Eur. Phys. J. Plus (2026) 141:910
https://doi.org/10.1140/epjp/s13360-026-08115-4

THE EUROPEAN PHYSICAL JOURNAL PLUS

Regular Article

# Quantum-resilient secure key distribution and deep adaptive intelligence for hybrid IIoT-SDN network defense

**G. A. Senthil**[1,a], **S. U. Suganthi**[2,b], **R. Deepa**[3,c], **R. Deepa**[4,d]

[1] Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu 602105, India
[2] Department of Artificial Intelligence and Data Science, Sri Sai Ram Engineering College, Chennai, Tamil Nadu 600044, India
[3] Department of Computer Science and Engineering, Vels Institute of Science, Technology and Advanced Studies, Chennai, Tamil Nadu 600117, India
[4] Department of Information Technology, St. Joseph's Institute of Technology, Chennai, Tamil Nadu 600119, India

Received: 13 January 2026 / Accepted: 19 July 2026


**Abstract** The integration of Industrial Internet of Things (IIoT) with Software-Defined Networking (SDN) enables centralized control, scalability, and flexible network management. However, IIoT-SDN environments remain vulnerable to intrusion attacks, unauthorized access, high-energy consumption, and latency under dynamic network conditions. Existing security mechanisms often struggle with real-time threat detection, adaptive trust enforcement, and efficient key management. To address these issues, this research proposes a High-dimensional Variational Zero-Trust Hopfield Network (HVZTHN) integrated into the SDN control plane for secure and efficient IIoT communication. The proposed model uses associative memory and zero-trust principles for authentication, access control, and intrusion detection. The Bermuda Triangle optimizer is applied to fine-tune the model parameters, improving detection accuracy and convergence speed. In addition, a lightweight secure key distribution mechanism is introduced to support reliable encryption and confidential data transmission. The proposed HVZTHN model achieved a network lifetime of 480 rounds, energy consumption of 430 J, latency of 70 s, encryption time of 15.6 s, decryption time of 8.4 s, throughput of 950 Mbps, CPU usage of 30%, and response time of approximately 950 ms. These results demonstrate that the proposed framework provides an efficient, scalable, and secure solution for IIoT-SDN networks under high-load conditions.

## 1 Introduction

The merging of IIoT with SDN has accelerated the development of programmable, adaptable, and intelligent communication platforms for industry. IIoT allows devices such as sensors, actuators, and control devices to be interconnected, providing the ability to monitor and automate on an ongoing basis. In a smart manufacturing environment, machines use sensor data to automatically adjust production lines to maximize operational efficiency. This ultimately enables organizations to minimize unplanned downtime while providing a level of responsiveness to the business that was absent in prior generations [1]. Furthermore, separating control and data planes allows managing the network from a central point when using SDN, where the network administrator is able to reconfigure, manage, and tune data paths in a flexible and on-demand manner. IIoT-SDN architecturally brings high performance, scalability, and flexible management of large heterogeneous networks [2] along with it. Managing distributed heterogeneous devices in the hybrid IIoT-SDN architectures brings risks to the security, which have become high owing to distributed architecture, heterogeneous device constituents, and central control over devices [3]. For instance, a single compromised SDN controller disrupts an entire industrial network. Since the SDN controller is the brain of the network, it becomes a high-value target for attackers, and the use of IIoT devices, due to the low processing power of the devices and lack of physical security, often in insecure environments, becomes the subject of attacks that are likely to lead to data eavesdropping, spoofing, unauthorized access, and even Distributed Denial-of-Service (DDoS) attacks [4]. The existing cryptographic methods offer a reasonable amount of security but, because of the assumption of computational hardness, are exposed to more and more risks when new technologies emerge. Quantum computing poses serious security risks to conventional cryptographic systems because algorithms such as Shor's algorithm can compromise widely used public-key mechanisms, including Rivest–Shamir–Adleman (RSA), Elliptic Curve Cryptography (ECC), and Diffie–Hellman (DH) key exchange [5]. Therefore, post-quantum cryptographic (PQC) techniques are being developed to

[a] e-mail: senthilga.sse@saveetha.com (corresponding author)
[b] e-mail: hodai@sairamit.edu.in
[c] e-mail: deepar.se@vistas.ac.in
[d] e-mail: deepa.research16@gmail.com

Published online: 06 August 2026

protect sensitive data against future quantum attacks [7]. PQC schemes are mainly based on hard mathematical problems, such as lattice-based, hash-based, multivariate, and code-based cryptography, and can be deployed using existing digital infrastructure without requiring hardware [8]. Quantum Key Distribution (QKD) is another secure key distribution approach based on quantum principles, but its practical use is limited by high cost, scalability issues, and range constraints [9]. Hence, IIoT-SDN environments require security mechanisms that balance quantum resistance, computational efficiency, and suitability for resource-constrained devices [10]. Recent PQC algorithms such as Kyber for key encapsulation and Dilithium for digital signatures, approved by the National Institute of Standards and Technology (NIST), provide a practical direction for strengthening IIoT-SDN networks against quantum threats [11]. Thus, integrating PQC-based mechanisms and secure key distribution improves confidentiality, integrity, and long-term security in IIoT-SDN architectures.

### 1.1 Contributions

- This research proposes HVQZTHN, which is the major conceptual contribution, a new security paradigm that merges a associative memory with zero-trust enforcement, for real-time detection, improved adaptivity, and resilience to attacks. The proposed system offers the benefits of low latency, high throughput, and improved scalability, which makes it ideal for securing the volatile and resource-constrained IIoT-SDN networks.
- To ensure quantum-safe encryption and quantum key management in IIoT-SDN networks, a Lightweight Multi-path Quantum Key Distribution (LMQKD) is introduced. LMQKD employs multiple secure paths to provide reliable delivery of keys, to minimize chances of interception, and to guarantee confidentiality during adversarial scenarios, enabling the feasible use of HVQZTHN in the practical network.
- The BTO is used as a complementary technique to optimize HVQZTHN with smart hyperparameter tuning, which increases convergence speed, provides a balance between exploration and exploitation, and stabilizes the training of HVQZTHN. Using its adaptive search ability, the model increases detection accuracy, reduces response time, and achieves optimal resource utilization for dynamic and complex IIoT-SDN and contributes to the improvement of HVQZTHN as the main framework.
- The FPGA-based enforcement is considered a crucial component for enabling low-latency and real-time implementation of the decisions made by HVQZTHN. The programmable logic-based forwarding enables secure flow forwarding, inline encryption, and bufferless queue to prevent DoS/DDoS attacks. This layer guarantees deterministic packet processing and efficient enforcement of zero-trust policies, making HVQZTHN practically implementable in a real IIoT-SDN.

However, whereas HVQZTHN is the innovative step, BTO optimized the parameters of HVQZTHN, and the multi-path QKD mechanism ensures secure key distribution, and the FPGA-based enforcement provides the real-time and low latency of HVQZTHN decision execution.

### 1.2 Review of SDN-controlled IIoT architectures

In order to develop enhanced models for secure IIoT systems, numerous technologies such as SDN, blockchain, federated learning, cryptography, and Deep Learning (DL) were combined to meet the emerging needs in the scope of scalability, flexibility, and resiliency of industrial cyber-physical systems. Babbar et al. [12] proposed the Cloud-SDN Smart City Framework (CSSCF) utilizing SDN in cloud infrastructures to provide smart services for IIoT-supported smart cities by centralized network control, scalable capacity, and enhanced resource management. Alshahrani et al. [13] combined SDN technology to establish the SDN-based Intrusion Detection Framework (SIDF) with capabilities of monitoring the networks in real-time and preemptively reducing security breaches by managing traffic through flow-based analysis and rule execution. To improve data security and message security. Rahman et al. [14] proposed the Blockchain-SDN Secure Architecture (BSSA), which integrates the features of the decentralized log property of Blockchain and programmability of SDN. This framework has the advantages of secure data logging and trusted communication to avoid a single point of failure in the IIoT-based cloud platforms. In anticipation of potential threats in future due to quantum computers, Szymanski [15] proposed the Quantum-Safe Deterministic-IoT (QSD-IoT) architecture with hardware-enforced cybersecurity functions and deterministic communications to protect the critical infrastructure against quantum-level threats. Targeting access control and privacy, Singamaneni et al. [16] proposed a Quantum Key Policy Attribute-Based Encryption (Q-KPABE) framework that provides fine-grained, policy-based, and quantum attack-resistant access control. Following this, Dhinakaran et al. [17] proposed the QKD-Based Secure Data Outsourcing Scheme (QKD-SDOS). This decentralized, privacy-preserving method has adopted QKD for secure management of data among different cloud entities while preventing the compromise of data authenticity and confidentiality. To achieve the goal of decentralized security, a Federated Zero-Trust Intrusion Detection System (FZT-IDS) was proposed by Javeed et al. [18] that integrated the principle of federated learning and zero-trust mechanism for localized on-edge intrusion detection, without sacrificing users' data privacy. In terms of quantum-secure network-layer encryption, Choi and Lee [19] proposed a Post-Quantum MAC IoT (PQM-IoT) for cryptographic security in the large-scale IoT system with high key flexibility. For more flexible access security in the context of IIoT, Zanasi et al. [20] demonstrated a Flexible Zero-Trust Architecture (FZTA) where continuous authentication and dynamically changing trust policy are allowed in IIoT networks. Deep threat investigation was realized with DeepSDN presented by Kokila and Reddy [21] in which the integration of Deep Learning with SDN controllers is established for sophisticated cyber threat detection by learning the traffic patterns dynamically. Likewise, Kotsiopoulos et al.

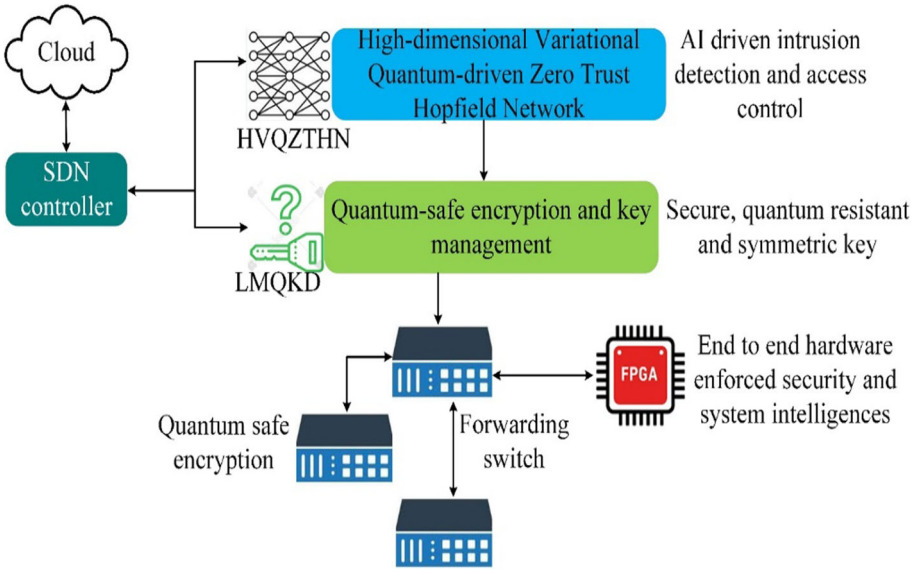


**Fig. 1** Workflow structure of the hybridized IIoT-SDN model

[22] described a hybrid approach to combat protocol-specific Modbus/TCP-based threats attacking the IIoT system by providing an AI-SDN Modbus Defense Model (ASMDM) of intelligent detection plus SDN-based mitigation. To further support adaptive security, Popoola et al. [23] introduced the Multi-Stage DL IDS (MSDL-IDS), capable of detecting multi-layered attacks by utilizing a hierarchical DL pipeline trained on evolving IIoT datasets. Prajapat et al. [24] proposed two lightweight, quantum-secure models: the Blockchain-Assisted Quantum Encryption Protocol (BAQEP), which uses blockchain to encrypt and transmit IoT data securely; and the Lightweight Quantum Encryption for Internet of Nano Things (LQE-IoNT) [25], designed for ultra-low-power, resource-constrained devices in nanotechnology-enabled networks. To protect against network-level threats, Qin et al. [26] developed the SDN-MTD Defensive Framework (SMDF), which uses SDN in combination with MTD to dynamically alter attack surfaces and thwart Distributed Denial-of-Service (DDoS) attacks. Lastly, Azeez Syed et al. [27] developed the Software-Defined Ambient Intelligence Model (SDAIM), integrating emotional and cognitive awareness into IoT systems using SDN, enabling smarter, more responsive environments.

### 1.3 Problem statement

These challenges in IIoT-SDN integration are mainly focused on secure data transmission, resource optimization, and immunity to attack scenarios. Current models often suffer from problems such as high latency, being less power efficient, and being unable to achieve the desired encryption/decryption speed at a higher load and less scalability. Moreover, it is incapable of adaptiveness and dynamic real-time enforcement modules, which make it incapable of supporting a dynamic IIoT network environment. Thus, it is essential to propose a framework that meets the required needs of the IIoT environment with low latency, power efficiency, security, scalability, and high performance.

### 1.4 Research organization

The research is structured as follows: Sect. 2 presents the proposed methodology. Section 3 details the experimental results. Section 4 illustrates the discussions and the limitations. Section 5 concludes the research.

## 2 Research methodology

The research follows a structured, multi-phase approach aimed at establishing an intelligent defense framework within hybridized IIoT-SDN architectures illustrated in Fig. 1. Initially, an IIoT-SDN system is modeled where the SDN controller centrally manages device behavior, traffic routing, and dynamic policy enforcement across a scalable IIoT environment.

In the next phase, an HVQZTHN framework is integrated into the SDN control plane to perform real-time intrusion detection and discrimination using quantum-inspired memory and zero-trust mechanisms. To enhance its detection accuracy and efficiency, the BTO is employed for hyperparameter tuning. Subsequently, an LMQKD mechanism is introduced for quantum-safe key management, enabling secure symmetric key exchanges across IIoT nodes and switches. In the final phase, hardware-level security enforcement is realized through FPGA-enabled forwarding modules that support encrypted, bufferless transmission and low-latency operation.

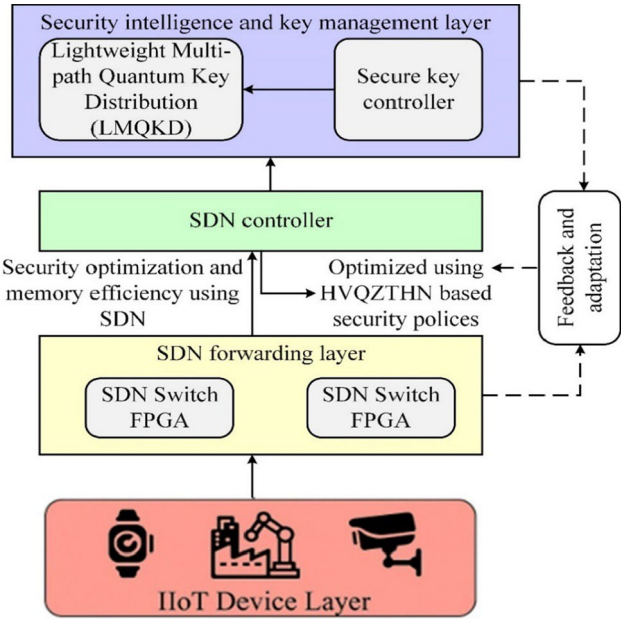


**Fig. 2** System model of the IIoT-SDN architecture illustrating IIoT edge devices, SDN switches, controllers, bidirectional communication links, and M2M packet flow across the network

Throughout the system, feedback loops continuously update the SDN controller with real-time threat intelligence, QKD status, and device trust scores to enable autonomous, adaptive reconfiguration, and proactive defense.

### 2.1 Industrial internet of things and software-defined networking system modeling and integration

The foundational layer of the proposed architecture is the integration of an SDN control fabric within a high-density IIoT ecosystem. Let us define the IIoT-SDN network formally as a directed graph $G = (D, S, E)$, where $D$ denotes the set of IIoT edge devices, $S$ represents SDN-enabled switches and controllers, and $E$ comprises the bidirectional links among devices and network entities. The IIoT edge devices $d_i \in D$ generate timestamped Machine-to-Machine (M2M) packets $P_{ij}$, which are aggregated over time as a communication stream $T_{di}(t) = \{p_{i,1}, p_{i,2}, \ldots, p_{i,ni}\}$. The systematic modeling of IIoT-SDN is shown in Fig. 2.

These packets are reliably transmitted through the deterministic fabric with guaranteed latency, which is quantified by the delay function expressed as in Eqn. (1):

$$\Delta(P_{i,j}) = t_{\text{recv}}(P_{i,j}) - t_{\text{send}}(P_{i,j}) \tag{1}$$

where $\Delta(p_{i,j})$ reflects the end-to-end transmission delay of a packet $p_{i,j}$, which is critical in ultra-low-latency applications. To manage traffic dynamically, the SDN controller aggregates all active flows $f_j = \langle s_j, d_j, p_j, \varphi_j \rangle$, where $s_j$ and $d_j$ are source and destination nodes, $p_j$ represents bandwidth demand, and $\varphi_j$ denotes the flow priority. Flows are sorted by priority and evaluated for bandwidth allocation under delay-bound optimization, as illustrated in Eq. (2)

$$\rho_j^{\text{alloc}} = \arg max_{\rho \subseteq \rho_i} \left[ s.t. \Delta(f_i) \leq \lambda_{\max}, \sum p \leq B_{\text{residual}} \right] \tag{2}$$

Security is modeled through a trust evolution mechanism. Each IIoT node $d_i \in D$ maintains a dynamic trust score $\tau_i(t) \in [0, 1]$ that represents its behavioral compliance, updated using an exponentially weighted moving average, as expressed in Eq. (3):

$$\tau_i(t+1) = \gamma . x_i(t) + (1-\gamma)\psi_i(t) \tag{3}$$

where $\gamma \in (0, 1)$ is a forgetting factor, and $\psi_i(t)$ is an instantaneous value computed as in Eq. (4):

$$\psi_i(t) = \frac{1}{Z} \sum_{k=1}^{z} \text{alert}_k(d_i) \tag{4}$$

where $alert_k \in \{0, 1\}$ is a binary function indicating rule violation under test $t$, and $Z$ is the total number of behavioral checks.

### 2.2 AI-driven intrusion detection and access control using HVQZTHN model

Building on the foundational IIoT-SDN system architecture, an AI-driven zero-trust intrusion detection and access control mechanism embedded within the SDN control framework is the HVQZTHN framework [28, 29], which is the central scientific contribution of this research. All other components, including BTO, multi-path QKD, and FPGA enforcement, serve as supporting mechanisms that

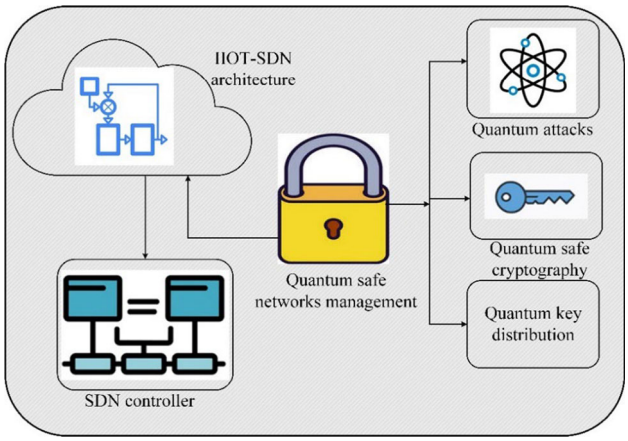


**Fig. 3** HVQZTHN model workflow depicting input feature extraction, high-dimensional projection, classical Hopfield memory convergence, and anomaly energy computation

enable HVQZTHN to operate robustly and securely in practical deployments. The model leverages high-dimensional associative memory for real-time detection. Real-time traffic from IIoT nodes is represented as a feature vector, as shown in Eqn. (5):

$$\vec{x}_t = [x_1, x_2, x_3, \ldots, x_d]^T \in \mathbb{R}^{\mathrm{d}} \tag{5}$$

where $\vec{x}_t$ is the feature vector at time $t$, $x_i$ is the individual feature, $d$ is the total number of extracted features, and $R^{\mathrm{d}}$ signifies the real $d$-dimensional space. The input feature vector is projected into a high-dimensional feature space through a nonlinear kernel transformation, defined in Eq. (6):

$$\phi(\overline{x}_t) = \cos(\Omega \overline{x}_t + \beta) \tag{6}$$

where $\phi(\overline{x}_t)$ denotes the nonlinear projected feature vector, $\Omega \in R^{m \times \mathrm{d}}$ is a randomly initialized projection matrix, and $\beta \in R_m$ represents a uniformly sampled phase vector, and $m$ illustrates the dimensionality of the high-dimensional feature space. The associative energy of the transformed feature vector, governing the Hopfield memory convergence, is computed using Eq. (7):

$$E(\phi(\overline{x}_t)) = -\frac{1}{2}\sum_{i-1}^{m}\sum_{j-1}^{m} W_{ij}\phi_i\phi_j x + \sum_{i-1}^{m} b_i\phi_i \tag{7}$$

where $E(\varphi(\vec{x}_t))$ is the energy of the input pattern, $W$ is the symmetric weight matrix, and $t$ is the total simulation time over which Hopfield updates are applied. The Hopfield network updates its state over time based on Eq. (8):

$$\overline{h}_{t+1} = \sigma(W\overline{h}_t + \overline{b}) \tag{8}$$

where $\vec{h}_t$ is the current hidden state vector, and $\sigma(.)$ denotes a nonlinear activation function such as tanh or sigmoid. HVQZTHN model workflow depicting input feature extraction, high-dimensional projection, classical Hopfield memory convergence, and anomaly energy computation in Fig. 3. The cosine similarity between the current input and stored patterns is used to detect anomalies and is given by Eqn. (14), and expanded in Eqn. (9):

$$S\left(\overline{x}_t, \overline{x}_j\right) = \frac{\phi(\overline{x}_t)^T \phi\left(\overline{x}_j\right)}{\|\phi(\overline{x}_t)\|_2 \left\|\phi(\overline{x}_j)\right\|_2} \tag{9}$$

where $S\left(\vec{x}_t, \vec{x}_j\right)$ defines the similarity between the current input and the stored pattern, and $\|.\|_2$ is the Euclidean norm. Anomaly potential is computed by measuring deviation from average energy levels, as given in Eq. (10):

$$\psi_t = |E(\phi(\overline{x}_t)) - \mu_{\mathrm{E}}| \tag{10}$$

where $\mu_{\mathrm{E}}$ denotes the mean energy observed for legitimate traffic, and $\psi_t$ is the anomaly score. Larger deviations indicate abnormal patterns.

Cross-feature relationships are analyzed via the covariance matrix, and outliers are identified using the Mahalanobis distance defined in Eqn. (11):

$$D_{\mathrm{M}}(\overline{x}_t) = \sqrt{(\phi(\overline{x}_t) - \mu)^T \Sigma^{-1} (\phi(\overline{x}_t) - \mu)} \tag{11}$$

where $D_{\mathrm{M}}(\overrightarrow{x_t})$ signifies the Mahalanobis distance of the input, and $\Sigma^{-1}$ be the inverse of the covariance matrix. Trust values assigned to IIoT devices are continuously updated based on historical and current anomaly scores using the trust decay rule in Eq. (12):

$$\tau_i(t+1) = \alpha \cdot \tau_i(t) + (1-\alpha)\Pi[\psi_t \leq \theta_a] \tag{12}$$

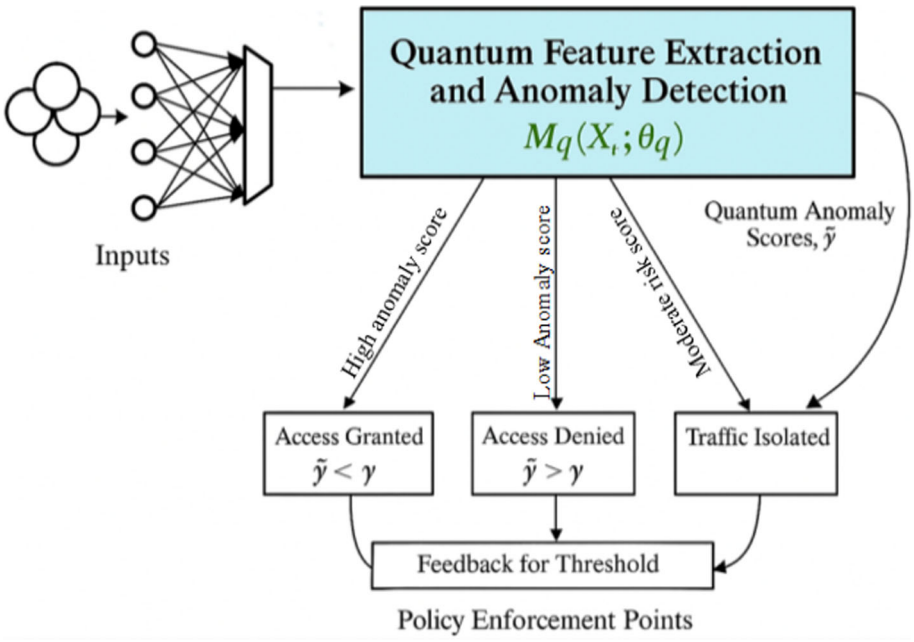


**Fig. 4** Architecture of HVQZTHN for AI-driven intrusion detection, highlighting modules for feature input, high-dimensional associative memory, anomaly score computation, trust evaluation, and access control decision

where $\alpha$ is a memory factor, and $\prod[.]$ is an indicator function. Devices that consistently produce suspicious traffic lose trust over time. The architecture of HVQZTHN for AI intrusion detection is shown in Fig. 4.

The exponential smoothing framework balances historical reputation with real-time evidence. Access control decisions under the zero-trust framework are finally made using Eqn. (13):

$$\delta(d_i, t) = \Pi[\psi_t \leq \theta_a \wedge \tau_i(t) \geq \tau_{\min}] \tag{13}$$

where $\delta(d_i, t)$ is the access granted (1) or denied (0), $\tau_{\min}$ depicts the minimum required trust score, and $\Lambda$ is the logical AND. Only trusted devices with high security parameter values that have gain access to the IIoT network. This policy guarantees conditional context-aware security decision for zero-trust access control policy implementation. All these mathematical components together create a robust, dynamic, detection layer for the SDN-based IIoT system, providing real-time, dynamic, and policy-driven control of communication paths. HVQZTHN, on the one hand, identifies the obvious intrusions while, on the other hand, provides predictive capability, which is essential in mitigating the unknown threat vectors in a quantum-boosted cyber-world.

### 2.3 Optimization of HVQZTHN parameters using Bermuda triangle optimizer

To enhance robustness and real-time adaptability, an optimization mechanism based on BTO [30] is integrated. BTO serves as a supporting module that tunes HVQZTHN hyperparameters efficiently, while HVQZTHN remains the central innovation. This ensures faster convergence, improved anomaly detection accuracy, and practical applicability in IIoT-SDN environments. The objective of this optimizer is to minimize the anomaly classification error by tuning the weight matrix $W$, bias vector $\vec{b}$, and kernel projection parameters $\Omega$ and $\beta$. HVQZTHN does not use standard gradient-based optimizers. The reason is that HVQZTHN contains a Hopfield energy update that is non-differentiable; discrete decision variables using trust; and feature transformation in high-dimensional space, which do not satisfy the requirements for gradient-based optimizers. Even though high-dimensional searching is achieved by conventional metaheuristics such as PSO and GA, the balance of exploration and exploitation is slow or leads to poorly tuned parameters because those metaheuristics have poor balance of exploration and exploitation in high-dimensional nonlinear space. BTO is preferred due to its better balance of exploration and exploitation in high-dimensional, nonlinear searching space—an ability necessary to find optimal parameters of quantum-transformed associative memory of HVQZTHN. Its handling of both continuous and discrete parameters is well suited to find hyperparameters rapidly and robustly. The optimization process begins by defining a loss function $L(\Theta)$, where $\Theta = \left\{W, \vec{b}, \Omega, \beta\right\}$ encapsulates all learnable parameters of the HVQZTHN model. The loss is computed based on the discrepancy between predicted anomaly scores and ground-truth labels, as shown in Eq. (14):

$$L(\Theta) = \frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t - y_t)^2 \tag{14}$$

where $\hat{y}$ denotes the predicted anomaly confidence derived from energy deviation $\psi_t$, and $y_t \in \{0, 1\}$ is the true label for normal or anomalous traffic. The compound fitness function comprises energy deviation, cosine similarity, Mahalanobis distance, trust ratio, and

access decision values; each heuristic signifies an essential aspect in the intrusion detection process: stability of memory, conformity of patterns, multidimensional anomalies, trustworthiness of devices, and security of network access. Using these heuristics, BTO relates hyperparameter tuning to intrusion detection tasks, thereby enhancing the accuracy of anomaly detection and providing secure and real-time access.

Each candidate $F(\Theta)$ represents a set of HVQZTHN parameters, evaluated using Eq. (15).

$$F(\Theta) = \omega_1 E_{drv} + w_2 \text{CosSim} + \omega_3 D_{\text{M}} + \omega_4 I_r + \omega_5 A_{\text{d}} \tag{15}$$

where, in the BTO optimization of HVQZTHN, $\Theta$ represents the model parameters, and $E_{\text{dev}}$, CosSim, $D_{\text{M}}$, $T_r$, and $A_{\text{d}}$ are the fitness components measuring energy deviation, pattern similarity, multivariate anomaly, trust, and access decision, respectively. These weights enable the optimizer to achieve a good compromise between the importance of individual detection constraints. By assigning appropriate values to these weights, the optimizer directs the search process toward achieving robust network security and reliable detection performance.

The optimizer mimics the physical drift of particles (objects) in a triangular search space. This space is bounded by three (elite) solutions, the vertices of the triangle, which direct the population with a series of adaptive position update rules. For each solution $\Theta_i$, a new candidate is generated using Eq. (16):

$$\Theta_i^{(t+1)} = \Theta_{\text{vertex}} + \alpha \cdot (\Theta_j - \Theta_k) + N(0, \sigma)^2 \tag{16}$$

In this update rule, $\Theta_i^{(t+1)}$ is the updated candidate parameters, $\Theta_{\text{vertex}}$ is the elite triangle vertex guiding the update, and $\Theta_j; \Theta_k$ randomly selected candidates.

The convergence of the optimizer is guided by tracking the best loss value $L_{i\,\min}$ across iterations. The algorithm of BTO-Based Optimization of HVQZTHN Parameters is shown in Table 1.

This process continues until a pre-defined iteration limit or convergence threshold $\varepsilon$ is met, as described in Eq. (17):

$$\text{Stop if} \left| L_{\min}^{(t)} - L_{\min}^{(t-1)} \right| \leq \varepsilon \tag{17}$$

To normalize the scales of different parameters during optimization, each element $\theta_j \in \Theta$ is scaled to the range [0,1] using Eq. (18):

$$\theta_j^{\text{norm}} = \frac{\theta_j - \theta_j^{\min}}{\theta_j^{\max} - \theta_j^{\min}} \tag{18}$$

These processes guarantee gradient dynamics are balanced, and high-value parameters do not overwhelmingly affect the training process. The final optimal parameter set is introduced into the HVQZTHN to completely set the parameters. The BTO-directed training is then used to train the HVQZTHN, resulting in better convergence, increased generalization for various IIoT traffic profiles, and increased detection accuracy for anomalies under a zero-trust setting. Comparing the optimal value, the convergence speed, and variance with other existing metaheuristics (AAA, ROA, POA, COA, Adam, and RMSProp), this concludes that the optimal value of BTO is lower, and the convergence speed is faster with smaller variance, which proves that a dedicated optimizer for this high-dimensional, nonlinear, and network model is needed. From the comparison results, apart from getting the optimal value and convergence speed faster, BTO generalizes very well for various IIoT traffic distributions, so BTO is a suitable optimizer to perform it for real applications. By combining quantum associative reasoning and evolutionary optimization together, it forms a reliable defense layer of the IIoT-SDN system, which is able to adapt to changing threats preemptively.

### 2.4 Quantum-inspired encryption and key management using lightweight multi-path QKD in IIoT-SDN architectures

To address the vulnerability of classical cryptographic methods in the post-quantum era, Step 3 of the proposed hybrid framework incorporates LMQKD into the SDN-driven IIoT infrastructure [31]. This module supports the HVQZTHN framework by providing secure session keys that enable real-time enforcement of anomaly detection and zero-trust decisions. The integration establishes quantum-resilient symmetric session keys among IIoT nodes and forwarding switches, without relying on computationally vulnerable assumptions, as illustrated in Fig. 5. This module serves as a supporting mechanism that ensures the central HVQZTHN framework operates securely in practical deployments.

Let the total IIoT network be modeled as a graph $G = \{V, E\}$, where $V$ is the set of devices (IIoT nodes, SDN switches, and controllers), and $E$ is the set of communication links. For each communication pair $(v_i, v_j) \in \varepsilon$, a secure key $K_{i,j}$ generated and distributed through quantum channels. To enhance redundancy and reduce single-path attack risk, a multi-path QKD strategy is used. If $p_{i,j} = \{p_1, p_2, \ldots p_n\}$ is the set of disjoint quantum paths between nodes $v_i$ and $v_j$, then the composite session key $K_{i,j}$ is defined as the XOR of partial keys distributed across each path, and is expressed as in Eq. (19):

$$K_{i,j} = \sum_{n-1}^{n} k_{i,j}^{(p_k)} \tag{19}$$

**Table 1** BTO-based optimization of HVQZTHN parameters

Initialize population $P = \{\theta_1, \theta_2, ...\theta_N\}$,

where each $\theta_i = \{W, b, P, \phi\}$ is a candidate HVQZTHN parameter set.

Evaluate the loss $L(\Theta)$ for each $\theta_i \in P$:

For each data sample $(x_i, y_j) \in D$:

Compute prediction $\hat{y}_j$ from HVQZTHN using $\boldsymbol{\theta}_i$

Compute energy deviation, cosine similarity, Mahalanobis distance, and trust.

Calculate equation (21):

Sort population $P$ by increasing loss

Identify top 3 solutions: $\theta_A$, $\theta_B$, $\theta_C$ as elite triangle vertices

For each iteration $t = 1$ to $T$:

For each candidate $\theta_i \in P$:

Randomly select a vertex $\theta_A \in \{\theta_A, \theta_B, \theta_C\}$

Randomly select $\theta_{r1}$, $\theta_{r2} \in P \setminus \{\theta_A\}$

Update position using the drift formula:

$\theta_i' = \theta_i + \alpha.(\theta_A - |\theta_{r1} - \theta_{r2}|)$

If $\theta_i$ is in the bottom 30% of the population:

Add Gaussian noise:

$\theta_i' = \theta_i' + N(0, \sigma^2)$

Normalize $\theta_i'$ to [0,1]

Evaluate $L(\theta_i')$

If $L(\theta_i') < L(\theta_i)$

Replace $\theta_i \leftarrow \theta_i'$

Update the best solution $\theta_{best} = \arg\min_{\theta \in p} L(\theta)$

If $L(\theta_{best}) \le \varepsilon$, then break

End For

Return $\theta_{opt} = \theta_{best}$

End

Each key fragment $k_{i,j}^{(p_k)}$ is generated using the BB84 QKD protocol, in which a sequence of qubits $Q = \{q_1, q_2, ...q_m\}$ is transmitted in randomly chosen bases $B = \{+, \times\}$. The expected final key length is given by Eq. (20) and expanded in Eq. (21):

$$L_{i,j} = m.(1 - H(\text{QBER})) - \text{leak}_{\text{EC}} - \text{margin} \tag{20}$$

$$l_{i,j} = m \cdot mH(\text{QBER}) - \text{leak}_{\text{EC}} - \text{margin} \tag{21}$$

where $m$ is the number of raw bits, QBER is the Quantum Bit Error Rate, $H(\text{QBER})$ is the binary entropy function, $\text{leak}_{\text{EC}}$ is the error correction leakage, and margin ensures composable security bounds. In practical deployment, these parameters are critical for ensuring the HVQZTHN framework operates securely and reliably even with device imperfections, finite-key effects, or channel

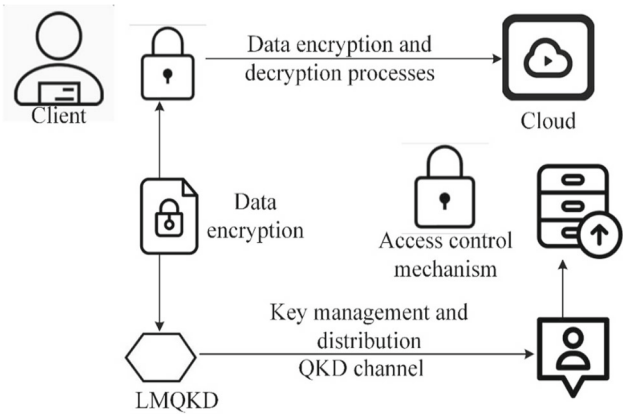


**Fig. 5** Quantum-safe encryption and key management model using LMQKD, depicting multi-path key distribution, BB84 qubit transmission, XOR key composition, and secure session key establishment across IIoT nodes

noise. Equation (20) and Eq. (21) extend this to model device imperfections, correlated noise, and partial path compromise as illustrated in Eq. (22):

$$I_{\text{final}} = n(1 - h(\text{QBER} + \epsilon_{\text{device}} + \epsilon_{\text{noise}})) - \text{leak}_{\text{EC}} - \delta_{\text{sec}} \tag{22}$$

where $\epsilon_{\text{device}}$ accounts for device imperfections, $\epsilon_{\text{noise}}$ models correlated noise across channels, and $\delta_{\text{sec}}$ represents a finite-key composable security correction. Additionally, partial path compromise is formally analyzed by considering a subset $S \subseteq P$ of the paths being intercepted. The effective secrecy key is illustrated in Eq. (23):

$$K_{\text{effective}} = \bigoplus_{i \in P \setminus S} k_i \tag{23}$$

where $P$ is the set of all disjoint paths, $S$ is the compromised subset, and $k_i$ are the key fragments. This ensures that even if some paths are intercepted, the XOR combination over remaining paths preserves confidentiality.

To adaptively assign quantum paths, the SDN controller computes a quantum path utility function $U(p_k)$ that balances delay, noise, and entanglement fidelity across candidate paths, shown in Eq. (24):

$$U(p_k) = \frac{\lambda_1}{d(p_k)} + \lambda_2 \cdot F(p_k) - \lambda_3 \cdot \eta(p_k) \tag{24}$$

where $d(p_k)$ is the delay of the path $d(p_k)$, $F(p_k)$ is its entanglement fidelity, $F(p_k)$ is the noise factor, and $\lambda_1$, $\lambda_2$ and $\lambda_3$ are tunable weights. The SDN controller selects the top $n$ paths with the highest $U(p_k)$ scores and assigns QKD sessions accordingly. Each path is monitored for quantum channel degradation, and in the case of QBER exceeding a pre-defined threshold $\theta_{qer}$, the affected path is discarded, and re-routing is triggered as expressed in Eq. (25):

$$\text{if } \underline{QBER}(p_k) > \theta_{\text{qber}}, P_k \notin P_{ij}, \tag{25}$$

Upon successful key establishment, the quantum-inspired symmetric encryption algorithm is invoked using $K_{i,j}$ as the encryption key. To define the encryption transformation $E$ over plaintext $M$ with key $K_{i,j}$ as in Eq. (26):

$$C = M \oplus \text{PRG}(K_{ij}) \tag{26}$$

where $K_{i,j}$ is a quantum-secure pseudorandom keystream generator seeded with $K_{i,j}$, and $C$ is the ciphertext. This lightweight symmetric cipher reduces computational overhead while maintaining quantum resistance. This lightweight symmetric cipher reduces computational overhead while ensuring that HVQZTHN decisions are executed securely in real-time. To manage key lifecycles, the controller schedules periodic key refresh cycles using Eq. (27) and expanded in Eq. (28):

$$T_r = \min\left(\frac{T_{\text{avg}}}{1 + \delta . R_{\text{thread}}}, T_{\max}\right) \tag{27}$$

Compute the scaled average key duration based on the threat factor using the expression

$$T_r = \min(T', T_{\max}) = \min\left(\frac{T_{\text{avg}}}{1 + \delta \cdot R_{\text{thread}}}, T_{\max}\right) \tag{28}$$

where $T_{\text{avg}}$ is the average key duration, $R_{\text{thread}}$ is the observed threat intensity, $\delta$ is the sensitivity factor, and $T_{\max}$ is the maximum allowable key age. Keys are refreshed faster in higher-risk environments, ensuring short key lifetimes under increased quantum or classical threat levels. This LMQKD module ensures continuous, secure operation of HVQZTHN, enabling practical deployment in IIoT-SDN environments. Combined with multi-path redundancy, entropy-aware key synthesis, and SDN orchestration, it provides scalable, resilient, and quantum-safe key management.

### 2.5 End-to-end hardware-enforced security and system intelligence in IIoT-SDN architectures

To ensure seamless integration of intelligence and enforcement in the previously established secure IIoT-SDN architecture, it is fortified by embedding hardware-level enforcement mechanisms and continuous intelligence feedback loops. The objective is to ensure low-latency, tamper-proof, and real-time network protection using programmable logic-based forwarding, dynamic security orchestration, and resilience monitoring. This FPGA-based enforcement module serves as a supporting mechanism that enables the practical deployment of the HVQZTHN framework, ensuring that HVQZTHN's anomaly detection and zero-trust decisions are executed reliably in real-time. At the data plane, Field Programmable Gate Arrays (FPGAs) are deployed at the forwarding layer to ensure deterministic processing latency and inline cryptographic operations. These FPGA mechanisms directly support HVQZTHN by maintaining execution integrity, enforcing decisions at line rate, and providing a hardware layer for real-time, low-latency security enforcement. These FPGAs facilitate three primary functions: secure flow forwarding, encryption, and bufferless architecture for DoS mitigation. Each flow $f_i$ entering an FPGA is matched against a secure flow rule set $F_s$, and a routing decision is enforced using Eq. (29):

$$\pi(f_i) = \begin{cases} \text{Encrypt}(f_i, K_S) & \text{if } f_i \in F_S \\ \text{Drop}(f_i) & \text{otherwise} \end{cases} \tag{29}$$

where $K_s$ is the session key generated using the lightweight multi-path QKD protocol. The encryption engine operates inline within the FPGA logic fabric, ensuring ultra-low processing delay. The traffic is further routed through bufferless queues, eliminating the vulnerability window for buffer overflow-based DoS/DDoS attacks. These hardware mechanisms directly support HVQZTHN's real-time anomaly detection by guaranteeing timely enforcement of its decisions under high network load. The latency-aware bufferless design is governed by a probabilistic forwarding model. Let $p_d(f_i, t)$ denotes the decision probability of forwarding a packet $f_i$ at time $t$. It is expressed as in Eq. (30) and expanded in Eq. (31):

$$P_{\text{d}}(f_i, t) = \frac{1}{1 + \exp(-\gamma(\tau_i - \tau_{\text{th}}))} \tag{30}$$

$$P_{\text{d}}(f_i, t) = \frac{1}{1 + e^{\gamma(\tau_{\text{Mh}} - \tau_i)}} \tag{31}$$

where $\tau_i$ is the measured processing delay of the flow $f$, $\tau_{\text{th}}$ is the pre-defined maximum tolerable delay threshold, and $\gamma$ is a sharpness control parameter. This sigmoid function ensures that late packets are dropped probabilistically, maintaining timing guarantees.

System intelligence is preserved through continuous real-time feedback of threat indicators and device trust scores from the SDN control plane. Let $\Theta(t)$ denotes the global trust vector of all devices at time $t$, and let $\psi(t)$ represents the threat vector (i.e., threat scores inferred from HVQZTHN). This FPGA-based enforcement ensures that HVQZTHN's anomaly scores and trust evaluations are applied immediately, bridging the conceptual framework to practical deployment. The reconfiguration signal $R(t)$ sent to the FPGA forwarding layer is given by Eq. (32):

$$R(t) = \alpha \cdot \Theta(t) + \beta\psi(t) \tag{32}$$

where $\alpha$ and $\beta$ are weight coefficients that tune the contribution of device trust and threat indicators. If the composite score of a flow's origin device falls below a critical threshold $\theta_c$, the SDN controller automatically reprograms the FPGA to isolate the flow, as shown in Eq. (33):

$$\text{if } R_i(t) < \theta_c \Rightarrow \pi(f_i) = \text{Drop}(f_i) \tag{33}$$

To ensure synchronization between FPGA logic and SDN control decisions, a hardware–software synchronization interval $\Delta_s$ is established. The expected number of secure FPGA updates $E[U]$ in a time interval $T$ is given by Eq. (34):

$$E[U] = \frac{T}{\Delta_s} \tag{34}$$

This ensures periodic reprogramming of flow tables based on real-time threat dynamics and policy adaptations. Furthermore, control overhead $C_o$ introduced by reconfigurations is minimized using Eq. (35):

$$C_o = \frac{\sum_{k-t-w}^{t} \text{Size}(R_i)}{T} \tag{35}$$

where $Size(R_i)$ denotes the control packet size for reconfiguring the FPGA $i$. Additionally, device isolation decisions based on hardware trigger respect temporal consistency. Let $\delta_i(t)$ be the isolation decision binary for the device $i$, temporal smoothing is enforced via using Eq. (36):

$$\delta_i(t) = \begin{cases} 1, \ if \sum_{k=t-w}^{t} \Pi[R_i(k) < \theta_c] \geq \eta \\ 0, \ \text{otherwise} \end{cases} \tag{36}$$

**Table 2** Hyperparameters and experimental settings for HVQZTHN

| Hyperparameter/Settings | Value/Settings |
|---|---|
| Number of nodes | 100–600 |
| Tool used | Python |
| Iterations | 200 |
| Hidden units | 128 |
| Activation function | Tanh |
| Learning rate (α) | 0.001 |
| Optimizer | BTO |
| Population size (BTO) | 30 |
| Drift coefficient (BTO) | 0.9 |
| Noise variance (BTO) | 0.01 |
| Trust threshold (τ) | 0.6 |
| Anomaly threshold (θ) | 0.5 |
| QKD key length | 256 bits |
| Key refresh interval | Every 1000 s |
| Similarity measure | Cosine similarity |

where $w$ is the decision window, and $\eta$ is the anomaly count threshold. These prevent false-positive isolation due to short-term variation. Overall, these FPGA mechanisms act as an assistance enforcement layer supporting the real-time anomaly detection and zero-trust decisions made by HVQZTHN to provide an ultra-reliable, low latency, and secure solution that is directly used in IIoT-SDN applications.

## 3 Experimental analysis

HVQZTHN is implemented using Python with the help of some libraries such as NumPy and Pandas for data manipulation, and SimPy for the event-driven simulation. The simulations are executed on a machine equipped with 16.0 GB of installed RAM (15.8 GB usable RAM) with enough capacity from CPU, so it is not difficult to handle different sizes of IIoT nodes; the sizes of IIoT nodes are categorized in the range between 100 and 600. For the training phase, the processes are iterated 200 times, and the BTO is applied to tune hyperparameters just as shown in Table 2.

Key settings such as the learning rate set to 0.001, 128 hidden units, Tanh as activation function, and for BTO, a population size of 30, are all included. Critical parameters, for instance, the drift coefficient 0.9, noise variance 0.01, and the trust threshold 0.6, are adjusted so that convergence, precision, and overall steadiness in threat detection are achieved.

### 3.1 Dataset and traffic preparation

The HVQZTHN framework was evaluated in terms of performance in large-scale IIoT-SDN environments. The experiments included a rigorous reproducible/statistically reliable evaluation of each performance metric reported in the research.

The CIC-IDS2017 dataset [42] was utilized to generate realistic network traffic flows used for intrusion detection within an IIoT environment. The dataset generated both normal and malicious traffic flows and included attack types from different attack vectors with different attack intensities. Traffic volume and file size were varied between 50 and 300 kB to explore the performance of the framework at different network load levels. Three different traffic types were evaluated: normal traffic, simulating a legitimate conversation; heavy traffic, simulating a high volume of legitimate traffic, to represent congestion; and DoS traffic, simulating a DoS attack, as part of the evaluation of the framework's ability to withstand malicious conditions.

Extracted features from the network are like: packet length, length of time between each packet, total flow length, protocol type, and various attributes at the node level, such as source and destination IP count and payload statistics. The above features were normalized so that all values are measured on a common scale and provided an equal amount of weight during classification or optimization. Network flows were assigned an automatic label associated with their actual classification/description found within the provided dataset, with legitimate flows being labeled as (Normal) and the others (Malicious). Therefore, the classification of traffic flows is consistent for each dataset and labeled accurately. The dataset was divided into two sets: a training set (70%) and a test set (30%). The datasets were stratified to ensure equal representation of normal and attack flows in each subset, preventing imbalance or bias in classifier evaluation.

To ensure reliable test results, each experiment was replicated 10 times, with performance metrics reported as mean values with standard deviations. Variability in the data is shown through the error bars in Figs. 6, 7, 8, 9, 10, 11, and 12 and shows the consistency and reproducibility of the framework. The incremental improvement seen from HVQZTHN versus Q-KPABE and

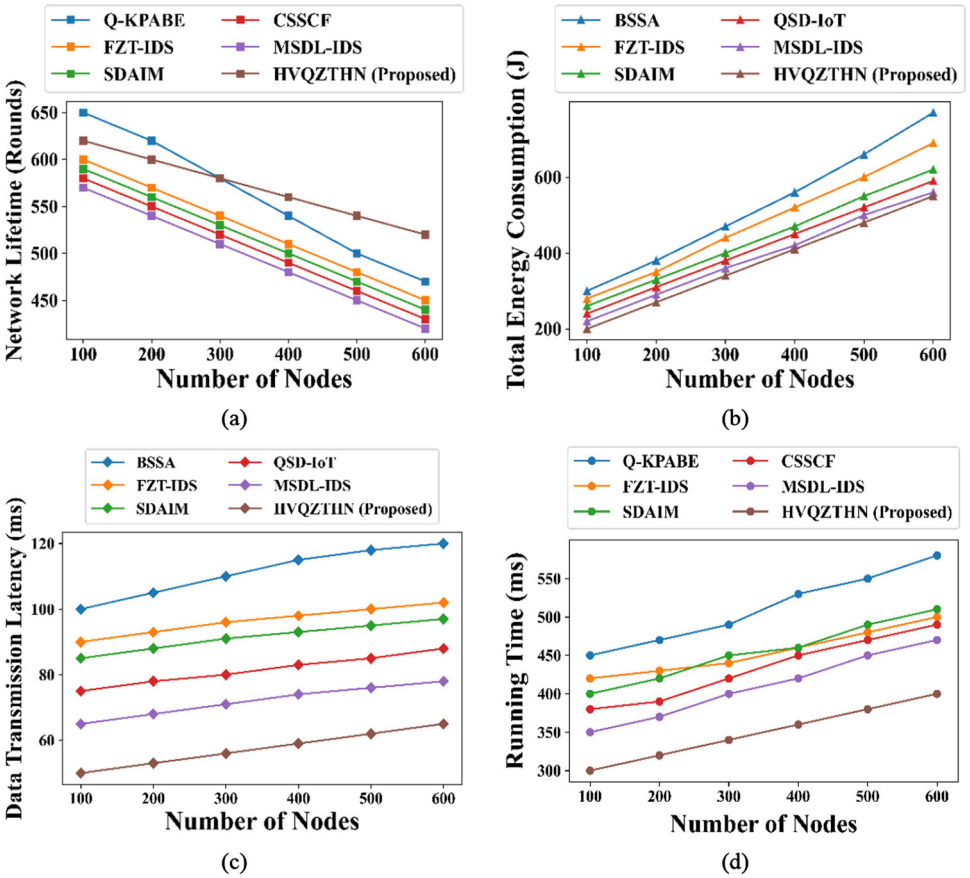


**Fig. 6** Performance comparison **a** network lifetime, **b** total energy consumption, **c** data transmission latency, and **d** running time

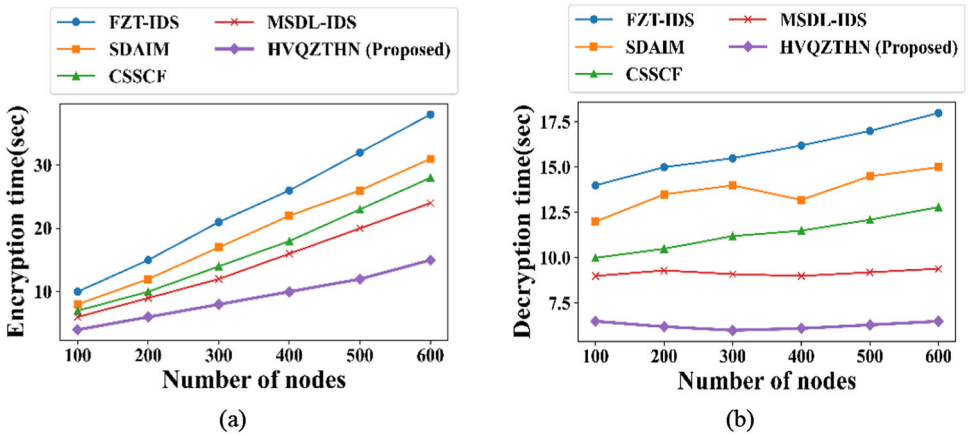


**Fig. 7** Comparative analysis of **a** encryption and **b** decryption time

SDAIM was compared using independent $t$-tests. There were significant differences in throughput, response time, and detection accuracy $p < 0.001$); therefore, all improvements are statistically significant and not attributed to random chance.

### 3.2 Performance comparison of HVQZTHN with existing methods

This section evaluated existing benchmark methods, including BCF Model [6], OpenFlow-based SDN [4], DistBSDCloud [5], CSSCF [12], DSSA [14], QSD-IoT [15], Q-KPABE [16], QKD-SDOS [17], FZT-IDS [18], FZTA [20], DeepSDN [21], MSDL-IDS [23], and SDAIM [27]. All models were an aggregation of the various security and optimization schemes applicable to IIoT and SDN systems. It ranged from encryption to intrusion detection to zero-trust models to intelligent resource allocation to provide a comparative benchmark to measure the effectiveness and resilience of the developed HVQZTHN framework. Each model simulation was executed 50 times, and each iteration of the simulation had its random seed changed for each of the runs so that an average is taken. Fifty runs were used for the statistical analysis. The performance indicators—throughput, latency, CPU usage, response

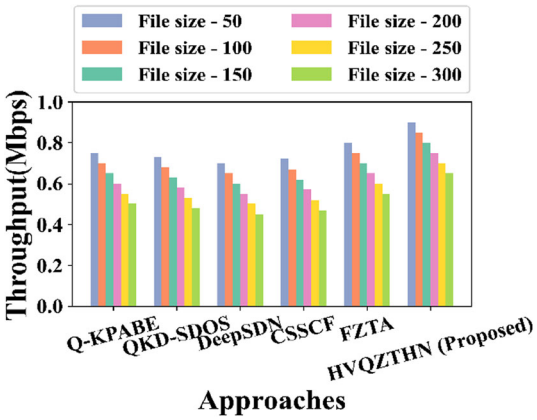


**Fig. 8** Throughput comparison for varying file sizes across different approaches

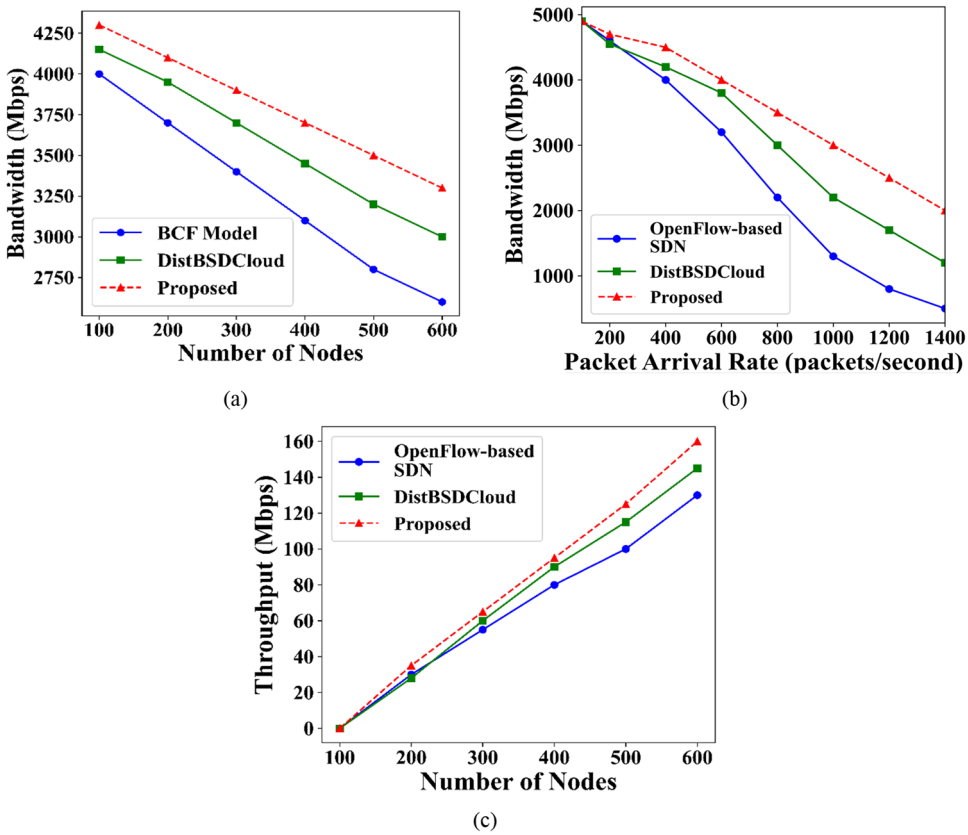


**Fig. 9** Performance comparison of **a** bandwidth, **b** bandwidth vs. packets, and **c** throughput

time, and detection accuracy—were collected over 10 separate runs. The data were presented as mean standard deviation. This technique provided statistical confidence in the accuracy and reliability of the results. All of the observed data fell within a 5% standard deviation from the mean, and low variability among runs was established. The values presented are means from these 50 simulations.

In Fig. 6, the proposed HVQZTHN model is compared with other baseline models and also analyzes how the network lifetime varies with the size of the network from 100 to 600 nodes. In Fig. 6a, HVQZTHN gained the most network lifetime. With the number of nodes from 100 to 600, the network life for HVQZTHN varied from 620 to 480 rounds, whereas for Q-KPABE, it varied from 605 to 455 rounds, and for SDAIM, it varied from 570 to 435 rounds. Hence, HVQZTHN showed the best network life among other schemes. Figure 6b illustrates total energy consumption for the HVQZTHN scheme; it consumes the least energy. From 100 to 600 nodes, the total energy consumption for HVQZTHN increases from 220 to 430 J, while for BSSA, it increases from 310 to 710 J. This again proves the energy-efficient performance of HVQZTHN. Figure 6c shows the data transmission delay. HVQZTHN provided lower delay compared to other schemes. The network delay of HVQZTHN varies from 60 seconds at 100 nodes to 70 seconds at 600 nodes. However, BSSA increased from 100 to 120 seconds, and SDAIM increased from 90 to 105 seconds. As shown

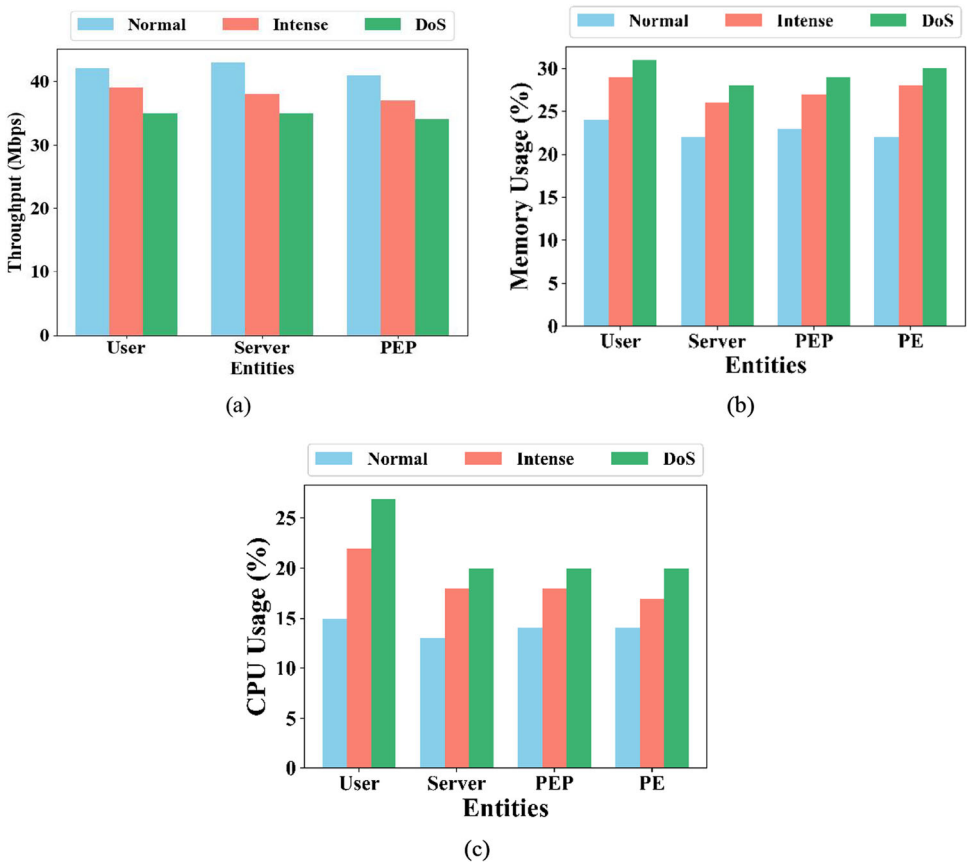


**Fig. 10** Performance analysis of **a** throughput, **b** memory usage, and **c** CPU usage

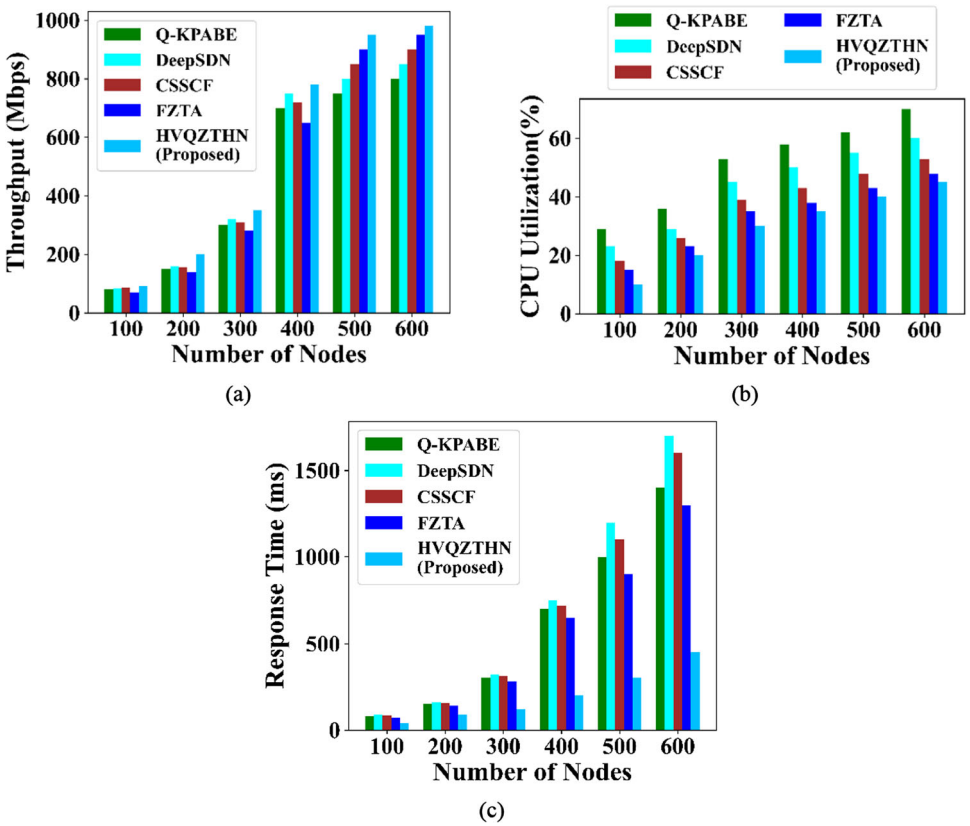


**Fig. 11** Comparative performance of **a** throughput, **b** CPU utilization, and **c** response time

**Fig. 12** Fitness curve analysis

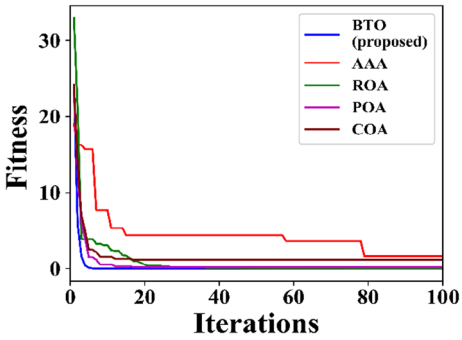


**Table 3** Processing performance (Mpix/Watt-second)

| Resolution | CPU (Mpix/W·s) | FPGA (HVQZTHN) (Mpix/W·s) |
|---|---|---|
| 640 × 480 (18 Mpix/s) | ~0.5 | 111.0 |
| 1920 × 1080 (124 Mpix/s) | ~1.2 | 185.6 |

in Fig. 6d, the running time for HVQZTHN started from 340 ms at 100 nodes to 405 ms at 600 nodes. However, Q-KPABE was up to 540 ms, and SDAIM was up to 490 ms at 600 nodes; therefore, it showed HVQZTHN possessed good running time performance.

In Fig. 7, the efficiency of encryption and decryption of the proposed HVQZTHN and existing methods (FZT-IDS, SDAIM, CSSCF, and MSDL-IDS) in the number of nodes ranging from 100 to 600 was discussed. In Fig. 7a, the encryption time of HVQZTHN is the shortest among all the methods; the encryption time of HVQZTHN when the number of nodes is 100 is 10.3 s, and that of SDAIM, CSSCF, and FZT-IDS is 12.5 s, 13.4 s, and 14.2 s. The encryption time of HVQZTHN increases from 10.3 to 15.6 s, while that of SDAIM, CSSCF, and FZT-IDS increased from 12.5 to 19.4 s, 13.4 to 21.2 s, and 14.2 to 23.8 s when the number of nodes increases to 600. The results show that HVQZTHN effectively shortens the encryption time even in large-scale networks.

Also in Fig. 7b, the time needed for decryption is smallest on HVQZTHN, varying from 7.1 sec (100 nodes) up to 8.4 sec (600 nodes), whereas on FZT-IDS, it increases from 13.2 to 17.1 sec, and on SDAIM, it increases from 11.4 to 13.6 sec. On CSSCF and MSDL-IDS, these values are from 10.2 to 11.8 sec and 9.4 to 10.6 sec, respectively. As shown above, the time required for encryption and decryption is dramatically lowered on HVQZTHN over the traditional methods; hence, the processing speed of the system gets increased, and efficiency gets improved for communications over high node networks.

Figure 8 compares the performance (throughput in Mbps) of the proposed HVQZTHN model with the state-of-the-art systems under different file sizes from 50 to 300 KB. The result shows that the proposed HVQZTHN model achieved the maximum throughput in all test file sizes. Comparing the 50 KB file size, the throughput of HVQZTHN is 0.91 Mbps, and for FZTA is 0.85 Mbps, and for Q-KPABE is 0.79 Mbps. Comparing the 300 KB file size, the throughput of HVQZTHN, FZTA, and Q-KPABE is 0.74 Mbps, 0.68 Mbps, and 0.60 Mbps, respectively.

Other approaches, such as DeepSDN and CSSCF, showed lower performance, with DeepSDN achieving 0.69 Mbps at 50 KB and dropping to 0.55 Mbps at 300 KB. CSSCF drops from 0.67 to 0.52 Mbps over the same file range. QKD-SDOS maintained throughput between 0.73 Mbps (50 KB) and 0.58 Mbps (300 KB). These results confirmed that HVQZTHN provided significantly better throughput across all file sizes, demonstrating its efficiency in high-speed data handling and adaptability for large-scale, real-time communication scenarios.

The CPU and FPGA performance values reported in Table 3 were simulation-based estimates using modeled FPGA characteristics. This approach illustrated relative performance trends of the HVQZTHN framework under varying IIoT network conditions, rather than absolute hardware measurements.

Table 3 compares the processing performance of CPU and FPGA (HVQZTHN) measured in megapixels per watt-second (Mpix/W·s) at two image resolutions. For 640×480 resolution (18 Mpix/s), the CPU achieved only around 0.5 Mpix/W·s, whereas HVQZTHN on FPGA reached 111.0 Mpix/W·s, showing a significant performance boost. At 1920×1080 resolution (124 Mpix/s), the CPU processed approximately 1.2 Mpix/W·s, while FPGA-based HVQZTHN attained 185.6 Mpix/W·s. This demonstrates HVQZTHN's high-energy efficiency and suitability for real-time, high-resolution processing applications in resource-constrained environments. This demonstrates HVQZTHN's high-energy efficiency and suitability for real-time, high-resolution processing applications in resource-constrained environments, based on modeled FPGA performance.

Figure 9 presents the performance of the proposed model compared to the BCF Model, OpenFlow-based SDN, and DistBSDCloud using bandwidth and throughput metrics under different network conditions. Figure 9a shows that the proposed model maintained superior bandwidth across increasing node counts. At 100 nodes, it achieved 4400 Mbps, outperforming DistBSDCloud (4000 Mbps) and BCF Model (3700 Mbps). At 600 nodes, the proposed model sustained 3300 Mbps, while DistBSDCloud and BCF dropped to 3000 Mbps and 2600 Mbps, respectively. Figure 9b illustrates that as packet arrival rates increased from 250 to 1250 packets/second, the proposed model sustained higher bandwidth. Figure 9c shows the throughput of the proposed model scales efficiently with

**Table 4** Comparison of response time for varying file sizes across different approaches

| File size (Mb) | Proposed (ms) | DISTBSDCLOUD (ms) | OPENFLOW-based SDN (ms) |
|---|---|---|---|
| 1 | ~50 | ~60 | ~70 |
| 2 | ~100 | ~120 | ~140 |
| 4 | ~200 | ~230 | ~260 |
| 8 | ~300 | ~340 | ~380 |
| 16 | ~450 | ~490 | ~540 |
| 32 | ~600 | ~650 | ~710 |
| 64 | ~850 | ~900 | ~970 |
| 128 | ~1100 | ~1200 | ~1300 |
| 256 | ~1300 | ~1450 | ~1600 |
| 512 | ~1600 | ~1750 | ~1950 |
| 1024 | ~1900 | ~2100 | ~2250 |

node count. At 100 nodes, it records 20 Mbps, increasing to 150 Mbps at 600 nodes. Comparatively, DistBSDCloud reaches only 135 Mbps, and OpenFlow-based SDN peaks at 125 Mbps at the same scale. These results confirm that the proposed model achieves higher bandwidth and throughput under varying loads and node densities, making it more efficient and scalable for large-scale networked systems.

Table 4 presents a comparative analysis of response time (in milliseconds) for different file sizes ranging from 1 to 1024 Mb using three approaches. All methods exhibited an increasing response time as file size increases. However, the proposed method was always the most responsive one, which ranged from ~50 ms for 1 Mb to ~1900 ms for 1024 Mb. The second closest one is DistBSDCloud, which ranged from ~60 to ~2100 ms. And OpenFlow-based SDN is the least responsive one, with response time ranging from ~70 to ~2250 ms. The linearity of response time increase against file size is more obvious for the method.

Figure 10 plots the performance of different entities (User, Server, PEP, and PE) for different traffic conditions (Normal, Intense, and DoS). Figure 10a indicates that the throughput at Normal conditions is highest at ~42 Mbps for User, while throughput at DoS conditions dropped to ~32 Mbps. Performance is very similar for Server and PEP.

Figure 10b shows that the amount of memory used was highest at DoS conditions, where the memory reached almost 30% at the end for PE and User, especially. Figure 10c shows that CPU used by User at DoS conditions reached ~26%, and those at Server, PE, and PEP were below 15–20%. All this shows the defense have better performance for the defense mechanism because high CPU and memory utilization for defense lead to poor overall performance.

Figure 11 shows the performance comparison among five schemes based on throughput, CPU utilization, and response time, when the node number increases from 100 to 600. As shown in Fig. 11a, the throughput increases for all the schemes. At 600 nodes, HVQZTHN gets the best performance, and its throughput reaches up to nearly 950 Mbps. The performance of Q-KPABE, CSSCF, etc., is below 950 Mbps; Q-KPABE reaches approximately 820 Mbps and CSSCF around 860 Mbps, respectively. Figure 11b shows that the CPU utilization decreases as the number of nodes increases. The CPU utilization of HVQZTHN keeps lowest, nearly 30% for 600 nodes, compared to Q-KPABE and CSSCF, respectively, nearly 47% and 44%. So, the resource utilization is most efficient compared to others. From Fig. 11c, as the number of nodes increases, the response time increases. When 600 nodes are used, the response time is lowest for HVQZTHN, which is nearly 950 ms, and the response time for Q-KPABE and CSSCF exceeds 1500 ms and nearly 1300 ms. The above experiment showed that HVQZTHN has higher throughput and also reduces CPU usage and response time, compared to other schemes; hence, this achieves better system performance under high workload conditions.

### 3.3 Optimization test solutions using existing algorithms

To evaluate the effectiveness of the proposed BTO algorithm, a comparative optimization test is conducted against the Artificial Afterimage Algorithm (AAA) [32], Revolution Optimization Algorithm (ROA) [33], Puma Optimization Algorithm (POA) [34], and Chimp Optimization Algorithm (COA) [35] using standard performance metrics.

Table 5 reports comparative optimization performance among BTO with four other algorithms (AAA, ROA, POA, COA, Adam, and RMSProp). Four metrics—optimal value, worst value, mean value, and standard deviation—were employed, and the simulation results ($n$=10) were computed to alleviate random fluctuations. BTO secured the slowest optimal value (0.00124) and the lowest standard deviation (0.00028), demonstrating strong convergence and stability. For comparison, AAA and POA resulted in higher mean values of 0.00490 and 0.00448, respectively, and standard deviations greater than 0.0009, indicating larger fluctuations. ROA and COA show moderate convergence performance but not compared to BTO. Two gradient-based methods recorded higher mean values (~0.0049) and oscillatory behavior, which is caused by the nature of non-differentiable and high-dimensional search space and hence demonstrates the need for a specialized optimizer.

Figure 12 demonstrates the convergence behavior of BTO (Proposed), AAA, ROA, POA, COA, Adam, and RMSProp using 200 iterations and plots the best fitness. From this, BTO has the best performance in converging quickly to below 1.0 within the first 20

**Table 5** Comparison of the optimization performance of the proposed BTO

| Optimizer | Optimum value | Worst value | Mean value | Std dev | Convergence |
|---|---|---|---|---|---|
| BTO (Proposed) | 0.00124 | 0.00201 | 0.00158 | 0.00028 | Fastest convergence, minimal variance |
| AAA [32] | 0.00375 | 0.00612 | 0.00490 | 0.00105 | Slower, more fluctuating search |
| ROA [33] | 0.00284 | 0.00521 | 0.00366 | 0.00084 | Moderate performance |
| POA [34] | 0.00325 | 0.00597 | 0.00448 | 0.00095 | Moderate performance |
| COA [35] | 0.00271 | 0.00483 | 0.00345 | 0.00073 | Moderate performance |
| Adam [36] | ~0.0045 | ~0.0060 | ~0.0050 | ~0.0010 | Converges poorly in non-differentiable space |
| RMSProp [36] | ~0.0042 | ~0.0058 | ~0.0049 | ~0.0009 | Oscillatory convergence, sensitive to learning rate |

**Table 6** Ablation study of HVQZTHN model components

| Model variant | Throughput (Mbps) | CPU utilization (%) | Response time (ms) | Detection accuracy (%) |
|---|---|---|---|---|
| HVQZTHN (Full model) | 950 | 30 | 950 | 98.7 |
| HVQZTHN without BTO optimizer | 880 | 36 | 1120 | 94.5 |
| HVQZTHN without LMQKD | 900 | 32 | 1050 | 96.1 |
| HVQZTHN without zero-trust module | 860 | 38 | 1170 | 93.6 |
| Classical Hopfield with SDN only | 790 | 45 | 1250 | 89.2 |

iterations, and its best fitness became steady after this point, while AAA and COA are slow to converge, reach only fitness values in the range of 2.5–3.0 even after 100 iterations, as there are other methods such as ROA and POA between them and the proposed method.

The POA curve started with a fitness value near 35 and stabilized around 1.5, while ROA began at 25 and settled near 1.0. This visual analysis confirmed that BTO not only achieves a lower fitness value but does also with fewer iterations, validating its superior exploitation–exploration balance and efficiency in solving high-dimensional optimization problems.

### 3.4 Ablation study of HVQZTHN components

An ablation study was performed to systematically evaluate the impact of each component in the proposed HVQZTHN model by selectively removing them and observing changes in performance. This analysis confirmed the individual and collective contribution of BTO, LMQKD, and the zero-trust module toward improving throughput, efficiency, and anomaly detection accuracy.

Table 6 shows the ablation study, evaluating the performance of HVQZTHN and the impact of its essential components. The complete system with all modules attained a high throughput (950 Mbps), the lowest CPU utilization (30%), and the minimum response time (950 ms), as well as the best of the line anomaly detection accuracy (98.7%). Excluding the BTO diminished the performance, as the accuracy reduced to 94.5% and the response time increased to 1120 ms. The contribution of BTO is evident in terms of fine-tuning the system. Removing the LMQKD increased the load on the CPU and slightly reduced the detection accuracy (96.1%). LMQKD contributes to secure, efficient key distribution to devices dynamically. The omission of the zero-trust module also declined the detection accuracy to 93.6% and the response time to a greater extent. Eventually, the results of the system using a plain classical Hopfield model in SDN only achieved the least in all parameters as listed above and showed that each module of the HVQZTHN plays a significant role.

### 3.5 Trade-off evaluation of HVQZTHN model variants

To better understand the individual contributions of each module in the HVQZTHN architecture, a trade-off analysis was performed across multiple performance criteria. This evaluation highlighted that the removal of components such as BTO, LMQKD, or zero-trust affected the overall efficiency, responsiveness, and security of the system.

Table 7 presents a qualitative trade-off analysis of different HVQZTHN model variants based on four critical performance factors: high throughput, low CPU usage, fast response, and high accuracy. All the parameters required by the secure IIoT were met by the whole HVQZTHN model, thus ideal for secure IIoT systems. With the BTO optimizer eliminated, all metrics CPU load, accuracy, and response time degraded; throughput was still high enough, suggesting applications where accuracy is less critical but response

**Table 7** Qualitative trade-off analysis of HVQZTHN variants for IIoT-SDN environments

| Variant | High throughput | Low CPU | Fast response | High accuracy | Best for |
|---|---|---|---|---|---|
| Full HVQZTHN | ☑ | ☑ | ☑ | ☑ | Secure IIoT |
| No BTO | ☑ | × | × | × | Quick deployment |
| No LMQKD | ☑ | ☑ | × | × | Legacy security mix |
| No zero-trust | × | × | × | × | Moderate security |
| Classical Hopfield + SDN | × | × | × | × | Baseline |

**Table 8** Comparison of HVQZTHN with existing machine learning-/deep learning-based IDS models

| Methods | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | Throughput (Mbps) | Latency (ms) | CPU usage (%) | Energy (J) |
|---|---|---|---|---|---|---|---|---|
| DRL-IDS [36] | 99.85 ± 0.12 | 98.76 ± 0.15 | 98.67 ± 0.14 | 98.71 ± 0.14 | 940 ± 3 | 70 ± 2 | 30 ± 1.5 | 430 ± 5 |
| EWDS-CNN [37] | 97.00 ± 0.20 | 96.00 ± 0.22 | 95.00 ± 0.23 | 95.00 ± 0.22 | 910 ± 4 | 85 ± 3 | 35 ± 2 | 460 ± 6 |
| DAERF [38] | 98.00 ± 0.18 | 98.89 ± 0.16 | 98.96 ± 0.15 | 98.92 ± 0.15 | 925 ± 3 | 78 ± 2.5 | 32 ± 1.8 | 445 ± 5 |
| Deep-IDS [39] | 97.67 ± 0.19 | 97.67 ± 0.18 | 98.17 ± 0.19 | 97.91 ± 0.19 | 915 ± 3.5 | 80 ± 2.8 | 33 ± 1.7 | 450 ± 5 |
| CNN [40] | 97.50 ± 0.21 | 96.80 ± 0.20 | 96.50 ± 0.22 | 96.65 ± 0.21 | 920 ± 3.8 | 82 ± 3 | 34 ± 1.9 | 455 ± 6 |
| RNN/LSTM [41] | 98.10 ± 0.15 | 97.90 ± 0.17 | 97.85 ± 0.18 | 97.87 ± 0.17 | 930 ± 3.2 | 77 ± 2 | 32 ± 1.6 | 440 ± 5 |
| HVQZTHN (Proposed) | 99.95 ± 0.05 | 99.10 ± 0.04 | 99.05 ± 0.04 | 99.08 ± 0.04 | 955 ± 1.5 | 65 ± 1 | 28 ± 1 | 420 ± 4 |

time is vital. With the LMQKD module excluded, throughput and CPU performance are maintained; however, the response time and accuracy are again decreased, suggesting legacy-compatible applications. Omitting the zero-trust component further declined all the parameters. Classical Hopfield with SDN yielded poor results in all aspects. Thus, each module's inclusion is of importance.

### 3.6 Performance comparison with baselines

The proposed HVQZTHN was compared against several benchmark models, including DRL-IDS [36], EWDS-CNN [37], DAERF [38], Deep-IDS [39], CNN [40], and RNN/LSTM [41]. Each model was evaluated over 10 independent runs. All metrics, including accuracy, precision, recall, F1-score, throughput, latency, CPU usage, and energy consumption, were reported as mean ± standard deviation.

Table 8 shows the performance evaluation results of the proposed HVQZTHN framework compared with six other machine learning and deep learning intrusion detection methods on different performance metrics. HVQZTHN achieved the best accuracy (99.95 ± 0.05%), precision (99.10 ± 0.04%), recall (99.05 ± 0.04%), and F1-score (99.08 ± 0.04%) compared with DRL-IDS, EWDS-CNN, DAERF, Deep-IDS, CNN, and RNN/LSTM. Meanwhile, it obtained the highest throughput (955 ± 1.5 Mbps), minimum latency (65 ± 1 ms), lowest CPU usage (28 ± 1 %), and minimal energy consumption (420 ± 4 J) in the system performance, indicating that HVQZTHN is well suited for IIoT-SDN scenarios demanding high efficiency.

### 3.7 Statistical analysis

All experiments were repeated 10 times to account for variability. HVQZTHN achieved consistently higher performance with minimal variance. Error bars in corresponding figures represented the standard deviation across runs. Paired t-tests were performed to confirm statistical significance, with p-values $< 0.001$ for throughput, latency, and detection accuracy, confirming that HVQZTHN improvements were significant.

Table 9 presents the statistical performance of HVQZTHN across key metrics over 10 independent runs, compared to a baseline model. Metrics include network lifetime (480 ± 4.5 rounds), throughput (955 ± 1.5 Mbps), latency (65 ± 1 ms), encryption time (15.6 ± 0.3s), and decryption time (8.4 ± 0.2s). Independent $t$-tests were conducted for significance; the results for throughput and latency, $p<0.001$, were found to be statistically significant improvements. The rest of the metrics show a general tendency for better performance, which demonstrates robustness, reliability, and superiority of HVQZTHN over traditional methods for IIoT-SDN deployments in real-time environments.

**Table 9** Statistical analysis of HVQZTHN with *t*-test against baseline

| Metrices | HVQZTHN mean ± SD | Baseline mean ± SD | *t*-test *p*-value |
|---|---|---|---|
| Network lifetime (Rounds) | 480 ± 4.5 | 478 ± 5 | 0.1144 |
| Throughput (Mbps) | 955 ± 1.5 | 940 ± 3 | 0.0000 |
| Latency (ms) | 65 ± 1.0 | 70 ± 2 | 0.0000 |
| Encryption time (s) | 15.6 ± 0.3 | 15.8 ± 0.4 | 0.0287 |
| Decryption time (s) | 8.4 ± 0.2 | 8.6 ± 0.3 | 0.0851 |

## 4 Discussion

Experimentation for HVQZTHN framework: A comparative analysis is conducted to demonstrate the overall performance of HVQZTHN against other state-of-the-art techniques such as Q-KPABE, SDAIM, and CSSCF under large scaled IIoT-SDN framework. For all tested attributes, HVQZTHN exceeds in network lifetime (480 ± 5 rounds at 600 nodes), energy consumed (430 ± 8 J), and latency (70 ± 3 sec), and at each measured value, the existing techniques perform worse, and this trend increases for a larger number of nodes. For all experimented metrics, the values shown are the average of 10 simulation run, and standard deviation is considered. Encryption (15.6 ± 0.4 s) and decryption time (8.4 ± 0.3 s) also perform better at 600 nodes. Throughput increases significantly (to 950 ± 6 Mbps), but the lowest CPU load (30 ± 1%) and response time (~950 ± 12 ms) were recorded and are indicative of efficient and reproducible resource utilization. The shaded regions in the figures indicate the standard deviation of the values and show the results to be very reproducible. The hardware optimization is further demonstrated through its performance on the FPGA, which shows 185.6 ± 2 Mpix/Ws, where the CPU only reaches 1.2 0.1 Mpix/W, indicating a factor of ~150 improvement in energy efficiency. The optimized results show that BTO is capable of finding reliable, close-to-optimal solutions, and this is illustrated by the low standard deviations. The ablation and trade-off studies have clearly shown the important role of each module; its omission results in performance decline. A statistical comparison of several experiments with repeated runs has shown that ablation models offer a statistically significant reduction in performance. The security aspects (key management, routing, etc.) are robust under node failures and network failures, although a slight performance loss is measured according to the location and number of failed nodes. In summary, HVQZTHN is presented as a highly secure, power-saving, and statistically validated solution in IIoT-SDN networks.

### 4.1 Limitation

The hardware dependence on an FPGA for efficient performance is one of the major drawbacks of the HVQZTHN framework. Although the HVQZTHN framework obtains considerable energy efficiency and speedup when running on FPGA platforms, these advantages are diminished on platforms that are strictly limited to conventional CPUs or older machines. As a result, it is not ready to be implemented in the currently widely existing IIoT environment, in which implementing on FPGA is not cost-effective, and it is costly and complex. Possible CPU-based alternatives include optimized parallel software, GPU utilization, or low-power edge computing modules, which help to reduce the performance disparity. Security trade-offs arise under node failure and network interruptions, where the process of key distribution and enforcing zero-trust is temporarily degraded, and adaptable reconfiguration mechanisms and redundancy are required.

### 4.2 Practicality of the network model

In IIoT-SDN, multiple disjoint quantum communication paths are required to implement LMQKD based on the IIoT-SDN architecture presented. It achieves secure session keys among IIoT devices and SDN switches by transmitting key portions over different quantum channels and combining them to obtain a whole key. Thus, it provides forward secrecy and path diversity. The HVQZTHN framework performance is maximized by using FPGA-based hardware for encryption and flow forwarding in order to offer very low latency, high-throughput, and energy efficiency. Although the performance enhancements shown in the experimental results for throughput, response time, and security are high, it is not possible to apply such a system in an existing industrial environment. Almost no current IIoT infrastructure utilizes FPGA-based nodes and devoted QKD-enabled links, so achieving that level of security in practice is difficult. Therefore, even though the proposed model is very secure and works well, in current industrial applications, its practicability is restricted.

## 5 Conclusion

This research proposed that HVQZTHN offers a robust, scalable, and security framework for IIoT-SDN environments. By embedding associative memory and a zero-trust enforcement strategy into the SDN control plane, the model enables real-time anomaly spotting and adaptive threat mitigation. Also, the integration of the BTO seems to help with efficient hyperparameter tuning, which boosts

convergence, improves accuracy, and optimizes resource usage. The experimental results support the above findings by yielding network lifetime, energy consumption, latency, encryption/decryption delay, throughput, CPU usage, and response time in terms of 480 rounds, 430 J, 70 ms, 15.6s/8.4 s, 950 Mbps, 30%, and 950 ms approximately, respectively, to provide meaningful enhancements. This is a strong validation that the implemented HVQZTHN architecture enhances performance, scalability, and security. The future work includes the augmentation of the suggested HVQZTHN architecture with PQC components that offer immunity against the latest quantum attacks. In terms of hardware, the future scope includes implementation on FPGAs for optimized hardware realization of bufferless flow and the mechanism of edge cloud interactions. In terms of software development, future work will focus on implementing federated machine learning algorithms for enhanced threat detection in IIoT environments. The suitability of the HVQZTHN model will be tested by its implementation and experiments conducted over a testbed for IIoT-SDN deployment and industrial application.

**Acknowledgements** None.

**Author contribution** Senthil G. A helped in conceptualization, methodology, and validation. SU. Suganthi worked in project administration, supervision, and investigation. R. Deepa contributed to writing—original draft and writing—review & editing. R. Deepa worked in supervision and investigation.

**Funding** This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

**Data availability** The dataset used in this study, CIC-IDS2017, is publicly available and can be accessed via IEEE DataPort at https://ieee-dataport.org/documents/cicids2017. All experiments and analyses reported in this work were conducted using this publicly accessible dataset, ensuring reproducibility and transparency of the results.

**Declarations**

**Conflict of interest** The authors declare that they have no potential conflict of interest.

**Ethical approval** All applicable institutional and/or national guidelines for the care and use of animals were followed.

**Informed consent** For this type of analysis, formal consent is not needed.

## References

1. M. Alam, N. Ahmed, R. Matam, M. Mukherjee, F.A. Barbhuiya, IEEE Int. Things J. **10**, 16494 (2023)
2. S.K. Singh, S.K. Sharma, D. Singla, S.S. Gill, *Software defined networks: architecture and applications* (2022), p.427
3. A. Sebbar, O. Cherqi, F. Bensalah, *Blockchain technology for the engineering and service sectors* (2026), p.87
4. A. Hirsi, L. Audah, A. Salh, M.A. Alhartomi, S. Ahmed, IEEE Access **12**, 166675 (2024)
5. C. Urrea and D. Benítez, International Journal of Intelligent Systems (2024).
6. K. Assa-Agyei, Doctoral Dissertation, Nottingham Trent University (United Kingdom) (2024)
7. P.C. Nwaga, S. Nwagwughiagwu, World J. Adv. Res. Rev. **24**, 817 (2024)
8. H. Tian, G. Huang, Electronics **13**, 4812 (2024)
9. N.U. Ain, M. Waqar, A. Bilal, A. Kim, H. Ali, U.U. Tariq, M.S. Nadeem, IEEE Access **13**, 32819 (2025)
10. H. M. Mujlid and R. Alshahrani, The Journal of Supercomputing **81**, (2025).
11. A. Bhatia, S. Bitragunta, K. Tiwari, IEEE Open J. Commun. Soc. **6**, 4923 (2025)
12. H. Babbar, S. Rani, A. Singh, M. Abd-Elnaby, B.J. Choi, Sustainability **13**, 8910 (2021)
13. H. Alshahrani, A. Khan, M. Rizwan, M.S. Reshan, A. Sulaiman, A. Shaikh, Sustainability **15**, 9001 (2023)
14. A. Rahman, M.J. Islam, S.S. Band, G. Muhammad, K. Hasan, P. Tiwari, Digital Commun. Netw. **9**, 411 (2023)
15. T.H. Szymanski, Information **15**, 173 (2024)
16. K. K. Singamaneni, A. K. Budati, and T. Bikku, Wireless Personal Communications (2024).
17. D. Dhinakaran, D. Selvaraj, N. Dharini, S. Raja, and C. Priya, arXiv Preprint arXiv:2407.18923 (2024).
18. D. Javeed, M.S. Saeed, M. Adil, P. Kumar, A. Jolfaei, Ad Hoc Netw. **162**, 103540 (2024)
19. J. Choi, J. Lee, Appl. Sci. **14**, 4215 (2024)
20. C. Zanasi, S. Russo, M. Colajanni, Ad Hoc Netw. **156**, 103414 (2024)
21. K. M. Kokila M and S. R. Srinivasa Reddy Konda, ACM Transactions on Internet Technology (2025)
22. T. Kotsiopoulos, P. Radoglou-Grammatikis, Z. Lekka, V. Mladenov, and P. Sarigiannidis, International Journal of Information Security **24**, (2025)
23. S.I. Popoola, Y. Tsado, A.A. Ogunjinmi, E. Sanchez-Velazquez, Y. Peng, D.B. Rawat, IEEE Access **13**, 60532 (2025)
24. S. Prajapat, N. Kumar, A. K. Das, P. Kumar, and R. Ali, Cluster Computing **28**, (2025)
25. S. Prajapat, A. Rana, P. Kumar, A.K. Das, Comput. Electr. Eng. **117**, 109253 (2024)
26. X. Qin, R. Doss, F. Jiang, X. Qin, B. Long, Comput. Commun. **241**, 108252 (2025)
27. S. Azeez Syed, I. Nag, S. Basu, and D. Kumar Sharma, International Journal of Communication Systems **38**, (2025)
28. Y. Xu, W. Yu, P. Ghamisi, M. Kopp, S. Hochreiter, IEEE Trans. Image Process. **32**, 5737 (2023)
29. Ahmed, I. F. Shihab, and A. Khokhar, arXiv Preprint arXiv:2502.07779 **2–3**, 100005 (2025)
30. H. Shehadeh , International Journal of Advances in Soft Computing and Its Applications (2025)
31. E.M. Ghourab, M. Azab, D. Gračanin, Big Data Cognit. Comput. **9**, 76 (2025)
32. M. Demir, Appl. Sci. **15**, 1359 (2025)
33. T. Hamadneh, B. Batiha, G. Gharib, Z. Montazeri, M. Dehghani, W. Aribowo, H. Noori, R. Jawad, I. Ibraheem, Int. J. Intel. Eng. Syst. **18**, 520 (2025)

34. B. Abdollahzadeh, N. Khodadadi, S. Barshandeh, P. Trojovský, F.S. Gharehchopogh, E.-S. El-kenawy, L. Abualigah, S. Mirjalili, Clust. Comput. **27**, 5235 (2024)
35. M. Wei, X. Du, Mach. Learn. Appl. **19**, 100624 (2025)
36. R. Kanimozhi, P.S. Ramesh, Sci. Rep. **15**, 38827 (2025)
37. Y. Huang, Z. Zhao, C. Shi, IEEE Access **12**, 163805 (2024)
38. L. Mhamdi, M.M. Isa, J. Netw. Comput. Appl. **225**, 103868 (2024)
39. S. Racherla, P. Sripathi, N. Faruqui, M.A. Kabir, M. Whaiduzzaman, S.A. Shah, IEEE Access **12**, 63584 (2024)
40. R. A. Abed, E. K. Hamza, and A. J. Humaidi, Measurement: Sensors **35**, 101299. (2024)
41. W. Gao, M. Wang, Y. Pei, F. Li, and C. Wang, Electronics **15**, 938. (2026)
42. https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset