



Hybrid quantum architecture for smart city security[☆]

Vita Santa Barletta^{*}, Danilo Caivano, Mirko De Vincentiis, Anibrata Pal, Michele Scalera

University of Bari Aldo Moro, Via Edoardo Orabona 4, Bari, 70125, BA, Italy

ARTICLE INFO

Keywords:

Hybrid quantum system
Quantum software engineering
Security engineering
Smart city

ABSTRACT

Currently and in the near future, Smart Cities are vital to enhance urban living, address resource challenges, optimize infrastructure, and harness technology for sustainability, efficiency, and improved quality of life in rapidly urbanizing environments. Owing to the high usage of networks, sensors, and connected devices, Smart Cities generate a massive amount of data. Therefore, Smart City security concerns encompass data privacy, Internet-of-Things (IoT) vulnerabilities, cyber threats, and urban infrastructure risks, requiring robust solutions to safeguard digital assets, citizens, and critical services. Some solutions include robust cybersecurity measures, data encryption, Artificial Intelligence (AI)-driven threat detection, public-private partnerships, standardized security protocols, and community engagement to foster a resilient and secure smart city ecosystem. For example, Security Information and Event Management (SIEM) helps in real-time monitoring, threat detection, and incident response by aggregating and analyzing security data. To this end, no integrated systems are operating in this context. In this paper, we propose a Hybrid Quantum-Classical Architecture for bolstering Smart City security that exploits Quantum Machine Learning (QML) and SIEM to provide security based on Quantum Artificial Intelligence and patterns/rules. The validity of the hybrid quantum-classical architecture was proven by conducting experiments and a comparison of the QML algorithms with state-of-the-art AI algorithms. We also provide a proof of concept dashboard for the proposed architecture.

1. Introduction

A smart city is an urban area that uses digital technology or information and communication technology (ICT), data, and various innovative solutions to enhance the quality of life for its residents and improve the efficiency of city operations. A Smart City encompasses a wide array of services that leverage advanced technologies to enhance urban living. This transformation into an intelligent urban ecosystem involves the extensive deployment of devices, sensors, and interconnected infrastructures, facilitating seamless interactions between objects and individuals across diverse service domains (Barletta et al., 2021).

Some of the technologies such as Big Data, IoT, Cloud computing, and Mobile Computing have in recent years become an integral part of smart cities (Sánchez-Corcuera et al., 2019). However, these technologies can store or exchange critical information that, if not secured properly, could be accessed or modified by an attacker. For example, considering the IoT, if an adversarial identifies a vulnerability, it could compromise the data and the overall security of the smart city. So, it is necessary to use efficient and advanced techniques in order to identify attacks and ensure higher confidentiality, integrity, and availability of

the data exchange in the smart city (Tariq et al., 2023; Gigante et al., 2023).

Some of these approaches use Machine Learning (ML), Deep Learning (DL), and Rule-based techniques. With rule-based techniques, it is necessary to create robust rules to identify attacks because an adversarial could evade them (Jia et al., 2023). In contrast, ML and DL algorithms can generalize the problem and identify unknown attacks, while rule-based detection techniques have a low false-positive rate and high efficiency (Rajapaksha et al., 2023). On the other hand, adversarial attacks can be exploited to make misclassification in a machine-learning pipeline (Tan et al., 2022; Zhou et al., 2021). In addition, some ML or DL algorithms require high time for training and prediction. In a critical context, like Smart City, it is necessary to make the prediction as quick as possible.

In addressing the escalating demands of data analysis and classification within the context of Smart Cities, the emergence of Intelligent Operations Centers (IOCs) has been a significant development (Zhuhadar et al., 2023). Among the prominent IOCs, IBM IOC has garnered substantial recognition (Bhowmick et al., 2012; Jiang et al., 2019). It operates by autonomously identifying conflicts between various city

[☆] Editor: Prof. Raffaella Mirandola.

^{*} Corresponding author.

E-mail addresses: vita.barletta@uniba.it (V.S. Barletta), danio.caivano@uniba.it (D. Caivano), mirko.devincentiis@uniba.it (M. De Vincentiis), anibrata.pal@uniba.it (A. Pal), michele.scalera@uniba.it (M. Scalera).

<https://doi.org/10.1016/j.jss.2024.112161>

Received 30 September 2023; Received in revised form 6 June 2024; Accepted 17 July 2024

Available online 19 July 2024

0164-1212/© 2024 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

agencies, leveraging reporting and monitoring mechanisms to optimize both planned and unplanned operations, and dynamically adapting its functionality based on the information it gathers. Importantly, IOCs play a pivotal role not only in streamlining city-wide operations but also in enforcing vital security and privacy controls for safeguarding sensitive data (Hwoij et al., 2021-04-05; Babar et al., 2020).

Parallel to IOCs, Security Operations Centers (SOCs) represent a specialized variant of these centers with a specific focus on shielding organizations from cyber threats (Barletta et al., 2023b). SOC teams, comprising seasoned security experts and SOC analysts, engage in continuous monitoring of an organization's assets. They employ a diverse toolkit to generate comprehensive reports and action plans. Notably, the cornerstone of SOC operations is the SIEM system, an indispensable tool for SOC analysts.

At this point, it is extremely crucial to understand the challenges for the implementation and integration of such systems for Smart Cities in the purview of security and privacy also considering the challenges and landscapes of Quantum Computing (QC). With the introduction of Quantum Computers, we can exploit this power to create new ML or DL algorithms that can make decisions faster than traditional algorithms. For example, in Barletta et al. (2023a), the authors proposed the use of QBoost (Neven et al., 2012), a QML algorithm, to reduce the time to identify attacks in an IoT scenario. The researchers compared the time performance with the Support Vector Machine (SVM) classifier in the training and testing. The results show that QBoost is faster than SVM, especially by increasing the example data.

On the other hand, quantum computers can be exploited to propose ML algorithms and increase security in cryptographic systems in that they offer not only computational benefits. In particular, the National Institute of Standards and Technology (NIST) proposed a program and competition to update their standards regarding *Post-Quantum Cryptography* that consists of developing cryptographic systems that are secure against both quantum and classical computers and can interoperate with existing communications protocols and networks (Anon., 2016). This is because if large-scale quantum computers are ever built, they will be able to break many of the public-key cryptosystems currently in use (Anon., 2016). Starting from this problem, several researchers have proposed solutions based on the NIST criteria to ensure security of the existing cryptosystems. For example, in Hekkala et al. (2023), the authors discussed the process of implementing a post-quantum cryptographic algorithm into a programming language library, choosing Crypto++.

An important concern is the lack of explanation of the predictions of some ML or DL algorithms. Therefore, it is necessary to integrate other mechanisms that can help the security analysts understand the attack, for example, how it was conducted. SIEM can help organizations recognize and address potential security threats and vulnerabilities before they disrupt business. SIEM can alert security analysts if an attack is detected and can help them reconstruct the kill chain to understand how it has been conducted. For example, consider IoT devices that send the data to the SIEM. If an attacker exploits a DDoS attack to disrupt the network, the security analyst could analyze the network events to identify how it was done and the impacted devices.

Therefore, there is a need for the development of software systems that integrate classical and quantum computation in order to improve threat identification. Moreover, considering the current challenges of quantum software engineering namely that of reworking and extending the whole of classical software engineering into the quantum domain Piattini and Murillo (2022) and Zhao (2020), the idea underlying the research work is to extend the design of systems with quantum technologies. Specifically, to integrate a quantum component that can communicate with traditional software systems and reduce data processing time where possible.

Having said that, in this paper, we propose a novel Reference Architecture, we henceforth call Hybrid Quantum-Classical Architecture, for Smart City Security that amalgamates QML through QBoost, a quantum

machine learning algorithm hosted in D-Wave Leap Quantum Cloud (DLQC), and IBM QRadar, a widely acclaimed SIEM system. In this preliminary work, the proposed hybrid architecture allows us to identify attacks against Smart City through quantum algorithms (QBoost) and traditional systems (SIEM) and show the results in a dashboard. This integrated approach enables security analysts to make timely decisions without needing to separately view the results from QBoost and QRadar. We evaluated these two identification mechanisms using the CICIOT2023 dataset (Neto et al., 2023), which accurately reflects real-world Smart City scenarios. In the remainder of the paper, we refer to the proposed reference architecture as "hybrid quantum-classical architecture" or just "architecture".

The rest of the paper is organized as follows: Section 2 presents the relevant literature on the topic, Section 3 discusses some basic information essential for the topic, Section 4 describes the proposed hybrid quantum-classical architecture in details, Section 5 illustrates the experimental setup and the experiments, Section 6 presents the results and discusses the implications, and finally concluding with remarks and future directions in Section 7.

2. Related work

2.1. Classical intrusion detection system

In the field of ML, the concept of Intrusion Detection has been consolidated in the literature. Researchers use classical ML or DL algorithms to identify attacks that can be conducted on the network such as Random Forest (RF), Support Vector Machine (SVM), Neural Networks, and so on. In Neto et al. (2023), the authors released a dataset containing 33 attack types executed in an IoT topology. To identify these attacks, they used different ML models: Logistic Regression, Perceptron, Adaboost, Random Forest, and Deep Neural Network. These models were tested considering two classes, eight classes, and 34 classes. In Alrashdi et al. (2019), the authors proposed an RF architecture called Anomaly Detection-IoT (AD-IoT) system to identify any suspicious activity at the distributed fog nodes. The results show that the system proposed by the researchers reaches 99.34% of accuracy with the lowest false positive rate. Interesting work was conducted by Bhavsar et al. (2023), where they developed an IDS based on a deep learning model called Pearson-Correlation Coefficient-Convolutional Neural Network (PCC-CNN) to detect network anomalies. The PCC-CNN model combines the important features obtained from the linear-based extractions followed by the CNN. The researchers evaluated the model using three datasets: NSL-KDD, CICIDS-2017, and IOTID20. They trained five different ML algorithms (Logistic Regression, Linear Discriminant Analysis, K Nearest Neighbor, Classification and Regression Tree, and Support Vector Machine). Then, the results were compared with the PCC-CNN model, which outperformed these traditional models.

Other works proposed signature-based IDS. These approaches detect attacks when a system or network behavior matches an attack signature stored in the IDS database (Zarpelão et al., 2017). For example, in Ioulaniou et al. (2018), the researchers proposed a signature-based IDS for protecting IoT networks from external and internal threats. This approach involves both centralized and distributed IDS modules. The system was tested using the Cooja simulator, implementing a DoS attack scenario in IoT devices.

With the introduction of quantum computers, some researchers proposed IDS exploiting quantum ML algorithms. Barletta et al. (2023a) used QBoost to identify malicious attacks in IoT networks. Instead, in Caivano et al. (2022), the authors have used the same quantum ML algorithm to identify attacks that can be conducted on the in-vehicle network and, in particular, on the Controller Area Network (CAN) bus. The CAN protocol is a standard protocol that allows communication between nodes installed in the vehicle.

2.2. Quantum-hybrid intrusion detection system

In the context of hybrid quantum computing, some researchers proposed works that exploit Quantum Computing in combination with classical machine learning and, in particular, Deep Learning algorithms. For example, in automotive security, to identify attacks that can be conducted on the in-vehicle network, Salek et al. (2023) proposed a framework using a classical neural network and a quantum-restricted Boltzmann machine (RBM). The classical neural network extracts the features from CAN images generated from the CAN bus. Instead, the RBM is used to reconstruct the CAN image and determine if the traffic is normal or if an attacker inject malicious messages on the bus. Exploiting the image classification, also Wang et al. (2023) designed a quantum-classical hybrid deep neural network (QHDNN) that aims to learn directly from normal raw images to train a normality model and exclude the images that are anomalies (excluded from the model).

In another work, Suryotrisongko and Musashi (Suryotrisongko and Musashi, 2022) implemented an anomaly detection system in the cybersecurity field to identify botnets. The authors compared their hybrid quantum-classical deep learning model with the classical model counterpart. The results show that in some cases, the hybrid model reached high performance. However, compared with the classical counterpart of the deep learning model, the overall performance remains lower for the remaining cases. A hybrid quantum-classical model of Long short-term memory (LSTM) was proposed in Chen et al. (2022) demonstrating that the model successfully learns several kinds of temporal data. In particular, the researchers have shown that the hybrid model converges faster and obtains better accuracy than classical LSTM.

In Herr et al. (2021), the authors replace the generator of a quantum-classical Wasserstein Generative Adversarial Network (WGAN) with a hybrid quantum-classical neural network without changing the classical discriminative model. This model was tested in an anomaly/fraudulent transaction field, obtaining performance on par with classical methods in terms of the F1-Score with credit card fraud dataset. Within the context of Smart City, in Barletta et al. (2023c), the authors proposed an architecture using Quantum as a Service for security incident detection and threat assessment. DLQC-based QBoost was implemented to support the threat classification in large Smart City IoT data.

The use of IDSs is not sufficient enough to identify attacks in the field of Smart City as they do not provide attack pathways but rather provide us with attack identification only. Therefore, the integration of a Quantum IDS along with classical approaches like SIEM is deemed necessary for a more robust security system. In this paper, we proposed a hybrid quantum-classical reference architecture that exploits the power of Quantum IDS and the critical capability to visualize and understand the attack path using SIEM. In addition to this, in this preliminary work, we develop a dashboard that combines the results from both QBoost (Quantum IDS that we chose based on the literature) and IBM QRadar SIEM, providing an extra layer of security. This integrated approach enables security analysts to make timely decisions without needing to view the results from QBoost and QRadar separately.

3. Background

In this section, we will describe the preliminary information necessary to understand the experimental phase. First, we will present a brief overview regarding the SIEM that has been used, then a brief recap of quantum theory around QBoost.

3.1. IBM QRadar SIEM

IBM QRadar SIEM collects, processes, aggregates, and stores network and event data in real-time to manage the security of an organization or different assets (or devices) by providing real-time information and monitoring, raising alerts and offenses, and executing responses

to network threats (Anon., 2023c). IBM QRadar SIEM is a modular architecture that can be used for threat detection and prioritization. The main functionalities of the SIEM are divided into three layers: **Data Collection**, **Data Processing**, and **Data Searches**.

In the Data Collection, the data (events or flows) are collected from the network (Anon., 2023c). QRadar accepts event logs created by log sources like firewalls, routers, IPS, and IDS by using different protocols, for example, the Syslog. The core functionality of the SIEM is focused on event data collection and flow collection. Event data pertains to occurrences within a user's environment at a specific moment, encompassing actions like user logins, email activities, firewall denials, and any other events. Instead, flow data is network activity information or session information between two hosts on a network, which QRadar translates as flow records. Flow records can be, for example, the messages exchanged in IoT networks.

The Data Processing is the second layer, where event and flow data are run through the Custom Rule Engine (CRE). This module generates offenses and alerts, and then the data is written in the *Ariel database*, which is a time-series database where event data is stored minute by minute when the event is processed.

Finally, the Data Search is the third or top layer, in which the data collected and stored by QRadar are available to be searched for analysis and reporting in order to take appropriate countermeasures.

We decided to use IBM QRadar SIEM because provides a free Community Edition. In addition, it is possible to use a default set of rules that can be customized.

3.2. Quantum computing

The journey from Quantum Mechanics to Quantum Computing has been a remarkable odyssey through the realms of fundamental physics, information theory, and cutting-edge technology. It represents a convergence of foundational insights from quantum mechanics, harnessed to revolutionize computation in ways previously unimaginable (Piattini et al., 2020).

The concept of a quantum computer emerged in the early 1980s when Richard Feynman proposed that quantum systems could simulate physical phenomena more efficiently than classical computers (Feynman, 1982). In 1994, Peter Shor devised a quantum algorithm that could factor large numbers exponentially faster than classical algorithms (Shor, 1997), posing a significant threat to classical cryptography. Simultaneously, Lov Grover developed Grover's algorithm, capable of quadratically speeding up database search, a critical computational problem (Grover, 1996). These breakthroughs underscored quantum computing's transformative potential.

Quantum computing's promise is not without challenges. Unlike traditional bits, quantum bits or qubits are inherently fragile and susceptible to decoherence. In classical computing, a bit can either be in a state of 0 or 1 at any given time. However, qubits can exist in a linear combination of both 0 and 1 states simultaneously. Superposition is one of the fundamental principles of quantum mechanics and is a key feature that makes this possible. Mathematically, this superposition is represented as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Where, $|\psi\rangle$ represents the state of the qubit, $\alpha|0\rangle$ and $\beta|1\rangle$ represent the classical states 0 and 1, respectively, and α and β are complex probability amplitudes, which determine the probability of measuring the qubit in the $|0\rangle$ or $|1\rangle$ state when a measurement is made. A qubit is often represented by a Bloch Sphere as shown in Fig. 1.

Apart from superposition, quantum computing also exhibits properties like entanglement, quantum tunneling/annealing, and quantum interference, enabling it to process information differently from classical computers (Farhi et al., 2014). It offers the potential for exponential speedup in solving specific problems, such as factoring large numbers

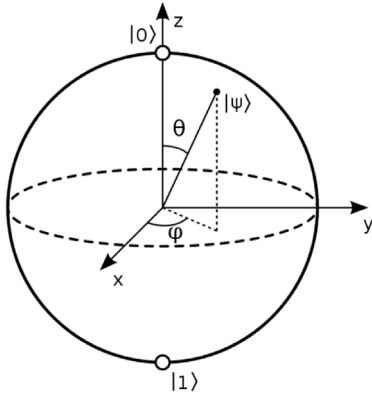


Fig. 1. Bloch Sphere.

and optimizing complex systems, making it a promising technology for various fields.

Quantum computing encompasses various approaches to harness quantum phenomena for performing computational tasks like Gate-based Quantum Computing, Topological Quantum Computing, and Adiabatic Quantum Computing.

1. *Gate-based quantum computing*: It employs quantum gates to manipulate qubits, which are the fundamental units of quantum information. These gates perform operations on qubits, enabling the creation of quantum circuits that implement algorithms. The operations include single-qubit gates (acting on individual qubits) and multi-qubit gates (acting on pairs of qubits). The quantum gates like Hadamard gate (H), Pauli gates (X, Y, Z), CNOT (Controlled-NOT) gate, and phase gate (S) can be composed in sequences to construct Quantum circuits. The quantum gates are applied sequentially to the qubits, with each gate affecting the quantum state of the qubits (for example, Fig. 2).
2. *Topological quantum computing*: Conceived by the physicist Alexei Kitaev in 1997, it employs anyons – quasiparticles in 2D systems – as logic gates, forming braids in 3D spacetime. Unlike traditional quantum computers relying on trapped particles prone to decoherence, topological quantum computers claim stability due to their robust topological properties. These properties, safeguarding against minor disruptions, make them promising for computational tasks. Microsoft's Station Q and Majorana fermions research initiatives are actively exploring this field.
3. *Adiabatic Quantum Computing (AQC)*: Based on the idea of Adiabatic Evolution (Farhi et al., 2000), this paradigm involves gradually transforming a quantum system from an initial Hamiltonian to a final Hamiltonian that encodes the solution to a problem. A popular approach to AQC uses the annealing process (Kadowaki and Nishimori, 1998), where the system starts in the ground state of the initial Hamiltonian and slowly evolves to the final Hamiltonian through a controlled annealing process via the application of external controls, such as electromagnetic fields or laser pulses. Therefore, by manipulating the system's Hamiltonian over time, the system can be driven to achieve the ground state of the final Hamiltonian, representing the solution to the problem. D-Wave Systems develops quantum annealers that implement adiabatic quantum computing.

Generally, a Hamiltonian of a physical system is an operator corresponding to the total energy of the system (Nielsen and Chuang, 2010).

Fig. 2. Quantum circuit for full adder (using IBMQ¹).

Considering a two-level quantum system, such as a qubit, this can be written as:

$$H(t) = \frac{\omega(t)}{2} \sigma_z + \frac{\Delta(t)}{2} \sigma_x \quad (1)$$

Where:

- $\omega(t)$ represents the time-dependent energy splitting between the two levels.
- $\Delta(t)$ represents the time-dependent strength of the interaction between the two levels.
- σ_z and σ_x are the Pauli matrices.

This transformation of this initial Hamiltonian (say H_0) is achieved by gradually changing the functions $\omega(t)$ and $\Delta(t)$ over time by applying a sequence of laser pulses with carefully designed amplitudes and frequencies to modulate these parameters. The time evolution of such quantum systems is given by the Schrödinger equation:

$$\hat{H}|\Psi(t)\rangle = i\hbar \frac{\partial}{\partial t} |\Psi(t)\rangle \quad (2)$$

Where:

- $|\Psi(t)\rangle$ is the quantum state of the system at time t .
- $H(t)$ is the Hamiltonian operator(observable) of the system at time t .
- $\hbar$ is the reduced Planck constant.

Thus, given $H(t)$, the goal of optimization is often to find the quantum state $|\Psi(t)\rangle$ that minimizes (or maximizes) an energy functional. This energy functional could be related to the total energy of the system, expectation values of certain observables, or other physical quantities of interest. Now, in complex potential landscape, the Hamiltonian $H(t)$ can be written as:

$$H(t) = \frac{\omega(t)}{2} \sigma_z + \frac{\Delta(t)}{2} \sigma_x + V(x, t) \quad (3)$$

Where:

- $V(x, t)$ represents the complex potential landscape, which can depend on both position x and time t .
- $\omega(t)$ and $\Delta(t)$ are time-dependent parameters as described earlier.

Considering an optimization problem where the aim is to find the quantum state $|\Psi(t)\rangle$ that minimizes the expectation value of the energy functional, denoted as $E[|\Psi(t)\rangle]$. This energy functional can be expressed as:

$$E[|\Psi(t)\rangle] = \langle \Psi(t) | H(t) | \Psi(t) \rangle \quad (4)$$

This is an optimization problem where the objective is to minimize $E[|\Psi(t)\rangle]$ with respect to $|\Psi(t)\rangle$ while satisfying any relevant constraints. Throughout the optimization process, the quantum state $|\Psi(t)\rangle$ is iteratively updated to approach the state that minimizes the energy functional. This involves evaluating the energy functional for different candidate states and adjusting the parameters of the quantum state accordingly. Once the optimization process converges, the resulting quantum state $|\Psi(t)\rangle$ represents the solution that minimizes the energy functional within the given constraints (Kaye et al., 2006; Boixo et al., 2016; Johnson et al., 2011).

¹ <https://quantum.ibm.com/>

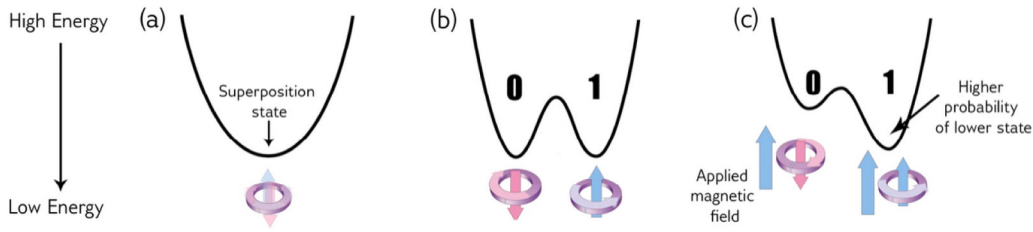


Fig. 3. Energy change diagram during Quantum Annealing.

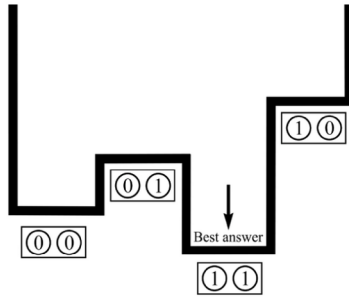


Fig. 4. Energy diagram — Best answer.

3.2.1. DWave quantum platform

Quantum annealing is a specialized paradigm tailored primarily for addressing optimization problems. D-Wave's quantum annealing systems, exemplified by the Advantage_system5.3, harness qubits to navigate the energy landscapes associated with these optimization challenges, seeking to identify the configurations with the lowest energy, which corresponds to optimal solutions.²

The intricate process of quantum annealing can be visualized through an energy diagram that evolves over time Fig. 3.³ Initially, the energy landscape resembles a single valley with a single minimum, representing a unique solution. As quantum annealing commences, a barrier is raised, transforming the landscape into a double-well potential. Here, the left valley corresponds to the 0 state, while the right valley signifies the 1 state. The qubit ultimately settles in one of these valleys at the end of the annealing process (Fig. 4⁴). By default, there is an equal probability of it ending up in either state (50 percent).

However, the quantum annealing process can be controlled by applying an external magnetic field to the qubit. This tilts the double-well potential, increasing the likelihood of the qubit ending up in the lower well. This programmable external magnetic field is called a bias, and the qubit adapts to minimize its energy under the influence of this bias.

The qubits are interconnected via devices called couplers to influence each other, leading to phenomena like entanglement. In an entangled state, two qubits collectively possess four possible states, determined by the biases and coupling strengths. During annealing, qubit states may span these states before settling into the optimal configuration.

Drawing reference from the Hamiltonian representations discussed previously, D-Wave's quantum annealing⁵ approach can also be described in terms of a Hamiltonian, where the ground state of the system

corresponds to the solution of the problem, which can be represented as:

$$\mathcal{H}_{ising} = \underbrace{-\frac{A(s)}{2} \left(\sum_i \hat{\sigma}_x^{(i)} \right)}_{\text{Initial Hamiltonian}} + \underbrace{\frac{B(s)}{2} \left(\sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j} \hat{\sigma}_z^{(i)} \hat{\sigma}_z^{(j)} \right)}_{\text{Final Hamiltonian}} \quad (5)$$

Where,

- $\hat{\sigma}_{x,z}^{(i)}$ are Pauli matrices operating on a qubit q_i .
- h_i and $J_{i,j}$ are the qubit biases and coupling strengths.⁶
- $A(s)$ representing the tunneling energy and the $B(s)$ the problem Hamiltonian energy at s , the normalized anneal fraction, an abstract parameter ranging from 0 to 1.
- Initial Hamiltonian represents the lowest-energy state with all qubits in superposition of 0 and 1, also known as tunneling Hamiltonian.
- Final Hamiltonian encodes the solution of the problem being addressed where lowest-energy end state is classical, also known as problem Hamiltonian.

In a linear annealing process, $s = t/t_f$, where t is time and t_f is the total annealing time. Initially, $A(0) \gg B(0)$, leading to a quantum ground state. As annealing progresses, A decreases and B increases until t_f , when the qubits reach a low-energy solution state. At the end, only the $B(s)$ term remains in the Hamiltonian.

3.2.2. Quantum Machine Learning (QML)

QML represents a compelling fusion of quantum principles and classical machine learning algorithms, promising breakthroughs in data analysis, optimization, and pattern recognition. One prominent player in this field is D-Wave Systems, which has championed quantum annealing technology. D-Wave Systems leads in quantum annealing with its Advantage_system5.3 by enabling QBoost (Neven et al., 2012), which is built on the principles of boosting algorithm and enhances binary classification using quantum computing. Quantum annealers complement QML by tackling optimization tasks and enhancing quantum machine learning models. IBM Quantum Computing is another notable contender, offering gate-based quantum computers providing QML algorithms like Variational Quantum Classifiers (VQC), Quantum Neural Network (QNN), and Quantum Support Vector Machine (QSVM). The synergy of quantum annealing, gate-based quantum computing, and innovative algorithms like QBoost holds promise for quantum machine learning and optimization tasks across various domains, including Smart City security.

4. Proposed hybrid quantum-classical architecture

The existing centralized IOCs discussed in the previous section offer a comprehensive approach to addressing challenges in Smart City conflict scenarios. This research introduces a novel approach to enhancing security incident detection within these established systems.

² <https://cloud.dwavesys.com/leap/>

³ https://docs.dwavesys.com/docs/latest/c_gs_2.html

⁴ https://docs.dwavesys.com/docs/latest/c_gs_2.html

⁵ https://docs.dwavesys.com/docs/latest/c_gs_2.html#id1

⁶ https://docs.dwavesys.com/docs/latest/c_gs_4.html#getting-started-topologies

We propose the adoption of a service-oriented architecture that has the potential to augment the capabilities of IOCs or SOCs. This innovative architecture leverages Quantum Machine Learning to expedite the prediction of security incidents and intrusions, thereby mitigating potential damage and minimizing their impact on interconnected systems. Consequently, this represents a significant advancement in addressing security and privacy concerns for Smart City residents and re-engineering traditional systems that integrate quantum components.

The envisioned architecture (Fig. 5) is characterized as a Hybrid-Quantum Computing system, bridging the realms of classical and quantum computing components. Within this framework, Quantum Machine Learning algorithms, referred to as QML algorithms, serve as a key component, accessible as a service from an external classical component.

This proposed architecture amalgamates two distinct security implementation systems: QRadar and the DWave Leap Quantum Computing platform. QRadar, a widely recognized SIEM system, identifies intrusions and issues by employing a rule-based approach. Security specialists configure a set of rules to analyze data received from various interfaces within the Smart City ecosystem. In contrast, the DWave quantum platform harnesses QML algorithms to train classification models for detecting attacks and intrusions within the system. It achieves this by processing data collected from the Smart City infrastructure.

In essence, this research introduces a novel approach to Smart City security, integrating classical and quantum computing elements to enhance incident detection and response, ultimately bolstering the security and privacy of Smart City inhabitants.

The envisioned system consists of a robust three-layer architecture, meticulously organized into six core components for optimal functionality. These components include: (a) *Smart City Data Source*, (b) *Dashboard*, (c) *Principal Application*, (d) *Database*, (e) *Quantum Cloud Service* and (f) *QRadar - SIEM*. In this context, the presentation, application, and data tiers were developed as distinct infrastructures, offering increased security and modularity. Each of these components contributes uniquely to the overall functionality of our system, ensuring a robust and comprehensive approach to data processing and security in the Smart City environment. The details of each of the components are presented in greater detail in the following subsections.

4.1. Smart City data source

The Smart City data source, as depicted in Fig. 5.a, represents the intelligence hub within the urban landscape. It serves as the nexus for seamless information exchange among various interconnected Smart City components, all aimed at enhancing the quality of life for city residents. These data sources are distributed throughout the city, forming a vital network that fuels the system's operations. Despite their physical location external to the system, they are extremely vital to the entire ecosystem. Data emanating from different facets of the Smart City arrives in various formats, which depend on the type of log source or data source integrated into the system. Generally, the data is received in diverse formats like Common Event Format (CEF), commonly used for network and security-related logs, Syslog (protocol for sending event messages and log data), JSON (JavaScript Object Notation) and XML (eXtensible Markup Language), SNMP (Simple Network Management Protocol), generated by network devices, database logs, and flat files such as CSV (comma-separated values) files among other.

4.2. Smart City security dashboard

The proposed architecture introduces a dynamic graphical user interface (GUI), as exemplified by the interactive dashboard featured in Fig. 5.b. This GUI serves as a real-time platform for monitoring and responding to a wide range of security incidents and attacks unfolding across various sectors within the Smart City infrastructure. The dashboard is intelligently structured so that different incidents associated

with each Smart City sector can be accessed through filters on the respective headers, presenting them in an accessible tabular format. In the spirit of immediate responsiveness, the dashboard promptly updates upon incident registration, furnishing comprehensive details regarding the encountered threat.

Each threat is either assigned a unique identifier if they do not come assigned with one, or else they carry forward their own unique identifiers. Further, the threats or offenses are also characterized by their timestamps. The dashboard's data stream is directly sourced from a RESTful API, which, in turn, retrieves data from two different sources: the database tables and the QRadar data output API. This API functions as the GUI's primary data conduit, presenting information in a format that allows security experts to swiftly evaluate and counter threats.

Moreover, the dashboard supports the review of historical incidents, enabling security specialists to reference prior occurrences for in-depth analysis and informed decision-making. Fig. 6 serves as an illustrative sample of the dashboard, a pivotal component within our proposed architecture, spotlighting real-time alerts stemming from incoming Smart City data. The dashboard systematically organizes attack incidents in rows, each tagged with a distinct incident ID and supplemented with relevant particulars such as incident timestamp, severity (when available), category, and attack type (also when available).

In addition to this, the GUI allows security experts to make decisions based on the received attacks. For example, considering the dashboard in Fig. 6, the *Incident IDs* from 15 to 20 correspond to the threats identified by IBM QRadar SIEM (see Section 5.3 for experimentation details). In this case, a security analyst can use the **Offense Source** IP to block the attacker and avoid the Distributed-Denial of Service (DDoS) attack (Attack Type column). On the other hand, for the attacks identified by the IDS, the security analysts could obtain the source IP from DWave (based on the prediction) and reconstruct the attack chain to identify the attacker IP and block the ones sending malicious content. In this preliminary work, the dashboard only shows information about the attack prediction with the **Offense Source** using a group-based typology of the type of attack, for instance, *IoT_Weather* in the dashboard screenshot (Fig. 6), owing to the limitation of the dataset in consideration. In the future, we plan to connect the dashboard to a real IoT device that provides IP and other detailed information.

The dashboard was developed using Angular (Version - 15.1.6) and Node.js (Version - 18.14.1).

4.3. Principal application

At the core of the proposed architecture, as depicted in Fig. 5.c, lies the main application, a critical component responsible for processing incoming incidents from various Smart City sectors. These sectors encompass areas such as government, mobility, living, environment, and people. Employing a well-structured Intrusion Detection System (IDS) pipeline, the application methodically sifts through this influx of data to pinpoint potential security incidents, a process illustrated in Fig. 7. Following the prescribed IDS methodology, the data undergoes initial scrutiny and cleansing, shedding irrelevant fields that hold no significance. Subsequently, a sequence of operations extracts and prioritizes the most pertinent features for further analysis. The system then embarks on the training of machine learning models, subjecting them to rigorous validation before entering the prediction phase. During this predictive stage, test data is meticulously processed and subjected to classification by the trained ML models. The outcomes of these predictions are then compared against existing data to gauge the efficiency of the models, a task often assessed through popular quality metrics such as accuracy, precision, recall, or the F1-score.

The principal application runs the core program that carries out all the functionalities of the proposed architecture. It curates the Smart City data, as mentioned before.

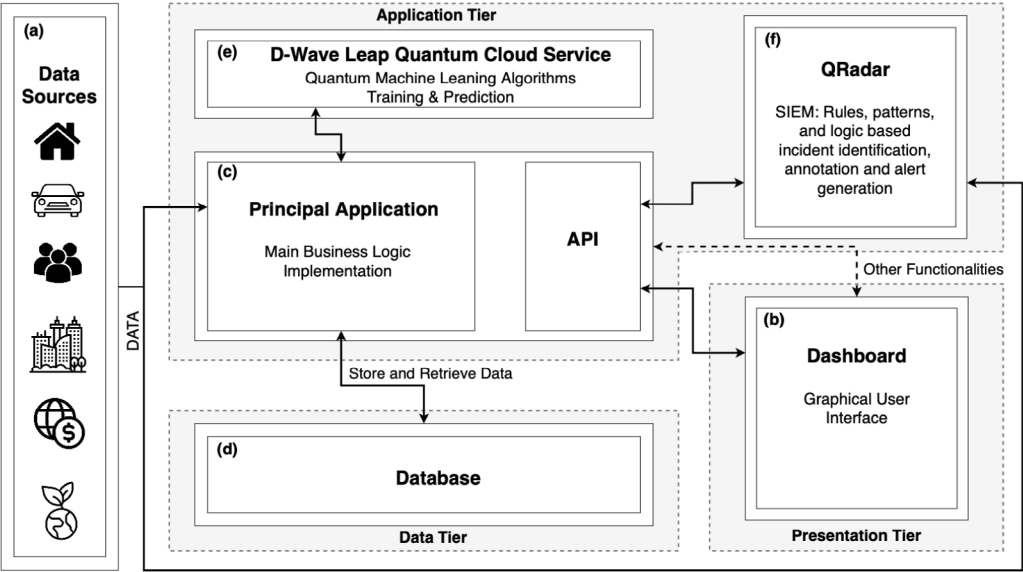


Fig. 5. Hybrid Quantum-Classical Architecture of Smart City Security.

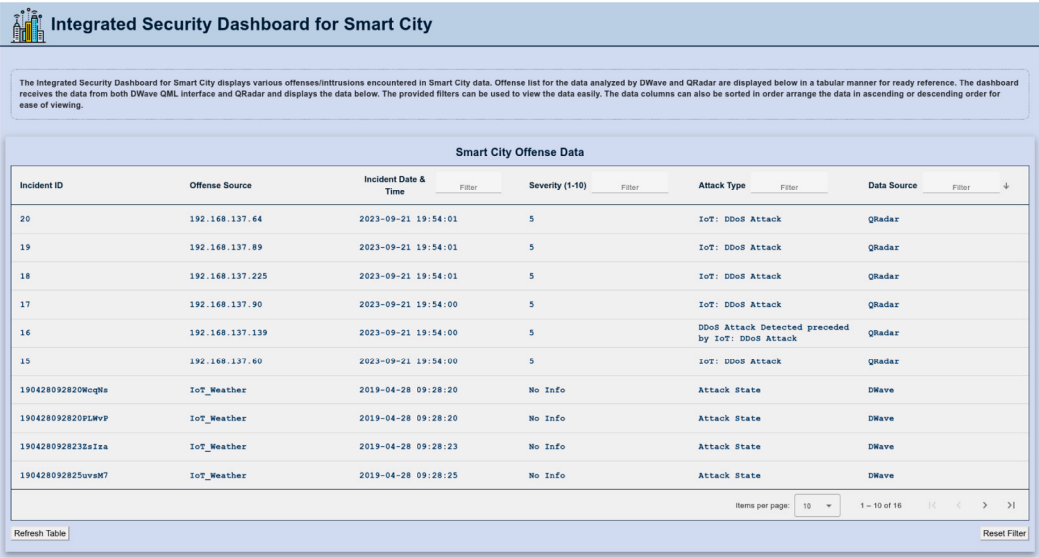


Fig. 6. Integrated Smart City security dashboard.

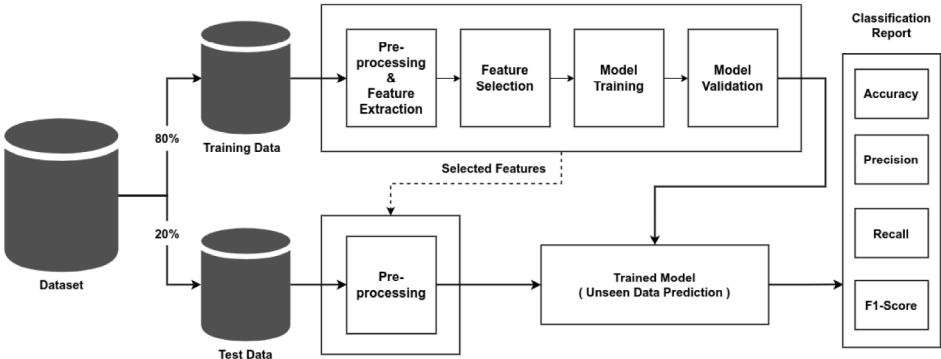


Fig. 7. IDS pipeline.

Within the proposed architecture, the training and testing of data are conducted within the D-Wave Leap Quantum Cloud (DLQC)⁷ (Fig. 5.e), which is external. The application further forms secure connections to DLQC instances through token-based native API. For DLQC, D-Wave Python libraries are harnessed to format data suitably for Quantum Machine Learning (QML), with a focus on employing the QBoost (Neven et al., 2012) algorithm. After the training and validation stages, QML models are preserved locally ready for use in real-time predictions. When the main application initiates real-time predictions, the prediction model is loaded into the Quantum Cloud.

The main application is capable of processing single or batch incident predictions received from data sources. Quantum Cloud assumes responsibility for incident-type predictions, leveraging QML algorithms to distinguish between normal and malicious incidents. Classification results are relayed back to the main application via the token-based API, subsequently updating the database with the most recent attacks, threats, or security incidents.

To maintain the model's effectiveness, the proposed system consistently trains QML models using a diverse array of datasets, including those sourced from reputable organizations or institutions. The best-performing model, determined through multiple runs, is retained for real-time predictions. Ensuring secure and efficient data access, our system employs a RESTful API interface, enabling seamless communication with the database instance. This API features distinct endpoints for data retrieval from various database tables and sources, converting the data into the JSON format for easy data fetch by the dashboard.

The proposed application was developed using Python 3.8, while Flask 2.2.2 was employed to establish the RESTful API for the retrieval of database and file data. The API's capabilities can be seamlessly expanded to encompass additional POST, PUT, PATCH, and DELETE operations in the future, allowing for flexibility to accommodate specific database needs.

4.4. Database

Situated at the core of the proposed architecture, the database (DB) depicted in Fig. 5.d serves as the basis of our data management infrastructure. It acts as the repository for the information streamed from our diverse data sources. Each sector of the Smart City contributes its unique data, which is meticulously organized into dedicated DB tables. These tables, akin to specialized vaults, are tailored to accommodate the distinctive features and information inherent to their respective sectors. This structured approach ensures not only efficient data management but also facilitates quick retrieval.

In order to facilitate seamless data retrieval, we have implemented a RESTful API that adeptly selects data from incident tables associated with each Smart City sector. Serving as a route between the user interface and the database, this API allows for the rapid and secure extraction of data. It acts as a vital link, enabling the GUI to provide real-time insights into incidents occurring across the Smart City's diverse sectors. Furthermore, the main application actively interacts with these DB tables, continuously updating them whenever an incident unfolds within a specific sector. This dynamic interaction ensures that the database remains dynamic, updating in real-time to reflect the continuously evolving events in Smart City.

PostgreSQL 15.2 was chosen as the database technology for the system, which offers the scalability and reliability required for the storage and retrieval of critical information within our Smart City security architecture.

4.5. Quantum Cloud services

The D-Wave Leap Quantum Cloud (DLQC) Service, as delineated in Fig. 5.e, occupies a pivotal role within the proposed architecture affording access to quantum computing capabilities via a secure native API mechanism, fortified by robust token-based encryption for authentication. While DLQC operates as an external service, its logical placement within the application tier is firmly rooted in its role as a strategic enabler of decision-making processes. Within this architectural context, DLQC takes receipt of meticulously pre-processed data originating from the principal application, sending those to a quantum computer encompassing data training and predictive analysis. DLQC harnesses the computational prowess of Quantum Unconstrained Binary Optimization (QUBO) algorithms, a cornerstone of quantum computing. This utilization extends not only to the training of quantum models but also to predicting outcomes, thus harnessing the computational supremacy intrinsic to quantum paradigms. Subsequently, DLQC returns the results of its training and prediction endeavors to the main application, which proceeds with further processing, thereby culminating in a cohesive decision-making framework.

The foundation of the architectural design is supported by the strategic deployment of QBoost (Neven et al., 2012), a quantum algorithm of repute. This algorithm is deployed on the QPU Hybrid Solvers residing on the Advantage_system5.3 platform (DWave QPU solvers for the European region). Additionally, the Leap Quantum Cloud service is available in diverse geographical locations, ensuring access to strategically positioned quantum hybrid solvers on a global scale. The selection of the most appropriate solver is executed judiciously, accounting for regional distinctions and the availability of solvers.

Notably, the solver possesses distinctive characteristics deeply rooted in the principles of quantum annealing, marked by its formidable ensemble of 5615 operational qubits. These qubits operate at an exceedingly low-temperature threshold of 16.4 ± 0.1 mK (millikelvin), a prerequisite for harnessing the unique computational capabilities in quantum mechanical phenomena. In summation, DLQC represents the veritable quantum computational powerhouse within our architectural expanse, endowing our infrastructure with advanced capabilities in data training and prediction, all emanating from the vantage point of quantum computational paradigms.

4.6. QRadar — SIEM

IBM QRadar (Fig. 5.f) forms another vital part of the main application, where it effectively incorporates external threat intelligence data to bolster its threat detection and incident response capabilities. It receives streams of data from a multitude of external sources, including commercial threat intelligence providers, open-source feeds, and customized data repositories. It also maintains reference data sets based on this external threat intelligence, which are continually updated to reflect the latest threat information, ensuring that the system operates with current knowledge. It correlates external threat intelligence with internal security events and logs, identifying patterns, anomalies, or matches that signify potential threats. Custom correlation rules trigger alerts when QRadar detects a potential threat or matches it with external threat indicators.

Risk scores and severity levels are assigned to incidents based on this data, helping security teams prioritize their responses effectively. The pattern and rules-based risk and threat identification are shared with the users over a secure API local to the QRadar implementation site. The use of the community edition of QRadar enables the proposed architecture to exploit the API-based data from both JSON dumps and local API.

QRadar's API empowers users to leverage the platform's rich security data and capabilities in a flexible and customized manner to meet their organization's specific security needs. In the purview of the proposed architecture, the cured and normalized data are fetched from the API to adapt the data to be displayed in the dashboard. The same functionality can also be implemented by taking JSON dumps at regular intervals and updating the dashboard via architecture-side API.

⁷ <https://cloud.dwavesys.com/leap/>

5. Experiments

To validate the proposed architecture, we used IBM QRadar SIEM by creating rules to identify the attacks and an IDS using QBoost with a state-of-the-art dataset called CICIOT2023 (Neto et al., 2023). We used IBM QRadar SIEM Community Edition v7.3.3 (Anon., 2023a) deployed on VirtualBox (Anon., 2023b) 7.0 executed on a machine with an Intel Core(TM) i7-11800H 2.30 GHz, 32 GB of RAM, with 16 GB RAM assigned to IBM QRadar. CICIOT2023 dataset consists of 34 labels, among which 33 are different types of IoT attacks executed in an IoT topology of 105 devices, and one label for *BenignTraffic* (normal data). These attacks are classified into seven categories, namely, DDoS, DoS, Recon, Web-based, brute force, spoofing, and Mirai.

Since the total number of labels is 33, we decided to consider only a part of these attacks to create the rules in IBM QRadar SIEM. In particular, we considered the DDoS, DoS, Recon, and Web-Based labels for the experiments. HTTP-Flood was considered for the DDoS and DoS categories, nmap port scanning for Recon, and XSS and SQL Inject attacks for the Web-Based category. We adopted this approach due to the impracticality of generating an extensive set of rules for our research objectives. The aim of this study is not to scrutinize the functionality of the involved systems but rather to propose a hybrid quantum-classical architecture integrating a quantum-based IDS and classical SIEM. Additionally, we have selected attacks commonly observed in real-world contexts.

On the other hand, in the context of DLQC, since QBoost functions as a binary classifier, the 33 various types of attacks are treated as a unified single *attack* class. During the training and testing of the IDS, attack data is paired against the normal data, which serves as the normal class.

5.1. CICIOT2023

The CICIOT2023 dataset (Neto et al., 2023) is an IoT dataset that consists of seven attack categories: **DDoS**, **DoS**, **Recon**, **Web-Based**, **Brute-Force**, **Spoofing**, and **Mirai**. For each of these categories, the authors have performed different types of attacks. For example, for DDoS, they performed attacks like *UDP Flood* and *HTTP Flood*. These attacks are generated with different tools like nmap, hping3, and golang-httpflood.

To perform these attacks, the researchers utilized an IoT platform comprising 105 IoT devices. Of these, 67 were directly involved in the attacks, while the remaining 38 Zigbee and Z-Wave devices were connected to five hubs (Neto et al., 2023). The network topology is divided into two parts. In the first part, an ASUS router links the network to the Internet, and a Windows 10 machine shares this connection. A Cisco switch is placed between this computer and a VeraPlus access point, connecting seven Raspberry Pi devices responsible for the attacks in the experiments. Traffic is captured using Wireshark via the Cisco switch that is connected to the second part of the network through a Gigamon Network Tap.

The CICIOT2023 has 46.686.579 rows of data with 46 independent variables (features). The dataset is 13.8 Gigabytes (GB) in size. The dataset is made available as a set of 169 different CSV files, which need to be concatenated before experimentation with the full dataset. The data are labeled with 33 attack types: *DDoS-ICMP_Flood*, *DDoS-UDP_Flood*, *DDoS-TCP_Flood*, *DDoS-PSHACK_Flood*, *DDoS-SYN_Flood*, *DDoS-RSTFIN_Flood*, *DDoS-SynonymousIP_Flood*, *DoS-UDP_Flood*, *DoS-TCP_Flood*, *DoS-SYN_Flood*, *Mirai-greeth_flood*, *Mirai-udpplain*, *Mirai-greip_flood*, *DDoS-ICMP_Fragmentation*, *MITM-ArpSpoofing*, *DDoS-ACK_Fragmentation*, *DDoS-UDP_Fragmentation*, *DNS_Spoofing*, *Recon-HostDiscovery*, *Recon-OSScan*, *Recon-PortScan*, *DoS-HTTP_Flood*, *VulnerabilityScan*, *DDoS-HTTP_Flood*, *DDoS-SlowLoris*, *DictionaryBrute-Force*, *BrowserHijacking*, *SqlInjection*, *CommandInjection*, *XSS*, *Backdoor_Malware*, *Recon-PingSweep*, *Uploading_Attack*; and one normal data label, *BenignTraffic*.

In the context of ML algorithms, the dataset is extremely unbalanced, which means that the distribution of classes or categories is significantly skewed or imbalanced. For example in a more granular level the dataset has about 15.5% records labeled as *DDoS-ICMP_Flood*, 11.6% of records labeled as *DDoS-UDP_Flood*, and 9.7% records labeled as *DDoS-TCP_Flood*, whereas there are only 0.002%, 0.005% and 0.006% records labeled as *Uploading_Attack*, *Recon-PingSweep*, and *Backdoor_Malware*. Further, normal messages (*BenignTraffic*) consist of only 2.3% of the dataset, which makes prediction difficult using ML classification algorithms.

5.2. DWave leap quantum machine learning

To validate the functionality of the proposed architecture, the CICIOT2023 dataset was subject to the following steps to prepare it for machine learning analysis using the QBoost QML algorithm from the DLQC:

1. *Categorical Feature Removal*: Categorical features, which represent non-numeric data, were removed from the dataset. This was done to reduce the dimensionality of the data, as dealing with high-dimensional data can potentially reduce the quality of prediction models.
2. *Redundant Feature Removal*: Features that had the same values across all instances were removed from the dataset. These features do not contribute to the prediction process and were thus eliminated to simplify the data.
3. *Handling Missing Values*: Any instances containing Infinity or NaN (Not-a-Number) values were processed by replacing them with the median values. This step ensures that the data is complete and suitable for analysis.
4. *Target Variable Transformation*: The dependent feature, which is the target variable, underwent a transformation. Normal or non-attack messages were assigned a value of -1, while attack messages were assigned a value of 1. This transformation is essential for compatibility with quantum machine learning algorithms.
5. *Data Normalization*: The independent variables, or features, in the dataset, were normalized using the *MinMaxScaler*⁸ from the *Scikit-Learn* library. Normalization scales the values to fall within the range of 0 and 1, ensuring uniformity in feature scaling.
6. *Balancing the Dataset*: Addressing class imbalance is critical in machine learning. To balance the dataset, a combination of techniques was applied, including Synthetic Minority Over-sampling Technique (SMOTE)⁹ and Random Under-Sampling, both from the *imblearn* library and the best parameters for balancing were decided by *CrossValidation* of the mean recall scores for the *DecisionTree* classifier. These methods were employed to achieve a more equitable distribution of class instances, ensuring that both attack and non-attack classes are adequately represented. After balancing, the resulting dataset was reduced to 10.257.385 rows.
7. *Dimensionality Reduction*: To mitigate the challenges posed by a high-dimensional dataset (consisting of 46 features), Principal Component Analysis (PCA)¹⁰ was applied (from *scikit-learn* library). PCA helps reduce the dimensionality while preserving the most informative features. The threshold method was used, based on the explained variance ratio, to retain the top 13 most relevant features.

⁸ <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>

⁹ https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

¹⁰ <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

8. **Data Splitting:** Finally, the processed and scaled data were split into training and testing datasets in an 80% to 20% ratio (Neto et al., 2023). This division is essential for evaluating the model's performance and ensuring that it generalizes well to unseen data.

In the final stage of the analysis, several key performance metrics were computed to evaluate and compare the effectiveness of the algorithm within the proposed architecture. These metrics include model accuracy, precision (quantifying the proportion of true positive predictions), recall (measuring the proportion of actual positives correctly anticipated), and the F1 score (a harmonic amalgamation of precision and recall). These metrics serve as critical benchmarks for assessing the predictive capabilities of the QBoost model in the context of Smart City Security.

Within the proposed architecture, particular emphasis was placed on the deployment of the QBoost (Neven et al., 2012) QML algorithm as the primary model for validation purposes. The operationalization of QBoost involved an iterative process that entailed training the model on the meticulously preprocessed and balanced dataset, which had previously undergone dimensionality reduction via PCA. Subsequently, the model was subjected to rigorous testing on a distinct dataset to evaluate its predictive capacity. The quality performance metrics collectively furnished a comprehensive understanding of QBoost's efficiency and suitability within the proposed architectural framework.

In the interest of comprehensiveness and impartiality, a comparative analysis was also conducted. To this end, the performance of QBoost was compared with that of Random Forest (RF) (Breiman, 2001), a state-of-the-art traditional machine learning algorithm. This comparison aimed to provide a holistic view of the strengths and weaknesses of both quantum and classical approaches in the context of Smart City Security.

For the sake of comprehensiveness and reproducibility, it is imperative to document the hyperparameters employed during the aforementioned experiments. These hyperparameters are provided herewith:

1. SMOTE Hyperparameters:

- (a) *sampling_strategy*: 0.1, indicating the desired ratio of the number of synthetic samples to be generated for the minority class compared to the majority class.
- (b) *k*: 7, determining the number of nearest neighbors considered during the synthetic sample generation process.

2. Random Undersampling Hyperparameters:

- (a) *sampling_strategy*: 0.8, presenting the desired ratio of the number of instances to be retained for the majority class compared to the minority class.

3. RepeatedStratifiedKFold Configuration:

- (a) *n_splits*: 10, indicating the number of splits or folds in the cross-validation process.
- (b) *n_repeats*: 3, denoting the number of times the cross-validation process was repeated.
- (c) *n_jobs*: -1, enabling parallel processing when calculating cross-validation scores.

4. PCA Configuration:

- (a) *Threshold method*: Relies on the explained variance ratio.
- (b) *Threshold value*: 0.99, indicating the proportion of variance that the selected principal components should collectively account for.

5. *QBoost λ value (λ)*: 0.085, introduced to control the complexity of the model and reduce the risk of overfitting.

This meticulous and thorough evaluation process, encompassing both quantum and classical machine learning paradigms, was designed to elucidate invaluable insights into the relative merits and demerits of these approaches within the distinctive precincts of Smart City Security.

5.3. IBM QRadar SIEM rules

The following subsection describes the rules created with QRadar to identify the considered attacks.

5.3.1. DDoS

A Distributed Denial of Service (DDoS) attack is a malicious attempt to disrupt the normal functioning of a network, service, or website by overwhelming it with a flood of traffic. The goal of a DDoS attack is to make the target system or network unavailable to its intended users. A large number of compromised computers typically carry out DDoS attacks, often referred to as a "botnet", that are controlled by the attacker. In the CICIOT23 dataset, the authors utilized multiple Raspberry Pi devices in an SSH-based master-client configuration to execute these attacks (Neto et al., 2023).

To create the rules that identify a DDoS attack, we first use a series of building blocks, and then we create the main rule that aggregates the building blocks in order to identify the attack. A building block (BB) group commonly uses tests to build complex logic so that it can be used in rules.

Fig. 8 shows the rules created to identify an HTTP Flood attack with the category DDoS. Figs. 8(c) and 8(d) show the BB rules. In particular, Fig. 8(d) is used to filter the flow network based on the TCP protocol and port 80 since the HTTP Flood is performed by sending a GET request on port 80. Fig. 8(c) shows the Superflow characteristics. IBM QRadar groups the individual flows into a Superflow. There are three types of Superflow: A (Network Scan), B (DDoS), and C (Port Scan) (Anon., 2023d). QRadar looks for flows where many hosts send data to one destination host and flags this activity as Superflow B.

Fig. 8(b) is used to identify two or more source IPs that send a high number of data to a single destination IP in two minutes. Finally, the rule *IoT: DDoS Attack* (Fig. 8(a)) aggregates the Superflow and the "Probable DDoS". In particular, if one of those rules is true, an offense will be raised. An important consideration is that it is possible to modify the rule in Fig. 8(b) to be a DoS and not a DDoS. For example, we could modify the source IP from two(2) to one(1).

Fig. 9 shows the offense generated when IBM QRadar SIEM identifies an HTTP Flood attack with a DDoS category. In the generated offense, it is possible to see that we have multiple *Offense Source* and *Source IPs* that correspond to the Raspberry Pi that performs the attack to one *Destination IP*. The events column indicates the number of event flows where a DDoS attack was identified. Since the CICIOT2023 does not provide the list of IP devices but only the Media Access Control (MAC) addresses, we have used this information to understand the attack messages by analyzing the PCAP file. A PCAP (short for Packet Capture) file is a type of file format used to store network traffic data. It is commonly associated with network analysis and troubleshooting tools.

5.3.2. Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) is a type of web security vulnerability that occurs when a web application allows users to inject malicious scripts into web pages viewed by other users. XSS attacks are one of the most common security vulnerabilities found in web applications. They can have serious consequences, including data theft, session hijacking, and defacement of websites (Wassermann and Su, 2008). To perform this attack, the researchers used DVWA (Anon., 2017), a vulnerable PHP/MySQL web application usually employed to test penetration testing techniques or for education purposes.

To create the rule that identifies an XSS attack, we first analyzed the PCAP file. We have seen that most of the messages are related to basic XSS injection techniques, such as using the tag *script* with *alert*. Fig. 10 shows the rule that matches several tags generally used to perform an XSS attack with a *regular expression*.

Figs. 11(a) and 11(b) show the generated offense when an XSS attack is identified by QRadar using the rule in Fig. 10.

Apply on flows which are detected by the system

   and when a flow matches any of the following BB:Threat:DDoS Superflow, IoT: Probable DDoS

(a)

Apply on flows which are detected by the system

   and when any of these BB:Threat:IoT D/DoS TCP with the same destination IP more than 3 times, across more than 2 source IP within 2 minutes

(b)

Apply on flows which are detected by the system

   and when the flow type is one of Superflow B

(c)

Apply on flows which are detected by the system

   and when the IP protocol is one of the following TCP:tcp_ip

   and when the destination port is one of the following 80

(d)

Fig. 8. The pictures show the rules to identify an HTTP Flood attack with the category DDoS. Figure (a) indicates the main rule. If one of the rules indicated in (a) is positive, it will raise an offense. Figures (c) and (d) show the BB rules.

Description	Offense Type	Offense Source	Magnitude	Source IPs	Destination IPs	Us	Lo So	Events
IoT: DDoS Attack	Source IP	192.168.137.90		192.168.137.90	192.168.137.139	N/A	Cus	113
IoT: DDoS Attack	Source IP	192.168.137.225		192.168.137.225	192.168.137.139	N/A	Cus	104
IoT: DDoS Attack	Source IP	192.168.137.89		192.168.137.89	192.168.137.139	N/A	Cus	118
IoT: DDoS Attack	Source IP	192.168.137.64		192.168.137.64	192.168.137.139	N/A	Cus	102

Fig. 9. Offense when a DDoS attack is identified.

Apply on flows which are detected by the system

   and when the source payload matches the regex (script|alert|onmouse[a-z]+|onkey[a-z]+|onload|onunload|ondragdrop|onblur|onfocus|onclick|ondblclick|onsubmit|onreset|onselect|onchange)

Fig. 10. Rule to identify an XSS attack.

Description	IoT: XSS Detection	Offense Type	Source IP
		Event/Flow count	<u>20 events</u> and <u>0 flows</u> in 1 categories
Source IP(s)	<u>192.168.137.147</u>	Start	Sep 20, 2023, 5:36:03 PM
Destination IP(s)	<u>Local (3)</u>	Duration	39m 47s
Network(s)	<u>Net-10-172-192_Net_192_168_0_0</u>	Assigned to	<u>Unassigned</u>

(a)

Description	Offense Type	Offense Source	Magnitude	Source IPs	Destination IPs
IoT: XSS Detection	Source IP	192.168.137.147		192.168.137.147	Local (3)

(b)

Fig. 11. Figure (a) shows information about the offense when an XSS attack is detected. Figure (b) shows more information about the offense.

Apply on flows which are detected by the system

   and when the source payload matches the regex (UNION|ALL|SELECT|OR|AND|CONCAT|WHEN|SELECT|FROM|NULL|SLEEP)

   and when the destination port is one of the following 80

Fig. 12. Rule to identify a SQL Injection attack.

5.3.3. SQL injection

A SQL injection is a web security vulnerability that occurs when an attacker is able to manipulate the input that is sent to a web application's SQL database. This manipulation can lead to unauthorized access, modification, or extraction of data from the database (Halfond et al., 2006). As with the XSS attack, the researchers used the DVWA application to perform the SQL Injection attack.

We first analyzed the PCAP file regarding the SQL Injection and noted that this attack had been performed using basic SQL Injection with UNION, SLEEP, and so on. So, based on this, we created a rule containing the most common SQL fields used in an SQL Injection attack. Fig. 12 shows the rule with the regular expression created.

Instead, Fig. 13 shows the generated offense. In particular, the field *Local (4)* is related to the number of the destination IPs involved

Description	Offense Type	Offense Source	Magr	Source IPs	Destination IPs	Use	Log Sou	Events
IoT: SQL Injection	Source IP	192.168.137.178	■■■	192.168.137.178	Local (4)	N/A	C...	170

Fig. 13. SQL Injection Offense generated.

Description	IoT SQL Injection		Offense Type	Source IP	
Source IP(s)	192.168.137.178 (192.168.137.178)		Event/Flow count	170 events and 0 flows in 1 categories	
Destination IP(s)	Local (4)		Start	Sep 29, 2023, 5:09:00 PM	
Network(s)	Net-10-172-192-Net, 192.168.0.0		Duration	48m 59s	
			Assigned to	Unassigned	

List of Local Destination IPs

Destination IP	Magnitude	Location	Vulnerability	User	MAC	Weight	Offenses	Source(s)	Last Event/Flow	Events/Flows
192.168.137.4		Net-10-172-192	No	?	?	0	4	1	51m 10s	209
192.168.137.217		Net-10-172-192	No	?	?	0	2	1	4m 11s	103
192.168.137.154		Net-10-172-192	No	?	?	0	2	1	4m 19s	4
192.168.137.86		Net-10-172-192	No	?	?	0	2	1	32m 10s	44

Fig. 14. SQL Injection with more detail obtaining the destination IPs involved in the attack.

Apply on flows which are detected by the system
 and when a flow matches any of the following Port Scanning: Over 10 Seconds

(a)

Apply on flows which are detected by the system
 and when all of these BB:Threat:IoT Port Scanning, in any order, from any source IP to the same destination IP, over 10 seconds

(b)

Apply on flows which are detected by the system
 and when the source byte/packet ratio is equal to 64 bytes/packet
 and when the IP protocol is one of the following TCP,tcp_ip

(c)

Fig. 15. The pictures show the rules to identify a port scan. Figure (a) indicates the main rule. If one of the rules indicated in (a) is positive, it will raise an offense. Figure (c) indicates a BB rule.

in the attack. We can obtain these IPs and other information by double-clicking the generated offense as Fig. 14 shows.

5.3.4. Port scan

A port scan is a reconnaissance technique to discover which network ports on a target system are open and listening for incoming connections (Bhuyan et al., 2011). Network administrators often employ port scanning for legitimate purposes such as network troubleshooting and security assessments. However, attackers can also use it maliciously to identify potential vulnerabilities in a target system. The researchers used a tool called *Nmap* (Lyon, 2009) to perform a port scan.

To create the rules (Fig. 15) to identify a port scan with XSS and SQL Injection attack, we analyze the PCAP file that corresponds to the Port Scan attack. In particular, we have shown that most messages sent with Nmap have a byte length of 64. So, with this information, we created the first rule (Fig. 15(c)). In addition, the rule will match only if a port scan is performed on the TCP protocol. Then, we created a second rule (Fig. 15(b)) that matches the flow traffic if any device sends a scanning packet (SYN, SYN-ACK, RST) at the same device (destination IP) for over 10 s. Finally, the SIEM (Fig. 15(a)) will raise an offense if the rule Port Scanning: Over 10 Seconds is true.

Fig. 16 shows the offense generated when a Port Scan attack is identified. We can see that we have one Source IP that performs the attack on one destination IP.

6. Results and discussion

In this section, we present and analyze the results obtained from our comprehensive experimentation, shedding light on the performance and efficacy of the proposed architecture. Apart from the key quality metrics like model accuracy, precision, recall, and F1-Score, training

and testing/prediction time efficiency were also taken into consideration.

Fig. 17 provides an overview of the experimental results on the dataset, focusing on the performance of RF and QBoost classifiers. Notably, both exhibited comparable and commendable classification quality. While RF performs excellently, QBoost, a quantum machine learning algorithm, also demonstrates competitive classification quality. This hints at the potential of quantum computing in enhancing data analysis and predictive tasks in Smart Cities. As quantum computing technology advances and becomes more accessible, it may open up new avenues for tackling complex Smart City challenges.

Table 1 shows the training and prediction times associated with the mentioned algorithms. Notably, these times provide valuable insights into the computational efficiency of each approach. While the primary objective of this study does not center on a direct comparison between QBoost and RF, the results substantiate the feasibility of considering QBoost as a viable candidate for implementation within a Smart City framework. It is important to underscore that both classifiers exhibited remarkably similar results regarding quality metrics, suggesting that QBoost can be a viable alternative to RF. Where the accuracy of RF was 99%, QBoost attained 96% accuracy in classifying the attacks. In other words, RF was ahead of QBoost, with a margin of a mere 3% for all quality metrics.

However, what sets QBoost apart is its notable advantage in terms of computational efficiency. Specifically, we observed an impressive reduction of approximately 80% in training time and a substantial 71% decrease in prediction times when comparing QBoost with RF. This substantial improvement in efficiency is particularly evident in the case of prediction times, where QBoost achieved a remarkable processing speed of just 2.24 s. Even when dealing with datasets containing over

Description	Offense Type	Offense Source	Magnitude	Source IPs	Destination IPs	Us	Lo So	Events
IoT: Possible Port Scanning	Source IP	192.168.137.41		192.168.137.41	192.168.137.178	N/A	Cus	1,192

(a)

Description	IoT: Possible Port Scanning	Offense Type	Source IP
Source IP(s)	192.168.137.41 (192.168.137.41)	Event/Flow count	1,192 events and 0 flows in 1 categories
Destination IP(s)	192.168.137.178 (192.168.137.178)	Start	Sep 21, 2023, 7:36:00 PM
Network(s)	Net-10-172-192.Net 192.168.0.0	Duration	1m 59s
		Assigned to	Unassigned

(b)

Fig. 16. Figure (a) shows the offense generated when a Port Scan attack is detected. Instead, Figure (b) shows the offense generated but with more details.

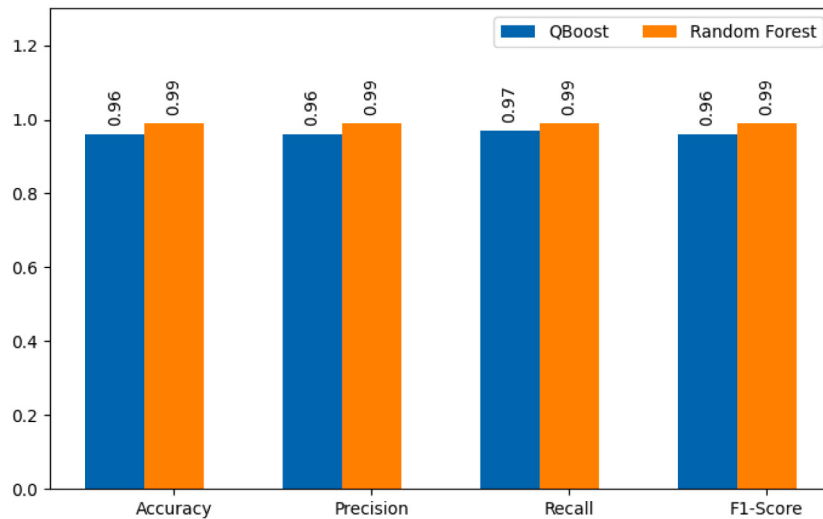


Fig. 17. Quality metrics of classifiers.

Table 1

Time performance analysis between QBoost and Random Forest.

Algorithm	Training time (s)	Prediction time (s)
QBoost	554.5	2.24
Random Forest	2667.5	7.83

a million data entries, the rapidity of QBoost's predictions remains a standout feature.

On the other hand, with IBM QRadar SIEM, we note that the time between the identification of an attack and the offense alert varies from about 2 to 3 min. This depends on whether we are using a version deployed on a virtual machine and not in a cloud environment. By using more efficient hardware, this time could decrease.

The coexistence of QML (QBoost) models and QRadar SIEM suggests the possibility of hybrid approaches. Future research could explore the synergy between classical and quantum computing to harness the strengths of both paradigms for improved Smart City solutions. It is evident from the results of the experiments that a hybrid approach of using a QML, QBoost in this case, and IBM QRadar reinforces the security aspects of Smart City incidents by detecting offenses or security incidents in real time.

The Integrated Security Dashboard provides a seamless means to monitor and study the security incidents captured by both QRadar and QBoost (DLQC). For example, considering Fig. 6, we can see samples of DDoS attacks identified by QRadar (Incident ID from 15 to 20) obtained with QRadar's API and by QBoost (the last four alphanumeric Incident IDs) using DWave. Specifically, the implemented dashboard

can support the SOC specialists for faster and more accurate decision-making. Also, the dataset used in the experiments likely represents a fraction of the complexity encountered in real-world Smart City scenarios. The positive results indicate that as Smart City data grows in scale and complexity, machine learning techniques can play a pivotal role in extracting valuable insights and addressing security and efficiency challenges.

In the proposed architecture, we have investigated a specific quantum platform (DWave) and an algorithm (QBoost), along with a traditional SIEM (IBM QRadar). To generalize the seamless integration of classical and quantum routines, it is necessary to identify the tasks that would benefit the most from quantum acceleration. Real-world data is classical, stipulating classical computational techniques for pre-processing, postprocessing, and error correction. As mentioned earlier (Section 3.2.2), QML algorithms like Variational Quantum Classifier (VQC), Quantum Neural Network (QNN), and Quantum Support Vector Machine (QSVM) can be used to carry out specific computationally intensive tasks, however, due to hardware constraints, quantum devices are still limited in scale and reliability.

Training of large and complex real-world datasets is not yet scalable on quantum hardware due to quantum decoherence and noise, leading to nonreliable predictions or optimizations. Apparently, quantum simulators can serve as an alternative to quantum hardware, but it is important to understand that it is only a simulation of quantum hardware on classical computers. Therefore, a quantum algorithm is actually using the power of the CPU and GPU only, and not quantum hardware.

However, simulating quantum algorithms classically can sometimes be better than classical methods. This behavior can be attributed to the

fact that quantum algorithms approach the problem in a paradigmatically different way and, thus, might prove effective in some scenarios in a quantum simulator. On many occasions, for high dimensional data, QML algorithms in quantum simulators exhaust available system memory in controlled experimental setups with considerable RAMs, for example, 64 GB. Moreover, training and prediction using large and complex datasets (more than 50 features) are slower, and the results are below par in comparison to classical ML algorithms.

Having said that, continuous refinement and optimization of this integration will be crucial for realizing the full potential of quantum computing algorithms in data analysis pipelines using real quantum hardware and state-of-the-art QML algorithms, along with classical techniques, to enhance data prediction quality.

7. Conclusion

A Smart City encompasses various sectors, including mobility, economy, government, and environment, and prioritizes the well-being of its residents. Such an interconnected urban environment generates vast volumes of data through IoT sensors, cloud computing, and mobile technologies. This data repository holds crucial information regarding these city sectors and records critical incident data. In this paper, we propose a Hybrid Quantum-Classical Architecture for Smart City Security that seamlessly integrates both D-Wave Leap Quantum Cloud (DLQC) and IBM QRadars SIEM for comprehensive security monitoring and control. Our proposed system features an Integrated Security Dashboard for Smart Cities, providing real-time insights into security incidents sourced from both QRadars and DLQC. The quality metrics of the Quantum Machine Learning (QML) algorithm, QBoost, implemented in DLQC, and the traditional RF algorithm yielded comparable results. Notably, QBoost demonstrated an impressive performance advantage, identifying security threats with over 70% greater speed than RF.

In practical terms, this enhanced computational efficiency holds significant promise for Smart City applications. The ability to swiftly process and predict data, even at such a large scale, is of paramount importance in the context of real-time monitoring and decision-making within a Smart City's dynamic environment. Thus, while our primary focus is on the architecture's feasibility and the integration of Quantum Machine Learning (QML) with IBM QRadars, the promising performance of QBoost in terms of speed opens up new possibilities for enhancing the overall responsiveness and effectiveness of Smart City security systems.

Smart City data can be highly complex to handle owing to the plethora of aspects from which it brings the data. To this end, although the literature boasts of intelligent algorithms and operations centers to manage complexities and conflicts, it would be extremely interesting to investigate the viability of quantum machine learning algorithms like Quantum Restricted Boltzmann Machines, Quantum Neural Networks, and Quantum Support Vector Machines to analyze such complex data. Also, we are working continuously on improving the Integrated dashboard to imbibe more functionalities for greater independence. Moreover, we want to work on implementing Quantum Key Distribution for secure data transmission in the current context.

CRedit authorship contribution statement

Vita Santa Barletta: Conceptualization, Data curation, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Danilo Caivano:** Conceptualization, Supervision, Writing – review & editing. **Mirko De Vincentiis:** Investigation, Software, Writing – original draft. **Anibrata Pal:** Methodology, Software, Writing – original draft. **Michele Scalera:** Data curation, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This study has been partially supported by the following projects: “QUASAR: QUAntum software engineering for Secure, Affordable, and Reliable systems”, grant 2022T2E39C, under the PRIN 2022 MUR program funded by the EU -NGEU; SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU; PON Ricerca e Innovazione 2014–2022 FSE REACT-EU, Azione IV.4 “Dottorati e contratti di ricerca su tematiche dell'innovazione” CUP:H99J21010060001.

References

- Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M., Ming, H., 2019. AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference. CCWC, pp. 0305–0310. <http://dx.doi.org/10.1109/CCWC.2019.8666450>.
- Anon., 2016. Post-quantum cryptography. <https://csrc.nist.gov/projects/post-quantum-cryptography>. (Accessed 10 May 2024).
- Anon., 2017. DVWA damn vulnerable web application. <https://github.com/digininja/DVWA>. (Accessed 28 September 2023).
- Anon., 2023a. IBM QRadars SIEM community edition. <https://www.ibm.com/community/qradar/ce/>. (Accessed 29 September 2023).
- Anon., 2023b. Oracle VM VirtualBox. <https://www.virtualbox.org/>. (Accessed 29 September 2023).
- Anon., 2023c. QRadars architecture overview. <https://www.ibm.com/docs/en/qsis/7.4?topic=deployment-qradar-architecture-overview>. (Accessed 27 September 2023).
- Anon., 2023d. Superflow. <https://www.ibm.com/docs/en/qsis/7.4?topic=monitoring-superflows>. (Accessed 28 September 2023).
- Babar, M., Tariq, M.U., Jan, M.A., 2020. Secure and resilient demand side management engine using machine learning for IoT-enabled smart grid. Sustainable Cities Soc. (ISSN: 2210-6707) 62, 102370. <http://dx.doi.org/10.1016/j.scs.2020.102370>.
- Barletta, V.S., Buono, P., Caivano, D., Dimauro, G., Pontrelli, A., 2021. Deriving smart city security from the analysis of their technological levels: A case study. In: 2021 IEEE International Conference on Omni-Layer Intelligent Systems. COINS, pp. 1–6. <http://dx.doi.org/10.1109/COINS51742.2021.9524268>.
- Barletta, V.S., Caivano, D., De Vincentiis, M., Magri, A., Piccinno, A., 2023a. Quantum optimization for IoT security detection. In: Julián, V., Carneiro, J.a., Alonso, R.S., Chamoso, P., Novais, P. (Eds.), Ambient Intelligence—Software and Applications—13th International Symposium on Ambient Intelligence. Springer International Publishing, Cham, ISBN: 978-3-031-22356-3, pp. 187–196.
- Barletta, V.S., Caivano, D., De Vincentiis, M., Ragone, A., Scalera, M., Martín, M.A.S., 2023b. V-SOC4AS: A vehicle-SOC for improving automotive security. Algorithms (ISSN: 1999-4893) 16 (2), <http://dx.doi.org/10.3390/a16020112>.
- Barletta, V.S., Caivano, D., Lako, A., Pal, A., 2023c. Quantum as a service architecture for security in a smart city. In: International Conference on the Quality of Information and Communications Technology. Springer, ISBN: 978-3-031-43703-8, pp. 76–89.
- Bhavsar, M., Roy, K., Kelly, J., Olusola, O., 2023. Anomaly-based intrusion detection system for IoT application. Discov. Internet of Things 3 (1), 5.
- Bhowmick, A., Francellino, E., Glehn, L., Loredi, R., Nesbitt, P., Yu, S.W., 2012. IBM Intelligent Operations Center for Smarter Cities Administration Guide. IBM Corporation, International Technical Support Organization.
- Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K., 2011. Surveying port scans and their detection methodologies. Comput. J. 54 (10), 1565–1581.
- Boixo, S., Smelyanskiy, V.N., Shabani, A., Isakov, S.V., Dykman, M., Denchev, V.S., Amin, M.H., Smirnov, A.Y., Mohseni, M., Neven, H., 2016. Computational multi-bit tunnelling in programmable quantum annealers. Nature Commun. (ISSN: 2041-1723) 7 (10327), 1–7. <http://dx.doi.org/10.1038/ncomms10327>.
- Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.
- Caivano, D., De Vincentiis, M., Nitti, F., Pal, A., 2022. Quantum optimization for fast CAN bus intrusion detection. In: Proceedings of the 1st International Workshop on Quantum Programming for Software Engineering. In: QP4SE 2022, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450394581, pp. 15–18. <http://dx.doi.org/10.1145/3549036.3562058>, URL <https://doi.org/10.1145/3549036.3562058>.
- Chen, S.Y.-C., Yoo, S., Fang, Y.-L.L., 2022. Quantum long short-term memory. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 8622–8626.
- Farhi, E., Goldstone, J., Gutmann, S., 2014. A quantum approximate optimization algorithm. [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- Farhi, E., Goldstone, J., Gutmann, S., Sipser, M., 2000. Quantum computation by adiabatic evolution. [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106).

- Feynman, R.P., 1982. Simulating physics with computers. *Internat. J. Theoret. Phys.* 21 (6/7), 467–488.
- Gigante, D., Pecorelli, F., Barletta, V.S., Janes, A., Lenarduzzi, V., Taibi, D., Baldassarre, M.T., 2023. Resolving security issues via quality-oriented refactoring: A user study. In: 2023 ACM/IEEE International Conference on Technical Debt, TechDebt. pp. 82–91. <http://dx.doi.org/10.1109/TechDebt59074.2023.00016>.
- Grover, L.K., 1996. A fast quantum mechanical algorithm for database search. In: STOC'96: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. ACM, pp. 212–219.
- Halfond, W.G., Viegas, J., Orso, A., et al., 2006. A classification of SQL-injection attacks and countermeasures. In: Proceedings of the IEEE International Symposium on Secure Software Engineering, Vol. 1. IEEE, pp. 13–15.
- Hekkala, J., Muurman, M., Halunen, K., Vallivaara, V., 2023. Implementing post-quantum cryptography for developers. *SN Comput. Sci.* 4 (4), 365.
- Herr, D., Obert, B., Rosenkranz, M., 2021. Anomaly detection with variational quantum generative adversarial networks. *Quantum Sci. Technol.* 6 (4), 045004.
- Hwoij, A., Khamaiseh, A., Ababneh, M., 2021-04-05. SIEM architecture for the Internet of Things and smart city. In: International Conference on Data Science, E-Learning and Information Systems 2021. ACM, ISBN: 978-1-4503-8838-2, pp. 147–152. <http://dx.doi.org/10.1145/3460620.3460747>.
- Iouliauou, P., Vasilakis, V., Moscholios, I., Logothetis, M., 2018. A signature-based intrusion detection system for the internet of things. *Inf. Commun. Technol. Form.*
- Jia, Y., Gu, Z., Du, L., Long, Y., Wang, Y., Li, J., Zhang, Y., 2023. Artificial intelligence enabled cyber security defense for smart cities: A novel attack detection framework based on the MDATA model. *Knowl.-Based Syst.* (ISSN: 0950-7051) 276, 110781. <http://dx.doi.org/10.1016/j.knsys.2023.110781>.
- Jiang, C., Qiu, Y., Gao, H., Fan, T., Li, K., Wan, J., 2019. An edge computing platform for intelligent operational monitoring in internet data centers. *IEEE Access* 7, 133375–133387. <http://dx.doi.org/10.1109/ACCESS.2019.2939614>.
- Johnson, M.W., Amin, M.H.S., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A.J., Johansson, J., Bunyk, P., Chapple, E.M., Enderud, C., Hilton, J.P., Karimi, K., Ladizinsky, E., Ladizinsky, N., Oh, T., Perminov, I., Rich, C., Thom, M.C., Tolkacheva, E., Truncik, C.J.S., Uchaikin, S., Wang, J., Wilson, B., Rose, G., 2011. Quantum annealing with manufactured spins. *Nature* (ISSN: 1476-4687) 473 (7346), 194–198. <http://dx.doi.org/10.1038/nature10012>, arXiv:21562559.
- Kadowaki, T., Nishimori, H., 1998. Quantum annealing in the transverse Ising model. *Phys. Rev. E* 58 (5), 5355–5363. <http://dx.doi.org/10.1103/PhysRevE.58.5355>.
- Kaye, P., Laflamme, R., Mosca, M., Kaye, P., Laflamme, R., Mosca, M., 2006. An Introduction to Quantum Computing. Oxford University Press, Oxford, England, UK, ISBN: 978-0-19857049-3, URL <https://global.oup.com/academic/product/an-introduction-to-quantum-computing-9780198570493>.
- Lyon, G.F., 2009. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure.
- Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., Ghorbani, A.A., 2023. CICIOT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors* (ISSN: 1424-8220) 23 (13), <http://dx.doi.org/10.3390/s23135941>.
- Neven, H., Denchev, V., Rose, G., Macready, W., 2012. QBoost: Large scale classifier training with adiabatic quantum optimization. *J. Mach. Learn. Res.* 25, 333–348.
- Nielsen, M.A., Chuang, I.L., 2010. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, Cambridge, England, UK, ISBN: 978-0-51197666-7, <http://dx.doi.org/10.1017/CBO9780511976667>.
- Piattini, M., Murillo, J.M., 2022. Quantum software engineering landscape and challenges. In: Quantum Software Engineering. Springer, pp. 25–38.
- Piattini, M., Peterssen, G., Pérez-Castillo, R., Hevia, J.L., Serrano, M.A., Hernández, G., de Guzmán, I.G.R., Paradela, C.A., Polo, M., Murina, E., et al., 2020. The talavera manifesto for quantum software engineering and programming. In: QANSWER. pp. 1–5.
- Rajapaksha, S., Kalutarage, H., Al-Kadri, M.O., Petrovski, A., Madzudzo, G., Cheah, M., 2023. AI-based intrusion detection systems for in-vehicle networks: A survey. *ACM Comput. Surv.* 55 (11), 1–40.
- Salek, M.S., Biswas, P.K., Pollard, J., Hales, J., Shen, Z., Dixit, V., Chowdhury, M., Khan, S.M., Wang, Y., 2023. A novel hybrid quantum-classical framework for an in-vehicle controller area network intrusion detection. *IEEE Access*.
- Sánchez-Corcuera, R., Nuñez-Marcos, A., Sesma-Solance, J., Bilbao-Jayo, A., Mulero, R., Zulaika, U., Azkune, G., Almeida, A., 2019. Smart cities survey: Technologies, application domains and challenges for the cities of the future. *Int. J. Distrib. Sens. Netw.* 15 (6), 1550147719853984.
- Shor, P.W., 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* 26 (5), 1484–1509.
- Suryotrisongko, H., Musashi, Y., 2022. Evaluating hybrid quantum-classical deep learning for cybersecurity botnet DGA detection. *Procedia Comput. Sci.* 197, 223–229.
- Tan, H., Wang, L., Zhang, H., Zhang, J., Shafiq, M., Gu, Z., 2022. Adversarial attack and defense strategies of speaker recognition systems: A survey. *Electronics* 11 (14), 2183.
- Tariq, U., Ahmed, I., Bashir, A.K., Shaukat, K., 2023. A critical cybersecurity analysis and future research directions for the Internet of Things: A comprehensive review. *Sensors* (ISSN: 1424-8220) 23 (8), <http://dx.doi.org/10.3390/s23084117>.
- Wang, M., Huang, A., Liu, Y., Yi, X., Wu, J., Wang, S., 2023. A quantum-classical hybrid solution for deep anomaly detection. *Entropy* 25 (3), 427.
- Wassermann, G., Su, Z., 2008. Static detection of cross-site scripting vulnerabilities. In: Proceedings of the 30th International Conference on Software Engineering. pp. 171–180.
- Zarpelão, B.B., Miani, R.S., Kawakani, C.T., de Alvarenga, S.C., 2017. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* 84, 25–37.
- Zhao, J., 2020. Quantum software engineering: Landscapes and horizons. *CoRR*, abs/2007.07047. arXiv:2007.07047.
- Zhou, X., Liang, W., Li, W., Yan, K., Shimizu, S., Kevin, I., Wang, K., 2021. Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system. *IEEE Internet Things J.* 9 (12), 9310–9319.
- Zuhadar, L., Thrasher, E., Marklin, S., de Pablos, P.O.n., 2023. The next wave of innovation—Review of smart cities intelligent operation systems. *Comput. Hum. Behav.* (ISSN: 07475632) 66, 273–281. <http://dx.doi.org/10.1016/j.chb.2016.09.030>.

Vita Santa Barletta Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

Vita Santa Barletta received the M.S. degree in computer science and the Ph.D. degree in computer science and mathematics from the University of Bari Aldo Moro, Italy, in 2017 and 2021, respectively. Since 2022, she has been an Assistant Professor with the Department of Computer Science, University of Bari Aldo Moro. Her research interests include quantum software engineering, secure software engineering, secure project management and cybersecurity.

Danilo Caivano Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

Danilo Caivano is currently a Full Professor of software engineering and project management with the Department of Computer Science, University of Bari “A. Moro”, and a Consultant for companies and organizations especially in the field of research and development projects. He is also the Head of the SERLAB Research Laboratory and the Director of the short master in cyber security. He contributed to the creation of The Hack Space, Cyber Security Laboratory, University of Bari. He is also a member of the Technical Scientific Committee of the Apulian Information Technology District and the IT Strategic Steering Committee

Mirko De Vincentiis Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

Mirko De Vincentiis is a Ph.D. Student in Computer Science at University of Bari Aldo Moro. He received the B.Sc. degree in computer science from the University of Bari Aldo Moro Italy in 2019 and M.Sc. degree in cyber security in 2021. His research interest includes cyber security in automotive and quantum software engineering.

Anibrata Pal Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

Anibrata Pal is a Ph.D. student at the Department of Computer Science, University of Bari, Italy. He has over 15 years of experience in information technology, having worked in diverse roles in project management, product management, and release management for Fortune 500 companies. His principal research areas are Software Engineering, Machine Learning, and Data Mining. He holds a Master's degree in Information Science with a specialization in Information Retrieval, Data Mining, and Machine Learning. His current research focuses on investigation on identification of frameworks and methodologies for Privacy and Security in Quantum Software Engineering.

Michele Scalera Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

Michele Scalera is graduated in Information Sciences at the University of Bari. He is assistant professor in Computer Science in the same university where he currently teaches courses on Computer Networks, Software Systems Integration and Test and Informatics. He is a member of the Scientific Technical Committee of Informatics Service Center and Delegated to ICT of the Jonian Department at the University of Bari. He currently is a reviewer of several journals and international congresses. His research fields are: security engineering, e-learning, network security, business intelligence and knowledge discovery.