to achieve the best performance.

Finally, the Support Vector Data Description (SVDD) algorithm was implemented with an *RBF (Radial Basis Function)* kernel and *gamma="scale"* parameter.

4.2.3.2.6 Evaluation of models

The evaluation procedure has been based on a *K-Fold cross-validation*. This technique splits the dataset into K subsets of equal size to train the model on the K-1 fold in rotation and test it on the remaining fold. According to *Teh et al.,* it is convenient to compute the different folds directly on the legitimate and illegitimate examples rather than calculating them on the whole dataset. In this work, five folds were chosen so that the model is trained on the concatenation of the k-1 folds of the legitimate user and the illegitimate user and is tested on the concatenation of the remaining folds. The only exception is the one-class models trained only on the k-1 folds of the legitimate user. The model is evaluated by averaging the results obtained over the number of folds k, improving the individual user's performance. The entire model's performance results for each user are averaged over the total number of users who participated in the experiment, i.e., 150.

4.2.3.2.7 Evaluation metrics

Several metrics have been able to evaluate the overall system's performance profitably. The Equal Error Rate (EER) has been calculated as the average between the values of *FAR (False Acceptance Rate)* and *FRR (False Rejection Rate)*. The FAR is calculated as the ratio of false positives (illegitimate users who are wrongly authenticated by the system) to the number of examples of the unlawful user used in the test; on the other hand, the FRR is calculated as the ratio of false negatives (genuine users classified as malicious by the system) to the number of examples of the actual user used in the test. *Equal Error Rate (EER)* is a general indicator of system performance and the best threshold for accepting a user into the system. Since it is an error rate, one aims to obtain the lowest possible values. In addition, metrics such as overall system accuracy, precision, recall, and f score were also computed, which is helpful in unbalanced contexts to calculate the true accuracy of the system on the different classes.

4.2.4 Results

Results obtained are reported in this section concerning the models and approaches developed. More specifically, results are presented when the classification problem has been considered as a binary or as a one-class problem. A direct comparison is finally provided.

4.2.4.1 Binary Classifiers

K-Nearest Neighbors Algorithm (KNN) and *Support Vector Machine (SVM)* obtained similar accuracy results, respectively, of 98.86% and 98.81%. Regarding EER, these two models achieved 0.6% and 1.72%; however, the lowest EER has been achieved using RF: 0.01%. It can be observed that the proposed and evaluated approaches consistently outperform state of the art on this dataset (Table 47).

	Accuracy	**Precision**	**Recall**	**F-Score**	**EER**	**EER_ROC**
KNN	98,86%	85,54%	100%	91,97%	**0,60%**	0,60%
SVM	98,81%	86,56%	97,65%	91,41%	**1,72%**	1,72%
RF	99,97%	99,62%	100%	99,80%	0,01%	0,01%
LR	96,47%	70,11%	98,49%	80,56%	2,58%	2,58%
***Teh et al.* [221]**	-	-	-	-	**8.7% (SVM)** **9.4% (KNN)**	

Table 47 - Results binary classification of the study carried out in this paper

It can be seen from Table 47 that although all models recorded satisfactory accuracy values, it is also helpful to consult the F-score to understand how the examples were ranked. The Logistic Regression model is the worst-performing model, not only from the standpoint of F Score but also on all other metrics. In contrast, the *K-Nearest Neighbors Algorithm (KNN)* and *Support Vector Machine (SVM)* perform similarly from the perspective of F-Score and EER. However, the best model proves to be Random Forest, which achieves almost the highest accuracy, precision, and recall values, but also on F Score registers a remarkable result of 99.80%. Even on EER values, it proves to be the one with the lowest error rate.

Looking at the results overall, it is interesting that the EER was calculated using the approach proposed by *Teh et al.*[221] results identical to those calculated with the *Receiver Operating Characteristic (ROC)* curve, probably because the *Receiver Operating Characteristic (ROC)* curve measures the system's performance precisely based on how the test examples are ranked in terms of false positives, false negatives, and true positives.

4.2.4.2 One-class Classifiers

Results are reported in Table 48.

	Accuracy	**Precision**	**Recall**	**F Score**	**EER**	**EER_ROC**
SVDD	93,09%	25,81%	64,78%	30,80%	20,87%	20,87%
OC_SVM	95,99%	34,38%	49,35%	34,27%	27,00%	27,00%

Table 48 - One-Class Results

As can be seen, the performance of the single-class models is relatively low. In addition, it can be determined based on EER that with the data used, a one-class approach based on Support Vector Data Description (SVDD) is the best compromise regarding the usability and security of an authentication system with touch dynamics. However, more efforts are required.

4.2.4.3 Comparison of binary and one-class approaches

Looking at the results commented on and shown above, it undoubtedly emerges that binary approaches are superior to single-class methods in every respect, both in terms of the usability of the system in ensuring that the legitimate user is correctly authenticated almost all the time, and also in terms of security because at the same time, any users trying to authenticate as genuine users fail with an average probability very close to 100 percent.

This result is due to the experiment setting, which allows binary classifiers to have information about illegitimate users, helping them classify. However, this is only sometimes the case, which is why one-class classifiers are widely used in this context, but based on the experiments conducted and the data used, such approaches still show some margin of error.

This work analyzed a PIN-based authentication method combined with touch dynamics biometrics. The study recorded excellent results, especially concerning the binary classification approaches that highly accurately classified legitimate users from illegitimate ones. The best result was obtained with the Random Forest model, which achieved an accuracy of 99.97% and an Equal Error Rate (EER) of 0.01%.

It might be interesting to study other single-class approaches in future developments, as they are more useful in real-world contexts since one needs to possess the data of potential illegitimate users with whom to train the binary classifiers. Or calculate innovative feature extraction functions based on residual number system data [227]. Achieving good performance with single-class models could be very important from the perspective of mobile device security. In the future, a security system could be implemented by encrypting with innovative algorithms [188] and using innovative models [211], [212].

4.3 Human Activity Recognition using Smartphone Sensors: Focusing on Fall Detection with the UNIBA HAR Dataset

Human Activity Recognition (HAR) identifies many techniques for recognizing daily life activities. This discipline has taken significant steps forward in recent years and has used several data sources, including smartphone sensors. Smartphones can be considered a simple but effective strategy due to

their popularity across different generations and people. Some of the activities usually analyzed in HAR can be very relevant because they can decree bullying situations. This paper introduces a new dataset including 19 users performing three "*neutral*" activities (Walking, Jumping, and Sitting) and five activities "at risk" for bullying detection (Falling Forward, Falling Backward, Falling Left, Falling Right, and Running). Moreover, the differentiation of types of falls has also been studied. Finally, highly effective Machine Learning Models have been considered for comparison purposes. These models have been trained and tested on accelerometer data collected through the smartphone.

The smartphone is a valuable source of data because of its built-in sensors, which are used for human activity recognition (HAR) [176], [228]. The HAR field has made significant progress in recent years, but only a few studies use smartphone sensors to recognize fall activities. The present study is particularly relevant because it focuses on human *activities at risk*, such as falling or running, and *neutral activities* characterized by lower risk [171].

The distinction between *neutral* and *risk* activities is justified by analyzing the risk of injury and the complexity associated with the activities [229]. *Neutral activities* tend to be less stressful on the body and have a significantly lower risk of causing injury. In contrast, risk activities involve more complex movements and a more significant potential for physical injury. This distinction helps plan exercise programs, injury prevention, and safety management in various settings, such as physical training and rehabilitation. Specifically, three neutral activities were identified: walking, jumping, and sitting, and five risk activities: falling forward, falling backward, falling left, falling right, *and running* (Table 49).

Classes	Activities	Impact	Risk of Accident
Neutral Activities	Jumping	Moderate, with a relatively low risk of injury if done in a controlled manner	Low under normal conditions, primarily if the activity is controlled and without obstacles or uneven surfaces.
	Walking	Low is considered one of the safest and most natural physical activities	Very low without obstacles, adverse environmental conditions, or pre-existing health issues.
	Sitting	Minimal, generally involving no movement that can cause injury	Very low, except for special conditions such as a sudden fall while moving to sit down.
Activities at Risk	Running	High, with increased stress on joints, especially knees and ankles	High compared with walking due to increased speed, impact force, and possibility of losing balance
	Fall [230]	Extreme, with a high risk of serious injury such as fractures, bruises, and trauma.	Very high, as a fall can cause significant damage, especially in older individuals or those with pre-existing health conditions.

Table 49 - Explanation of activity classification

Activities at risk can also be considered activities of alleged physical bullying since most bullying actions involve significant changes in the physical state of the person being bullied, resulting in sudden movements and accelerations, such as falling after being pushed or running to escape. These activities have been incorporated into the new UNIBA HARDataset [5], [90]. The *UNIBA HAR Dataset* contains triaxial accelerometer data related to the activities mentioned in Table 49, plus three types of falls: *Backward, Forward,* and *Lateral*. The activities were performed by 19 users, a sample number in line with similar datasets studied at the state of the art.

What has been said so far covers the first experiment. The second experimentation deals exclusively with fall activity, divided into three forms: *Backward Fall, Forward Fall,* and *Lateral Fall*. In the second trial, the best model for fall-type recognition, a crucial aspect of *Human Activity Recognition (HAR)*, was also identified. The ability to distinguish between different types of falls can improve the accuracy of monitoring systems and provide vital information for timely and personalized interventions in health care. Despite the importance of this topic, it is surprising to note that few datasets and scientific articles have fully explored the *"fall type."* Most research focuses on simply detecting falling without differentiating the different modes. This gap represents a significant opportunity for improvement in the science and technology community.

Available datasets are often not detailed enough to distinguish between forward, backward, lateral, or other variant falls. In addition, scientific articles dealing with fall-type recognition are relatively scarce and do not always provide a complete picture of the methodologies used or the results obtained. To address these challenges, new datasets need to be developed that include a variety of fall scenarios and are annotated as accurately as the UNIBA HAR Dataset created. At the same time, there is a need to stimulate research toward creating machine learning models capable of processing this information effectively. Advanced models such as deep neural networks and supervised and unsupervised learning techniques could offer promising solutions.

The Sub-chapter 4.3.1 State of the Art, offers a comprehensive state-of-the-art review, highlighting recent advancements and critical research in the field.
The Sub-chapter 4.3.2 Design, details the design of the proposed system, describing the architecture and methodology. The Sub-chapter 4.3.3 Experimental Design, elaborates on the experimental design and is divided into two main experiments. The first experiment includes three sub-experiments: raw data with five classes, raw data with the sitting action removed, and raw data with an increased sliding window for five classes. The second experiment consists of two sub-experiments: a fall detection experiment with three classes using raw data and another using a leave-one-out approach.

4.3.1 State of the Art

Several papers were reviewed, as shown below. A first inspection has been related to the action considered. Table 50 shows the complete list of activities in the considered papers.

Activities	Paper Referenced
Walking	[231], [232], [137], [233], [141], [142], [133], [57], [234], [143], [235], [145], [144], [236], [237], [83], [139], [87], [238], [239], [240], [241], [242], [243], [49], [244], [245], [246], [27], [146], [247], [248], [249], [250], [132], [251], [252], [33], [253], [254], [140], [67], [255], [101], [138], [73], [256], [134], [41], [233], [257]
Downstairs	[232], [137], [133], [234], [143], [235], [236], [237], [83], [139], [87], [238], [239], [241], [242], [49], [244], [246], [27], [146], [247], [248], [249], [250], [132], [251], [33], [253], [254], [140], [138], [73], [256], [134], [41], [233], [257]
Upstairs	[232], [137], [133], [234], [143], [235], [236], [237], [83], [139], [87], [238], [239], [241], [242], [49], [244], [246], [27], [146], [247], [248], [249], [250], [132], [251], [33], [253], [254], [140], [138], [73], [256], [134], [41], [233], [257]
Standing	[231], [232], [137], [142], [133], [234], [143], [144], [237], [83], [139], [87], [239], [241], [242], [243], [244], [146], [248], [249], [250], [132], [251], [253], [140], [255], [138], [73], [256], [257]
Sitting	[231], [232], [137], [9], [234], [237], [83], [139], [87], [238], [239], [242], [243], [244], [246], [146], [248], [249], [250], [132], [251], [252], [253], [140], [67], [138], [73], [256], [41]
Running	[231], [137], [233], [133], [57], [143], [235], [145], [144], [237], [83], [87], [240], [244], [246], [146], [247], [249], [250], [33], [253], [140], [41], [233]
Lie down	[231], [232], [133], [237], [139], [87], [238], [239], [241], [242], [246], [146], [248], [132], [251], [252], [253], [140], [73]
Falling	[141], [234], [143], [235], [145], [144], [146], [247], [250], [252], [256], [134], [41], [233]
Jogging	[141], [139], [87], [239], [49], [246], [252], [138], [73], [256], [134], [257]
Jumping	[142], [143], [145], [253], [256], [134], [41], [233]
Standing still	[233], [57], [235], [246], [252], [254], [140]
Falling to the left	[133], [146], [247], [256], [134], [41]
Falling in front	[133], [146], [247], [256], [134], [41]
Push	[141], [142], [234], [143], [144]
Hit	[141], [142], [234], [143], [144]
Falling back	[133], [146], [247], [134], [41]
Fall to the right	[133], [146], [247], [256], [41]
Transition sitting – standing.	[243], [247], [248], [134], [41]
Standing-sitting transition	[243], [247], [248], [134], [41]
Riding a bike	[246], [249], [252], [253], [255]
Transition lying down - standing	[27], [247], [248], [134], [41]
Transition standing - lying down.	[27], [247], [248], [134], [41]
Falling with protective strategies	[41]
Falling back from the chair	[41]

Table 50 - List of activities related to the work

It can be stated that there are often no actions of interest for Risk activity. The most recognized activities are overwhelmingly Neutral activities (Figure 110 *(left)*). However, studies have also focused on risk activity (Figure 110 *(right)*). This image shows that the most significant actions for recognition of bullying activities are *Falling* and *Running*. These are immediately followed by actions such as *Hitting, Pushing,* and *Pushing to the Ground*. *Falling* and *running* can be direct consequences of these previous actions. Based on these considerations, *Running* and *Falling* are considered in this study because they are the most used (Figure 110).

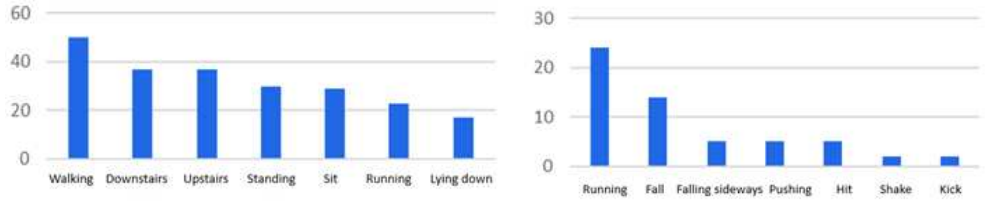


Figure 110 - (left) The most recognized activities in the studies. (right) “at risk” Activities

The search for suitable datasets started from HAR in general. There are a variety of public datasets concerning Neutral activities. Some examples are *HAR using smartphones.* [101], *WISDM* [47], and *Actitracker* [73]. These datasets contain simple, everyday life actions such as walking, running, sitting, going down, and upstairs. Usually, the datasets in this field consist of a small number of activities, typically around 5 or 6 actions per dataset.

When examining datasets specific to the field of Bullying Detection or Physical Violence Detection, it becomes evident that these datasets are often created through a roleplaying phase [145]. A group of users is assigned two roles, one as a victim and one as a bully/offender. Once they have found a way to record their data, a series of everyday life actions are performed and then classified as opposed to bullying or violent actions. To record data from these actions, users are attached to inertial measurement units at specific strategic locations to collect this data or are given cell phones to put in their pockets. The authors have not released this type of dataset to the public. Nonetheless, some datasets have activities previously mentioned as being related to bullying. Many of these, however, use inertial units instead of smartphones.
This study has needed a phase of searching datasets and skimming publicly available ones. The conditions were to have the activities described above as possibly resulting from bullying, i.e., *falling* and *running*, and to record values via smartphone. Next, as shown in Table 51Table 51 - Datasets obtained through smartphones that have actions and are obtained from smartphones., the publicly available datasets studied have these characteristics.

Name	Details	Users	Conditions
MobiFall [256]	87Hz. Samsung S3	24, 7 females, 17 males 22 to 47 years old, 160 to 189 cm tall 50 to 103 kg.	Smartphones can be placed in a pants pocket in any position. For falls pocket opposite side of fall. These took place on a mattress.
UMAFall [134]	200Hz Samsung S5 LG G4.	19, 8 females 11 males 18 to 67 years old 155 to 195 cm tall 50 to 93 kg.	Domestic environment falls made on a mattress. Phone stowed in front pockets of pants.
UniMiB SHAR [41]	50Hz Samples of 51 or 151 values. Samsung Galaxy Nexus I9250	30, 24 females, 6 males 18 to 60 years old 160 to 190 cm tall 58 to 82 kg.	Smartphones are placed in pants pockets, half samples on the left and half on the right.

Table 51 - Datasets obtained through smartphones that have actions and are obtained from smartphones.

4.3.1.1 Related Studies

This chapter illustrates an in-depth and general study of HAR systems. Before conducting a study, the *"Research Question, Dataset, Human Actions, Pre-processing, Feature Extraction, Classification Models, and Evaluation Metrics"* are analyzed.

The research question addressed in the selected studies deals with recognizing and classifying human activities from sensors. In fact, in the literature, sensor approaches have addressed smartphone and smartwatch devices [137] [139]. These works aim to detect suitable activity through data analysis performed by machine learning and deep learning models. They tried to discriminate even very similar human actions. Modern Deep Learning models can discriminate similar actions in the right way. The research question leads to the subsequent identification of Datasets. Cho et al. [133] Uses several Datasets that can be easily found on the web, namely, UMAFall [50], UniMiB SHAR [51], and SisFall [52]. These Datasets deal with fall activities using sensors like accelerometers, gyroscopes, etc. In addition, Gupta et al. [137] focuses their research on the Dataset called WISDM [138], which includes various activities and is also publicly available on the Internet. Finally, Ismail et al. used a smartwatch dataset consisting of 12 daily life actions [139]. Proposes a classification with a dataset containing five daily life classes. In general, the most recognized state-of-the-art activities are *Walking, Standing, Jogging, Sitting, Driving, Running, Writing, Typing, Brushing Teeth, Clapping Hands, Folding Clothes, Playing With A Tennis Ball, Eating, Drinking, and More* [137] [139]. These activities are considered in the studies analyzed for this paper.

It is helpful to carry out the Pre-processing phase after acquiring the dataset. The sliding window mechanism for data extraction is commonly used in the analyzed studies. In the study by Cho et al., the window size generally varies from one second to nine seconds. [133]. The overlap, when implemented, ranges from 33 %to 60 percent [133] [5]. Cho et al. [133] considered features such as signal vector magnitude, singular value decomposition, kernel principal component analysis, and sparse principal component analysis in the selected studies [133]. The mentioned studies mainly use raw data. Regarding classification models, neural networks have been used mainly in many works. In the work of Gupta et al. [137], *Convolutional Neural Networks (CNN),* and *Deep Convolutional Long Short-Term Memory (DeepConv LSTM)* [137] are used. Cho et al. [9] also use CNN networks in their research. Gupta et al. [5] use CNN and *DeepConv LSTM*.

Some use Shallow Learning Approaches, for example, in the work of Concone et al. [136] K-Nearest Neighbor models, comparing the results with the MosT framework and Google API [136]. Lee et al. [193] use Random Forest [193]. Ismail et al. [139] uses Adam and RMSprop [139]. The evaluation results of the models have been analyzed. In the study by Concone et al. [53], satisfactory results are obtained with 95.43% accuracy with the KNN model [136]. In work done by Gupta et al. [137] with CNN, an accuracy of 96.54% is obtained, and with DeepConv LSTM, an accuracy of 87-88% (smartwatch) [137]. Lee et al. [193] use a CNN and achieve 92.71% accuracy, exceeding Random Forest by 3% [193]. Ismail et al. [19], with RMSprop, an accuracy of 95.83% is obtained [139]. The basis of this analysis has been circumscribed to select datasets and classification models to carry out our bullying experiment.

4.3.2 Design

The structure of the work pipeline was adopted for the dataset analysis and testing of the study. This chapter opens with a presentation of the datasets used, including a new ad hoc created dataset named "UNIBA HAR Dataset" and two public datasets: UMAFall and UniMiB SHAR.

4.3.2.1 Datasets

A new ad hoc dataset was created: "*UNIBA HAR Dataset."* Moreover, two publicly available datasets were picked: UMAFall and UniMiB SHAR. Although the data sampling is radically different, with

200 values every second for 15 seconds for UMAFall and 151-value windows for UniMiB SHAR, these two datasets were chosen because they both provide truth labels for each task, something that the excluded one, MobiFall, did not guarantee with its *"Fall - Not Fall"* distinction.

First, the newly created dataset, the UNIBA HAR Dataset, is presented; second, the selected public datasets, UMAFall and UniMiB SHAR, are illustrated.

UNIBA HAR Dataset

A *recruitment protocol* was developed to ensure the representativeness and quality of the data collected to realize the UNIBA HAR dataset. Nineteen participants were selected, including 13 males and 6 females, ranging in age from 18 to 59 years and in height from 160 cm to 190 cm. The selection of such a diverse sample was critical to ensure that the data collected represented a wide range of physical and demographic characteristics.

Recruitment was done through personal contacts, inviting candidates to participate in the study voluntarily. The experiment occurred in a laboratory designed to create a controlled and safe environment. Each participant used a smartphone with a specific application for sensor data collection, set at a sampling rate of 200Hz. Users were asked to place the smartphone in their fitting pants pocket, with the screen facing their body, to equalize data collection.

During the experiment, participants performed eight specific actions: *walking, running, jumping, sitting,* and *falling forward, backward, right,* and *left*. Each action was repeated two or three times, with a duration of 15 seconds for each execution, to ensure sufficient data for each activity. For the falling actions, a mat was used to allow participants to perform the falls safely and without fear, thus ensuring movements were as natural as possible.

The data collected by the accelerometer were initially saved in .txt format and sent to a server. Subsequently, this raw data was reprocessed to clean and convert it into .csv format, which is more practical for analysis. In the .csv file, the first column contains the user ID, the second the activity performed, the third the timestamp in milliseconds from the start of the action, and the following three columns contain the triaxial accelerometer values (x, y, z). This recruitment protocol and detailed description of the experiment were designed to ensure the collection of accurate, representative, and valuable data for analyzing human movement patterns.

Including participants of different ages and heights and conducting the experiment in a controlled and safe environment helps create a robust and reliable dataset, which is essential for studying movement and fall phenomena Table 52.

Id	Gender	Age	Height (cm)
1	Male	23	180
2	Male	24	182
3	Male	24	177
4	Female	23	160
5	Male	23	178
6	Male	19	170
7	Male	32	182
8	Female	25	165
9	Female	24	173
10	Male	59	167
11	Male	27	171
12	Female	58	172
13	Female	23	170
14	Female	35	171
15	Male	22	175
16	Male	18	169
17	Male	40	190

18	Male	24	183
19	Male	25	184

Table 52 - Master of subjects participating in the construction of the dataset

UMAFall

The activities performed by the 19 users are as follows: *Fall(Forward, Backward, Lateral), Walking, running, squats, jumping, going up and down stairs, lying down and getting out of bed, sitting, and getting up from a chair.*

This dataset was recorded with a sampling rate of 200Hz. The sensors used are accelerometer, gyroscope, and magnetometer. All movements were recorded for 15 seconds. It comprises 747 comma-separated value files with names indicating subject ID, movement type (*FALL* or *not*), activity performed, experiment number, and date. Within each .csv file, in addition to the data, there is the time in milliseconds from the start of the action, sample number, three real numbers indicating sensor values (x, y, z), and an integer specifying which sensor that measurement came from (Accelerometer is 0, Gyroscope 1, and Magnetometer 2). The sensor sorts the values. Each action was repeated by the subjects up to eighteen times. However, this leads to a disproportion per user, as some users did not experiment with each type of action (for example, the user with ID 3 did not perform even one type of fall, as observed from the .csv files). Another example of the imbalance can be observed by viewing the tuples derived only from the accelerometer of the used smartphones collected per user. The total is 2,234,277 rows; the user with ID 18 is responsible for almost a quarter. UMAFall, in the supplementary material, provides videos of how the actions took place.

UniMiB SHAR

A dataset consisting only of values obtained from the accelerometer. Recognized activities: *Falling (Falling forward, backward, right, left, hitting an obstacle, with protective strategies, without protective strategies, Syncope), Walking, running, climbing stairs, descending stairs, jumping, lying down from standing, sitting.*

The dataset includes 11,771 human activities and falls performed by 30 subjects aged 18 to 60. For each activity, there are 2 to 6 experiments for each user. For the actions with two experiments, the smartphone is placed in the right pocket in the first experiment and the left pocket in the second. For actions with 6 experiments, the first three have the smartphone in the right pocket and the others in the left pocket. Data is provided in windows of 51 or 151 samples around an original signal peak higher than 1.5g, with g being the acceleration of gravity. The best experimentation results performed from this dataset are obtained with a K-NN in the ADL-only category. The code has various data divisions: total, ADL-only, fall-only, or ADL vs. Fall. The sampling rate used is 200Hz.

4.3.2.2 Pre-Processing

UMAFall and UniMiB SHAR datasets were ported to the same form as the UNIBA HAR Dataset, selecting only the values from the smartphone accelerometer for UMAFall. The datasets had discordance of signs on the y-axis, probably due to how the cell phone was placed in the pocket during the construction of the dataset itself. Therefore, the marks in this column were equalized where appropriate. In addition, accelerometer data was brought into the same scale of values through a division. The data was then normalized with Robust Scaler, and a sliding window with overlap was applied to all datasets.

4.3.2.3 Feature Extraction and Selection

Raw data without shallow feature computation was used as the feature. Then, in a second experiment, the Signal Vector Magnitude feature – explained and mentioned earlier – was introduced to see how it would affect the performance.

4.3.2.4 ML Models

This study uses four different classifiers. These classifiers were chosen because they stood out as being state-of-the-art in the HAR world. Two are Shallow Learning algorithms (*Support Vector Machine, K-nearest neighbors*), and two are Deep Learning (*Convolutional Neural Network (CNN), Bi-directional Long Short-Term Memory (Bi-LSTM)*).

The *CNN* or *ConvNet* is an artificial neural network that flows forward. It is composed of layers that contain neurons. There must be at least three layers in this type of neural network. The first layer is the input layer; then, there are one or more hidden layers that perform calculations through activation functions, and finally, an output layer that deals with classification. Among the hidden layers are the convolution layers, which differentiate the convolutional neural network from a normal multilayer perceptron. These oversee the use of filters and the extraction of the features of the data to be classified. This process of receiving input from the neurons in the previous layer and giving it to the neurons in the next layer is done through a convolutional operation. Figure 111 shows the convolutional neural network architecture used for the ADL vs. No-ADL experiments.

Figure 111 - CNN architecture used in this study

In contrast, LSTM deals with networks that include neurons connected to form loops. This allows LSTMs to process single data points and entire sequences of data. An LSTM also has a set of cyclically connected blocks, known as memory blocks, that allow context to be stored so that dependencies between temporally distant data points are also captured. This work used a Bi-LSTM that processes data in both directions with two hidden layers flowing into the same output layer.

Figure 112 shows the architecture of the Bi-LSTM classifier used in this study.

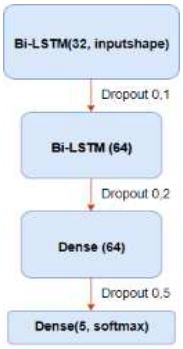


Figure 112 - The architecture of the BI-LSTM used in this study.

4.3.3 Experimental Design

The experiments were designed to correctly identify and classify different activities a person may perform, distinguishing between everyday actions and those potentially indicative of dangerous situations.

The **first experiment** assesses the discernment ability between activities: *walking, jumping, sitting, running, and falling*. In this experiment, the three different types of falls were unified into one class named "*Falling*." The testing strategy is 70/30 for the same datasets and cross-dataset for different datasets in Train and Test. The 70/30 split (70% of the data for training and 30% for testing) is a standard practice for evaluating model performance. This split allows enough data for training the model and, simultaneously, a significant portion to test its ability to generalize to data not seen during training *(Tests T1, T2, T3, Table 5).* A cross-dataset testing technique was performed for Tests T4, T5, T6, T7, and T8 (Table 5). This technique is essential to assess the model's generalization ability. This type of test checks whether the model can recognize activities and potential bullying incidents in varied contexts that it never saw during training. A model performing well on cross-dataset tests demonstrates significant robustness, indicating that its discernment capabilities are not limited to a particular data type or context. This is crucial for practical applications where the actual data may vary considerably.

For the **first experiment** only, several sub-tests were conducted, which are named after the sub-chapters of the **first experiment**:

1. **Raw Data**: This phase is critical for understanding the initial nature of the data collected and identifying any underlying issues;
2. **Raw Data and Sitting Action Removed**: The falling action can easily be confused with the sitting action. Therefore, at this stage, remove the sitting action from the raw data to observe how this affects the clarity of the fall data;
3. **Raw Data and Increased Sliding Window**: In this phase, examine whether increasing the sliding time window can improve the disambiguation between the two activities examined without removing the sitting action from the data. This approach involves using a more significant time window for data analysis to capture more contextual information. The goal is to improve the robustness and accuracy of the data-derived models, thereby increasing the ability to clearly distinguish between the falling and sitting actions (four-second sliding window with 75 %overlap).

On the other hand, the **second experiment** focuses exclusively on distinguishing between the different types of falls, trying to understand whether the fall occurred forward, backward, or sideways; therefore, only classes related to the three types of falls were considered. This approach allows a more detailed analysis of the dynamics of falls, providing helpful information to correctly identify the type of accident and the circumstances that may have caused it. The sub-tests identified renaming the sub-chapters of the second experiment are:

1. **Raw Data**: This phase is critical to understand the initial nature of the data regarding the falls and identify any problems;
2. **Leave One Out**: For the second experiment only, in addition to the test type (Tests T1, T2, T3, T4, T5, T6, T7, T8, Table 53), another type, namely Leave One Out (LOO) per user was considered. The choice of LOO for this experiment allows us to maximize the use of available data, provide a robust and detailed evaluation of model performance, and obtain specific and helpful information for fall prevention, considering the individual characteristics of each user. Each user may have unique characteristics in how they fall, depending on factors such as age, physical condition, and other personal variables. Using LOO, one can evaluate how the model performs for each user, identifying any specific issues and improving the accuracy of fall classification for

different people. A second aim is to implement the following model on a smartphone device with real-world activity.

Experiment	Dataset Training	Dataset Testing
T1	UMAFall	UMAFall
T2	UniMiB SHAR	UniMiB SHAR
T3	UNIBA HAR Dataset	UNIBA HAR Dataset
T4	UMAFall	UniMiB SHAR
T5	UniMiB SHAR	UMAFall
T6	UMAFall	UNIBA HAR Dataset
T7	UniMiB SHAR	UNIBA HAR Dataset
T8	UMAFall + UniMiB SHAR	UNIBA HAR Dataset

Table 53 - The eight sub-experiments for each experiment category

Eight sub-experiments (Table 53) were conducted based on the datasets used for these two experiments. These sub-experiments aimed to test different configurations and improve the accuracy of activity and fall recognition, thus helping to develop more effective systems for preventing and identifying potentially dangerous situations. In this phase, the experiments were conducted with the four chosen classifiers: K-NN, SVM, CNN, and Bi-LSTM.

4.3.3.1 First Experiment

Experiment with 5 classes - Raw Data

Table 6 shows data from the first experiment. Analyzing the data presented can make some general remarks about the machine learning models used and their performance across different experiments (T1-T8).

Starting with CNN, note that it generally performs well, especially for the T2 test, with an accuracy of 93.39%. Accuracy, recall, and F1-score are also high for T2, 85.70%, 89.18%, and 87.07%, respectively. This suggests that CNN is particularly effective when the training and testing data come from the same dataset, UniMiB SHAR. However, CNN performs less when the model is trained and tested on different datasets, as evidenced by the lower results for T4, T5, and T6. This indicates a potential generalization problem. The T3 test is in line with the T2 test.

The KNN shows variable performance, with generally lower results than the CNN. The best performance is observed in T3, with an accuracy of 83.24%, but in other tests, such as T5, accuracy drops to 27.07%. The other metrics also follow a similar trend, suggesting that KNN may not be the most suitable model for scenarios where training and testing data differ. The low precision and recall in T4 and T5 indicate a poor ability to distinguish classes in these cases correctly.

The BI-LSTM model shows an exciting performance, especially in T2, with an accuracy of 92.46%, comparable to CNN. The other metrics for T2 are also high, with accuracy at 74.15%, recall at 73.66% and F1-score at 73.80%. However, as with CNN, performance decreases significantly when the training and testing data differ, as in T4, T5, and T8. This model might have advantages in contexts where capturing temporal relationships in the data is essential, but like the others, it suffers from generalization problems. Finally, SVM is the model that shows the worst performance of all. The highest accuracy is observed in T7, with 50.22%, but in general, the metrics are significantly lower than those of the other models in almost all tests. This suggests that SVM may not be suitable for this problem or that significant parameter optimization is needed to improve performance.

Model	Metric	Same dataset			Cross-dataset				
		T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**

CNN	Accuracy	73.02	**93.39**	**85.54**	66.44	57.27	**77.93**	54.34	**73.10**
	Precisione	63.97	**85.70**	**87.63**	47.78	37.76	**79.88**	44.75	**63.10**
	Recall	58.01	**89.18**	**80.19**	40.59	40.96	**65.69**	49.78	**67.16**
	F1-score	59.91	**87.07**	**83.24**	41.36	33.64	**63.33**	42.07	**63.20**
KNN	Accuracy	61.44	61.39	**83.24**	53.39	27.07	69.18	47.21	37.09
	Precisione	50.75	57.60	**87.05**	29.35	26.93	66.54	32.63	52.53
	Recall	43.94	54.48	**75.67**	34.30	26.60	57.96	38.96	33.18
	F1-score	44.21	46.24	**79.52**	29.88	19.57	57.06	29.24	24.04
BI-LSTM	Accuracy	66.70	**92.46**	**83.04**	54.43	64.32	71.80	69.11	50.24
	Precisione	56.20	**74.15**	**71.75**	22.78	36.92	67.09	64.63	50.24
	Recall	52.00	**73.66**	**73.28**	30.56	44.86	62.31	53.85	50.68
	F1-score	49.80	**73.80**	**72.42**	25.24	37.24	63.91	58.76	47.15
SVM	Accuracy	60.31	36.12	53.03	30.77	29.64	55.95	50.22	29.92
	Precisione	36.35	24.51	21.65	26.24	19.49	39.04	25.73	17.13
	Recall	25.76	23.38	28.31	25.70	17.28	36.63	29.85	16.56
	F1-score	22.39	23.18	22.73	22.34	14.29	34.43	22.64	13.56

Table 54 - Experimentation with raw accelerometer data

Experiment with 5 classes - Raw Data and Sitting Action Removed

Comparing the performance of the models in Table 7 with those in Table 54, observe some significant differences and similarities that deserve in-depth analysis.

In the case of CNN, a general improvement in performance was achieved. Accuracy, for example, has increased in all tests, with T2 showing an increase from 93.39% to 96.03% and improvement in other metrics such as accuracy, recall, and F1-score. This suggests that changes to the model or data have led to a better ability of the model to make accurate predictions. However, generalization performance continues to show variation, with T4 and T5 remaining relatively low compared to other tests, although slightly improved from Table 54.

Table 7 also shows significant improvement for the KNN, especially in T3, where accuracy has increased to 92.66 % from 83.24 % in Table 54. Other tests, such as T1 and T2, improved accuracy, recall, and F1-score metrics. However, KNN continues to show considerable variation between tests, suggesting that its generalization ability has improved but is still not optimal.
The BI-LSTM in Table 55 shows generally stable or improved performance. For example, accuracy in T2 has slightly increased to 95.37% from Table 54, which is 92.46%. However, some metrics, such as accuracy and recall, show variability across tests, indicating that although the model is effective under specific conditions, there are still areas where it can be improved, especially in terms of accuracy and generalization in tests such as T4 and T5. The SVM slightly improves accuracy and precision metrics but continues to underperform compared to the other models. Accuracy, for example, reached 68.77 %in T1 compared to 60.31 % in Table 54 but remains low in other tests such as T2 and T8. This suggests that the SVM is still not the best model for this problem despite some improvements.

Model	**Metric**	**Same dataset**			**Cross-dataset**				
		T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**
CNN	Accuracy	**84.88**	**96.03**	**88.96**	67.72	67.88	**84.90**	74.95	**85.10**
	Precision	**73.54**	**96.03**	**93.57**	56.67	44.97	**86.73**	63.51	**78.91**
	Recall	**71.08**	**94.41**	**81.34**	51.06	49.05	**78.14**	62.85	**80.88**
	F1-score	**72.18**	**95.18**	**85.79**	46.06	37.12	**77.81**	59.04	**79.37**
KNN	Accuracy	76.63	81.05	**92.66**	62.80	34.95	**81.65**	56.21	49.76
	Precision	59.07	82.96	**94.12**	38.66	38.27	**83.71**	35.99	54.02
	Recall	52.73	78.52	**88.17**	49.84	36.24	**72.17**	43.43	43.89
	F1-score	52.78	77.50	**90.67**	41.99	29.23	**71.70**	34.06	45.33
BI-LSTM	Accuracy	75.10	**95.37**	75.84	49.74	63.36	**81.23**	69.11	**83.50**
	Precision	60.79	**96.05**	65.67	26.14	62.12	**75.12**	64.63	**85.12**
	Recall	63.13	**93.55**	60.32	32.88	60.67	**74.96**	53.85	**82.48**
	F1-score	54.29	**94.67**	61.24	27.52	59.03	**79.23**	52.49	**84.98**
SVM	Accuracy	68.77	39.45	55.44	36.22	34.07	63.15	47.76	37.93
	Precision	39.66	26.23	53.23	33.44	19.27	41.84	37.95	51.65
	Recall	30.08	25.74	39.70	30.75	39.66	46.27	27.23	42.13
	F1-score	29.46	23.98	35.59	30.85	32.32	42.33	26.56	42.32

Table 55 - Experimental results with raw data without sitting action

Experiment with 5 classes - Raw Data and Increased Sliding Window

When the results in Table 56 are compared with those in Table 54, significant differences emerge due to the different data configurations and time windows used. Table 55 used a four-second sliding window method with 75% overlap and applied to the UMAFall and UNIBA HAR datasets. This differs from the first experiment, where data from the UniMiB SHAR dataset, which had predefined time windows, were also considered.

CNN continues to be the best model in terms of overall performance. In T1, the accuracy increased to 79.08% from 73.02% in the first experiment. In T3, accuracy is 86.90%, slightly higher than the 85.54% in Table 54. However, for T6 and T8, performance is like that of the first experiment, with accuracy hovering around 76-78%. This indicates that the KNN benefits from using the sliding window modification, slightly improving its generalization ability.

The KNN shows a general decrease in performance in Table 55. For example, the accuracy in T1 decreased to 56.73% compared to 61.44% in the first experiment. In T3, accuracy is comparable (84.24% compared to 83.24%), but in T6 and T8, performance is significantly worse, with accuracy dropping to 40.79% in T8. This suggests that the KNN fails to take advantage of the use of sliding windows, probably due to the nature of the data or the model's lack of ability to adapt to the temporal characteristics of the data.

The BI-LSTM shows mixed results. In T1, accuracy increased to 72.66% from 66.70% in the first experiment, indicating an improvement due to sliding windows. However, in T3, performance is the same (83.04% vs. 83.24%). For T6 and T8, the performance is comparable to that of the first experiment. This suggests that the BI-LSTM benefits somewhat from the increased sliding windows, but not uniformly. Finally, the SVM continues to show the worst performance among all models.

Model	Metrics	Same dataset	Cross-dataset		
		T1	T3	T6	T8
CNN	Accuracy	**79.08**	**86.90**	**78.64**	76.65
	Precision	**71.76**	**87.63**	**71.78**	69.71
	Recall	**63.63**	**80.19**	**70.07**	67.16
	F1-score	**65.20**	**83.24**	**67.21**	66.04
KNN	Accuracy	56.73	**84.24**	61.75	40.79
	Precision	43.46	**86.05**	31.17	44.23
	Recall	34.70	**78.69**	45.40	33.03
	F1-score	32.36	**79.92**	36.37	23.08
BI-LSTM	Accuracy	72.66	**83.04**	70.78	60.73
	Precision	61.74	**71.75**	62.98	40.56
	Recall	50.61	**73.28**	56.39	40.84
	F1-score	44.32	**72.42**	49.37	36.66
SVM	Accuracy	48.46	54.41	39.59	35.92
	Precision	28.73	55.39	33.54	32.88
	Recall	25.57	32.29	31.90	27.46
	F1-score	25.57	44.70	30.85	28.43

Table 56 - Experimental results with a sliding window of 4 seconds

4.3.3.2 Second Experiment

Fall Experiment with 3 Classes - Raw Data

Analyzing Table 57, CNN shows good performance, particularly in the T3 test (Training UNIBA HAR Dataset, Testing UNIBA HAR Dataset), with an accuracy of 70.61% and precision of 78.58%, a sign of solid learning ability when the training and testing sets coincide. However, its performance

drops significantly when tested on datasets different from the training dataset, showing limited generalization ability.

The KNN model performs well, especially in T2 (*Training UniMiB SHAR, Testing UniMiB SHAR*) and T3, with accuracies of 58.00% and 72.51%, respectively. However, the KNN shows some weakness in tests involving mixed or other training datasets, as indicated by the lower performance in T6 and T8, suggesting that it may be less effective in contexts with high data variability.

The BI-LSTM model, known for capturing long-term dependencies in sequential data, shows high accuracy in T2 (73.63%) and T3 (68.91%), confirming its effectiveness in test scenarios where data sequences are like training sequences. However, the Bi-LSTM struggles in tests on datasets other than training datasets, as evidenced by lower scores in T4 and T8. This suggests that, although powerful, the BI-LSTM may need regularization techniques or additional data to improve its generalization.

Finally, the SVM model shows variable performance with a relatively high accuracy in T7 (*Training UniMiB SHAR, Testing UNIBA HAR Dataset*) of 61.19% and a good F1-score of 61.21%. However, its accuracy and other metrics are lower than the other models, especially in T3 (45.81%) and T5 (44.80%). This suggests that the SVM may be less effective in dealing with complex or nonlinear data than the other models. Finally, the results of the latest experiment regarding Leave One Out are shown.

Model	Metrics	Same Dataset			Cross-dataset				
		T1	T2	T3	T4	T5	T6	T7	T8
CNN	Accuracy	46.46	36.16	**70.61**	39.40	44.22	37.42	38.05	**72.41**
	Precision	46.10	22.71	**78.58**	41.54	63.09	39.06	53.23	**75.08**
	Recall	45.84	34.51	**62.64**	39.81	43.49	37.18	43.73	**71.83**
	F1-score	45.87	22.56	**65.62**	39.47	35.88	37.77	37.27	**70.34**
KNN	Accuracy	48.31	58.00	**72.51**	37.26	46.48	32.03	51.80	39.55
	Precision	48.53	57.89	**72.11**	37.41	42.75	32.42	54.68	37.62
	Recall	48.28	61.13	**69.65**	39.80	43.60	34.33	56.21	39.50
	F1-score	48.15	58.69	**70.01**	38.23	41.20	33.05	52.24	38.29
BI-LSTM	Accuracy	44.37	**73.63**	**68.91**	35.23	46.86	32.23	49.21	39.22
	Precision	42.77	**73.40**	**73.43**	38.62	40.46	29.96	49.89	45.12
	Recall	44.23	**76.33**	**62.35**	39.02	42.87	30.14	50.72	38.21
	F1-score	42.44	**73.79**	**64.10**	35.68	37.31	30.33	49.74	35.24
SVM	Accuracy	36.99	51.26	45.81	38.12	44.80	41.32	60.25	38.92
	Precision	37.88	48.74	46.74	38.51	51.13	45.51	61.19	38.84
	Recall	36.77	48.94	48.20	38.76	45.49	45.78	66.31	39.07
	F1-score	36.85	48.70	46.53	37.98	44.26	41.39	61.21	37.54

Table 57 - Fall experimental results with raw data

Fall Experiment with Three Classes - Leave One Out

Table 58 contains the observations inherent in LOO analysis by users. CNN models demonstrate the best overall performance, especially in the UniMiB SHAR and UNIBA HAR datasets. CNN outperforms the other models with 95.24% and 90.32% accuracy, respectively. CNN's accuracy and recall are also high, indicating a remarkable ability to classify falls consistently, although the performance is relatively lower in the UMAFall dataset.

The KNN model, on the other hand, shows considerable variability in performance. On the UniMiB SHAR dataset, its accuracy is only 58.22%, while on the UNIBA HAR dataset, it reaches 89.77%. This shows that the KNN may be effective in specific contexts but lacks general robustness. The recall and F1 scores are consistent with these results, showing a discrepancy between the datasets.

The BI-LSTM, a model based on recurrent neural networks, offers intermediate performance. While not matching the CNN, it shows good accuracy on the UniMiB SHAR (92.08%) and UNIBA

(88.16%) datasets. However, its accuracy and recall are slightly lower than CNN's, suggesting difficulty capturing all fall instances.

Finally, the SVM model ranks as the least performing across all datasets, with accuracy values that do not exceed 44.55%. The SVM's accuracy, recall, and F1-score are low, indicating a poor ability to distinguish between different fall classes correctly. These results suggest that the SVM may not be suitable for this type of classification in comparison with other models.

Data analysis was performed with a user LOO test, which means that for each iteration, a user is excluded from the training set and used for the test. This method allows for a robust evaluation of the model because it simulates an actual application where the model must generalize to data from new users not seen before. This methodology is particularly relevant for evaluating the ability of models to generalize to new individuals, which is crucial for real-world applications such as fall monitoring.

Model	Metrics	UMAFall	UniMiB SHAR	UNIBA HAR Dataset
CNN	Accuracy	**74.43%**	**95.24%**	**90.32%**
	Precision	**77.31%**	**90.81%**	**91.29%**
	Recall	**71.66%**	**89.01%**	**87.99%**
	F1-score	**71.70%**	**88.60%**	**88.44%**
KNN	Accuracy	59.08%	58.22%	**89.77%**
	Precision	63.17%	54.08%	**90.05%**
	Recall	47.90%	55.19%	**87.85%**
	F1-score	49.21%	44.14%	**88.00%**
BI-LSTM	Accuracy	70.77%	**92.08%**	**88.16%**
	Precision	62.59%	**90.00%**	**85.92%**
	Recall	62.99%	**83.06%**	**82.97%**
	F1-score	60.47%	**83.72%**	**83.21%**
SVM	Accuracy	43.61%	32.74%	44.55%
	Precision	32.40%	19.91%	31.88%
	Recall	25.29%	20.97%	27.33%
	F1-score	23.91%	19.10%	25.20%

Table 58 - LOO average results

The HAR systems use machine learning algorithms to identify human activities, but few studies use smartphone sensors to recognize different falls. This study focuses on risk activities such as falls and neutral activities such as walking and jumping. A new dataset, the UNIBA HAR Dataset, was created to include different types of falls to improve recognition patterns and provide timely interventions in health care. The UNIBA HAR dataset performs excellently, even in the context of falls alone.

The following are critical observations on the experiments conducted to identify and classify different hazardous activities and situations correctly:

- **Best Model**: The CNN performed best on homogeneous training and test datasets, as in the case of the T2 test, with an accuracy of 93.39%. Even with cross-dataset data, CNN maintained significant robustness, albeit with a drop in performance compared to testing on homogeneous data. In the LOO experiment, CNN excelled in the UniMiB SHAR and UNIBA HAR datasets with accuracies of 95.24% and 90.32%, respectively;
- **Model with Good Performance**: The B-LSTM model showed good performance, especially in the T2 (accuracy of 92.46%) and T3 tests. It is particularly effective in capturing temporal dependencies in sequential data. However, it suffers from generalization problems with data other than training data, as evidenced in cross-dataset tests. In the LOO experiment, it achieved high accuracies in the UniMiB SHAR and UNIBA HAR datasets but slightly lower than CNN;
- **Performance of the K-Nearest Neighbors (KNN)**: The KNN showed variable performance. It achieved good results in some tests, such as T3 (accuracy of 83.24%), and significant improvements in specific conditions. However, its generalization ability was limited, with deficient

performance in cross-dataset tests (e.g., accuracy of 27.07% in T5). In the LOO experiment, it showed significant variability, with accuracies ranging from 58.22% to 89.77%;

- **Support Vector Machine (SVM) performance**: The SVM was the worst-performing model. Accuracies were generally low, with a maximum of 50.22% in the T7 test. It failed to stand out in cross-dataset tests and showed lower metrics than the other models in almost all tests;
- **Effects of Data Changes**: Removing the sitting action improved the performance of all models, suggesting that the confusion between falling and sitting was significant. The use of more oversized sliding windows had positive effects. The CNN benefited most from this technique, while the KNN saw a slight decrease in performance;
- **Generalization of Models**: The models generally showed better performance on homogeneous datasets than cross-dataset tests, indicating the need for additional regularization techniques and diverse training data to improve robustness;
- **Unification of Different Falls**: The unification of different falls into one class, given the results of the falls experiment, undoubtedly contributed to the increase in classification metrics;
- **Recognition of Activities**: The most accessible classes to recognize are walking, running, and falling, as these are better recognized and are not confused with other activities. In some classifiers, the activity of jumping is confused with running, given the similar variation on the y-axis, and similarly, the activity of sitting behaves like falling;
- **Performance of the Proposed Dataset**: The dataset proposed by this study performs better than one of the two state-of-the-art datasets, UMAFall, and ranks just below the other one considered by this study and the state-of-the-art, UniMiB SHAR;
- **False Positive Issues**: The underperforming results seen in the previous chapter for UMAFall confirm Khojasteh et al. [11] 's concerns regarding false positives. One explanation is the imbalance of samples in UMAFall for some users and the fact that many users needed to be sampled better in some tasks. Balanced datasets, such as the UNIBA HAR dataset proposed in this study or UniMiB SHAR, perform better and, in the case of UniMiB SHAR, also in previous studies.

The CNN model emerged as the top performer, achieving an impressive 93.39% accuracy in homogeneous data and demonstrating robustness even with cross-set data. It particularly shone in the UniMiB SHAR and UNIBA HAR datasets. The B-LSTM model also delivered good results, especially in T2 and T3 tests, but struggled with generalization beyond the training data, which could limit its practical use. The KNN model's performance was variable, with some tests yielding good results (83.24% in T3), but it showed limited generalization ability, which could also impact its practical use. Unfortunately, the SVM model was the weakest performer, consistently delivering low accuracies and a maximum of 50.22% in T7, further limiting its practical use.

In *Future developments*, it would be beneficial to extend the UNIBA HAR dataset with additional users, focusing on studies to categorize different types of falls, a topic that has not yet been extensively explored in the literature. Another potential direction could be the implementation of new methodologies to distinguish between fall-sit and jump-run using smartphone sensors. Furthermore, to protect the dataset and users' privacy, asymmetric key cryptography schemes such as Elliptic Curve Cryptography (ECC) or AES [258] could be adopted. At the same time, techniques for detecting cyberattacks through behavioral biometrics models are still being refined, especially regarding the analysis of human interactions in real-world contexts. Future developments will focus on expanding behavioral investigations using advanced sensors, integrating more sophisticated machine learning algorithms, and applying these models to different contexts.

Additionally, the dataset has some *limitations*, such as the bias in age differentiation between male students in the university dataset and the predominantly female, younger age group in the school dataset. However, the dataset remains overall balanced. It is also important to note that the data was collected during the COVID period, which may influence some of the behaviors observed.

5. Contribution

In this brief chapter, I will outline the contributions (papers) for this PhD thesis. The titles of the chapters have been set as the names of the papers to better facilitate referencing. The detailed description of the works is in the chapters mentioned below.

Subchapter 1.2.1: "*Leveraging Artificial Intelligence to Fight (Cyber)Bullying for Human Well-being: The Bully-Buster Project*"
Status: Paper Published
Description: This paper discusses the use of artificial intelligence to combat cyberbullying, part of the "Bully-Buster" project, an initiative developed by an interdisciplinary team of Italian universities aimed at detecting cyberbullying content in real-time through behavioral biometrics and social network analysis techniques. References to AI for crowd analysis and behavioral biometric detection systems are extensively covered. It is a project publicly presented at Italian tech fairs, such as the Maker Faire in Rome in 2021.
Link: https://iris.unica.it/handle/11584/377764

Subchapter 1.2.2: "*Development of Technologies for the Detection of (Cyber)Bullying Actions: The Bully-Buster Project*"
Status: Paper Published
Description: This study delves into the technologies and models developed to detect acts of cyberbullying. The Bully-Buster project, funded and presented in academic and scientific contexts, leverages smartphone sensors and advanced behavioral recognition techniques to identify bullying actions, aiming to promote the social safety of young people on social networks.
Link: https://www.mdpi.com/2078-2489/14/8/430

Subchapter 1.2.3: "*Cyber Aggression and Cyberbullying Identification on Social Networks*"
Status: Paper Published
Description: The work presents an automatic system for identifying cyberbullying by analyzing textual comments from Italian Twitter to detect aggression. Two experiments were conducted to identify cyber aggression and cyberbullying, achieving the best results with the Random Forest classifier, trained on a specific dataset of labeled comments. Although the system is a valid tool for addressing cyberbullying, it requires further improvements to optimize performance.
Link: https://www.researchgate.net/publication/358686276_Cyber_Aggression_and_Cyberbullying_Identification_on_Social_Networks

Chapter 2: "*Human Activity Recognition with Smartphone-Integrated Sensors: A Survey*"
Status: Paper Published
Description: Human Activity Recognition (HAR) is an important research area focusing on smartphones' ability to recognize human activities through integrated sensors. This work provides a practical overview of the sensors in modern smartphones and the most cited machine learning models for activity recognition. Through summary tables, methods, datasets, co-occurrences between activities and sensors, and obtained performances are analyzed, aiming to present the current state of the art in this field.
Link: https://www.sciencedirect.com/science/article/pii/S0957417424000083

Subchapter 3.3.2.1: "*Fixed Tasks for Continuous Authentication via Smartphone*"
Status: Paper Published
Description: The document analyzes the effectiveness of various machine learning algorithms for user authentication on mobile devices, highlighting the vulnerabilities of traditional authentication methods such as PINs and passwords. It proposes the use of fixed tasks that simulate daily interaction with the device, utilizing motion sensors and touch behavior. Additionally, a social issue related to

identity verification is explored, assessing whether a group of subjects has completed the assigned tasks correctly without the intervention of others.
Link: https://www.researchgate.net/publication/369015921_Fixed_Tasks_for_Continuous_Authentication_via_Smartphone

Subchapter 3.3.2.2: "*Anomaly Detection Using Smartphone Sensors for a Bullying Detection App*"
Status: Paper Published
Description: Anomaly detection is a fundamental process for identifying situations that deviate from the norm. This work analyzes anomalies in human behavior during the completion of a bullying and cyberbullying questionnaire, using data from smartphone sensors. Psychology and computer science are integrated to uncover latent patterns in the dataset, aiming to identify abnormal behaviors among users of the Android application.
Link: https://www.researchgate.net/publication/375006021_Anomaly_Detection_using_smartphone_Sensors_for_a_Bullying_Detection

Subchapter 3.3.2.3: "*Human Activity Recognition for the Identification of Bullying and Cyberbullying Using Smartphone Sensors*"
Status: Paper Published
Description: This work leverages data from smartphone sensors and Machine Learning techniques to analyze human behavior during the completion of a bullying and cyberbullying questionnaire. Using Human Activity Recognition (HAR) models, the goal is to classify users as Bullies, Cyberbullies, Bullying Victims, and Cyberbullying Victims, recognizing five daily activities: walking, jumping, sitting, running, and falling. The analysis is based on a model designed to discriminate daily actions from those related to bullying behaviors.
Status: Paper Published
Link: https://www.mdpi.com/2079-9292/12/2/261

Subchapter 3.3.2.4: "*Classification Bullying/Cyberbullying through Smartphone Sensor and a Questionnaire Application*"
Status: Paper Published
Description: This study explores the correlation between computer science and psychology, focusing on the use of smartphone sensors and the users' personality index. An Android application for a bullying and cyberbullying questionnaire has been developed, utilizing sensor data collected in the "UNIBA HAR Dataset" and analyzed with artificial intelligence algorithms. The results show that the Bayesian Bridge model achieves an average accuracy of 0.94, while the LSTM model achieves 0.89, highlighting the importance of these findings for future research in the field.
Link: https://link.springer.com/article/10.1007/s11042-023-17609-7

Subchapter 3.3.3.1: "*Human Activity Recognition for Identifying Bullying and Cyberbullying: A Comparative Analysis Between School and University Students*"
Status: Paper Published
Description: This work uses Human Activity Recognition (HAR) models to identify activities performed during the completion of a questionnaire, classifying users as bullies, cyberbullies, bullying victims, and cyberbullying victims. The analysis aims to recognize activities beyond simple sitting, focusing on movements such as walking, jumping, running, and falling. The best model identified for activity recognition, a CNN, was applied to an experimental dataset obtained via smartphones.
Link: https://www.scitepress.org/publishedPapers/2024/125788/pdf/index.html

Subchapter 4.1: "*Touch Events and Human Activities for Continuous Authentication via Smartphone*"
Status: Paper Published

Description: The security of modern smartphones relies on continuous authentication techniques through touch events and human activities, which provide valuable data for machine learning algorithms. This study develops a continuous authentication method while the user is seated and scrolling through documents on the phone, utilizing data from the H-MOG dataset with the Signal Vector Magnitude as an additional feature. The results show that the 1-class SVM model achieves an accuracy of 98.9% and an F1-score of 99.4%.
Link: https://www.nature.com/articles/s41598-023-36780-3

Subchapter 4.2: "*Two-factor Authentication by Combining PIN and Biometrics Touch Dynamics*"
Status: Paper Published
Description: Mobile devices are increasingly widespread, but their use exposes users to the risk of unintentional sharing of sensitive information. This study implements and tests a user verification system based on touch dynamics during PIN entry, improving the state of the art in binary classification with an EER of 0.01% on the considered dataset.
Link: https://ieeexplore.ieee.org/document/10386880

Subchapter 4.3: "*Human Activity Recognition using Smartphone Sensors: Focusing on Fall Detection with the UNIBA HAR Dataset*"
Status: Paper Submitted
Description: Human Activity Recognition (HAR) uses smartphone sensors to detect everyday activities, distinguishing between "neutral" activities (e.g., walking) and "risky" activities (e.g., falling), which are relevant for detecting bullying situations. The study introduces the UNIBA HAR Dataset, containing accelerometer data collected from 19 users performing different activities, including three fall variants. The differentiation of fall types and the application of advanced machine learning models aim to improve health monitoring and intervention systems, filling a gap in current research.

6. Conclusion

The PhD thesis is conducted as part of the Ph.D. program in Computer Science and Mathemat-ics and has been funded by the PON Research and Innovation 2014-2020 ESF REACT-EU, Action IV.4 "*Ph.D. Programs and Research Contracts on Innovation Topics*" (CUP H99J21010060001). Academic supervision was provided by Prof. Antonio Piccinno, with the support of Co-Tutor Prof. Donato Impedovo.
The **PhD thesis** represents a pioneering contribution to the field of behavioral biometrics, emphasizing the potential of these technologies to enhance safety and well-being through precise and vigilant monitoring of daily interactions. Through an in-depth and interdisciplinary analysis, this research demonstrates how integrating behavioral biometric models can foster innovative assessment tools, such as the BullyBuster Questionnaire, effectively aiding in the prevention and control of complex phenomena like bullying and cyberbullying.

The **thesis structure** unfolds across various chapters, not only exploring the theoretical and practical understanding of behavioral biometrics but also extending the application potential of current technologies, proposing methodologies and tools applicable across diverse disciplinary and professional contexts. Among the significant contributions is the creation of the BullyBuster Questionnaire, a unique application that combines biometric data analysis with targeted questionnaires, allowing for detailed data collection and behavioral analysis, which identifies human behaviors and signs of distress.

The **objective** of the thesis lies in implementing behavioral biometric models that accompany a structured questionnaire on bullying and cyberbullying. These models, developed through advanced data

analysis, enable the identification of abnormal behaviors correlated with sensitive questionnaire items, creating personalized behavioral profiles capable of signaling potential threats or psychological distress indicators. A **key aspect** of the research was the development of dedicated behavioral datasets for the BullyBuster Questionnaire (BBQuest) used in school and university settings and the Human Activity Recognition (HAR) dataset produced in a laboratory environment. These datasets have played a crucial role in training and validating machine learning models, ensuring high levels of accuracy and reliability in solutions applied across various real-world contexts.

In examining daily activities, the chapter on **Human Activity Recognition (HAR)** with smartphone-integrated sensors highlights the importance of technologies like accelerometers and gyroscopes in monitoring habitual movements and activities. Advanced machine learning techniques, including Convolutional Neural Networks (CNN) and *Hidden Markov Models (HMM)*, have demonstrated effectiveness in comprehending behavioral dynamics, and finding relevant applications in health and safety sectors. Despite the complexity of these models and the risk of overfitting, proposals to extend activity recognition analysis to everyday contexts signify an advancement in understanding human behavior. Comparative studies conducted on HMM and CNN models underscored their high performance in activity recognition, with the **UNIBA HAR dataset** significantly improving critical event classification, such as falls, compared to other reference datasets by reducing false positives. These findings highlight the importance of balanced datasets for training robust and reliable models, essential for behavioral analysis.

The development of the **BullyBuster Questionnaire**, available as both an Android application and a Web App, has expanded bullying detection and monitoring capabilities, offering an easily accessible technological platform in educational settings. This tool has been utilized in real educational environments, including the University of Bari and high schools in Cagliari and Avellino, allowing the collection of high-quality and authentic behavioral data. The dual nature of the questionnaire, accessible on mobile devices and as a web application, allows adaptable implementation in various educational settings, overcoming logistical limitations and increasing participation accessibility. Integrating advanced technologies and sensor-based data collected through smartphone sensors has introduced a new dimension to understanding bullying and cyberbullying dynamics. With movement and touch sensors, it has been possible to capture not only questionnaire responses but also data linked to behavioral patterns during interaction, creating a comprehensive and multimodal dataset. This data structure, enriched by touch and movement details, enables a more precise analysis and detection of behavioral anomalies that may be associated with specific questionnaire items currently being completed by the user.

The **experimental chapters** presented key studies that analyze the behavioral dataset of the BBQuestionnaire to compare university and high school students. The first analysis shows the use of human activity recognition (HAR) models to study high school and university students' behavior while administering a questionnaire. Utilizing accelerometer sensors integrated into smartphones, the research identified *"abnormal"* behaviors such as running and falling, revealing significant differences between the two groups. The results indicate that high school students tend to be more active and involved in bullying dynamics, suggesting the need for advanced cybersecurity measures to protect sensitive information. The convolutional neural network (CNN) achieved the best performance among the tested models. The second comparative analysis focuses on anomaly detection and classification algorithms to analyze high school and university students' behaviors. The Elliptic Envelope algorithm demonstrated the best anomaly detection capabilities, particularly among high school students. The results indicate a high frequency of bullying and cyberbullying-related behaviors among more younger groups. Furthermore, the Random Forest proved the best classification algorithm in both groups, highlighting the influence of external factors during questionnaire administration. The analysis suggests that the methodology is valid but could benefit from additional discriminative

behavioral features. High school students exhibited greater vulnerability to abnormal behaviors and a higher frequency of bullying-related situations. The most relevant performances were recorded by algorithms such as CNN and Random Forest, which proved robust and reliable for behavioral analysis and anomaly detection. These results emphasize the importance of monitoring and understanding youth behaviors in educational settings to implement effective bullying and cyberbullying prevention strategies.

This PhD thesis demonstrated how behavioral biometrics could play a key role in enhancing safety and well-being, identifying anomalies indicative of psychological distress, and enabling timely interventions. Besides advancing research in behavioral biometrics, the thesis fostered academic collaborations and disseminated results through scientific publications, enriching the field of tools and methodologies for future studies. Research perspectives include optimizing monitoring techniques and psychological analysis of attack dynamics, opening new possibilities for applying biometric technologies in diverse sectors. Indeed, future developments recommend adopting advanced extraction techniques for interpreting temporal data, such as genetic programming methods and other automated methodologies. This enhancement can allow for a deeper understanding of temporal dynamics in human activities, increasing model precision and reliability. An essential aspect for research progress lies in expanding datasets. This development would enhance the versatility of HAR systems, making them applicable to a wide range of contexts. The BullyBuster project represents just one of many potential developments; in fact, a new project, BullyBuster2, is continuing this research. Future studies could focus on analyzing the frequency and nature of bullying episodes to identify recurring patterns useful for improving monitoring technologies. In parallel, psychological insights into the motivations behind aggressive behaviors could support the development of more targeted and effective preventive interventions. The implementation of encryption schemes, such as Elliptic Curve Cryptography (ECC) or AES, could improve the security of sensitive data, especially in datasets used for training and evaluating HAR models. On the methodological front, exploring under-researched approaches such as multimodal and Multiview learning would be interesting. These methods could enhance model generalizability across heterogeneous datasets and real-life situations, increasing the reliability of behavioral analyses.

The work presented in **Chapter 2** has led to careful discussion and analysis of methodologies in the **Human Activity Recognition (HAR)** domain. While the presented study offers a well-structured overview and a **solid experimental foundation** in HAR, several **important methodological considerations** must be highlighted. In particular, the **selection and application** of methods were driven by **data availability** and their inherent characteristics, resulting in coherent but **potentially improvable** solutions. A **critical expansion** of the methodological section, including a deeper discussion of **alternative approaches**, could have **strengthened the overall value** of the study.

Among the most effective methodologies identified, **Convolutional Neural Networks (CNNs)** stand out for their ability to automatically extract discriminative features from raw signals, along with **two-stage hierarchical models based on Coupled Hidden Markov Models (CHMMs)**, which demonstrated high accuracy in capturing the sequential and temporal structure of human activities. Moreover, the integration of sensors such as accelerometers and gyroscopes, combined with **Principal Component Analysis (PCA)** for dimensionality reduction, enabled an effective data representation, facilitating classification.

A particularly relevant aspect was the feature extraction and selection process, which emerged as one of the key strengths of the entire study. However, the challenge of robustly identifying significant and generalizable behavioral metrics remains open to a challenge that is still the focus of active research within the scientific community. This is a highly complex ("*super-hard*") problem, requiring increasingly adaptive and data-driven approaches. Considering the above, the current work represents an

initial step toward the integration of HAR techniques into real-world and personalized scenarios, with promising applications in health monitoring, personal safety, continuous authentication, and mobile device personalization.

Future directions include the optimization of automatic feature extraction pipelines, the use of unsupervised pre-training strategies, the development of lightweight and interpretable models for mobile deployment, and the expansion of datasets through realistic recordings in uncontrolled environments. In my research grant I am pursuing the HAR topic in the area of Bullying/cyber using newer deep-learning approaches to the problem.

In conclusion, this research makes a significant contribution to understanding and applying behavioral biometrics to urgent social issues such as bullying and cyberbullying. The proposed synergy between technology and ethics outlines a promising path for developing increasingly sophisticated tools that respect the dignity and rights of users, establishing a solid foundation for future scientific and social advancements.

7. References

[1] S. Hinduja and J. W. Patchin, "Offline consequences of online victimization: School violence and delinquency," *J Sch Violence*, vol. 6, no. 3, pp. 89–112, 2007, doi: 10.1300/J202V06N03_06.

[2] S. Woods and D. Wolke, "Direct and relational bullying among primary school children and academic achievement," J Sch Psychol, vol. 42, no. 2, pp. 135–155, Mar. 2004, doi: 10.1016/J.JSP.2003.12.002.

[3] M. A. Campbell, "Cyber Bullying: An Old Problem in a New Guise?," *Australian Journal of Guidance and Counselling*, vol. 15, no. 1, pp. 68–76, Jul. 2005, doi: 10.1375/AJGC.15.1.68.

[4] D. Olweus, *Bullying at School: Long-Term Outcomes for the Victims and an Effective School-Based Intervention Program.* Springer, Boston, MA, 1994. doi: 10.1007/978-1-4757-9116-7_5.

[5] S. Hinduja and J. W. Patchin, "Bullying, cyberbullying, and suicide," *Archives of Suicide Research*, vol. 14, no. 3, pp. 206–221, Jul. 2010, doi: 10.1080/13811118.2010.494133.

[6] P. K. Smith, J. Mahdavi, M. Carvalho, S. Fisher, S. Russell, and N. Tippett, "Cyberbullying: its nature and impact in secondary school pupils," *Journal of Child Psychology and Psychiatry*, vol. 49, no. 4, pp. 376–385, Apr. 2008, doi: 10.1111/J.1469-7610.2007.01846.X.

[7] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, Oct. 2007, doi: 10.1111/J.1083-6101.2007.00393.X.

[8] R. E. Petty, S. G. Harkins, K. D. Williams, and B. Latane, "The Effects of Group Size on Cognitive Effort and Evaluation," *http://dx.doi.org/10.1177/014616727700300406*, vol. 3, no. 4, pp. 579–582, Jul. 1977, doi: 10.1177/014616727700300406.

[9] H. Blumenfeld, "Cellular and Network Mechanisms of Spike-Wave Seizures," *Epilepsia*, vol. 46, no. SUPPL. 9, pp. 21–33, Nov. 2005, doi: 10.1111/J.1528-1167.2005.00311.X.

[10] Ö. Erdur-Baker, "Cyberbullying and its correlation to traditional bullying, gender and frequent and risky usage of internet-mediated communication tools," *New Media Soc*, vol. 12, no. 1, pp. 109–125, Feb. 2010, doi: 10.1177/1461444809341260.

[11] T. Beran and L. I. Qing, "Cyber-harassment: A study of a new method for an old behavior," *Journal of Educational Computing Research*, vol. 32, no. 3, pp. 265–277, 2005, doi: 10.2190/8YQM-B04H-PG4D-BLLH.

[12] C. Katzer, D. Fetchenhauer, and F. Belschak, "Cyberbullying: Who Are the Victims? A Comparison of Victimization in Internet Chatrooms and Victimization in School," *J Media Psychol*, vol. 21, no. 1, pp. 25–36, 2009, doi: 10.1027/1864-1105.21.1.25.

[13] K. J. Mitchell, M. Ybarra, and D. Finkelhor, "The relative importance of online victimization in understanding depression, delinquency, and substance use," *Child Maltreat*, vol. 12, no. 4, pp. 314–324, Nov. 2007, doi: 10.1177/1077559507305996.

[14] P. M. Valkenburg, J. Peter, and A. P. Schouten, "Friend networking sites and their relationship to adolescents' well-being and social self-esteem," *Cyberpsychol Behav*, vol. 9, no. 5, pp. 584–590, Oct. 2006, doi: 10.1089/CPB.2006.9.584.

[15] L. A. Jackson, A. Von Eye, E. A. Witt, Y. Zhao, and H. E. Fitzgerald, "A longitudinal study of the effects of Internet use and videogame playing on academic performance and the roles of gender, race and income in these relationships," *Comput Human Behav*, vol. 27, no. 1, pp. 228–239, Jan. 2011, doi: 10.1016/J.CHB.2010.08.001.

[16] K. R. Williams and N. G. Guerra, "Prevalence and Predictors of Internet Bullying," *Journal of Adolescent Health*, vol. 41, no. 6 SUPPL., Dec. 2007, doi: 10.1016/J.JADOHEALTH.2007.08.018.

[17] J. M. Cénat, M. Hébert, M. Blais, F. Lavoie, M. Guerrier, and D. Derivois, "Cyberbullying, psychological distress and self-esteem among youth in Quebec schools," *J Affect Disord*, vol. 169, pp. 7–9, Dec. 2014, doi: 10.1016/J.JAD.2014.07.019.

[18] U. Orth, R. W. Robins, K. F. Widaman, and R. D. Conger, "Is low self-esteem a risk factor for depression? Findings from a longitudinal study of mexican-origin youth," *Dev Psychol*, vol. 50, no. 2, pp. 622–633, Feb. 2014, doi: 10.1037/A0033817.

[19] U. Orth, R. W. Robins, and K. F. Widaman, "Life-span development of self-esteem and its effects on important life outcomes," *J Pers Soc Psychol*, vol. 102, no. 6, pp. 1271–1288, Jun. 2012, doi: 10.1037/A0025558.

[20] V. Gattulli, D. Impedovo, G. Pirlo, and L. Sarcinella, "Cyber Aggression and Cyberbullying Identification on Social Networks," *11th International Conference on Pattern Recognition Applications and Methods*, pp. 644–651, Feb. 2022, doi: 10.5220/0010877600003122.

[21] V. Vyas, K. Walse, and R. Dharaskar, "A Survey on Human Activity Recognition using Smartphone," *nternational Journal of Advance Research in Computer Science and Management Studies*, vol. 5, Jan. 2017.

[22] S. O. Slim, A. Atia, M. M. A. Elfattah, and M. S. M. Mostafa, "Survey on Human Activity Recognition based on Acceleration Data," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, pp. 84–98, Spring 2019, doi: 10.14569/IJACSA.2019.0100311.

[23] A. Gupta, K. Gupta, K. Gupta, and K. Gupta, "A Survey on Human Activity Recognition and Classification," *Proceedings of the 2020 IEEE International Conference on Communication and Signal Processing, ICCSP 2020*, pp. 915–919, Jul. 2020, doi: 10.1109/ICCSP48568.2020.9182416.

[24] D. Thakur and S. Biswas, "Smartphone based human activity monitoring and recognition using ML and DL: a comprehensive survey," *J Ambient Intell Humaniz Comput*, vol. 11, no. 11, pp. 5433–5444, Nov. 2020, doi: 10.1007/S12652-020-01899-Y.

[25] A. KH and L. Ibrahim, "Survey on Human Activity Recognition using Smartphone," *AL-Rafidain Journal of Computer Sciences and Mathematics*, vol. 15, no. 1, pp. 55–67, Jun. 2021, doi: 10.33899/CSMJ.2021.168253.

[26] H. F. Nweke, Y. W. Teh, G. Mujtaba, and M. A. Al-garadi, "Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions," *Information Fusion*, vol. 46, pp. 147–170, Mar. 2019, doi: 10.1016/J.INFFUS.2018.06.002.

[27] A. Jordao, L. A. B. Torres, and W. R. Schwartz, "Novel approaches to human activity recognition based on accelerometer data," *Signal Image Video Process*, vol. 12, no. 7, pp. 1387–1394, Oct. 2018, doi: 10.1007/S11760-018-1293-X/TABLES/6.

[28] E. M. Tapia *et al.*, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," *Proceedings - International Symposium on Wearable Computers, ISWC*, pp. 37–40, 2007, doi: 10.1109/ISWC.2007.4373774.

[29] A. Wang, G. Chen, J. Yang, S. Zhao, and C. Y. Chang, "A Comparative Study on Human Activity Recognition Using Inertial Sensors in a Smartphone," *IEEE Sens J*, vol. 16, no. 11, pp. 4566–4578, Jun. 2016, doi: 10.1109/JSEN.2016.2545708.

[30] A. S. Abdull Sukor, A. Zakaria, and N. Abdul Rahim, "Activity recognition using accelerometer sensor and machine learning classifiers," *Proceedings - 2018 IEEE 14th International Colloquium on Signal Processing and its Application, CSPA 2018*, pp. 233–238, May 2018, doi: 10.1109/CSPA.2018.8368718.

[31] Y. J. Kim, B. N. Kang, and D. Kim, "Hidden Markov Model Ensemble for Activity Recognition Using Tri-Axis Accelerometer," *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 3036–3041, Jan. 2016, doi: 10.1109/SMC.2015.528.

[32] N. Hnoohom, S. Mekruksavanich, and A. Jitpattanakul, "Human activity recognition using triaxial acceleration data from smartphone and ensemble learning," *Proceedings - 13th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2017*, vol. 2018-January, pp. 408–412, Apr. 2018, doi: 10.1109/SITIS.2017.73.

[33] A. Bayat, M. Pomplun, and D. A. Tran, "A Study on Human Activity Recognition Using Accelerometer Data from Smartphones," *Procedia Comput Sci*, vol. 34, pp. 450–457, Jan. 2014, doi: 10.1016/J.PROCS.2014.07.009.

[34] J. Wannenburg and R. Malekian, "Physical Activity Recognition from Smartphone Accelerometer Data for User Context Awareness Sensing," *IEEE Trans Syst Man Cybern Syst*, vol. 47, no. 12, pp. 3142–3149, Dec. 2017, doi: 10.1109/TSMC.2016.2562509.

[35] A. D. Ignatov and V. V. Strijov, "Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer," *Multimed Tools Appl*, vol. 75, no. 12, pp. 7257–7270, Jun. 2016, doi: 10.1007/S11042-015-2643-0/TABLES/5.

[36] B. Kolosnjaji and C. Eckert, "Neural network-based user-independent physical activity recognition for mobile devices," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9375 LNCS, pp. 378–386, 2015, doi: 10.1007/978-3-319-24834-9_44.

[37] K. H. Walse, R. V. Dharaskar, and V. M. Thakare, "PCA based optimal ANN classifiers for human activity recognition using mobile sensors data," *Smart Innovation, Systems and Technologies*, vol. 50, pp. 429–436, 2016, doi: 10.1007/978-3-319-30933-0_43/COVER.

[38] R. Akhavian and A. H. Behzadan, "Smartphone-based construction workers' activity recognition and classification," *Autom Constr*, vol. 71, no. Part 2, pp. 198–209, Nov. 2016, doi: 10.1016/J.AUTCON.2016.08.015.
[39] C. A. Ronao and S. B. Cho, "Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models," *Int J Distrib Sens Netw*, vol. 13, no. 1, Jan. 2017, doi: 10.1177/1550147716683687/ASSET/IMAGES/LARGE/10.1177_1550147716683687-FIG2.JPEG.
[40] C. Catal, S. Tufekci, E. Pirmit, and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," *Appl Soft Comput*, vol. 37, pp. 1018–1022, Dec. 2015, doi: 10.1016/J.ASOC.2015.01.025.
[41] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences (Switzerland)*, vol. 7, no. 10, Oct. 2017, doi: 10.3390/APP7101101.
[42] L. Xu, W. Yang, Y. Cao, and Q. Li, "Human activity recognition based on random forests," *ICNC-FSKD 2017 - 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 548–553, Jun. 2018, doi: 10.1109/FSKD.2017.8393329.
[43] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical Human Activity Recognition Using Wearable Sensors," *Sensors 2015, Vol. 15, Pages 31314-31338*, vol. 15, no. 12, pp. 31314–31338, Dec. 2015, doi: 10.3390/S151229858.
[44] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep Learning Models for Real-time Human Activity Recognition with Smartphones," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, Apr. 2020, doi: 10.1007/S11036-019-01445-X/FIGURES/7.
[45] A. Jain and V. Kanhangad, "Human Activity Classification in Smartphones Using Accelerometer and Gyroscope Sensors," *IEEE Sens J*, vol. 18, no. 3, pp. 1169–1177, Feb. 2018, doi: 10.1109/JSEN.2017.2782492.
[46] L. M. Rodrigues and M. Mestria, "Classification methods based on bayes and neural networks for human activity recognition," *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016*, pp. 1141–1146, Oct. 2016, doi: 10.1109/FSKD.2016.7603339.
[47] L. Zhang, X. Wu, and Di. Luo, "Human activity recognition with HMM-DNN model," *Proceedings of 2015 IEEE 14th International Conference on Cognitive Informatics and Cognitive Computing, ICCI*CC 2015*, pp. 192–197, Sep. 2015, doi: 10.1109/ICCI-CC.2015.7259385.
[48] R. San-Segundo, H. Blunck, J. Moreno-Pimentel, A. Stisen, and M. Gil-Martín, "Robust Human Activity Recognition using smartwatches and smartphones," *Eng Appl Artif Intell*, vol. 72, pp. 190–202, Jun. 2018, doi: 10.1016/J.ENGAPPAI.2018.04.002.
[49] M. Zeng *et al.*, "Convolutional Neural Networks for human activity recognition using mobile sensors," *Proceedings of the 2014 6th International Conference on Mobile Computing, Applications and Services, MobiCASE 2014*, pp. 197–205, Jan. 2015, doi: 10.4108/ICST.MOBICASE.2014.257786.
[50] Y. Chen and Y. Xue, "A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer," *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 1488–1492, Jan. 2016, doi: 10.1109/SMC.2015.263.
[51] B. Almaslukh, A. M. Artoli, and J. Al-Muhtadi, "A Robust Deep Learning Approach for Position-Independent Smartphone-Based Human Activity Recognition," *Sensors (Basel)*, vol. 18, no. 11, Nov. 2018, doi: 10.3390/S18113726.
[52] M. Panwar *et al.*, "CNN based approach for activity recognition using a wrist-worn accelerometer," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 2438–2441, Sep. 2017, doi: 10.1109/EMBC.2017.8037349.
[53] S. Ha and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2016-October, pp. 381–388, Oct. 2016, doi: 10.1109/IJCNN.2016.7727224.
[54] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing," *26th International World Wide Web Conference, WWW 2017*, pp. 351–360, Nov. 2016, doi: 10.48550/arxiv.1611.01942.
[55] W. Xu, Y. Pang, Y. Yang, and Y. Liu, "Human Activity Recognition Based on Convolutional Neural Network," *Proceedings - International Conference on Pattern Recognition*, vol. 2018-August, pp. 165–170, Nov. 2018, doi: 10.1109/ICPR.2018.8545435.
[56] M. O. Mario, "Human activity recognition based on single sensor square HV acceleration images and convolutional neural networks," *IEEE Sens J*, vol. 19, no. 4, pp. 1487–1498, Feb. 2019, doi: 10.1109/JSEN.2018.2882943.
[57] S. M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using Convolutional Neural Network," *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*, pp. 131–134, Mar. 2017, doi: 10.1109/BIGCOMP.2017.7881728.
[58] A. Ignatov, "Real-time human activity recognition from accelerometer data using Convolutional Neural Networks," *Appl Soft Comput*, vol. 62, pp. 915–922, Jan. 2018, doi: 10.1016/J.ASOC.2017.09.027.
[59] B. Zhou, J. Yang, and Q. Li, "Smartphone-Based Activity Recognition for Indoor Localization Using a Convolutional Neural Network," *Sensors 2019, Vol. 19, Page 621*, vol. 19, no. 3, p. 621, Feb. 2019, doi: 10.3390/S19030621.

[60] S. Matsui, N. Inoue, Y. Akagi, G. Nagino, and K. Shinoda, "User adaptation of convolutional neural network for human activity recognition," *25th European Signal Processing Conference, EUSIPCO 2017*, vol. 2017-January, pp. 753–757, Oct. 2017, doi: 10.23919/EUSIPCO.2017.8081308.

[61] T. Zebin, M. Sperrin, N. Peek, and A. J. Casson, "Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2018-July, pp. 1–4, Oct. 2018, doi: 10.1109/EMBC.2018.8513115.

[62] F. Hernández, L. F. Suárez, J. Villamizar, and M. Altuve, "Human Activity Recognition on Smartphones Using a Bidirectional LSTM Network," *2019 22nd Symposium on Image, Signal Processing and Artificial Vision, STSIVA 2019 - Conference Proceedings*, Apr. 2019, doi: 10.1109/STSIVA.2019.8730249.

[63] S. Yu and L. Qin, "Human activity recognition with smartphone inertial sensors using bidir-LSTM networks," *Proceedings - 2018 3rd International Conference on Mechanical, Control and Computer Engineering, ICMCCE 2018*, pp. 219–224, Nov. 2018, doi: 10.1109/ICMCCE.2018.00052.

[64] W. H. Chen, C. A. Betancourt Baca, and C. H. Tou, "LSTM-RNNs combined with scene information for human activity recognition," *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services, Healthcom 2017*, vol. 2017-December, pp. 1–6, Dec. 2017, doi: 10.1109/HEALTHCOM.2017.8210846.

[65] D. Tao, Y. Wen, and R. Hong, "Multicolumn Bidirectional Long Short-Term Memory for Mobile Devices-Based Human Activity Recognition," *IEEE Internet Things J*, vol. 3, no. 6, pp. 1124–1134, Dec. 2016, doi: 10.1109/JIOT.2016.2561962.

[66] M. Milenkoski, K. Trivodaliev, S. Kalajdziski, M. Jovanov, and B. R. Stojkoska, "Real time human activity recognition on smartphones using LSTM networks," *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, pp. 1126–1131, Jun. 2018, doi: 10.23919/MIPRO.2018.8400205.

[67] G. Ogbuabor and R. La, "Human activity recognition for healthcare using smartphones," *ACM International Conference Proceeding Series*, pp. 41–46, Feb. 2018, doi: 10.1145/3195106.3195157.

[68] R. A. Voicu, C. Dobre, L. Bajenaru, and R. I. Ciobanu, "Human Physical Activity Recognition Using Smartphone Sensors," *Sensors (Basel)*, vol. 19, no. 3, Feb. 2019, doi: 10.3390/S19030458.

[69] I. M. Pires *et al.*, "Pattern Recognition Techniques for the Identification of Activities of Daily Living Using a Mobile Device Accelerometer," *Electronics 2020, Vol. 9, Page 509*, vol. 9, no. 3, p. 509, Mar. 2020, doi: 10.3390/ELECTRONICS9030509.

[70] N. Twomey *et al.*, "A Comprehensive Study of Activity Recognition Using Accelerometers," *Informatics 2018, Vol. 5, Page 27*, vol. 5, no. 2, p. 27, May 2018, doi: 10.3390/INFORMATICS5020027.

[71] G. Acampora, P. Foggia, A. Saggese, and M. Vento, "Combining neural networks and fuzzy systems for human behavior understanding," *Proceedings - 2012 IEEE 9th International Conference on Advanced Video and Signal-Based Surveillance, AVSS 2012*, pp. 88–93, 2012, doi: 10.1109/AVSS.2012.25.

[72] M. Inoue, S. Inoue, and T. Nishida, "Deep Recurrent Neural Network for Mobile Human Activity Recognition with High Throughput," *Artif Life Robot*, vol. 23, no. 2, pp. 173–185, Nov. 2016, doi: 10.48550/arxiv.1611.03607.

[73] G. M. Weiss, J. W. Lockhart, T. T. Pulickal, P. T. McHugh, I. H. Ronan, and J. L. Timko, "Actitracker: A smartphone-based activity recognition system for improving health and well-being," *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pp. 682–688, Dec. 2016, doi: 10.1109/DSAA.2016.89.

[74] C. Dobbins and R. Rawassizadeh, "Towards Clustering of Mobile and Smartwatch Accelerometer Data for Physical Activity Recognition," *Informatics 2018, Vol. 5, Page 29*, vol. 5, no. 2, p. 29, Jun. 2018, doi: 10.3390/INFORMATICS5020029.

[75] Y. Lu, Y. Wei, L. Liu, J. Zhong, L. Sun, and Y. Liu, "Towards unsupervised physical activity recognition using smartphone accelerometers," *Multimed Tools Appl*, vol. 76, no. 8, pp. 10701–10719, Apr. 2017, doi: 10.1007/S11042-015-3188-Y/TABLES/4.

[76] Ó. D. Lara, A. J. Prez, M. A. Labrador, and J. D. Posada, "Centinela: A human activity recognition system based on acceleration and vital sign data," *Pervasive Mob Comput*, vol. 8, no. 5, pp. 717–729, Oct. 2012, doi: 10.1016/J.PMCJ.2011.06.004.

[77] T. Sztyler, H. Stuckenschmidt, and W. Petrich, "Position-aware activity recognition with wearable devices," *Pervasive Mob Comput*, vol. 38, pp. 281–295, Jul. 2017, doi: 10.1016/J.PMCJ.2017.01.008.

[78] T. Yu, J. Chen, N. Yan, and X. Liu, "A Multi-Layer Parallel LSTM Network for Human Activity Recognition with Smartphone Sensors," *2018 10th International Conference on Wireless Communications and Signal Processing, WCSP 2018*, Nov. 2018, doi: 10.1109/WCSP.2018.8555945.

[79] Z. Chen, C. Jiang, and L. Xie, "A Novel Ensemble ELM for Human Activity Recognition Using Smartphone Sensors," *IEEE Trans Industr Inform*, vol. 15, no. 5, pp. 2691–2699, May 2019, doi: 10.1109/TII.2018.2869843.

[80] Q. Zhu, Z. Chen, and Y. C. Soh, "A Novel Semisupervised Deep Learning Method for Human Activity Recognition," *IEEE Trans Industr Inform*, vol. 15, no. 7, pp. 3821–3830, Jul. 2019, doi: 10.1109/TII.2018.2889315.

[81] W. Qi, H. Su, C. Yang, G. Ferrigno, E. De Momi, and A. Aliverti, "A Fast and Robust Deep Convolutional Neural Networks for Complex Human Activity Recognition Using Smartphone," *Sensors 2019, Vol. 19, Page 3731*, vol. 19, no. 17, p. 3731, Aug. 2019, doi: 10.3390/S19173731.

[82] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, pp. 307–313, Apr. 2018, doi: 10.1016/J.FUTURE.2017.11.029.

[83] M. Gholamrezaii and S. AlModarresi, "A time-efficient convolutional neural network model in human activity recognition," *Multimed Tools Appl*, vol. 80, no. 13, pp. 19361–19376, May 2021, doi: 10.1007/S11042-020-10435-1/TABLES/8.

[84] T. Zebin, P. J. Scully, N. Peek, A. J. Casson, and K. B. Ozanyan, "Design and Implementation of a Convolutional Neural Network on an Edge Computing Smartphone for Human Activity Recognition," *IEEE Access*, vol. 7, pp. 133509–133520, 2019, doi: 10.1109/ACCESS.2019.2941836.

[85] B. A. Mohammed Hashim and R. Amutha, "Human activity recognition based on smartphone using fast feature dimensionality reduction technique," *J Ambient Intell Humaniz Comput*, vol. 12, no. 2, pp. 2365–2374, Feb. 2021, doi: 10.1007/S12652-020-02351-X/TABLES/10.

[86] S. Deep and X. Zheng, "Hybrid Model Featuring CNN and LSTM Architecture for Human Activity Recognition on Smartphone Sensor Data," *Proceedings - 2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2019*, pp. 259–264, Dec. 2019, doi: 10.1109/PDCAT46702.2019.00055.

[87] V. Ghate and C. Sweetlin Hemalatha, "Hybrid deep learning approaches for smartphone sensor-based human activity recognition," *Multimed Tools Appl*, vol. 80, no. 28–29, pp. 35585–35604, Nov. 2021, doi: 10.1007/S11042-020-10478-4/TABLES/4.

[88] R. Jansi and R. Amutha, "Sparse representation based classification scheme for human activity recognition using smartphones," *Multimed Tools Appl*, vol. 78, no. 8, pp. 11027–11045, Apr. 2019, doi: 10.1007/S11042-018-6662-5/FIGURES/4.

[89] M. Ullah, H. Ullah, S. D. Khan, and F. A. Cheikh, "Stacked Lstm Network for Human Activity Recognition Using Smartphone Data," *Proceedings - European Workshop on Visual Information Processing, EUVIP*, vol. 2019-October, pp. 175–180, Oct. 2019, doi: 10.1109/EUVIP47703.2019.8946180.

[90] V. Gattulli, D. Impedovo, G. Pirlo, and L. Sarcinella, "Human Activity Recognition for the Identification of Bullying and Cyberbullying Using Smartphone Sensors," *Electronics (Basel)*, vol. 12, no. 2, p. 261, Jan. 2023, doi: 10.3390/ELECTRONICS12020261.

[91] V. N. Convertini, V. Gattulli, D. Impedovo, and G. Terrone, "Classification bullying/cyberbullying through smartphone sensor and a questionnaire application," *Multimed Tools Appl*, vol. 83, no. 17, pp. 51291–51320, May 2024, doi: 10.1007/S11042-023-17609-7/FIGURES/7.

[92] D. Garcia-Gonzalez, D. Rivero, E. Fernandez-Blanco, and M. R. Luaces, "Deep learning models for real-life human activity recognition from smartphone sensor data," *Internet of Things*, vol. 24, p. 100925, Dec. 2023, doi: 10.1016/J.IOT.2023.100925.

[93] A. W. Sardar, F. Ullah, J. Bacha, J. Khan, F. Ali, and S. Lee, "Mobile sensors based platform of Human Physical Activities Recognition for COVID-19 spread minimization," *Comput Biol Med*, vol. 146, Jul. 2022, doi: 10.1016/J.COMPBIOMED.2022.105662.

[94] T. H. Tan, J. Y. Wu, S. H. Liu, and M. Gochoo, "Human Activity Recognition Using an Ensemble Learning Algorithm with Smartphone Sensor Data," *Electronics 2022, Vol. 11, Page 322*, vol. 11, no. 3, p. 322, Jan. 2022, doi: 10.3390/ELECTRONICS11030322.

[95] C. A. Ronao and S. B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Syst Appl*, vol. 59, pp. 235–244, Oct. 2016, doi: 10.1016/J.ESWA.2016.04.032.

[96] L. Zhang, X. Wu, and D. Luo, "Recognizing human activities from raw accelerometer data using deep neural networks," *Proceedings - 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA 2015*, pp. 865–870, Mar. 2016, doi: 10.1109/ICMLA.2015.48.

[97] C. A. Ronao and S. B. Cho, "Human activity recognition using smartphone sensors with two-stage continuous hidden markov models," *2014 10th International Conference on Natural Computation, ICNC 2014*, pp. 681–686, 2014, doi: 10.1109/ICNC.2014.6975918.

[98] S. Ha, J. M. Yun, and S. Choi, "Multi-modal Convolutional Neural Networks for Activity Recognition," *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 3017–3022, Jan. 2016, doi: 10.1109/SMC.2015.525.

[99] L. Zhang, X. Wu, and D. Luo, "Real-Time Activity Recognition on Smartphones Using Deep Neural Networks," in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, 2015, pp. 1236–1242. doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.224.

[100] M. Bernaś, B. Płaczek, and M. Lewandowski, "Ensemble of RNN Classifiers for Activity Detection Using a Smartphone and Supporting Nodes," *Sensors 2022, Vol. 22, Page 9451*, vol. 22, no. 23, p. 9451, Dec. 2022, doi: 10.3390/S22239451.

[101] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," *Proceedings of the 21th International European Symposium on Artificial Neural Networks*, 2013, Accessed: Jan. 10, 2023. [Online]. Available: http://www.i6doc.com/en/livre/?GCOI=28001100131010.

[102] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster, "Wearable activity tracking in car manufacturing," *IEEE Pervasive Comput*, vol. 7, no. 2, pp. 42–50, Apr. 2008, doi: 10.1109/MPRV.2008.40.

[103] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, Mar. 2011, doi: 10.1145/1964897.1964918.

[104] D. Roggen *et al.*, "Walk-through the OPPORTUNITY dataset for activity recognition in sensor rich environments," *Project: OPPORTUNITY (Activity and Context Recognition with Opportunistic Sensor Configurations)*, Accessed: Jan. 10, 2023. [Online]. Available: https://www.researchgate.net/publication/229001920_Walk-through_the_OPPORTUNITY_dataset_for_activity_recognition_in_sensor_rich_environments

[105] A. Stisen *et al.*, "Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition," *SenSys 2015 - Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 127–140, Nov. 2015, doi: 10.1145/2809695.2809718.

[106] D. P. Ong, E. J. L. San Pedro, M. E. M. Valenzuela, and N. M. C. Tiglao, "BrainSmart: Ambient Assisted Living System Smartphone App Prototype for Parkinson's Disease Patients," *GHTC 2018 - IEEE Global Humanitarian Technology Conference, Proceedings*, Jan. 2019, doi: 10.1109/GHTC.2018.8601563.

[107] M. Janidarmian, A. R. Fekr, K. Radecka, and Z. Zilic, "A Comprehensive Analysis on Wearable Acceleration Sensors in Human Activity Recognition," *Sensors 2017, Vol. 17, Page 529*, vol. 17, no. 3, p. 529, Mar. 2017, doi: 10.3390/S17030529.

[108] F. Balducci, D. Impedovo, N. Macchiarulo, and G. Pirlo, "Affective states recognition through touch dynamics," *Multimed Tools Appl*, vol. 79, no. 47–48, pp. 35909–35926, Dec. 2020, doi: 10.1007/S11042-020-09146-4.

[109] B. E. Palladino, A. Nocentini, and E. Menesini, "Evidence-based intervention against bullying and cyberbullying: Evaluation of the NoTrap! program in two independent trials," *Aggress Behav*, vol. 42, no. 2, pp. 194–206, Mar. 2016, doi: 10.1002/AB.21636.

[110] B. E. Palladino, A. Nocentini, and E. Menesini, "Psychometric properties of the Florence CyberBullying-CyberVictimization Scales," *Cyberpsychol Behav Soc Netw*, vol. 18, no. 2, pp. 112–119, Feb. 2015, doi: 10.1089/CYBER.2014.0366.

[111] O. Lopez-Fernandez *et al.*, "Cross-Cultural Validation of the Compulsive Internet Use Scale in Four Forms and Eight Languages," *Cyberpsychol Behav Soc Netw*, vol. 22, no. 7, pp. 451–464, Jul. 2019, doi: 10.1089/CYBER.2018.0731.

[112] E. Menesini, P. Calussi, and A. Nocentini, "Cyberbullying and Traditional Bullying: Unique, Additive, and Synergistic Effects on Psychological Health Symptoms," *Cyberbullying in the Global Playground: Research from International Perspectives*, pp. 245–262, Jul. 2012, doi: 10.1002/9781119954484.CH12.

[113] V. Gattulli, D. Impedovo, T. Palmisano, and L. Sarcinella, "Fixed Tasks for continuous authentication via smartphone," *12 th International Conference on Pattern Recognition Applications and Methods* , 2023.

[114] P. Lamb, A. Millar, and R. Fuentes, "Swipe Dynamics as a Means of Authentication: Results From a Bayesian Unsupervised Approach".

[115] P. Vaishnav, M. Kaushik, and L. Raja, "DESIGN AN ALGORITHM FOR CONTINUOUS AUTHENTICATION ON SMARTPHONE THROUGH KEYSTROKE DYNAMICS AND TOUCH DYNAMICS," *Indian Journal of Computer Science and Engineering*, vol. 13, no. 2, pp. 444–455, Mar. 2022, doi: 10.21817/INDJCSE/2022/V13I2/221302111.

[116] Y. Ku and L. H. Park, "Draw It As Shown: Behavioral Pattern Lock for Mobile User Authentication", doi: 10.1109/ACCESS.2019.2918647.

[117] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, 2013, doi: 10.1109/TIFS.2012.2225048.

[118] M. Levi, I. Hazan, N. Agmon, and S. Eden, "Behavioral embedding for continuous user verification in global settings," *Comput Secur*, vol. 119, p. 102716, Aug. 2022, doi: 10.1016/J.COSE.2022.102716.

[119] O. D. Incel *et al.*, "DAKOTA: Sensor and Touch Screen-Based Continuous Authentication on a Mobile Banking Application," *IEEE Access*, vol. 9, pp. 38943–38960, 2021, doi: 10.1109/ACCESS.2021.3063424.

[120] P. M. A. B. Estrela, R. de O. Albuquerque, D. M. Amaral, W. F. Giozza, and R. T. de Sousa Júnior, "A framework for continuous authentication based on touch dynamics biometrics for mobile banking applications," *Sensors*, vol. 21, no. 12, Jun. 2021, doi: 10.3390/S21124212.

[121] A. Z. Zaidi, C. Y. Chong, Z. Jin, R. Parthiban, and A. S. Sadiq, "Touch-based continuous mobile device authentication: State-of-the-art, challenges and opportunities," *Journal of Network and Computer Applications*, vol. 191, Oct. 2021, doi: 10.1016/J.JNCA.2021.103162.

[122] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, Jul. 2012, doi: 10.1109/TIFS.2012.2225048.
[123] O. D. Incel *et al.*, "DAKOTA: Sensor and Touch Screen-Based Continuous Authentication on a Mobile Banking Application," *IEEE Access*, vol. 9, pp. 38943–38960, 2021, doi: 10.1109/ACCESS.2021.3063424.
[124] M. Smith-Creasey and M. Rajarajan, "A novel word-independent gesture-typing continuous authentication scheme for mobile devices," *Comput Secur*, vol. 83, pp. 140–150, Jun. 2019, doi: 10.1016/J.COSE.2019.02.001.
[125] O. D. Incel *et al.*, "DAKOTA: Sensor and Touch Screen-Based Continuous Authentication on a Mobile Banking Application," *IEEE Access*, vol. 9, no. 99, pp. 38943–38960, 2021, doi: 10.1109/ACCESS.2021.3063424.
[126] D. Reichinger, E. Sonnleitner, M. Kurz, and R. Duque, "Continuous Mobile User Authentication Using Combined Biometric Traits," 2021, doi: 10.3390/app112411756.
[127] M. Zhao, J. Chen, and Y. Li, "A Review of Anomaly Detection Techniques Based on Nearest Neighbor," *Proceedings of the 2018 International Conference on Computer Modeling, Simulation and Algorithm (CMSA 2018)*, vol. 151, Jul. 2018, doi: 10.2991/CMSA-18.2018.65.
[128] C. C. Aggarwal, "Probabilistic and Statistical Models for Outlier Detection," *Outlier Analysis*, pp. 35–64, 2017, doi: 10.1007/978-3-319-47578-3_2.
[129] N. R. Prasad, S. Almanza-Garcia, and T. T. Lu, "Anomaly detection," *Computers, Materials and Continua*, vol. 14, no. 1, pp. 1–22, 2009, doi: 10.1145/1541880.1541882.
[130] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, Jan. 2016, doi: 10.1016/J.JNCA.2015.11.016.
[131] S. Mokhtari, K. K. Yen, S. Mokhtari, and K. K. Yen, "Measurement data intrusion detection in industrial control systems based on unsupervised learning," *Applied Computing and Intelligence 2021 1:61*, vol. 1, no. 1, pp. 61–74, 2021, doi: 10.3934/ACI.2021004.
[132] A. E. Minarno, W. A. Kusuma, H. Wibowo, D. R. Akbi, and N. Jawas, "Single Triaxial Accelerometer-Gyroscope Classification for Human Activity Recognition," *2020 8th International Conference on Information and Communication Technology, ICoICT 2020*, Jun. 2020, doi: 10.1109/ICOICT49345.2020.9166329.
[133] H. Cho and S. M. Yoon, "Applying singular value decomposition on accelerometer data for 1D convolutional neural network based fall detection," *Electron Lett*, vol. 55, no. 6, pp. 320–322, Mar. 2019, doi: 10.1049/EL.2018.6117.
[134] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection," *Procedia Comput Sci*, vol. 110, pp. 32–39, 2017, doi: 10.1016/J.PROCS.2017.06.110.
[135] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "SisFall: A Fall and Movement Dataset," *Sensors (Basel)*, vol. 17, no. 1, Jan. 2017, doi: 10.3390/S17010198.
[136] F. Concone, S. Gaglio, G. Lo Re, and M. Morana, "Smartphone data analysis for human activity recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10640 LNAI, pp. 58–71, 2017, doi: 10.1007/978-3-319-70169-1_5/COVER.
[137] S. Gupta, "Deep learning based human activity recognition (HAR) using wearable sensor data," *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100046, Nov. 2021, doi: 10.1016/J.JJIMEI.2021.100046.
[138] G. M. Weiss, K. Yoneda, and T. Hayajneh, "Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living," *IEEE Access*, vol. 7, pp. 133190–133202, 2019, doi: 10.1109/ACCESS.2019.2940729.
[139] K. Ismail and K. Özacar, "Human activity recognition based on smartphone sensor data using CNN," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 44, no. 4/W3, pp. 263–265, Nov. 2020, doi: 10.5194/ISPRS-ARCHIVES-XLIV-4-W3-2020-263-2020.
[140] M. B. Dehkordi, A. Zaraki, and R. Setchi, "Feature extraction and feature selection in smartphone-based activity recognition," *Procedia Comput Sci*, vol. 176, pp. 2655–2664, Jan. 2020, doi: 10.1016/J.PROCS.2020.09.301.
[141] L. Ye, H. Ferdinando, T. Seppänen, and E. Alasaarela, "Physical Violence Detection for Preventing School Bullying," *Advances in Artificial Intelligence*, vol. 2014, pp. 1–9, Aug. 2014, doi: 10.1155/2014/740358.
[142] L. Ye, H. Ferdinando, T. Seppanen, T. Huuki, and E. Alasaarela, "An instance-based physical violence detection algorithm for school bullying prevention," *IWCMC 2015 - 11th International Wireless Communications and Mobile Computing Conference*, pp. 1384–1388, Oct. 2015, doi: 10.1109/IWCMC.2015.7289284.
[143] L. Ye, P. Wang, L. Wang, H. Ferdinando, T. Seppänen, and E. Alasaarela, "A Combined Motion-Audio School Bullying Detection Algorithm," *Intern J Pattern Recognit Artif Intell*, vol. 32, no. 12, Dec. 2018, doi: 10.1142/S0218001418500465.
[144] L. Ye, J. Shi, H. Ferdinando, T. Seppänen, and E. Alasaarela, "A Multi-sensor School Violence Detecting Method Based on Improved Relief-F and D-S Algorithms," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1655–1662, Oct. 2020, doi: 10.1007/S11036-020-01575-7.
[145] Z. Zihan and Z. Zhanfeng, "Campus bullying detection based on motion recognition and speech emotion recognition," *J Phys Conf Ser*, vol. 1314, no. 1, Nov. 2019, doi: 10.1088/1742-6596/1314/1/012150.

[146] M. I. Amara, A. Akkouche, E. Boutellaa, and H. Tayakout, "A Smartphone Application for Fall Detection Using Accelerometer and ConvLSTM Network," *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-Being, IHSH 2020*, pp. 92–96, Feb. 2021, doi: 10.1109/IHSH51661.2021.9378743.
[147] -L ; Nguyen *et al.*, "A Novel Feature Set Extraction Based on Accelerometer Sensor Data for Improving the Fall Detection System," *Electronics 2022, Vol. 11, Page 1030*, vol. 11, no. 7, p. 1030, Mar. 2022, doi: 10.3390/ELECTRONICS11071030.
[148] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1331–1338, 2011, doi: 10.1109/ICCV.2011.6126386.
[149] M. Z. Uddin and A. Soylu, "Human activity recognition using wearable sensors, discriminant analysis, and long short-term memory-based neural structured learning," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/S41598-021-95947-Y.
[150] D. Impedovo, A. Longo, T. Palmisano, L. Sarcinella, and D. Veneto, "An investigation on voice mimicry attacks to a speaker recognition system," *CEUR Workshop Proc*, vol. 3260, pp. 114–123, 2022.
[151] A. Sharaff, N. K. Nagwani, and A. Dhadse, "Comparative Study of Classification Algorithms for Spam Email Detection," *Emerging Research in Computing, Information, Communication and Applications*, pp. 237–244, 2016, doi: 10.1007/978-81-322-2553-9_23.
[152] N. K. Nagwani and A. Sharaff, "SMS spam filtering and thread identification using bi-level text classification and clustering techniques," *J Inf Sci*, vol. 43, no. 1, pp. 75–87, Feb. 2017, doi: 10.1177/0165551515616310/ASSET/IMAGES/LARGE/10.1177_0165551515616310-FIG2.JPEG.
[153] L. Minh Dang, K. Min, H. Wang, M. Jalil Piran, C. Hee Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognit*, vol. 108, p. 107561, Dec. 2020, doi: 10.1016/J.PATCOG.2020.107561.
[154] F. Luo, S. Khan, Y. Huang, and K. Wu, "Activity-based person identification using multimodal wearable sensor data," *IEEE Internet Things J*, 2022, doi: 10.1109/JIOT.2022.3209084.
[155] M. Straczkiewicz, E. J. Huang, and J. P. Onnela, "A 'one-size-fits-most' walking recognition method for smartphones, smartwatches, and wearable accelerometers," *npj Digital Medicine 2023 6:1*, vol. 6, no. 1, pp. 1–16, Feb. 2023, doi: 10.1038/s41746-022-00745-z.
[156] Q. Wang *et al.*, "A smartphone-based zero-effort method for mitigating epidemic propagation," *EURASIP J Adv Signal Process*, vol. 2023, no. 1, Feb. 2023, doi: 10.1186/S13634-023-00984-6.
[157] M. Hu, K. Zhang, R. You, and B. Tu, "AuthConFormer: Sensor-based Continuous Authentication of Smartphone Users Using A Convolutional Transformer," *Comput Secur*, p. 103122, Apr. 2023, doi: 10.1016/J.COSE.2023.103122.
[158] P. K. Rayani and S. Changder, "Sensor-based continuous user authentication on smartphone through machine learning," *Microprocess Microsyst*, vol. 96, Feb. 2023, doi: 10.1016/J.MICPRO.2022.104750.
[159] S. Alzahrani, J. Alderaan, D. Alatawi, and B. Alotaibi, "Continuous Mobile User Authentication Using a Hybrid CNN-Bi-LSTM Approach," *Computers, Materials and Continua*, vol. 75, no. 1, pp. 651–667, 2023, doi: 10.32604/CMC.2023.035173.
[160] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "A Survey on Touch Dynamics Authentication in Mobile Devices," *Comput Secur*, vol. 59, pp. 210–235, 2016.
[161] D. Impedovo, A. Longo, T. Palmisano, L. Sarcinella, and D. Veneto, "An investigation on voice mimicry attacks to a speaker recognition system," *ITASEC'22: Italian Conference on Cybersecurity*, 2022.
[162] M. Nerini, E. Favarelli, and M. Chiani, "Augmented PIN Authentication through Behavioral Biometrics," *Sensors*, vol. 22, no. 13, Jul. 2022, doi: 10.3390/S22134857.
[163] P. S. Teh, N. Zhang, S. Y. Tan, Q. Shi, W. H. Khoh, and R. Nawaz, "Strengthen user authentication on mobile devices by using user's touch dynamics pattern," *J Ambient Intell Humaniz Comput*, vol. 11, no. 10, pp. 4019–4039, Oct. 2020, doi: 10.1007/S12652-019-01654-Y/FIGURES/12.
[164] R. Zaccagnino, C. Capo, A. Guarino, N. Lettieri, and D. Malandrino, "Techno-regulation and intelligent safeguards: Analysis of touch gestures for online child protection," *Multimed Tools Appl*, vol. 80, no. 10, pp. 15803–15824, Apr. 2021, doi: 10.1007/S11042-020-10446-Y.
[165] D. Ozkul, "Children's mobile communicative practices and locational privacy," *Journal of Computer-Mediated Communication*, vol. 27, no. 5, Sep. 2022, doi: 10.1093/JCMC/ZMAC015.
[166] V. Gattulli, D. Impedovo, and L. Sarcinella, "Anomaly Detection using smartphone Sensors for a Bullying Detection ," *WorldCist23*, 2023.
[167] B. E. Palladino, A. Nocentini, and E. Menesini, "Psychometric properties of the florence cyberbullying-cybervictimization scales," *Cyberpsychol Behav Soc Netw*, vol. 18, no. 2, pp. 112–119, Feb. 2015, doi: 10.1089/CYBER.2014.0366.
[168] J. Makhoul, "A Fast Cosine Transform in One and Two Dimensions," *IEEE Trans Acoust*, vol. 28, no. 1, pp. 27–34, 1980, doi: 10.1109/TASSP.1980.1163351.
[169] M. Ptaszynski, P. Dybala, T. Matsuba, F. Masui, R. Rzepka, and K. Araki, "Machine Learning and Affect Analysis Against Cyber-Bullying," *Research Project on Automatic Cyberbullying Detection*, 2010.

[170] M. Chimienti, I. Danzi, D. Impedovo, G. Pirlo, G. Semeraro, and D. Veneto, “MIRROR: Methodological Innovation to Remodel the Electric Loads to Reduce Economic OR Environmental Impact of User,” *Algorithms 2023, Vol. 16, Page 1*, vol. 16, no. 1, p. 1, Dec. 2022, doi: 10.3390/A16010001.
[171] V. Dentamaro, D. Impedovo, and G. Pirlo, “Fall detection by human pose estimation and kinematic theory,” *Proceedings - International Conference on Pattern Recognition*, pp. 2328–2335, 2020, doi: 10.1109/ICPR48806.2021.9413331.
[172] L. Minh Dang, K. Min, H. Wang, M. Jalil Piran, C. Hee Lee, and H. Moon, “Sensor-based and vision-based human activity recognition: A comprehensive survey,” *Pattern Recognit*, vol. 108, p. 107561, Dec. 2020, doi: 10.1016/J.PATCOG.2020.107561.
[173] Y. Zhang, L. Wang, H. Chen, A. Tian, S. Zhou, and Y. Guo, “IF-ConvTransformer: A Framework for Human Activity Recognition Using IMU Fusion and ConvTransformer,” *Proc ACM Interact Mob Wearable Ubiquitous Technol*, vol. 6, no. 2, p. 88, Jul. 2022, doi: 10.1145/3534584.
[174] B. Thomas, M. L. Lu, R. Jha, and J. Bertrand, “Machine Learning for Detection and Risk Assessment of Lifting Action,” *IEEE Trans Hum Mach Syst*, vol. 52, no. 6, pp. 1196–1204, Dec. 2022, doi: 10.1109/THMS.2022.3212666.
[175] K. Twyman, C. Saylor, L. A. Taylor, and C. Comeaux, “Comparing children and adolescents engaged in cyberbullying to matched peers,” *Cyberpsychol Behav Soc Netw*, vol. 13, no. 2, pp. 195–199, Apr. 2010, doi: 10.1089/CYBER.2009.0137.
[176] V. Gattulli, D. Impedovo, G. Pirlo, and L. Sarcinella, “Human Activity Recognition for the Identification of Bullying and Cyberbullying Using Smartphone Sensors,” *Electronics 2023, Vol. 12, Page 261*, vol. 12, no. 2, p. 261, Jan. 2023, doi: 10.3390/ELECTRONICS12020261.
[177] A. E. Minarno, W. A. Kusuma, H. Wibowo, D. R. Akbi, and N. Jawas, “Single Triaxial Accelerometer-Gyroscope Classification for Human Activity Recognition,” *2020 8th International Conference on Information and Communication Technology, ICoICT 2020*, Jun. 2020, doi: 10.1109/ICOICT49345.2020.9166329.
[178] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, “UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection,” in *Procedia Computer Science*, Elsevier B.V., 2017, pp. 32–39. doi: 10.1016/j.procs.2017.06.110.
[179] D. Micucci, M. Mobilio, and P. Napoletano, “UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones,” *Applied Sciences (Switzerland)*, vol. 7, no. 10, Oct. 2017, doi: 10.3390/app7101101.
[180] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, “SisFall: A fall and movement dataset,” *Sensors (Switzerland)*, vol. 17, no. 1, Jan. 2017, doi: 10.3390/s17010198.
[181] M. B. Dehkordi, A. Zaraki, and R. Setchi, “Feature extraction and feature selection in smartphone-based activity recognition,” in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 2655–2664. doi: 10.1016/j.procs.2020.09.301.
[182] L. Minh Dang, K. Min, H. Wang, M. Jalil Piran, C. Hee Lee, and H. Moon, “Sensor-based and vision-based human activity recognition: A comprehensive survey,” *Pattern Recognit*, vol. 108, Dec. 2020, doi: 10.1016/j.patcog.2020.107561.
[183] Z. Zihan and Z. Zhanfeng, “Campus bullying detection based on motion recognition and speech emotion recognition,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Nov. 2019. doi: 10.1088/1742-6596/1314/1/012150.
[184] L. Ye, P. Wang, L. Wang, H. Ferdinando, T. Seppänen, and E. Alasaarela, “A Combined Motion-Audio School Bullying Detection Algorithm,” *Intern J Pattern Recognit Artif Intell*, vol. 32, no. 12, Dec. 2018, doi: 10.1142/S0218001418500465.
[185] M. I. Amara, A. Akkouche, E. Boutellaa, and H. Tayakout, “A Smartphone Application for Fall Detection Using Accelerometer and ConvLSTM Network,” in *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, IEEE, Feb. 2021, pp. 92–96. doi: 10.1109/IHSH51661.2021.9378743.
[186] F. Castro, V. Dentamaro, V. Gattulli, and D. Impedovo, “Fall Detection with LSTM and Attention Mechanism,” *WAMWB 2023 Advances of Mobile and Wearable Biometrics 2023*, 2023.
[187] V. Gattulli, D. Impedovo, and L. Sarcinella, “Anomaly Detection using smartphone Sensors for a Bullying Detection,” *WAITT 2023 - 1st Workshop on Artificial Intelligence for Technology Transfer*, 2023.
[188] F. Castro, D. Impedovo, and G. Pirlo, “A Medical Image Encryption Scheme for Secure Fingerprint-Based Authenticated Transmission,” *Applied Sciences 2023, Vol. 13, Page 6099*, vol. 13, no. 10, p. 6099, May 2023, doi: 10.3390/APP13106099.
[189] V. Gattulli, D. Impedovo, G. Pirlo, and F. Volpe, “Touch events and human activities for continuous authentication via smartphone,” *Scientific Reports 2023 13:1*, vol. 13, no. 1, pp. 1–7, Jun. 2023, doi: 10.1038/s41598-023-36780-3.
[190] F. B. Shaikh, M. Rehman, and A. Amin, “Cyberbullying: A Systematic Literature Review to Identify the Factors Impelling University Students towards Cyberbullying,” *IEEE Access*, vol. 8, pp. 148031–148051, 2020, doi: 10.1109/ACCESS.2020.3015669.

[191] Md. M. Islam, S. Nooruddin, F. Karray, and G. Muhammad, "Human Activity Recognition Using Tools of Convolutional Neural Networks: A State of the Art Review, Data Sets, Challenges and Future Prospects," *Comput Biol Med*, vol. 149, Feb. 2022, doi: 10.1016/j.compbiomed.2022.106060.
[192] J. Zhu, P. Wu, X. Wang, and J. Zhang, "SenSec: Mobile security through passive sensing," *2013 International Conference on Computing, Networking and Communications, ICNC 2013*, pp. 1128–1133, 2013, doi: 10.1109/ICCNC.2013.6504251.
[193] W.-H. Lee and R. Lee, "Multi-sensor authentication to improve smartphone security," in *International Conference on Information Systems Security and Privacy*, Mar. 2017. Accessed: Dec. 15, 2022. [Online]. Available: https://www.researchgate.net/publication/282785492_Multi-sensor_Authentication_to_Improve_Smartphone_Security
[194] S. Amini, S. Gupte, V. Noroozi, P. S. Yu, A. Pande, and C. Kanich, "Deepauth: A framework for continuous user re-authentication in mobile apps," *International Conference on Information and Knowledge Management, Proceedings*, pp. 2027–2036, Oct. 2018, doi: 10.1145/3269206.3272034.
[195] M. Ehatisham-ul-Haq *et al.*, "Authentication of Smartphone Users Based on Activity Recognition and Mobile Sensing," *Sensors 2017, Vol. 17, Page 2043*, vol. 17, no. 9, p. 2043, Sep. 2017, doi: 10.3390/S17092043.
[196] M. Abuhamad, T. Abuhmed, D. Mohaisen, and D. Nyang, "AUToSen: Deep-learning-based implicit continuous authentication using smartphone sensors," *IEEE Internet Things J*, vol. 7, no. 6, pp. 5008–5020, Jun. 2020, doi: 10.1109/JIOT.2020.2975779.
[197] S. Mekruksavanich and A. Jitpattanakul, "Deep Learning Approaches for Continuous Authentication Based on Activity Patterns Using Mobile Sensing," *Sensors 2021, Vol. 21, Page 7519*, vol. 21, no. 22, p. 7519, Nov. 2021, doi: 10.3390/S21227519.
[198] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," *IEEE 4th International Conference on Biometrics: Theory, Applications and Systems, BTAS 2010*, 2010, doi: 10.1109/BTAS.2010.5634532.
[199] M. P. Centeno, A. van Moorsel, and S. Castruccio, "Smartphone continuous authentication using deep learning autoencoders," *Proceedings - 2017 15th Annual Conference on Privacy, Security and Trust, PST 2017*, pp. 147–155, Sep. 2018, doi: 10.1109/PST.2017.00026.
[200] Y. Li, H. Hu, Z. Zhu, and G. Zhou, "SCANet: Sensor-based Continuous Authentication with Two-stream Convolutional Neural Networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 16, no. 3, Jul. 2020, doi: 10.1145/3397179.
[201] A. Garbuz, A. Epishkina, and K. Kogos, "Continuous authentication of smartphone users via swipes and taps analysis," *Proceedings of the 2019 European Intelligence and Security Informatics Conference, EISIC 2019*, pp. 48–53, Nov. 2019, doi: 10.1109/EISIC49498.2019.9108780.
[202] C. Shen, Y. Zhang, Z. Cai, T. Yu, and X. Guan, "Touch-interaction behavior for continuous user authentication on smartphones," *Proceedings of 2015 International Conference on Biometrics, ICB 2015*, pp. 157–162, Jun. 2015, doi: 10.1109/ICB.2015.7139046.
[203] H. C. Volaka, G. Alptekin, O. E. Basar, M. Isbilen, and O. D. Incel, "Towards Continuous Authentication on Mobile Phones using Deep Learning Models," *Procedia Comput Sci*, vol. 155, pp. 177–184, Jan. 2019, doi: 10.1016/J.PROCS.2019.08.027.
[204] M. Smith-Creasey and M. Rajarajan, "A continuous user authentication scheme for mobile devices," *2016 14th Annual Conference on Privacy, Security and Trust, PST 2016*, pp. 104–113, 2016, doi: 10.1109/PST.2016.7906944.
[205] Z. Tian *et al.*, "Real-Time Lateral Movement Detection Based on Evidence Reasoning Network for Edge Computing Environment," *IEEE Trans Industr Inform*, vol. 15, no. 7, pp. 4285–4294, Jul. 2019, doi: 10.1109/TII.2019.2907754.
[206] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep Reinforcement Learning for Partially Observable Data Poisoning Attack in Crowdsensing Systems," *IEEE Internet Things J*, vol. 7, no. 7, pp. 6266–6278, Jul. 2020, doi: 10.1109/JIOT.2019.2962914.
[207] L. Lv, Z. Wu, J. Zhang, L. Zhang, Z. Tan, and Z. Tian, "A VMD and LSTM Based Hybrid Model of Load Forecasting for Power Grid Security," *IEEE Trans Industr Inform*, vol. 18, no. 9, pp. 6474–6482, Sep. 2022, doi: 10.1109/TII.2021.3130237.
[208] L. Lv, Z. Wu, L. Zhang, B. B. Gupta, and Z. Tian, "An Edge-AI Based Forecasting Approach for Improving Smart Microgrid Efficiency," *IEEE Trans Industr Inform*, vol. 18, no. 11, pp. 7946–7954, Nov. 2022, doi: 10.1109/TII.2022.3163137.
[209] Z. Tian *et al.*, "User and Entity Behavior Analysis under Urban Big Data," *ACM Transactions on Data Science*, vol. 1, no. 3, pp. 1–19, Sep. 2020, doi: 10.1145/3374749.
[210] Q. Yang *et al.*, "Poster abstract: A multimodal data set for evaluating continuous authentication performance in smartphones," *SenSys 2014 - Proceedings of the 12th ACM Conference on Embedded Networked Sensor Systems*, pp. 358–359, Nov. 2014, doi: 10.1145/2668332.2668366.
[211] M. Cheriet, V. Dentamaro, M. Hamdan, D. Impedovo, and G. Pirlo, "Multi-speed transformer network for neurodegenerative disease assessment and activity recognition," *Comput Methods Programs Biomed*, vol. 230, Mar. 2023, doi: 10.1016/J.CMPB.2023.107344.

[212] V. Dentamaro, P. Giglio, D. Impedovo, L. Moretti, and G. Pirlo, "AUCO ResNet: an end-to-end network for Covid-19 pre-screening from cough and breath," *Pattern Recognit*, vol. 127, p. 108656, Jul. 2022, doi: 10.1016/J.PATCOG.2022.108656.
[213] E. Ellavarason, R. Guest, F. Deravi, R. Sanchez-Riello, and B. Corsetti, "Touch-dynamics based Behavioural Biometrics on Mobile Devices – A Review from a Usability and Performance Perspective," *ACM Computing Surveys (CSUR)*, vol. 53, no. 6, Dec. 2020, doi: 10.1145/3394713.
[214] I. Kim, "Keypad against brute force attacks on smartphones," *IET Inf Secur*, vol. 6, no. 2, pp. 71–76, Jun. 2012, doi: 10.1049/IET-IFS.2010.0212.
[215] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan, "Shoulder surfing defence for recall-based graphical passwords," *SOUPS 2011 - Proceedings of the 7th Symposium on Usable Privacy and Security*, 2011, doi: 10.1145/2078827.2078835.
[216] C. Giuffrida, K. Majdanik, M. Conti, and H. Bos, "I sensed it was you: Authenticating mobile users with sensor-enhanced keystroke dynamics," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8550 LNCS, pp. 92–111, 2014, doi: 10.1007/978-3-319-08509-8_6/COVER.
[217] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "A survey on touch dynamics authentication in mobile devices," *Comput Secur*, vol. 59, pp. 210–235, Jun. 2016, doi: 10.1016/J.COSE.2016.03.003.
[218] V. Gattulli, D. Impedovo, G. Pirlo, and F. Volpe, "Touch events and human activities for continuous authentication via smartphone," *Sci Rep*, vol. 13, no. 1, Jun. 2023, doi: 10.1038/S41598-023-36780-3.
[219] V. Dentamaro, D. Impedovo, and G. Pirlo, "Gait Analysis for Early Neurodegenerative Diseases Classification through the Kinematic Theory of Rapid Human Movements," *IEEE Access*, vol. 8, pp. 193966–193980, 2020, doi: 10.1109/ACCESS.2020.3032202.
[220] D. Impedovo, V. Dentamaro, G. Pirlo, and L. Sarcinella, "TrafficWave: Generative deep learning architecture for vehicular traffic flow prediction," *Applied Sciences (Switzerland)*, vol. 9, no. 24, Dec. 2019, doi: 10.3390/APP9245504.
[221] P. S. Teh, N. Zhang, S. Y. Tan, Q. Shi, W. H. Khoh, and R. Nawaz, "Strengthen user authentication on mobile devices by using user's touch dynamics pattern," *J Ambient Intell Humaniz Comput*, vol. 11, no. 10, pp. 4019–4039, Oct. 2020, doi: 10.1007/S12652-019-01654-Y/FIGURES/12.
[222] I. de Mendizabal-Vázquez, D. de Santos-Sierra, J. Guerra-Casanova, and C. Sánchez-Ávila, "Supervised classification methods applied to keystroke dynamics through mobile devices," *Proceedings - International Carnahan Conference on Security Technology*, vol. 2014-October, no. October, Dec. 2014, doi: 10.1109/CCST.2014.6987033.
[223] S. Sen and K. Muralidharan, "Putting 'pressure' on mobile authentication," *2014 7th International Conference on Mobile Computing and Ubiquitous Networking, ICMU 2014*, pp. 56–61, 2014, doi: 10.1109/ICMU.2014.6799058.
[224] M. Trojahn, "Authentication with Keystroke Dynamics on Touchscreen Keypads-Effect of different N-Graph Combinations," *Computer Science*, 2013.
[225] C. Shen, T. Yu, S. Yuan, Y. Li, and X. Guan, "Performance Analysis of Motion-Sensor Behavior for User Authentication on Smartphones," *Sensors (Basel)*, vol. 16, no. 3, Mar. 2016, doi: 10.3390/S16030345.
[226] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy," *IEEE Trans Pattern Anal Mach Intell*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005, doi: 10.1109/TPAMI.2005.159.
[227] G. Pirlo and D. Impedovo, "A new class of monotone functions of the residue number system," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 7, no. 9, pp. 802–809, 2013.
[228] V. Dentamaro, V. Gattulli, D. Impedovo, and F. Manca, "Human activity recognition with smartphone-integrated sensors: A survey," *Expert Syst Appl*, vol. 246, p. 123143, Jul. 2024, doi: 10.1016/j.eswa.2024.123143.
[229] M. Alema Khatun and M. Abu Yousuf, "Human Activity Recognition Using Smartphone Sensor Based on Selective Classifiers," *2020 2nd International Conference on Sustainable Technologies for Industry 4.0, STI 2020*, Dec. 2020, doi: 10.1109/STI50764.2020.9350486.
[230] A. Chelli and M. Patzold, "A Machine Learning Approach for Fall Detection and Daily Living Activity Recognition," *IEEE Access*, vol. 7, pp. 38670–38687, 2019, doi: 10.1109/ACCESS.2019.2906693.
[231] H. Rezaie and M. Ghassemian, "Comparison Analysis of Radio_Based and Sensor_Based Wearable Human Activity Recognition Systems," *Wirel Pers Commun*, vol. 101, no. 2, pp. 775–797, Jul. 2018, doi: 10.1007/S11277-018-5715-4/TABLES/7.
[232] M. K, A. Ramesh, R. G, S. Prem, R. A A, and Dr. M. P. Gopinath, "1D Convolution approach to human activity recognition using sensor data and comparison with machine learning algorithms," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 130–143, Jun. 2021, doi: 10.1016/J.IJCCE.2021.09.001.
[233] S. B. Khojasteh, J. R. Villar, E. de la Cal, V. M. González, and J. Sedano, "Fall Detection Analysis Using a Real Fall Dataset," *Advances in Intelligent Systems and Computing*, vol. 771, pp. 334–343, 2019, doi: 10.1007/978-3-319-94120-2_32/COVER.

[234] J. I. Pilataxi Piltaxi, M. F. Trujillo Guerrero, V. C. Benavides Laguapillo, and J. A. Rosales Acosta, "Human Activity Recognition Using an Accelerometer Magnitude Value," *Communications in Computer and Information Science*, vol. 1194 CCIS, pp. 462–472, 2020, doi: 10.1007/978-3-030-42520-3_37/COVER.
[235] M. Hou, H. Wang, Z. Xiao, and G. Zhang, "An SVM fall recognition algorithm based on a gravity acceleration sensor," *http://mc.manuscriptcentral.com/tssc*, vol. 6, no. 3, pp. 208–214, Sep. 2018, doi: 10.1080/21642583.2018.1547888.
[236] F. J. Ordóñez, D. Roggen, Y. Liu, W. Xiao, H.-C. Chao, and P. Chu, "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition," *Sensors 2016, Vol. 16, Page 115*, vol. 16, no. 1, p. 115, Jan. 2016, doi: 10.3390/S16010115.
[237] J. Shi, D. Zuo, and Z. Zhang, "An Energy-Efficient Human Activity Recognition System Based on Smartphones," *2020 7th International Conference on Soft Computing and Machine Intelligence, ISCMI 2020*, pp. 177–181, Nov. 2020, doi: 10.1109/ISCMI51676.2020.9311585.
[238] A. Alruban, H. Alobaidi, and N. C. F. Li, "Physical Activity Recognition by Utilising Smartphone Sensor Signals," *ICPRAM 2019 - Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods*, pp. 342–351, Jan. 2022, doi: 10.5220/0007271903420351.
[239] M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, M. Ryan, M. Elhoseny, and H. Song, "ST-DeepHAR: Deep Learning Model for Human Activity Recognition in IoHT Applications," *IEEE Internet Things J*, vol. 8, no. 6, pp. 4969–4979, Mar. 2021, doi: 10.1109/JIOT.2020.3033430.
[240] E. Brophy, W. Muehlhausen, A. F. Smeaton, and T. E. Ward, "Optimised Convolutional Neural Networks for Heart Rate Estimation and Human Activity Recognition in Wrist Worn Sensing Applications," Mar. 2020, doi: 10.48550/arxiv.2004.00505.
[241] E. Fridriksdottir and A. G. Bonomi, "Accelerometer-Based Human Activity Recognition for Patient Monitoring Using a Deep Neural Network," *Sensors 2020, Vol. 20, Page 6424*, vol. 20, no. 22, p. 6424, Nov. 2020, doi: 10.3390/S20226424.
[242] Ankita, S. Rani, H. Babbar, S. Coleman, A. Singh, and H. M. Aljahdali, "An Efficient and Lightweight Deep Learning Model for Human Activity Recognition Using Smartphones," *Sensors 2021, Vol. 21, Page 3845*, vol. 21, no. 11, p. 3845, Jun. 2021, doi: 10.3390/S21113845.
[243] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, and T. Kechadi, "Human Activity Recognition with Convolutional Neural Netowrks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11053 LNAI, pp. 541–552, Jun. 2019, doi: 10.1007/978-3-030-10997-4_33.
[244] B. Oluwalade, S. Neela, J. Wawira, T. Adejumo, and S. Purkayastha, "Human Activity Recognition using Deep Learning Models on Smartphones and Smartwatches Sensor Data," *HEALTHINF 2021 - 14th International Conference on Health Informatics; Part of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2021*, pp. 645–650, Feb. 2021, doi: 10.48550/arxiv.2103.03836.
[245] N. Hnoohom, A. Jitpattanakul, and S. Mekruksavanich, "Real-life Human Activity Recognition with Tri-axial Accelerometer Data from Smartphone using Hybrid Long Short-Term Memory Networks," *Proceedings - 2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing, iSAI-NLP 2020*, Nov. 2020, doi: 10.1109/ISAI-NLP51646.2020.9376839.
[246] A. K. M. Masum, E. H. Bahadur, A. Shan-A-Alahi, M. A. Uz Zaman Chowdhury, M. R. Uddin, and A. Al Noman, "Human Activity Recognition Using Accelerometer, Gyroscope and Magnetometer Sensors: Deep Neural Network Approaches," *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*, Jul. 2019, doi: 10.1109/ICCCNT45670.2019.8944512.
[247] X. Li, Y. Wang, B. Zhang, and J. Ma, "PSDRNN: An Efficient and Effective HAR Scheme Based on Feature Extraction and Deep Learning," *IEEE Trans Industr Inform*, vol. 16, no. 10, pp. 6703–6713, Oct. 2020, doi: 10.1109/TII.2020.2968920.
[248] S. Fan, Y. Jia, and C. Jia, "A Feature Selection and Classification Method for Activity Recognition Based on an Inertial Sensing Unit," *Information 2019, Vol. 10, Page 290*, vol. 10, no. 10, p. 290, Sep. 2019, doi: 10.3390/INFO10100290.
[249] M. Shoaib, S. Bosch, O. Durmaz Incel, H. Scholten, and P. J. M. Havinga, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition," *Sensors 2014, Vol. 14, Pages 10146-10176*, vol. 14, no. 6, pp. 10146–10176, Jun. 2014, doi: 10.3390/S140610146.
[250] L. Atallah, B. Lo, R. King, and G. Z. Yang, "Sensor positioning for activity recognition using wearable accelerometers," *IEEE Trans Biomed Circuits Syst*, vol. 5, no. 4, pp. 320–329, Aug. 2011, doi: 10.1109/TBCAS.2011.2160540.
[251] A.-M. Mandong and U. Munir, "Smartphone Based Activity Recognition using K-Nearest Neighbor Algorithm," *Conference: International Conference on Engineering Technologies*.
[252] W. T. D. souza and K. R, "Human Activity Recognition Using Accelerometer and Gyroscope Sensors," *International Journal of Engineering and Technology*, vol. 9, no. 2, pp. 1171–1179, Apr. 2017, doi: 10.21817/IJET/2017/V9I2/170902134.
[253] C. Hu, Y. Chen, L. Hu, and X. Peng, "A novel random forests based class incremental learning method for activity recognition," *Pattern Recognit*, vol. 78, pp. 277–290, Jun. 2018, doi: 10.1016/J.PATCOG.2018.01.025.

[254] P. Casale, O. Pujol, and P. Radeva, “Human activity recognition from accelerometer data using a wearable device,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6669 LNCS, pp. 289–296, 2011, doi: 10.1007/978-3-642-21257-4_36.
[255] S. Mehrang *et al.*, “Human activity recognition using a single optical heart rate monitoring wristband equipped with triaxial accelerometer,” *IFMBE Proc*, vol. 65, pp. 587–590, 2017, doi: 10.1007/978-981-10-5122-7_147.
[256] G. Vavoulas, M. Pediaditis, C. Chatzaki, E. G. Spanakis, and M. Tsiknakis, “The MobiFall Dataset,” *Int J Monit Surveill Technol Res*, vol. 2, no. 1, pp. 44–56, Jan. 2014, doi: 10.4018/IJMSTR.2014010103.
[257] M. Webber and R. F. Rojas, “Human Activity Recognition with Accelerometer and Gyroscope: A Data Fusion Approach,” *IEEE Sens J*, vol. 21, no. 15, pp. 16979–16989, Aug. 2021, doi: 10.1109/JSEN.2021.3079883.
[258] F. Castro, D. Impedovo, and G. Pirlo, “A Hybrid Protection Scheme for the Gait Analysis in Early Dementia Recognition,” *Sensors 2024, Vol. 24, Page 24*, vol. 24, no. 1, p. 24, Dec. 2023, doi: 10.3390/S24010024.