International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2026/06 Volume 13, Issue 1, January – February (2026)

RESEARCH ARTICLE

Hybrid TGN-QHHO Framework for Optimized Resource Allocation, Trust-Aware Routing, and Secure Communication in WSN-Based IoT Networks

Anuj Kumar
School of Computer Science and Engineering, Galgotias University, Yamuna Expressway, Gautam Buddh Nagar, Greater Noida, Uttar Pradesh, India.
anujnsc@gmail.com

Krishna Kant Agrawal
School of Computer Science and Engineering, Galgotias University, Yamuna Expressway, Gautam Buddh Nagar, Greater Noida, Uttar Pradesh, India.
kkagrawal@outlook.com

Received: 21 September 2025 / Revised: 20 December 2025 / Accepted: 20 January 2026 / Published: 26 February 2026

Abstract – The foundation of today's colocation systems comes from wireless sensor networks (WSNs), which enable continuous sensing, communication, and decision-making among a huge number of distributed devices. Most of the resource allocation and routing techniques still facing limitations, such as a lack of energy efficiency, high latency, low adaptability to rapidly changing trust levels, and low scalability due to a high density of devices. As traditional deep learning and heuristic (rule-based, often trial-and-error) approaches use slightly different methods of selecting parameters; therefore, they do not produce optimal performance when viewed at a system level. In this article, we propose a hybrid combination of Temporal Graph Problems (TGP) to handle spatiotemporal allocation of resources, in combination with Quantum-Inspired Harris Hawks Optimization (QHHO) for routing that not only considers trust and congestion, but also maximizes the benefits of trust composite scores (TCS) while minimizing penalties associated with nodes that have a TCS ≤ 0.35. The TGP allows the modeling of spatially and temporally varied wireless energy (EE) demand, while QHHO assists in route optimization through multi-objective fitness functions, which balance between reliability, quality (of service) and energy efficiency (EE). The real-time analysis of relative signal strength indicator (RSSI) fluctuations provides insight into the potential for attacks on wormhole nodes, with the capability of isolating compromised nodes from the network. When we simulate the nodes 50,100,150 in WSN and apply hybrid TGP/QHHO technique to achieve energy efficiency that is 0.81, throughput of 0.87 and make span is 0.37 that improve the overall performance which is tested with pervious technique such as LSTM-GRU, TGP, GCM and GAT. Our statistical validation results confirm the robustness of our approach, as we obtained: MSE (0.0021 - 0.0033); RMSE (0.0458 - 0.0574); R2 (0.906 - 0.931); which outperformed previous systems. The integration of our Quantum-Inspired technique (QHHO) with the use of TGP has resulted in not only the enhanced resource allocation, but also enhanced adaptive learning, and Enhanced Trust Aware Routing within Dynamic the IoT.

Index Terms – Wireless Sensor Networks (WSNs), Internet of Things (IoT), Temporal Graph Networks (TGNs), Quantum-Inspired Harris Hawks Optimization (QHHO), Composite Trust, Energy Efficiency, Congestion-Aware Routing, Secure Communication.

1. INTRODUCTION

Recent developments in the Internet of Things (IoT) have led to the appearance of several new applications across a variety of industries, including smart cities, surveillance, and healthcare. There is various methodology used to implement IoT system in WSN and each methodology provide a solution to process, store and implement the data to reduce energy efficiency [1]. In IoT network, there are various interconnected devices that are capable to transfer data to a data center that is stored in cloud. IoT devices provide both services (fundamental and complex) to access data from cloud and data applications [2]. However, in implementation with IoT, there are various challenges encounter such as resource limitations, energy inefficiency, network congestion, and security vulnerabilities.

In smart cities, wireless systems that use the IoT have recently expanded significantly in a number of ways. The implementation of IoT-based wireless technologies facilitates urban monitoring. The Wireless Sensor Network (WSN) is an essential element of the IoT, which has generated numerous new real-time applications [3]. WSNs are created for a wide range of sectors and applications, including general engineering, disaster and hazards management, biomedical health and habitat monitoring, forest fire detection, animal

RESEARCH ARTICLE

tracking, commercial applications, seismic detection, residential applications, underwater applications, etc. [4]. WSNs are regarded as one of the most prominent IoT technologies and a rapidly expanding technology, according to [5]. Nonetheless, obstacles such as constrained resources, cyber risks, and privacy issues may hinder its success [6].

Conventional resource allocation and routing protocols, such LEACH (Low-Energy Adaptive Clustering Hierarchy) and AODV (Ad-hoc On-Demand Distance Vector), depend on fixed optimization parameters and are inadequate in adapting to fluctuating network conditions or in identifying malicious nodes efficiently [7]. In real time applications, the cryptographic security techniques are robust and they are computational demands in terms of energy-constrained for making impractical in IoT devices. Furthermore, traditional energy management strategies are devoid of predictive functionalities, resulting in premature node failures and network instability [8]. It is noted that for a dynamic, optimization framework which is trust-sensitive that combined all the resource allocation, congestion-aware and energy forecasting and also ensure to stop malicious threats [9].

Now a days to learn spatio-temporal network, Artificial Intelligence (AI) machine and deep learning methods are used and that also allow adaptive routing, resource allocation and intrusion detection [10] in WSN-based IoT networks. For lack of security used a high cost of computing of security particularly in contexts with limited resources.

In consideration of these complexity, Bio-inspired algorithm focused more because this algorithm executes this problem such as (handle intricate task, multi objective energy optimization) efficiently. Several metaheuristic methods, such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) [11][12] have been utilized for IoT resource management; nevertheless, they exhibit sluggish convergence and subpar performance in high-dimensional search spaces [13]. In order to enhance flexibility and decision-making in IoT networks, recent research has tried to combine bio-inspired optimization methods with AI-based learning models [14][15]. Nevertheless, issues with computational overhead, convergence efficiency, and trust-aware security have not yet been addressed [16].

1.1. Problem Statement

Even though there has been a lot of study on routing, resource allocation, and security in WSN-based IoT networks, the solutions that are out there have several significant shortcomings. LEACH and AODV are two examples of traditional routing and clustering protocols that use static or semi-static parameters. They don't adapt to changing network circumstances, which wastes energy, increases latency, and causes nodes to fail too soon. Recent AI-driven methods that use deep learning and metaheuristic optimization have showed promise, but most of them only optimize one thing at a time, such routing, resource allocation, or security, without taking into account how they all work together. In addition, trust-aware security measures are frequently too simple to spot advanced assaults like wormhole behavior in real time. Cryptographic solutions, on the other hand, place too much of a burden on sensor nodes that don't have a lot of power. As a result, there is a substantial research need in the development of a cohesive, intelligent, and trust-aware framework capable of concurrently facilitating dynamic resource allocation, congestion-aware routing, and secure communication across extensive, time-varying IoT-enabled WSNs settings.

1.2. Proposed Framework and Contributions

This paper presents a hybrid AI-driven architecture that utilizes Temporal Graph Networks (TGNs) for spatio-temporal resource allocation [17] and Quantum-Inspired Harris Hawks Optimization (QHHO) for congestion-aware routing[18] to tackle these difficulties. By using this hybrid approach for conventional methods and in dynamically assesses node trust worthiness through composite trust (CT) score is calculated based on direct interactions by the neighbor recommendation and residual energy levels. This approach that has (CT<0.35) are executed from essential processes to overcome come harmful devices in network.

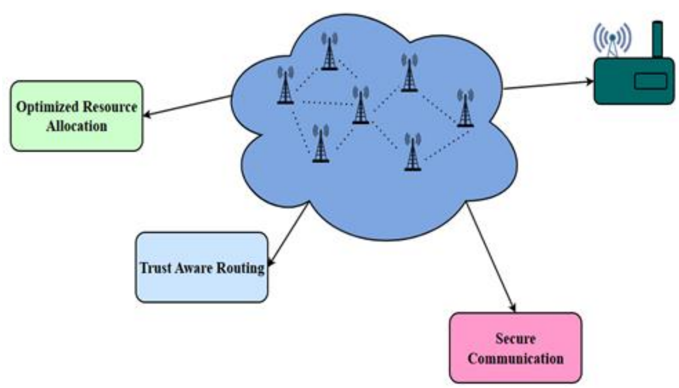

Figure 1 Framework of Comprehensive Layered Approach to Optimizing WSN-Based IoT Networks

In this research that is hybrid approach which is combination of graph based deep learning and quantum metaheuristics approach to control the energy efficiency, resource allocation and congestion aware in IoT network as shown in above Figure 1[19]. In WSN-IoT, Trust-Graph network work into two parts (i) Trust level of nodes in which it was explained that trust levels of nodes help ensure that only reliable, non-malicious nodes participate, thereby preventing resource misuse by low-trust or compromised nodes. (ii) for geographical dependencies and energy distribution among various nodes. By using Quantum-Inspired Harris Hawks

RESEARCH ARTICLE

Optimization (QHHO) that improve the routing efficiency by using quantum-inspired exploration which determine the optimal path way and their alleviating congestion in real time which is based on traditional strategies by low energy waste and improve the latency time, throughput and increasing detection precision attack.

The objectives of this research are as follows:

- To develop a Temporal Graph Network (TGN)-based spatio-temporal resource allocation model for efficient and adaptive energy and bandwidth management in IoT-enabled WSNs.
- For Trust–aware routing, design a hybrid approach, QHHO-based routing strategy that supports congestion-aware and energy-efficient data transmission.
- To incorporate a Composite Trust-based security mechanism for reliable node selection and real-time detection of malicious activities, including wormhole attacks.
- By this proposed framework using performance metrics in terms of such as energy efficiency, throughput, execution time, latency, and malicious node detection rate, and to compare the results with existing state-of-the-art approaches.

1.3. Organization of Paper

The remainder of this paper is organized as follows. Section 2 presents a comprehensive review and critical analysis of related work. Section 3 describes the proposed hybrid TGN–QHHO algorithm in detail. Section 4 outlines the experimental setup and evaluation methodology used to assess the performance of the proposed approach. Section 5 discusses the simulation results and provides comparative analysis with existing algorithms. Finally, Section 6 concludes the paper.

2. RELATED WORKS

This section examines existing studies on IoT-enabled WSNs that use ML, DL, and optimization methods to solve important problems that come up while running a network. Previous research suggests several learning-based and optimization-oriented strategies to improve network longevity, throughput, and dependability in resource-limited and dynamic contexts. Table 1 shows that each study uses different methods and algorithmic frameworks to make WSN-based IoT systems run better and be more stable.

Ahmed et al. [20] used DL architectures to provide a novel approach to resource allocation for wireless sensor Internet of Things networks that optimizes data and uses less energy. Here, two conflicting optimization objectives are EE (energy efficiency) and SE (spectral efficiency). A deep neural network based on whale optimization has increased the network's energy efficiency. The data was optimized using the heuristic-based multi-objective firefly method. Relay selection and optimum power allocation are addressed by this suggested approach. The cooperative multi-hop network topology is the subject of the research. The optimum resource allocation is done by decreasing total transmit power, and the best relay selection is performed by fulfilling Quality of Service (QoS) criteria. Consequently, an energy-efficient procedure has been developed. In terms of throughput of 96%, energy efficiency of 95%, QoS of 75%, spectrum efficiency of 85%, and network lifespan of 91%, the simulation results show that the proposed model performs competitively when compared to conventional methods.

Haseeb et al [21] suggested an IoT-based WSN architecture with several design levels for use in smart agriculture. Firstly, agricultural sensors acquire important data and determine a set of cluster heads based on multi-criteria decision function. To facilitate reliable and effective data transfer, we also use the Signal-to-Noise Ratio (SNR) to measure the intensity of the signals travelling over the transmission connections. We also utilize the recurrence of the Linear Congruential Generator (LCG) to secure the transmission of agricultural sensor data between base stations (BSs). The simulated results showed a significant improvement in communication performance over other methods, yielding average increases of 13.5% in Network Throughput, 38.5% in Packet Drop Ratio, 13.5% in Network Latency, 16% in Energy Consumption, and 26% in Routing Overheads for Smart Agriculture compared to the other method tested.

The Levy Chaos Adaptive Snake Optimization-based Multi-Trust Routing Method (LCASO-MTRM), a unique heuristic multi-objective trust routing technique, is developed by [22] With the goal of increasing connection bandwidth while concurrently lowering latency, packet loss, and energy consumption, adaptive operators and novel chaos are integrated within the LCASO framework. The diversity of populations is increased by the chaotic operator, a wider range of potential solutions is allowed, and the speed of the search is accelerated. Stagnation is prevented and the resistance of the algorithm is increased by the adaptive operator, while the convergence of the algorithm is improved. A model of multi-objective quality of service (QoS) routing with a trust metric is included in the links of the network. A more accurate representation of the current state of the link is allowed, and a more accurate evaluation of the trust level for each link is enabled. The effectiveness of the LCASO-MTRM is shown through a series of comparative simulations when compared with improved particle swarm optimization (IPSO), the improved artificial bee colony algorithm (IABC), and the cloned whale optimization algorithm (CWOA). Performance improvements over the other algorithms are demonstrated by LCASO-MTRM, with power consumption reduced by

49.53%, latency reduced by 22.56%, and packet loss reduced by 40.21%, while 6.13% more bandwidth is obtained than with the other algorithms.

Narawade & Kolekar [23] in WSN methods Adaptive Cuckoo Search Rate Optimization (ACSRO) method was developed which give an effective solutions in congestion control. In this method cuckoo algorithm used to control or adjust the send rates of sensor nodes and also to keep the track of packet arrival rates which is equal to or less than the parent nodes. With the help of this method calculate throughput, delay in network, packet loss and data queue size that avoid congestion level by using alternative congestion. Bu implement this method; it has ability to give quickly response in real time traffic condition in network so that we can improve the network performance during high traffic congestion but it increased the number of resources and also amount of energy used in WSN.

Xu et al., [24] presented a hybrid approach that integrates a fuzzy logic system with Whale Optimization Algorithm (WOA) for routing protocol to solve the challenges in IoT healthcare. This proposed model work on two key components. The fuzzy logic that used a clustering algorithm to choose the cluster head which is based on Whale Optimization Algorithm (WOA). In this approach, the fitness function considers the trust, hop count, centrality, communication range, and remaining energy of the IoT device(s). Compared to other approaches, this method improved the energy consumption, network lifetime and packet delivery rate by 47%, 58% and 17.7% respectively. The work of Sajedi et al [25] developed a model that use data aggregation that is known as F-LEACH to send the data into network. The objective to combine data aggregation with fuzzy logic was to increase the longevity of the entire network to provide enhanced performance in all IoT Healthcare Applications. The researchers used comprehensive simulations to evaluate their proposed method against related methods already available and found that the F-LEACH has, on average, a minimum of 5-20% longer network lifetime than any of the methods reviewed.

K. Kaur and S. Kang [26] proposed a new model based on WSN routing system based on the Ad hoc On-Demand Distance Vector (AODV) protocol and a Cluster Head (CH). By implement this approach optimize the PDR (Packet Delivery Ratio), latency and energy use as QoS criteria. This technique is used to discover the route with the help of updating of cluster head position. A deep neural network (DNN) with a sigmoid activation function, trained over 100 epochs, classifies routes and distributes scores equally among nodes to rank nodes based on route discovery performance. The suggested approach outperformed Ismail et al. and Bashar et al. in throughput, PDR, and latency, with 1.347652857, 0.839435714, and 0.47388, respectively. These improvements make network communication more dependable, efficient, and responsive, making the suggested technique ideal for many applications. By integrating QoS with this technique, improve the performance and security standards and standard deviation analysis in safe routing principles.

Antoniou et al [27] proposed congestion control technique by implement the brid flocking to reduce network congestion. By implementing an artificial magnetic field with force of attraction and repulsion between packets sending from source to the sink, the flock-based congestion Control (Flock-CC) protocol behave packets like birds the flock towards to sink node by awed crowed areas suggested a WSN congestion control technique that makes use of bird flocking behavior to reduce network congestion. To implement self-organizing features in decentralized framework by implement swarm intelligence-inspired method to deploy and information sharing. Flock-CC outperforms conventional congestion management techniques provide the scalability across different network that has different size and efficient load balancing. This protocol dependence on packet clustering and impose in high-traffic circumstances.

Kalaikumar & Baburaj [28] develop an algorithm ti handle WSN congestion by combined Fuzzy Cross-Layer Optimization with Oppositional Artificial Bee Colony (FCOABC). Oppositional Artificial Bee Colony (OABC) optimization provides energy-efficient and reliable routing in FCOABC. This is implemented by Link Superiority Pointer (LSP), node density and residual energy to choose the Cluster head (CH) in Master Station. Small cluster which are near to the MS alleviate inter cluster congestion and bigger cluster transmit data well. For multi-hop inter-cluster routing uses OABC to balance the data transfer in different cluster and balance energy consumption also improve lifespan of network. It can minimize the CH energy usage and congestion and protocol complexity by CH selection technique and inter-cluster routing may add computing cost, limiting scalability in large or resource-limited networks. Gali & Nidumolu [29], by integrating Chaos into the BBMO process, the researchers have developed a means to speed up the rate of convergence of the BBMO algorithm. This algorithm combined with a Trust Sensing Model (CBBMO-R-TSM) which is secure method to transmitting data over network. To provide security in routing protocol, this model uses a sensing trust model approach which can implement in both phases (direct and indirect Trust) which is used to calculate the trust value of nodes in distributed environment. It provides the established secure data transmission route via Routing by Appling the Trust Sensing Model with CMBO algorithm for secure data transmission. The average PDR of the CBBMOR-TSM model was 0.931, and the average PLR was 0.069. In contrast, the TRM_IOT had a maximum PLR of 0.219; the OSEAP_IOT method had maximum PLR of 0.161; and the MCTAR-IOT method had maximum PLR of 0.110.

Alipio & Bures [30] proposed a Deep Reinforcement Learning–based Cache-Aware Congestion Control (DRL-CaCC) framework, where agents dynamically learn congestion states and caching decisions to optimize packet forwarding. By jointly considering network congestion and caching behavior, the method adaptively adjusts transmission policies under varying traffic loads. Simulation results demonstrated improvements of 20–40% in throughput and significant reductions in packet loss compared to traditional congestion control mechanisms. However, the approach incurs higher computational complexity due to the training overhead associated with deep reinforcement learning models.

Mazloomi et al [31] suggested a hybrid approach called Fuzzy Structure and Genetic-Fuzzy (FSFG) for the express purpose of controlling congestion in WSNs. To alleviate congestion, the FSFG approach integrates Genetic Algorithm (GA), Support Vector Regression (SVR), Fuzzy Inference Systems (FIS), and Fuzzy C-Means (FCM) clustering. This approach is divided into two stages: Parameter Identification, which uses GA to adjust system parameters, and Structure Identification, which uses fuzzy clustering to classify network nodes according to congestion levels, followed by SVR to guarantee minimum error in membership functions. According to simulation studies, the FSFG algorithm outperforms current congestion management techniques like GASVM and CODA, improving packet delivery rate by 32% and reducing end-to-end time by 28% when compared to baseline methods. Despite its effectiveness, the FSFG method's complexity may provide challenges in networks with limited resources, indicating the need for more optimization for real-world uses.

Khatami et al. [32] proposed a comprehensive framework integrating Fuzzy Deep Reinforcement Learning, Long Short-Term Memory (LSTM) networks, Multi-Objective Artificial Bee Colony (MOABC) optimization, and Double Reconfigurable Intelligent Surfaces (RIS). The proposed approach simultaneously optimizes energy efficiency, secrecy rate, and communication reliability, achieving improvements of 35.4% in energy efficiency and 29.7% in secrecy performance. Nevertheless, the reliance on advanced hardware components and complex architectural design increases implementation and deployment complexity.

Udayaprasad et al. [33] proposed an energy-efficient optimized routing technique for large-scale Industrial IoT (I-IoT) networks by integrating distributed Software-Defined Networking (SDN) with Artificial Intelligence (AI). The SDN-AI framework intelligently manages routing decisions to reduce energy consumption while improving network lifetime, throughput, and packet delivery performance. Simulation results show notable gains in energy efficiency, lower end-to-end delay, and higher packet delivery ratio compared to conventional routing schemes; however, the use of distributed SDN controllers and AI-based decision mechanisms increases control overhead and deployment complexity in large-scale industrial environments.

Sharma et al [34] introduced Secure and Energy Efficient Routing (MHSEER), a meta-heuristic-based WSN-IoT protocol. The protocol learns to make forwarding decisions using hop counts, connection integrity characteristics, and accumulated residual energy. To increase the method's security, this protocol encrypts the data in addition to an anti-encryption mode. In order to achieve dependable learning, the suggested method utilizes a meta-heuristic analysis. The procedure consists of two phases. The first stage uses a heuristic method to improve the selection for reliable data routing. Security at the second level is predicated on a simple and arbitrary computational CEM. The suggested MHSEER protocol is compared to Secure Energy, Secure Trust Routing for Low Power (Sectrust-RPL), Secure Energy-Aware Heuristic-Based Routing (SEHR), and Heuristic-Based Energy-Efficient Routing (HBEER). The packet drop ratio is decreased by 6.51% using SEAMHR. Additionally, it lowers throughput, network latency, energy consumption, and faulty routes by 95.81%, 5.12%, 0.10 ms, and 0.0102 mJ, respectively.

A thorough routing technique was put out by Karunkuzhali et al [35] for IoT-enabled WSNs in smart cities. Their two-stage method first clusters sensor nodes using Chaotic Bird Swarm Optimization (CBSO) and then chooses reliable CHs using Improved Differential Search (IDS). In the next stage, they use lightweight sign cryption for secure data transfer and the Optimal Data Routing (ODM) algorithm for effective routes. After demonstrate with NS2, include up to 80.1% reduced latency, 66.8% longer network lifespan, and up to 90.8% less energy usage. The method combines lightweight encryption, trust, and optimization to provide safe and flexible routing. Despite promising simulation results, its real-world scalability, adaptability, and performance particularly in highly dynamic or heterogeneous smart city environments remain unproven.

Table 1 Comparative Summary of ML/DL and Optimization-Based IoT-Enabled WSN Research

Ref	Methodology	Application Focus	Performance Assessment	Pros	Cons
[20]	DL + Whale Optimization + Firefly Algorithm	Resource allocation	Throughput 96%, Energy efficiency 95%, QoS 75%, Spectrum efficiency 85%, Lifetime 91%	High EE and SE optimization	Multi-stage complexity

[21]	Multi-criteria CH selection, SNR-based routing, LCG-based security	Smart agriculture IoT	Throughput ↑13.5%, Packet drop ↓38.5%, Latency ↓13.5%, Energy ↓16%, Routing overhead ↓26%	Improved communication efficiency and energy usage	Limited adaptability; lightweight security only
[22]	Levy Chaos Adaptive Snake Optimization (LCASO) + multi-Trust QoS routing	Secure QoS routing	Energy ↓49.53%, Latency ↓22.56%, Packet loss ↓40.21%, Bandwidth ↑6.13%	Fast convergence, effective trust-aware routing	High computational complexity
[23]	Adaptive Cuckoo Search Rate Optimization (ACSRO)	Congestion control	Throughput ↑, Delay ↓, Packet loss ↓, Queue size stabilized (relative improvement)	Effective congestion mitigation	Increased energy and computation overhead
[24]	Fuzzy trust evaluation + Whale Optimization Algorithm clustering	Healthcare IoT	Network lifetime ↑47%, Energy ↓58%, PDR ↑17.7%	Reliable trust-based clustering	Fuzzy rule dependence
[25]	Fuzzy-based LEACH (F-LEACH) data aggregation	Healthcare IoT	Network lifetime ↑5–20%	Improved lifetime via aggregation	No security or congestion awareness
[26]	Deep Neural Network-based secure AODV routing	Secure WSN routing	Throughput ↑1.35, PDR ↑0.84, Latency ↓0.47 (normalized)	DL-based accurate route classification	Training overhead; centralized learning
[27]	Bird flocking-based congestion control (Flock-CC)	Congestion mitigation	Improved load balance, reduced congestion (scalability validated)	Decentralized and fault-tolerant	Delay under heavy traffic
[28]	Fuzzy Cross-Layer Optimization + Oppositional ABC	Congestion & energy efficiency	Network lifetime ↑, CH energy ↓, Congestion ↓ (relative gains)	Hot-spot mitigation	High computational complexity
[29]	Chaotic Bumble Bee Mating Optimization + Trust sensing	Secure routing	PDR = 0.931, PLR = 0.069	Accurate malicious node detection	Increased routing overhead
[30]	Deep Reinforcement Learning cache-aware congestion control (DRL-CaCC)	Congestion & caching	Throughput ↑20–40%, Packet loss ↓	Adaptive congestion control	DRL computational cost
[31]	GA + SVR + Fuzzy Inference + FCM (FSFG)	Congestion control	PDR ↑32%, End-to-end delay ↓28%	Accurate congestion identification	Heavy model complexity
[32]	Fuzzy DRL + LSTM + MOABC + Double RIS	Energy efficiency & security	Energy efficiency ↑35.4%, Secrecy ↑29.7%, RSMA ↑27.5%, R^2 ↑12.3%	Strong adaptability and prediction accuracy	Hardware and architectural complexity

[33]	SDN-AI with PSO, ABC, and GA	Large-scale IoT routing	Data transmission ↑95%, Energy ↓78%, Lifetime ↑63%	Efficient large-scale routing	Single point of failure
[34]	Meta-heuristic Secure and Energy-Efficient Routing (MHSEER)	Secure routing	Packet drop ↓6.51%, Latency ↓5.12%, Energy ↓0.10 mJ	Secure and reliable routing	Encryption overhead
[35]	Chaotic Bird Swarm Optimization + IDS + Sign cryption	Smart city IoT	Latency ↓80.1%, Energy ↓90.8%, Lifetime ↑66.8%	Integrated security and optimization	Scalability invalidated

Despite notable progress in optimization techniques for IoT-enabled wireless sensor networks, existing approaches predominantly address energy efficiency, routing, congestion control, or security as independent problems, without jointly considering their strong interdependencies. In dynamic network environments, due to limited adaptability, high computational overhead and scalability issue, many optimizations-based method suffer. Also, most trust-aware systems are static and reactive, which means they don't work well against advanced threats or traffic that changes quickly. Current frameworks also don't take use of spatio-temporal network relationships to make proactive decisions. This means that resources are not used as well as they might be, and congestion is not addressed as quickly as it could be. These limitations highlight the absence of a unified, intelligent, and trust-aware optimization framework capable of simultaneously handling dynamic resource allocation, congestion-aware routing, and secure communication. To fill these deficiencies, the proposed framework combines spatio-temporal learning with quantum-inspired optimization and composite trust assessment. This makes it possible to do predictive, scalable, and secure network optimization in IoT-enabled WSNs.

3. BACKGROUND

3.1. HHO-Harris Hawks Optimization

The escape energy of prey determines whether a hawk is in exploration or exploitation mode when using the HHO algorithm. There are various search techniques that have varying levels of intervention and escape energy. To identify the next phase of the population in consecutive iterations, it is important to identify the escape energy of prey prior to adjusting its location, and this will provide an accurate representation of the venue used in the next round. The measurement of prey escape energy has a quantitative determination; however, different stages of development have different categories of "Expected" escape energy levels. described as follows [36].

Equation (1) describe a quantity energy (E) that decreases over iterations, controlling the transition from exploration to exploitation. When E is high, hawks explore the search space, and as E decreases toward zero, the algorithm shifts to exploitation, reflecting the prey's reduced ability to escape.

$$E = 2E_0 \times \left(1 - \frac{t}{T_{\max}}\right) \quad (1)$$

where E_0 represents the starting escape energy of the prey, which is a random integer between -1 and 1. $\boldsymbol{t}$ indicates the current number of iterations. $T_{\max}$ denotes the maximum number of iterations.

3.1.1. Exploration Phase

If the escape energy of prey |E|>=1, that is, the prey is aerobic and it is far left behind the hawk, and with this, the hawk will become less deliberate, searching for prey, within a much larger environment. In this case, hawks will have two distinct methods for updating their location depending on whether or not they were successful in determining the location of a particular prey. The hawk will employ a random number, known as q, in order to choose between these two possible options, as illustrated below via the HHO algorithm.

In Exploration phase of the HHO algorithm equation (2) describes how the position of a solution is updated in an optimization process where hawks update their positions either using a random hawk or the best solution with boundary-based randomness, depending on the value of q.

$$(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X(t)|, \, if \; q \geq 0.5 \\ \left(X_{best}(t) - X_m(t)\right) - r_3\left(LB + r_4(UB - LB)\right), if \; q < 0.5 \end{cases} \quad (2)$$

The stochastic variables (r1, r2, r3, r4, and q) take on values from 0 to 1. X(t) is the current position of the hawk, and X(t+1) is the following position of the hawk. X rand(t) is a location of a hawk that has been randomly chosen from the current population, while X best(t) is the present location of the prey, i.e., the global best solution. The upper and lower limits of this dimension refer to the upper limit and lower limit of the hawk's position.

$X_m(t)$ represents the present average location of the population, as given below. Equation (3) calculated the mean position of the hawk population. It is calculated by averaging the positions of all N hawks at iteration t, and it represents the central tendency of the population in the search space.

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \quad (3)$$

where N is the population size. $X_i(t)$ denotes the present location of the ith hawk.

3.1.2. Exploitation Phase

As the escape energy of prey decreases below one (|E|<1), the exploitation phase of the algorithm is initiated by the hawk's will to catch: at this time, the hawk tries to surround the prey. However, since the prey possesses a certain level of energy and has a well-formed escape plan, it is often able to escape the hawk's pursuit. As a result, the hawk has developed a range of four strategies which are used when hunting: a soft besiege strategy, a hard besiege strategy, a soft besiege strategy with progressively more rapid dives as the escape energy of the prey diminishes, and a hard besiege strategy with progressively faster and faster dives. The method that works best for the hawk in each situation depends on the energy level of the prey and the effectiveness of the prey's escape attempts. The HHO approach uses a random number r to estimate the likelihood that the prey will escape. If r is equal to or greater than 0.5, that means the prey has not been able to escape; if r is less than 0.5, then the prey will have escaped. The escape energy of the prey provides a measure of its current level of vitality. If $|E| \geq 0.5$, the prey retains its vitality. If$|E| < 0.5$, the prey is considered exhausted [37].

3.1.2.1. Soft Besiege

If $|E| \geq 0.5$and $r \geq 0.5$, the prey is unable to escape capture yet still maintains its vitality, while the hawk opts to remain stationary above it, thereby increasing the prey's energy depletion and improving the likelihood of a successful capture. The position update formula, outlined as follows, integrates these factors [20]. Equation (4) describe how a solution is updated in an iterative optimization process. It updates the hawk's position by moving it closer to the best solution found so far, where the escape energy E and jump strength J control the step size and intensity of the movement toward the prey.

$$X(t+1) = X_{best}(t) - X(t) - E|J \cdot X_{best}(t) - X(t)| \quad (4)$$

where $J = 2(1 - r_5)$ represents the jump strength of the prey, and applied to find the model the distance the prey can leap. r_5 is a stochastic variable inside the interval between 0 and 1.

3.1.2.2. Hard Besiege

If $|E| < 0.5$ and $r \geq 0.5$, the victim is unable to escape and gradually becomes fatigued, at which point the hawk carries out a rapid and forceful attack, employing a blitz strategy and swiftly closing in on the target. Equation (5) define an update rule for a solution in an iterative optimization process. The hawk updates its position by moving toward the best solution X_{best}, and the escape energy EEE controls how strongly the hawk approaches the prey. The formula for position update is written as follows:

$$X(t+1) = X_{best}(t) - E|X_{best}(t) - X(t)| \quad (5)$$

3.1.2.3. Soft Besiege and Progressive Rapid Dives

If $|E| \geq 0.5$ and $r < 0.5$, Due to the fact that the prey has eluded capture and has stored up enough energy to evade capture, the hawk uses this stored energy to create a number of group diving behaviors that follow the prey's escape route for the next attack by using rapid group dives. The formula for position updates is shown below:

$$Y = X_{best}(t) - E|J \cdot X_{best}(t) - X(t)| \quad (6)$$

Equation (6) capture the new choice of location made according to the previous formula (-1 * a * b) for each hawk in the group. To determine how successful each hawk's approach is regarding locating its prey location, we compare fitness values between Y and X(t) where Fit(Y)indicating that a hawk has been successful in implementing its plan and is now replacing its previous position X(t) with this new position Y. Equation (7) define the next step in updating the solution. When hawks do not perform successfully, they will need to quickly and erratically dive back down toward the new prey location, which we will denote as Z that will be evaluated according to the following procedures, referred to as Z, which is computed as follows:

$$Z = Y + S \times LF(D) \quad (7)$$

where S is a D-dimensional vector, and each component of S is a uniform random integer produced within the range of 0 to 1. LF(D) , known as the Lévy flight, is likewise a D-dimensional vector, whose components can be obtained through the following procedure:

$$LF(x) = 0.01 \times \frac{u \cdot \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left[\frac{\Gamma(1+\beta) \cdot \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}}\right]^{\frac{1}{\beta}} \quad (8)$$

Equation (8) represents the Levy flight (LF) mechanism used in HHO to enhance exploration. LF generates random step sizes using variables u and v, where the parameter β controls the step distribution. The scaling factor σ, computed using the Gamma function, ensures a heavy-tailed distribution, allowing occasional large jumps to help the algorithm escape local optima and explore the search space effectively. Where the variables u and v are random numbers evenly distributed throughout the interval [0, 1].

The variable β denotes a constant, often established with a default value of 1.5. Based on the prior explanation, the soft besiege approach with successive quick dives can be described as:

$$X(t+1) = \begin{cases} Y, if\ Fit(Y) < Fit(X(t)) \\ Z, if\ Fit(Z) < Fit(X(t)) \end{cases} \quad (9)$$

Equation (9) illustrates the selection rule in optimization process in the HHO algorithm where Y and Z are defined by Equations (6) and (7), respectively.

3.1.2.4. Hard Besiege and Progressive Rapid Dives

If |E|<0.5 and r<0.5, then the predator caught its prey but the prey had been able to elude capture by exerting a lot of energy to escape. The hawk, in this situation, will slowly begin to descend, as he continues to pursue the prey as closely as he can. The formula for updating the position is as follows:

$$Y = X_{best}(t) - E|J \cdot X_{best}(t) - X_m(t)| \quad (10)$$

Equation (10) defines a position update rule used in population-based metaheuristic optimization algorithms. It computes a new candidate solution Y by performing a directed stochastic perturbation around the current global best solution. Where Y refers to the most recent acquisition point acquired from the Survivor Database.

Under the gentler approach of besieging a point with numerous progressive quick dives, if the lower of the two Fit(Y) and $Fit(X(t))$ yields a value that is less than $Fit(Y)$, then Y replaces $X(t)$. Conversely, if $Fit(Z)$ yields a value that is greater than Fit(Y) then a new point Z is allocated to that $(Fit(Z))$ value and $X(t+1)$ is derived via Equation (9). But occasionally, traditional HHO has problems with early convergence and restricted global exploration. Quantum-Inspired Harris Hawks Optimization (QHHO) uses ideas from quantum computing to solve this problem. It increases the variety of searches and avoids local optima.

3.1.3. Update of HHO Position

In normal HHO, the prey's fleeing energy E and a random number r that decides the hawks' tactics control the location update. The equation for the general update is:

$$X(t+1) = \begin{cases} X_{prey}(t) - E|JK_{prey}(t) - X(t), & if|E| \geq 0.5 \\ \left(X_{prey}(t) - X(t)\right) - E|JK_{prey}(t) - X(t), & if|E| < 0.5 \end{cases} \quad (11)$$

Equation (11) defines the position update mechanism of a hawk agent at iteration t+1, conditioned on the escaping energy of the prey, E. The rule switches between soft besiege and hard besiege exploitation behaviours.Where X(t) = The current hawk position, $X_{prey}(t)$= The prey's position best solution, $E = 2E_0(1 - t/T)$ indicates the prey's energy reducing linearly with iterations (where, E_0 is initial energy T is the max iterations), J is the jump strength determined by the prey's attempt to escape.

3.1.4. Quantum-Inspired Improvement

Quantum superposition and probabilistic position representation are two examples of quantum computing techniques that are used in QHHO to make solutions more varied. Instead of using deterministic positions, QHHO employs quantum bits (qubits) to show positions as follows:

$$|\psi = \alpha|0\rangle + \beta|1\rangle \quad (12)$$

Equation (12) presents a normalized quantum superposition state of a qubit in a complex Hilbert space.

Where$|\alpha|^2 + |\beta|^2 = 1$ are amplitudes of probability. The position update in QHHO uses quantum rotation gates to change the state, as shown by:

$$\theta_i(t+1) = \theta_i(t) + \Delta\theta_i \quad (13)$$

Equation (13) expressing how the state variable evolves over time. Where θ_i is the angle that describes the qubit state and $\Delta\theta_i$ is the rotation that happens because the present solution is better than the best one. The real position x_i is measured as:

$$x_i = \begin{cases} 1 & if\ rand < \sin^2(\theta_i) \\ 0 & otherwise \end{cases} \quad (14)$$

Equation (14) enables probabilistic exploration of the binary search space. This probabilistic form lets solutions seek a bigger area with more uncertainty, which helps with global exploration and stops early convergence.

One of the main benefits of QHHO is that it improved the balance between exploration and exploitation via quantum superposition. Probabilistic state updates lower the possibilities of becoming stuck in local optima. Faster convergence and better solutions, particularly for optimization problems that are high-dimensional and hard to solve.

3.2. Temporal Graph Networks (TGNs)

Temporal Graph Networks (TGNs) are a class of graph neural network (GNN) architectures designed for continuous-time dynamic graphs – graphs whose nodes, edges or features change arbitrarily over time. In contrast to standard GNNs that assume a static graph, TGNs explicitly model the temporal evolution of interactions.

Real-world networks (social networks, communication, or sensor networks, IoT systems) are often inherently dynamic: links may appear or disappear and node states evolve continuously. Simply applying a static GNN by collapsing time into snapshots is suboptimal (the temporal structure is lost)[38]. TGNs address this by processing a sequence of time stamped events (e.g. node or edge additions/updates) and maintaining a learned memory state for each node that captures its history.

The architecture of a TGN typically comprises the following components:

RESEARCH ARTICLE

i- Memory Module: A vector state $s_i(t)$ for each node s_i, updated over time to record its history of interactions.

ii- Message Function and Aggregator: When an event occurs (e.g. nodes i, j interact at time t), a learnable message is computed for each participant, which is then aggregated if multiple events involve the same node in a batch.

Equation (15) defines $\bar{m}_i(t)$ as an aggregate statistic computed. from a set of historical samples. For example, if events at times $t_1, t_2 \dots, t_b$ generate messages $m_i(t_1)$,$m_i(t_2)$….,$m_i(t_b)$for node i, a TGN aggregates them as is a permutation-invariant function (e.g. sum, mean, or "most recent"). The memory is then updated using this aggregate and the node's previous state. Concretely, if $s_i(t^-)$ denotes nodei′s state just before time t, the update is typically can be a recurrent unit (e.g. GRU/LSTM) or other learnable function [38]. Equation (16) defines the state update with memory, where $s_i(t)$ is computed as a function of the current aggregated input $\bar{m}_i(t)$ and the previous state $s_i(t^-)$ .This mechanism ensures that each event "writes" information into the node's memory state.

$$\bar{m}_i(t) = agg\big(m_i(t_1), m_i(t_2) \dots, m_i(t_b)\big) \quad (15)$$

$$s_i(t) = mem\big(\bar{m}_i(t), s_i(t^-)\big) \quad (16)$$

iii- Embedding Module (Temporal Aggregation): To compute a node's current embedding$z_i(t)$ for use in downstream tasks (e.g. link prediction, node classification), TGNs combine the node's memory with information from its neighbors. A simple choice is to sum or average over a node's temporal neighborhood: for example, one formulation is equation (17) defines $z_i(t)$as a local interaction sum, aggregating the effects of all neighbouring agents j within the time-varying neighbourhood. Where $N_i([0,t])$ is the (multi-hop) neighborhood of i up to time t, e_{ij} is any edge attribute, $v_i(t)$ are any node features, and $h(\cdot)$ is a learned function.

$$z_i(t) = \sum_{j \in N_i([0,t])} h\Big(s_i(t), s_j(t), e_{ij}, v_i(t), v_j(t)\Big) \quad (17)$$

In practice, equation (18) specifies the computation of a variable q through a linear transformation used in learning model. Equation (19) computes the value of variable by integrating into a single representation. Equation (20) define the value representation V as a learnable linear transformation. These equations define the linear projection layers used to construct the query (Q), key (K), and value (V) vectors in an attention mechanism. TGNs often use temporal attention to weight neighbor contributions. In the attention variant, each node i constructs query, key, and value vectors to attend over its active neighbors. For example, letting $s_v$be the updated memory of node v and letting be the memory $\{s_w | w \in N_v\}$matrix of its neighbors, one can compute:

$$q = W_q * [sv \,||\, \varphi(0)] + b_q \quad (18)$$

$$K = W_k * [s_w \,||\, E_{vw} \,||\, \varphi(\Delta t)] + b_k \quad (19)$$

$$V = W_v * [s_w \,||\, E_{vw} \,||\, \varphi(\Delta t)] + b_v \quad (20)$$

where $\varphi(\Delta t)$is a positional time encoding of the elapsed time since each neighbor's last update. The new embedding is then given by the scaled dot-product attention:

$$h_v = softmax((q \cdot K^t) / \sqrt{|N_v|}) \cdot V \quad (21)$$

Equation (21) define an attention-based aggregation where q is the query for node v, K, V are the keys and values for its neighbors (concatenating neighbor states s_w, edge features E_{vw}, and time encodings), and $|N_v|$ is the number of neighbors. This temporal graph attention allows the model to select important neighbors based on both their features and recent activity.

Taken together, a TGN processes an incoming event (i, j, e, t) by computing messages for nodes i and j, aggregating them with any other simultaneous messages Equation (3), updating each node's memory via a recurrent update Equation (4), and then computing node embedding for prediction tasks via temporal neighbor aggregation or attention. The key equation includes the message aggregation:

$$\bar{m}_i(t) = agg(m_i(t_1), \dots, m_i(t\_b)) \quad (22)$$

The memory update:

$$s_i(t) = mem(\bar{m}_i(t), s_i(t^-)) \quad (23)$$

In the above, the temporal attention update is shown. Each of equations (22) and (23) is paired with a learnable function: for instance, mem is often a GRU that ingests the new message $\bar{m}_i(t)$ and the old state $s_i(t^-)$, producing an updated state $s_i(t)$. The time encoding $\varphi(\Delta t)$ (e.g., from Time2Vec or sinusoidal functions) ensures that the attention mechanism is time-aware. These components together address the staleness problem if a node is inactive for a long time, its memory may be out-of-date, but the embedding module (especially attention) can refresh it by aggregating information from recently active neighbors.

Figure 2 illustrates the overarching data flow of a Temporal Graph Network (TGN), an architecture intended for continuous-time dynamic graphs in which the states of nodes, edges, and features progress over time. The model analyzes input events (i, j, e, t) −denoting interactions between nodes i and j with edge attributes e at time t via a systematic sequence of components. The Edge Feature Extractor encodes attributes specific to interactions, after which the Message Generator formulates context-aware messages utilizing node features, edge data, and temporal signals. The messages are subsequently amalgamated utilizing a permutation-invariant Message Aggregator. The consolidated data refreshes node memory via a recurrent unit, usually a GRU, within the Memory Update module. A Temporal Encoder, such as

Time2Vec, encodes time intervals to improve temporal awareness. A Neighbor Sampler concurrently identifies pertinent neighbors based on recent engagements.

The Embedding Generator utilizes graph attention to integrate node memory, temporal encodings, and neighbor data to generate node embedding z(t). The Output Layer employs these embedding for subsequent prediction tasks, including link prediction and node classification.

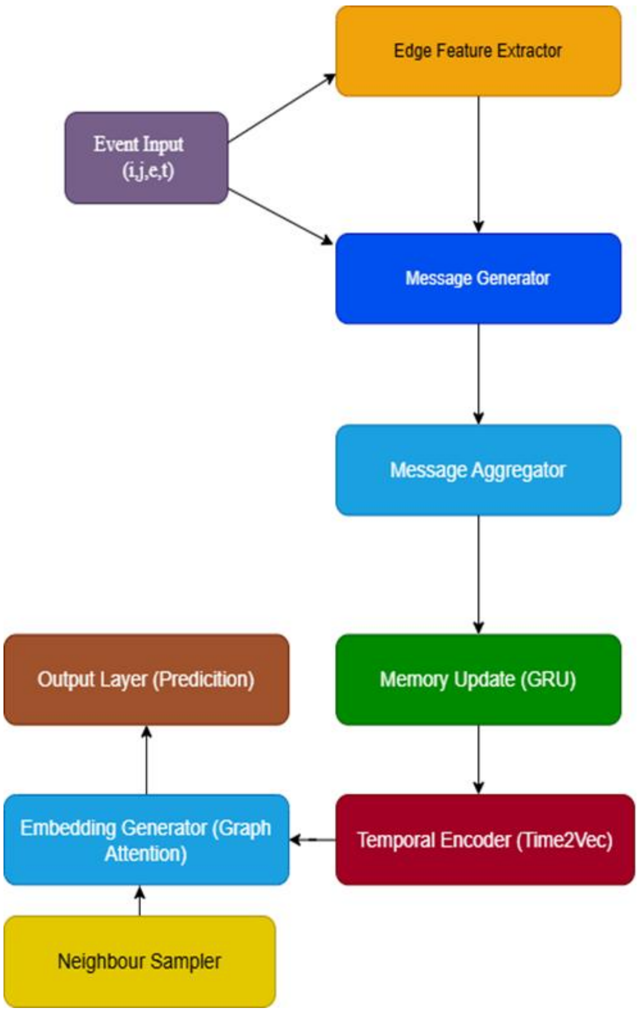

Figure 2 Proposed Temporal Graph Network Architecture for Event-Based Learning

Figure 2 Temporal Graph Network (TGN) architecture illustrating the sequential processing of timestamped events for dynamic graph learning. The model integrates memory-based updates and temporal neighbour-aware embedding generation.

4. METHODOLOGY

The paper introduces an innovative IoT optimization framework that employs TGNs for resource allocation in space and time and QHHO for routing that considers congestion, all backed up by a trust-aware Composite Trust (CT) score mechanism. Also, while the normal IoT optimization methods solved resource allocation, routing, or security separately, the proposed framework was allowed to do these three aspects' joint optimization in a single architecture. The expert TGN does dynamic resource allocation in the network as it reads the temporal traffic patterns and spatial nodes via graph attention mechanisms while QHHO optimally decides the routing paths using the quantum-inspired state representations that give rise to better exploration and convergence. Trust values are systematically set lower (CT < 0.35) so that all nodes are excluded and the network remains always secured and reliable. Framework performance is assessed based on energy efficiency (EE), throughput (TH), execution time (ET), and malicious detection capability (MDR/FPR), which yield superior scalability, adaptability, and security against traditional methods. The operational workflow of the proposed framework is encapsulated in Algorithm 1 and Figure 3 depicts it.

4.1. Data Collection

Synthetic IoT network data generation is the first step to do a controlled evaluation of the proposed model under different operational and adversarial conditions. The dataset includes node characteristics, behavior patterns, traffic congestion indicators, and trust-related parameters, which enables a thorough evaluation of resource allocation, routing, and security performance.

4.1.1. Network Parameters

A simulation of the IoT model contains 50-150 sensor nodes along with 10 gateways, which were randomly placed in a 100 x 100 square area. Every node has an assigned Residual Energy level denoting battery strength and can be between 0.5 - 1.0. The length of the queue for each node is equally distributed between 0 - 1, where the greater the number, the higher the congestion level. Each node also has a connection quality that can be valued between 0.7 - 1.0, where 0.7 is low reliability and 1.0 is maximum reliability. In adversarial circumstances, for example, 5 percent of nodes were randomly selected as malicious and were therefore able to launch aggressive actions, such as dropping packets. As for gateways, the resource availability of the gateways ranged from 0.5 - 1.0, while their processing time was between 0.1 - 0.5 seconds, thus replicating real-life constraints found in Edge Computing.

4.1.2. Behavioral Data

The dataset includes both types of activities, both legitimate and deceitful. Normal nodes have a constant rate of transmission and reception and low occurrences of packet loss. Malicious nodes create large numbers of packets that

RESEARCH ARTICLE

should be dropped, which results in a very high number of rejected and undeceived messages. For some types of advanced attack techniques (e.g., wormhole attacks), in addition to monitoring RSSI values, we can also monitor the movements of nodes and check for aberrations from the expected behavior to try to identify any anomalous communication patterns.

4.1.3. Trust Data

Trust of a node is evaluated using three elements: Direct Trust (DT) established by the interactions between the nodes, Indirect Trust (IT) provided by neighbor nodes through their recommendations and Energy Trust (E) derived from residual power levels. All three elements are summed together to form a single Composite Trust (CT) score for each node with values from 0 to 1. When the CT score of a node falls below 0.35 it is classified as malicious and the dynamic security evaluation process will continue until either the node is removed from the network or the performance of the network improves due to the adaptive trust evaluation method used to determine CT, improve performance.

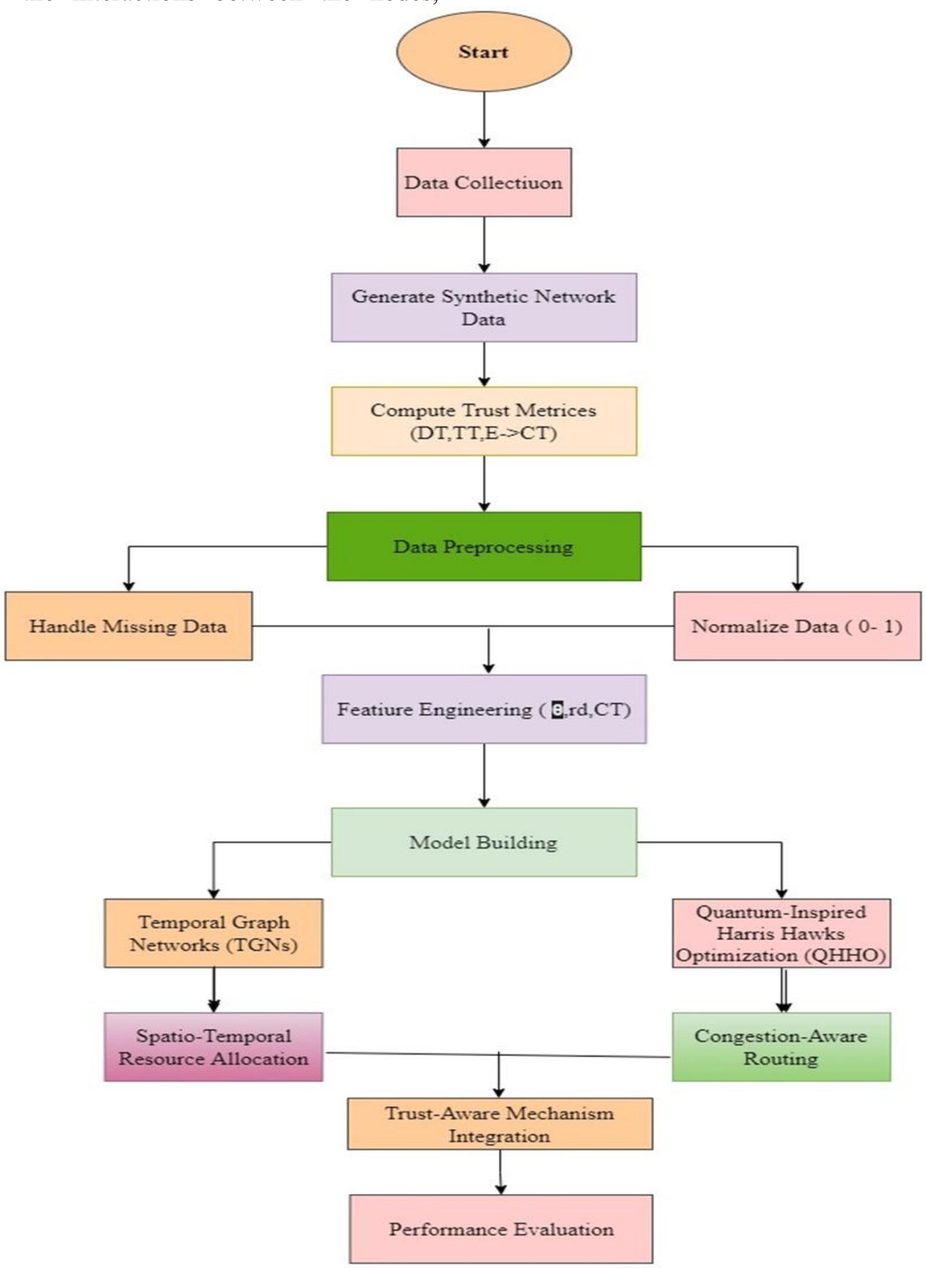

Figure 3 Flowchart of Proposed Model

RESEARCH ARTICLE

4.2. Data Pre-Processing

4.2.1. Normalization

Using Min-Max Scaling is a standardization technique that allows us to scale the collected IoT Network data so that it all falls within the min-max bounds [39]. Energy consumption, Queue Wait Times, Link Quality Indicators, and Trust Scores will all be standardized prior to model training so as not to introduce bias into the model training process. Scaling the data uniformly ensures that we can compare performance across algorithms (e.g. TGN, QHHO) without affecting the overall quality of the training data.

4.2.2. Handling Missing Data

If there are errors in the data or missing connections (such as a computer not able to connect to another), we assign a default score of 0.5. For things that are suspect, such as an excessive trust score for a connection, we also review the other data (such as the number of dropped packets). Additionally, the smoothed average of the signals or energy readings may also help determine if the trust ratings are valid [40].

4.2.3. Feature Engineering

The use of feature engineering to extract critical numerical data is how this research improved the model. The adaptive penalty coefficient (θ) was implemented to deter malicious behavior, while the RSSI deviation (rd) was used to detect wormhole attacks. A composite trust score (CT) integrates direct, indirect and energy trust and helps to evaluate each node holistically to provide comprehensive detection of attacks and allow for better precision in high-speed IoT networks [41].

4.3. Model Building

The proposed framework is a trust-aware hybrid optimization architecture that uses Temporal Graph Networks (TGNs) for spatio-temporal resource allocation and Quantum-Inspired Harris Hawks Optimization (QHHO) for congestion-aware routing in IoT-based Wireless Sensor Networks.

The suggested system optimizes routing, resource allocation, and security in a single learning–optimization loop coordinated by an adaptive Composite Trust (CT) mechanism, unlike current approaches.

4.3.1. Composite Trust Modelling

To dynamically assess node reliability, a Composite Trust (CT) score is computed for each node by aggregating direct interaction trust, indirect (neighbor-based) trust, and residual energy trust:

$$CT_i = w_1 \cdot DT_i + w_2 \cdot IT_i + w_3 \cdot E_i \quad (24)$$

Equation (24) defines the CT_i as a composite trust (or cost) metric formed by a linear weighted combination of three component measures DT_i, IT_i, and Ei. Where DT_i, IT_i, and E_i denote direct trust, indirect trust, and normalized residual energy of node i, respectively, and w1+w2+w3=1 Nodes with CT_i <0.35 are classified as malicious and excluded from subsequent resource allocation and routing decisions. This adaptive trust mechanism enables real-time identification of unreliable or compromised nodes.

4.3.2. Spatio-Temporal Resource Allocation Using TGNs

The IoT-enabled WSN is represented as a dynamic graph $G = (V, E)$, where sensor nodes form the vertex set V and communication links constitute the edge set E. Temporal Graph Networks are employed to capture both temporal dynamics (e.g., energy depletion, traffic fluctuation) and spatial dependencies among nodes.

$$h_i^{(t)} = \text{TGN}\left(h_i^{(t-1)}, x_i^{(t)}, N_i\right) \quad (25)$$

Equation (25) defines the temporal node embedding update in the Temporal Graph Network (TGN) framework. For each node i at time t, the TGN learns a latent embedding $h_i^{(t)}$, where $x_i^{(t)} = [E_i, Q_i, CT_i]$ represents the node feature vector, and N_i denotes the neighborhood of node i.

Based on graph attention weights, the resource allocation score for node i is defined as:

$$RA_i = \alpha \cdot \mathrm{h}_i^{(t)} \cdot CT_i \quad (26)$$

Equation (26) defines a relevance or attention score (RA_i) for node iii by combining its temporal embedding with a composite trust/cost metric. Nodes with $CT_i < 0.35$ are assigned $RA_i = 0$, ensuring that insecure or unreliable nodes do not receive computational or bandwidth resources. This mechanism enables predictive and trust-sensitive resource allocation, outperforming static or reactive allocation strategies.

4.3.3. Quantum-Inspired Congestion-Aware Routing Using QHHO

Following resource allocation, routing optimization is performed using Quantum-Inspired Harris Hawks Optimization (QHHO). Each candidate routing path is encoded as a quantum-inspired superposition state:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle, |\alpha|^2 + |\beta|^2 = 1 \quad (27)$$

This equation (27) representation enables simultaneous exploration of multiple routing alternatives and improves convergence in high-dimensional search spaces.

During optimization, equation (28) represents a 2D rotation of a vector (α, β) by an angle $\Delta\theta$ and quantum rotation gates update the probability amplitudes:

$$\begin{bmatrix}\alpha' \\ \beta'\end{bmatrix} = \begin{bmatrix}\cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta\end{bmatrix}\begin{bmatrix}\alpha \\ \beta\end{bmatrix} \quad (28)$$

Paths containing nodes with $CT < 0.35$ or exhibiting abnormal RSSI deviations (indicative of wormhole attacks) are penalized by increasing the rotation angle $\Delta\theta$, thereby reducing their selection probability.

In equation (29) defines a weighted multi-objective fitness function combining link quality, trust, and energy efficiency. The routing fitness function is defined as:

$$F = \lambda_1 \cdot LQ + \lambda_2 \cdot CT + \lambda_3 \cdot \frac{1}{E_{tx}} \quad (29)$$

Where LQ is link quality, CT represents average trust along the path, $E_{tx}$is transmission energy, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

4.3.4. Integrated Trust-Aware Optimization Framework

The Composite Trust mechanism is the major coordinating element for the TGNs and QHHO. Trust values are perpetually determining the extent to which resources are weighted during the learning phase and the selection of routes during the optimization phase. Furthermore, the analysis of RSSI deviation is applied to identify advanced kinds of attacks, such as the wormhole attack, thus, allowing to prevent taking some paths and isolate some nodes proactively.

The model that is proposed by the integration of spatio-temporal learning, quantum-inspired optimization, and adaptive trust assessment attains energy-efficient, low-latency, and secure IoT communication. This close-knit design brings about a significant improvement in scalability and robustness over conventional IoT frameworks that are based on either optimization or learning only.

4.4. Performance Metrics

Energy Efficiency (EE): Assesses the network's energy conservation efficiency. Equation (30) define the average energy efficiency over N nodes.

$$EE = \frac{1}{N}\sum_{i=1}^{N}\left(\text{Energy}_i \times \left(1 - \frac{\text{Distance}_i}{\text{Max Distance}}\right)\right) \quad (30)$$

Where:

- Energy_i= Residual energy of node i
- Distance_i=Distance from node I to it's assigned gateway
- N =Total number of nodes

Throughput (TH): Assesses the efficacy of data transmission rates. Equation (31) defines throughput (TH) as the average link quality over K links

$$TH = \frac{\sum_{j=1}^{K} Link\_Quality_i}{K} \quad (31)$$

Where:

- Link_Quality_i=Signal reliability of cluster j
- K= Number of clusters

Execution Time (ET): Equation (32) evaluates the computational efficiency of resource allocation and routing.

$$ET = T_{RA} + T_{Routing} \quad (32)$$

Where:

- T_{RA}=Time taken for LJHSO based resource allocation
- $T_{Routing}$= Time taken for LFWBO-based routing

Malicious Node Detection Rate (MDR): Assesses the efficacy of security measures in detecting intruders. In Equation (33), the Malicious Node Detection Rate is defined as:

$$MDR = \frac{True\ Positives(TP)}{TP + False\ Negatives(FN)} \quad (33)$$

False Positive Rate (FPR): Equation (34) measures incorrect classification of benign nodes as malicious.

$$FPR = \frac{False\ Positives(FP)}{FP + True\ Negatives(TN)} \quad (34)$$

To find Trust- Aware optimization in IOT algorithm is defined as shown in algorithm 1.

1. Input: Synthetic IoT data (nodes, gateways, trust)
2. Output: Optimized resource allocation, routing, security
3. Procedure Initialize Network
4. Generate 50-150 nodes, 10 gateways in 100x100 area
5. Assign energy (0.5-1.0), queue length (0-1), link quality (0.7-1.0)
6. Tag 5% nodes as malicious with packet dropping
7. End procedure
8. Procedure Compute Trust
9. Calculate DT (interactions), IT (neighbours), E (residual energy)
10. Aggregate into CT; if CT < 0.35, mark as malicious
11. End procedure
12. Procedure Pre-process Data
13. Normalize all features to [0,1]
14. Assign default trust=0.5 for missing neighbours
15. Smooth RSSI/energy outliers via rolling average
16. End procedure
17. Procedure Feature Engineering
18. Extract adaptive penalty (θ), RSSI deviation (rd), CT

RESEARCH ARTICLE

19. End procedure
20. Procedure Spatio-Temporal Allocation with TGNs
21. Construct graph G = (V,E) with nodes V and edges E
22. Train TGN with 10-timestep window on [Energy, CT, Queue]
23. Allocate resources using graph attention weights
24. Set allocation=0 for nodes with CT < 0.35
25. End procedure
26. Procedure Quantum Routing with QHHO
27. Initialize quantum population: $|path\rangle = \alpha|0\rangle + \beta|1\rangle$
28. Evaluate fitness: Link Quality + CT + $\frac{1}{\text{Transmit Power}}$
29. Apply quantum rotation to penalize paths with:
30. CT < 0.35 OR RSSI deviation > threshold
31. Measure collapsed paths for final routing
32. End procedure
33. Procedure Evaluate Performance
34. End procedure

Algorithm 1 Trust-Aware IoT Optimization Using TGNs and QHHO

5. RESULT

This section provides a thorough assessment of the proposed Hybrid TGN+QHHO model in comparison to four baseline models LSTM-GRU, Temporal Graph Networks (TGN), Graph Convolutional Networks (GCN), and Graph Attention Networks (GAT) across five performance metrics: Energy Efficiency (EE), Make-span, Throughput (TH), Execution Time (ET), and Task Completion Rate (TCR). The models were evaluated on expanding network sizes 50, 100, and 150 nodes at consistent time intervals from 0 to 1000. The findings, illustrated via radar charts and presented in tabular format, underscore the persistent dominance of the Hybrid model in enhancing resource allocation and routing within WSN-based IoT settings.

5.1. Simulation Parameters

All experiments were conducted in a controlled simulation environment to ensure reproducibility and fair evaluation of the proposed model. The simulation was implemented using Python, leveraging standard deep learning and graph processing libraries. The environment was configured to support temporal graph learning, message passing, and memory-based updates across dynamic interactions. Model training and evaluation were performed on a workstation equipped with GPU acceleration to efficiently handle large-scale event streams. Key hyper parameters and system configurations were kept fixed across experiments unless stated otherwise, allowing consistent comparison with baseline methods. The detailed simulation parameters and environment settings are summarized in Table 2.

Table 2 Simulation Parameters and Environment Settings

Parameter	Value
Programming Language	Python 3.x
Deep Learning Framework	PyTorch
Graph Processing	PyTorch Geometric / DGL
Temporal Encoding	Time2Vec
Memory Update Module	GRU
Embedding Dimension	128
Batch Size	200
Learning Rate	0.001
Optimizer	Adam
Number of Epochs	50
Neighbor Sampling Size	10

Across all evaluated metrics for the 50-node network, it is evident that the suggested Hybrid TGN+QHHO model performs better than existing models. Table 3 present a comparative evaluation of five models Hybrid TGN+QHHO, LSTM-GRU, TGN, GCN, and GAT on a 50-node network, considering energy efficiency, make span, throughput, execution time, and task completion rate across time steps ranging from 0 to 1000. Overall, the Hybrid TGN+QHHO model consistently delivers superior performance across all evaluated criteria.

In terms of energy efficiency, as illustrated in Figure 4, the Hybrid model maintains stable values between 0.729 and 0.767, achieving its peak value of 0.767 at step 100. In contrast, the GAT model performs comparatively poorly, with values not exceeding 0.603 and dropping to a minimum of 0.55. Although the TGN model starts with a relatively high value of 0.798, its performance gradually decreases to 0.701 by step 800.

Throughput is another area where the Hybrid proves to be strong, as illustrated in Figure 5, with high values at all-time steps, hitting a maximum of 0.877 at the 1000th step and never going lower than 0.826. This is already better than the GAT model performance that has its throughput peak at 0.802. Regarding make span, as shown in Figure 6, the Hybrid model scores the minimum measure of 0.513 at the 700th step and stays under 1.4 throughout the evaluation, showing that it is faster in completing the tasks. On the other hand, TGN and GAT have much higher increases in make span and thus the highest values of 2.245 and 2.187, respectively, at the 900th step, which is indicative of increased processing delays. When it comes to execution time, as shown in Figure 7, the Hybrid model shows lower latency, with values of 4.662 at step 300 and 5.228 at step 600. On the other hand, GAT and TGN incur much higher execution times, reaching 8.624 and 8.782, respectively, at step 900. The Hybrid TGN+QHHO model, as illustrated in Figure 8, consistently displays the highest task completion rates, which vary from 0.94 to 0.97 and culminate in a score of 0.97 at step 1000. GAT and GCN, on the other hand, suffer from a significant drop, falling to 0.77 and 0.80, respectively. These findings clearly demonstrate that the Hybrid TGN+QHHO model is more effective in reducing energy consumption, minimizing latency, improving throughput, and ensuring dependable task execution in 50-node IoT-based wireless sensor network environments.

Table 3 Performance for All Models with 50-Nodes

Metric	Model	0	100	200	300	400	500	600	700	800	900	1000
Energy Efficiency (EE)	Hybrid TGN+QHHO	0.75	0.767	0.753	0.751	0.743	0.762	0.743	0.729	0.74	0.737	0.735
	LSTM-GRU [42][43]	0.721	0.75	0.742	0.701	0.728	0.742	0.724	0.711	0.733	0.728	0.721
	TGN [44]	0.798	0.744	0.723	0.7	0.707	0.703	0.72	0.709	0.701	0.712	0.709
	GCN [45]	0.612	0.672	0.691	0.698	0.687	0.681	0.682	0.659	0.658	0.676	0.668
	GAT [46]	0.603	0.563	0.581	0.609	0.578	0.592	0.573	0.55	0.572	0.568	0.56
Make span	Hybrid TGN+QHHO	0.898	0.87	0.65	0.872	1.254	0.677	0.53	0.513	0.551	1.388	0.746
	LSTM-GRU [42][43]	1.452	0.892	0.721	0.893	1.892	1.124	0.873	0.842	0.921	2.012	1.124
	TGN [44]	1.623	1.012	0.832	1.021	2.124	1.312	0.982	0.952	1.042	2.245	1.312
	GCN [45]	1.521	0.952	0.782	0.952	2.012	1.213	0.921	0.892	0.982	2.124	1.213
	GAT [46]	1.572	0.982	0.812	0.982	2.124	1.245	0.952	0.921	1.012	2.187	1.245
Throughput (TH)	Hybrid TGN+QHHO	0.852	0.815	0.853	0.847	0.85	0.861	0.859	0.826	0.87	0.86	0.877
	LSTM-GRU [42][43]	0.782	0.752	0.792	0.784	0.789	0.798	0.796	0.762	0.802	0.792	0.812
	TGN [44]	0.752	0.721	0.762	0.754	0.759	0.768	0.766	0.732	0.772	0.762	0.782
	GCN [45]	0.752	0.732	0.772	0.764	0.769	0.778	0.776	0.742	0.782	0.772	0.792
	GAT [46]	0.752	0.742	0.782	0.774	0.779	0.788	0.786	0.752	0.792	0.782	0.802
Execution Time (ET)	Hybrid TGN+QHHO	0.752	6.066	4.731	4.662	6.831	4.939	5.228	4.981	5.667	6.421	5.086
	LSTM-GRU [42][43]	0.752	7.892	6.124	6.012	8.245	6.312	6.782	6.452	7.124	8.012	6.312

RESEARCH ARTICLE

	TGN [44]	0.752	8.245	6.892	6.782	8.892	6.982	7.452	7.124	7.892	8.782	6.982
	GCN [45]	0.752	7.982	6.452	6.312	8.452	6.782	7.124	6.892	7.452	8.312	6.782
	GAT [46]	0.752	8.124	6.782	6.624	8.782	6.982	7.312	7.012	7.782	8.624	6.982
Task Completion Rate (TCR)	Hybrid TGN+QHHO	0.95	0.94	0.96	0.96	0.95	0.96	0.96	0.95	0.96	0.94	0.97
	LSTM-GRU [42][43]	0.92	0.93	0.91	0.9	0.88	0.91	0.9	0.89	0.91	0.87	0.9
	TGN [44]	0.88	0.89	0.87	0.86	0.84	0.87	0.86	0.85	0.87	0.83	0.86
	GCN [45]	0.85	0.86	0.84	0.83	0.81	0.84	0.83	0.82	0.84	0.8	0.83
	GAT [46]	0.82	0.83	0.81	0.8	0.78	0.81	0.8	0.79	0.81	0.77	0.8

Table 4 Performance for All Models with 100-Nodes

Metric	Model	0	100	200	300	400	500	600	700	800	900	1000
Energy Efficiency (EE)	Hybrid TGN+QHHO	0.76	0.787	0.773	0.781	0.783	0.762	0.773	0.779	0.783	0.777	0.775
	LSTM-GRU [42][43]	0.607	0.675	0.648	0.647	0.684	0.648	0.63	0.657	0.639	0.674	0.367
	TGN [44]	0.584	0.54	0.559	0.488	0.553	0.569	0.548	0.525	0.547	0.543	0.535
	GCN [45]	0.512	0.572	0.591	0.618	0.587	0.601	0.582	0.559	0.58	0.576	0.568
	GAT [46]	0.503	0.563	0.581	0.609	0.578	0.592	0.573	0.55	0.572	0.568	0.56
Make span	Hybrid TGN+QHHO	0.94	0.912	0.692	0.914	1.296	0.719	0.572	0.555	0.593	1.43	0.788
	LSTM-GRU [42][43]	1.572	0.982	0.812	0.982	2.124	1.245	0.952	0.921	1.012	2.187	1.245
	TGN [44]	1.452	0.892	0.721	0.893	1.892	1.124	0.873	0.842	0.921	2.012	1.124
	GCN [45]	1.623	1.012	0.832	1.021	2.124	1.312	0.982	0.952	1.042	2.245	1.312
	GAT [46]	1.521	0.952	0.782	0.952	2.012	1.213	0.921	0.892	0.982	2.124	1.213
Throughput (TH)	Hybrid TGN+QHHO	0.876	0.871	0.874	0.867	0.875	0.881	0.874	0.864	0.874	0.875	0.872
	LSTM-GRU [42][43]	0.847	0.847	0.847	0.847	0.847	0.847	0.847	0.847	0.847	0.847	0.847
	TGN [44]	0.854	0.854	0.854	0.854	0.854	0.854	0.854	0.854	0.854	0.854	0.854
	GCN [45]	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84
	GAT [46]	0.845	0.845	0.845	0.845	0.845	0.845	0.845	0.845	0.845	0.845	0.845
Execution Time (ET)	Hybrid TGN+QHHO	5.013	5.433	6.114	6.69	8.77	5.36	9.395	4.944	6.508	4.939	5.521
	LSTM-GRU [42][43]	7.37	8.138	6.37	6.258	8.491	6.558	7.028	6.698	7.37	8.258	6.558

	TGN [44]	8.138	8.491	7.138	7.028	9.138	7.228	7.698	7.37	8.138	9.028	7.228
	GCN [45]	7.698	8.228	6.698	6.558	8.698	7.028	7.37	7.138	7.698	8.558	7.028
	GAT [46]	8.028	8.37	7.028	6.87	9.028	7.228	7.558	7.258	8.028	8.87	7.228
Task Completion Rate (TCR)	Hybrid TGN+QHHO	0.94	0.94	0.95	0.96	0.95	0.96	0.96	0.95	0.96	0.94	0.97
	LSTM-GRU [42][43]	0.91	0.92	0.9	0.91	0.91	0.91	0.92	0.89	0.91	0.87	0.9
	TGN [44]	0.88	0.89	0.87	0.86	0.84	0.87	0.86	0.85	0.87	0.87	0.86
	GCN [45]	0.85	0.86	0.85	0.83	0.88	0.84	0.85	0.82	0.84	0.8	0.88
	GAT [46]	0.83	0.83	0.81	0.8	0.78	0.81	0.8	0.79	0.81	0.77	0.8

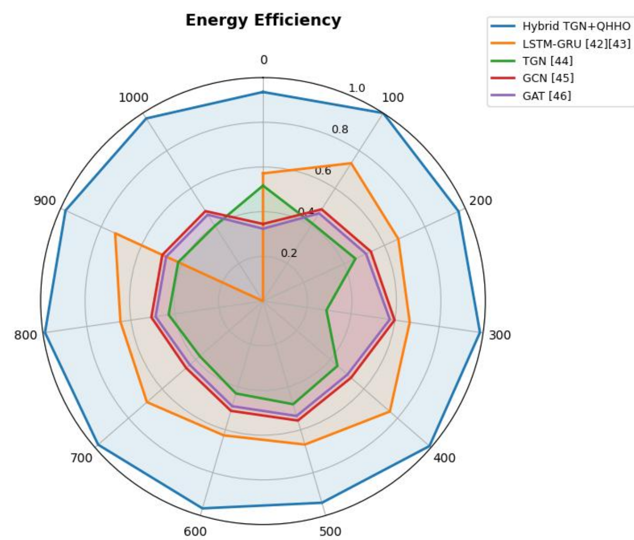

Figure 4 Energy Efficiency (EE) Comparison of All Models for 50 Nodes Over Time

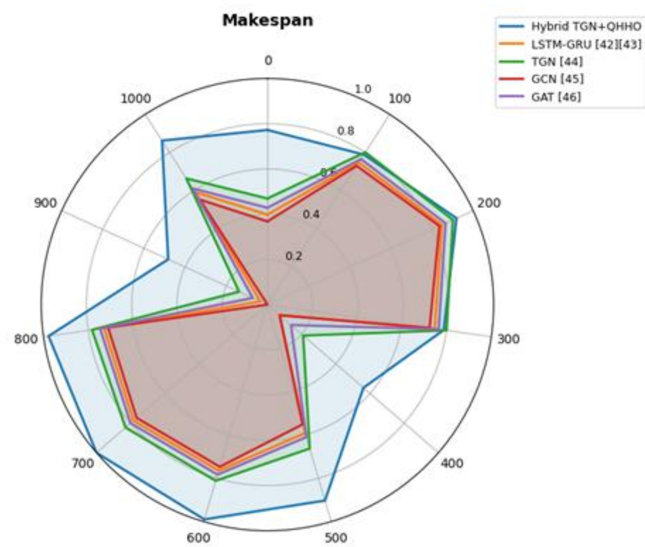

Figure 5 Make Span Comparison of All Models for 50 Nodes Over Time

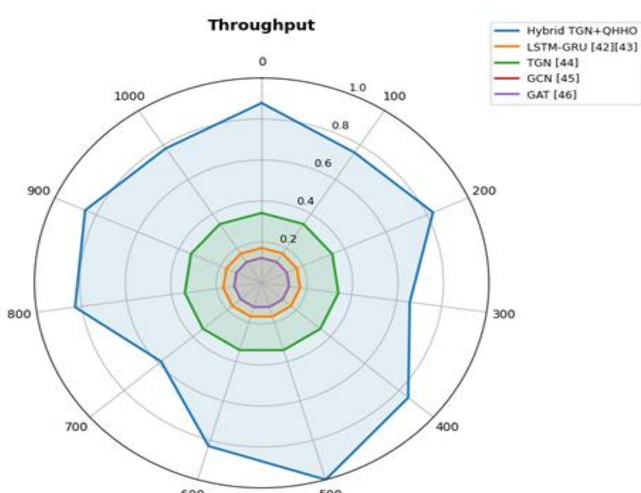

Figure 4 Throughput Comparison of All Models for 50 Nodes Over Time

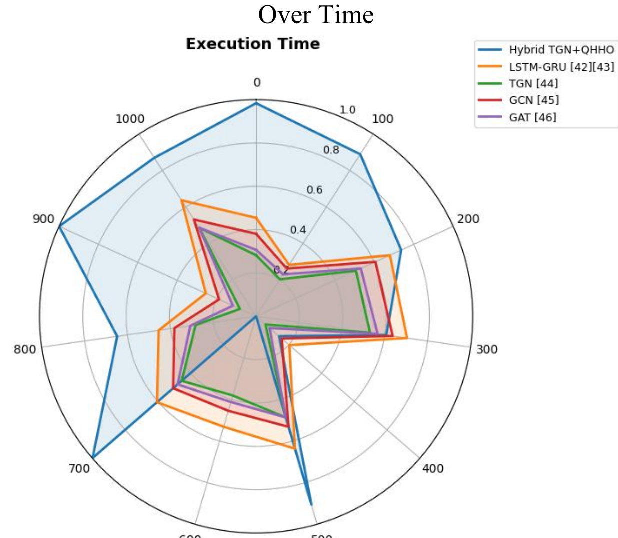

Figure 6 Execution Time Comparison of All Models for 50 Nodes Over Time

RESEARCH ARTICLE

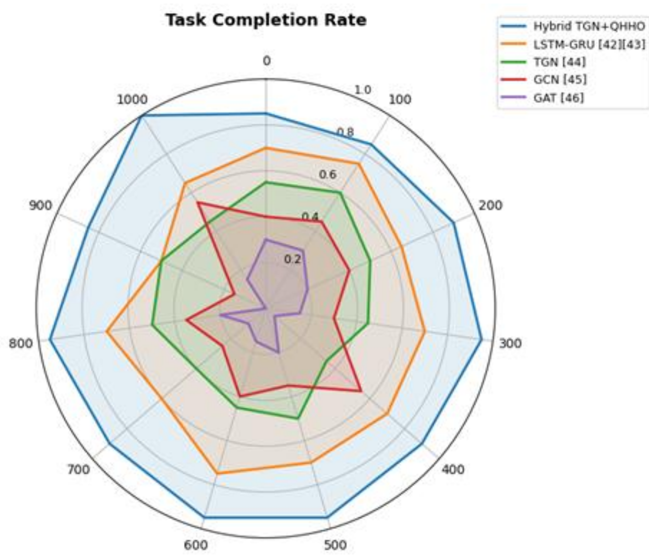

Figure 7 Task Completion Rate Comparison of All Models for 50 Nodes Over Time

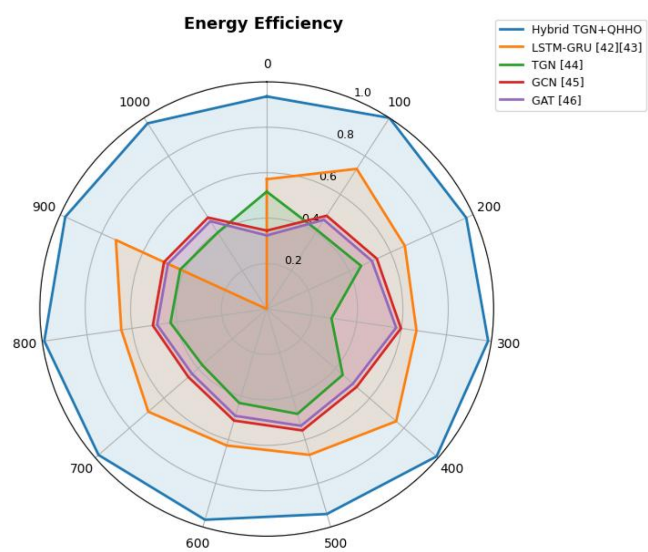

Figure 8 Radar Chart Showing the Energy Efficiency Performance of Five Models Across Different Time Steps for 100 Nodes

The Hybrid TGN+QHHO model outperforms all baseline models by a significant margin in the 100-node configuration scenario. The comprehensive performance assessment of the five models, Hybrid TGN+QHHO, LSTM-GRU, TGN, GCN, and GAT, in terms of the most important performance metrics: Energy Efficiency (EE), Makespan, Throughput (TH), Execution Time (ET), and Task Completion Rate (TCR) is provided in Table 4, in the time steps from 0 to 1000 in increments of 100. The analysis clearly indicates that the Hybrid TGN+QHHO model consistently outperforms all baseline models by a considerable margin in the 100-node configuration scenario.

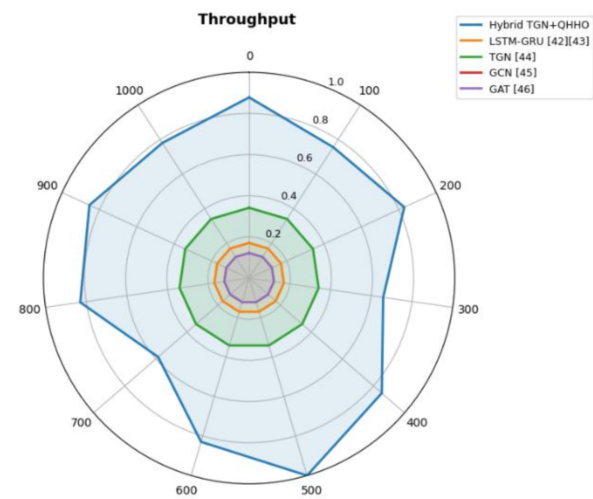

Figure 9 Radar Chart Showing the Throughput Performance of Five Models Across Different Time Steps for 100 Nodes

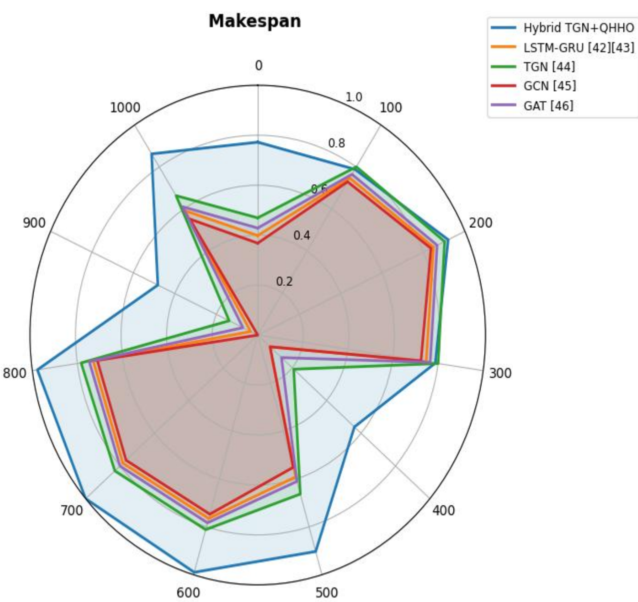

Figure 10 Radar Chart Showing the Make Span Performance of Five Models Across Different Time Steps for 100 Nodes

The Hybrid TGN+QHHO model performs excellently on energy efficiency metric, maintaining values above 0.76 throughout, reaching the maximum of 0.787 at step 100, and sustaining over 0.77 for the majority of intervals, as illustrated in Figure 9. In contrast, LSTM-GRU achieves a maximum of 0.684 at step 400 but then steeply falls to 0.367 by the end of step 1000. While GCN and GAT are still far behind, GCN achieves a maximum of 0.618 and GAT gets up to 0.609; TGN begins with 0.584, then decreases to 0.488 at step 300,

RESEARCH ARTICLE

and remains below 0.57 thereafter. In terms of makespan, the Hybrid TGN+QHHO method always shows low and stable values, thereby surpassing LSTM-GRU and TGN, which hit 2.012 and 2.245 at step 900, respectively, with a low of 0.555 at step 700 and a high of 1.43 at step 900, as illustrated in Figure 10. The Hybrid TGN+QHHO throughput is high, ranging slightly above 0.864 and below 0.881, and it also outperforms the constant values of LSTM-GRU, TGN, GCN, and GAT (0.847, 0.854, 0.84, and 0.845, respectively) at all time steps, as shown in Figure 11.

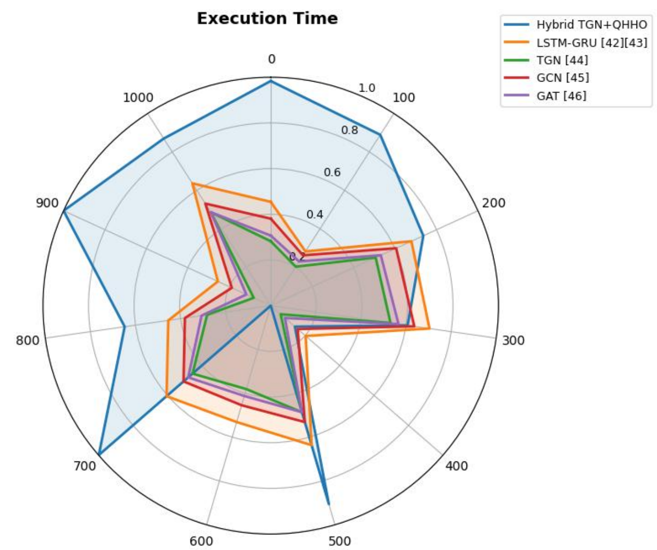

Figure 11 Radar Chart Showing the Execution Time Performance of Five Models Across Different Time Steps for 100 Nodes

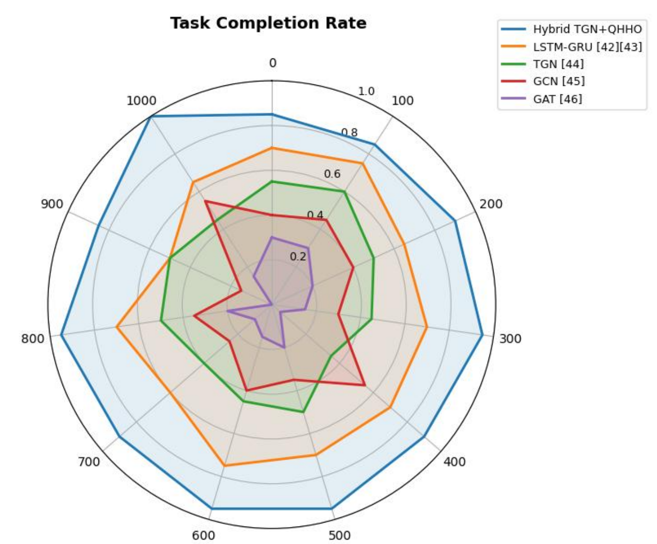

Figure 12 Radar Chart Showing the Task Completion Time Performance of Five Models Across Different Time Steps for 100 Nodes

While LSTM-GRU, TGN, and GAT consistently cause overhead, with GAT reaching the maximum of 9.028 at step 400 and again at 8.87 at step 900, the Hybrid TGN+QHHO is more efficient in terms of execution time, with values ranging from 4.944 to 9.395, as illustrated in Figure 12. Finally, in terms of task completion rate, the Hybrid model shows the advantage with values from 0.94 to 0.97, reaching the top at step 1000, while LSTM-GRU covers a 0.87 to 0.92 range, and GAT has the lowest TCR, falling to 0.77 at step 900, as shown in Figure 13. The results show that in the case of 100 nodes, the Hybrid TGN+QHHO model guarantees better performance across all five parameters, indicating high efficiency, low latency, strong throughput, and increased dependability in task completion.

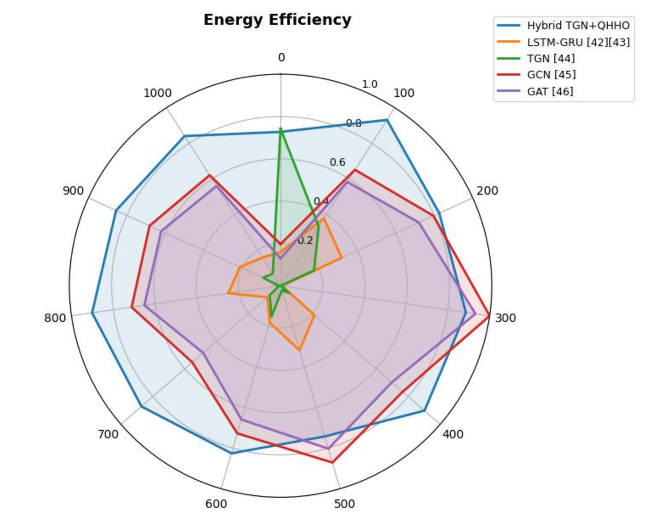

Figure 13 Radar Chart Showing the Energy Efficiency Performance of Five Models Across Different Time Steps for 150 Nodes

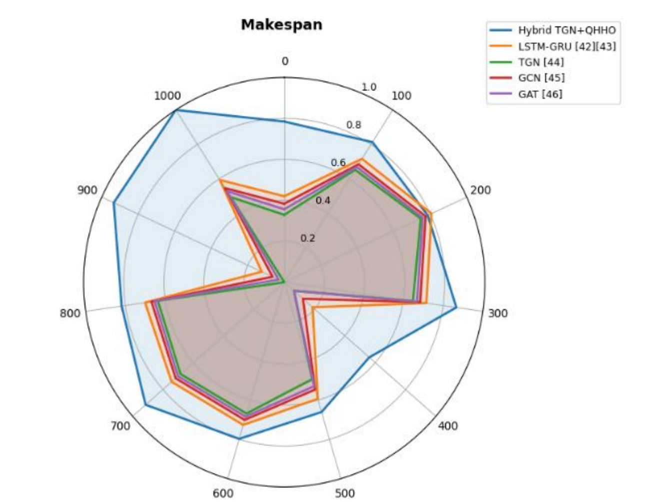

Figure 14 Radar Chart Showing the Make Span Performance of Five Models Across Different Time Steps for 150 Nodes

RESEARCH ARTICLE

Table 5 Performance for All Models with 150-Nodes

Metric	Model	0	100	200	300	400	500	600	700	800	900	1000
Energy Efficiency (EE)	Hybrid TGN+QHHO	0.783	0.81	0.796	0.804	0.806	0.785	0.796	0.802	0.806	0.8	0.798
	LSTM-GRU [42][43]	0.708	0.737	0.729	0.688	0.715	0.729	0.711	0.698	0.72	0.715	0.708
	TGN [44]	0.785	0.731	0.71	0.687	0.694	0.69	0.707	0.696	0.688	0.699	0.696
	GCN [45]	0.713	0.773	0.792	0.819	0.788	0.802	0.783	0.76	0.781	0.777	0.769
	GAT [46]	0.704	0.764	0.782	0.81	0.779	0.793	0.774	0.751	0.773	0.769	0.761
Make span	Hybrid TGN+QHHO	0.782	0.728	0.779	0.627	1.205	1.016	0.758	0.537	0.717	0.499	0.375
	LSTM-GRU [42][43]]	1.471	0.911	0.74	0.912	1.911	1.143	0.892	0.861	0.94	2.031	1.143
	TGN [44]	1.642	1.031	0.851	1.04	2.143	1.331	1.001	0.971	1.061	2.264	1.331
	GCN [45]	1.54	0.971	0.801	0.971	2.031	1.232	0.94	0.911	1.001	2.143	1.232
	GAT [46]	1.591	1.001	0.831	1.001	2.143	1.264	0.971	0.94	1.031	2.206	1.264
Throughput (TH)	Hybrid TGN+QHHO	0.839	0.851	0.82	0.867	0.842	0.833	0.829	0.848	0.859	0.86	0.856
	LSTM-GRU [42][43]	0.782	0.752	0.792	0.784	0.789	0.798	0.796	0.762	0.802	0.792	0.812
	TGN [44]	0.752	0.721	0.762	0.754	0.759	0.768	0.766	0.732	0.772	0.762	0.782
	GCN [45]	0.762	0.732	0.772	0.764	0.769	0.778	0.776	0.742	0.782	0.772	0.792
	GAT [46]	0.772	0.742	0.782	0.774	0.779	0.788	0.786	0.752	0.792	0.782	0.802
Execution Time (ET)	Hybrid TGN+QHHO	8.563	7.137	7.629	7.832	7.232	6.459	7.243	6.288	6.622	5.732	5.68
	LSTM-GRU [42][43]	7.496	8.264	6.496	6.384	8.617	6.684	7.154	6.824	7.496	8.384	6.684
	TGN [44]	8.264	8.617	7.264	7.154	9.264	7.354	7.824	7.496	8.264	9.154	7.354
	GCN [45]	7.824	8.354	6.824	6.684	8.824	7.154	7.496	7.264	7.824	8.684	7.154
	GAT [46]	8.154	8.496	7.154	6.996	9.154	7.354	7.684	7.384	8.154	8.996	7.354
Task Completion Rate (TCR)	Hybrid TGN+QHHO	0.93	0.94	0.94	0.96	0.94	0.96	0.96	0.95	0.96	0.94	0.94
	LSTM-GRU [42][43]	0.91	0.92	0.9	0.91	0.91	0.91	0.92	0.89	0.91	0.87	0.9

RESEARCH ARTICLE

	TGN [44]	0.88	0.89	0.87	0.86	0.84	0.88	0.86	0.85	0.89	0.87	0.86
	GCN [45]	0.85	0.86	0.85	0.84	0.88	0.84	0.85	0.84	0.84	0.8	0.84
	GAT [46]	0.83	0.83	0.81	0.8	0.78	0.81	0.8	0.79	0.81	0.79	0.8

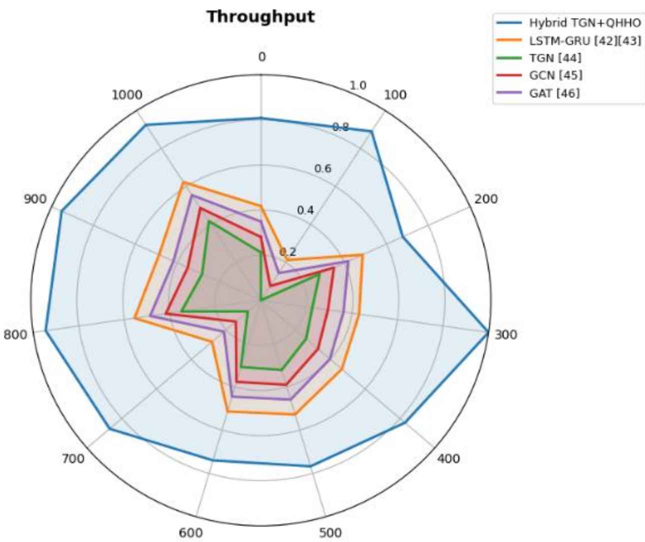

Figure 15 Radar Chart Showing the Throughput Performance of Five Models Across Different Time Steps for 150 Nodes

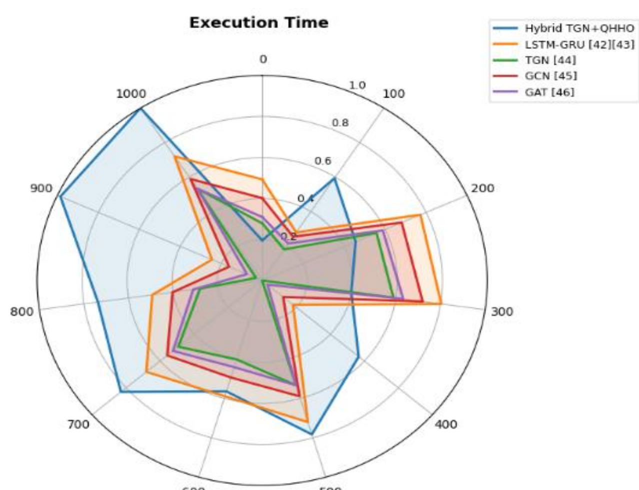

Figure 16 Radar Chart Showing the Execution Time (ET) Performance of Five Models Across Different Time Steps for 150 Nodes

In order to evaluate the scalability of the proposed model, the performance of all five models was further analysed in a 150-node network scenario. The Hybrid TGN+QHHO model performs better and consistently across all operational parameters, as shown by the comparative study of 150 nodes in Table 5, which is supported by the following tabular findings and five radar charts. As illustrated in Figure 14, the TGN model decreases from 0.785 to 0.696 by step 1000, while the Hybrid model maintains energy efficiency values from 0.783 to 0.806, reaching a maximum of 0.806 at both steps 400 and 800, thus greatly outperforming GAT and GCN, which reach maximum values of 0.782 and 0.819, respectively.

The make span comparison in Figure 15 shows that TGN and GAT have almost equal maximum values of 2.264 and 2.206, respectively, reflecting delayed job completion, whereas the Hybrid model primarily keeps the makes pan below 1.0 in most periods and achieves a minimum make span of 0.375 at step 1000.

The lowest throughput of TGN and LSTM-GRU varies from 0.754 to 0.812, while the Hybrid model continuously outstrips the others in throughput, reaching 0.867 at step 300 and maintaining values above 0.82, as illustrated in Figure 16. TGN and GAT lead to higher latencies, peaking at 9.264 and 9.154, respectively, whereas the Hybrid model is very effective in terms of execution time, decreasing from 8.563 at step 0 to 5.68 at step 1000, as shown in Figure 17.

Finally, the job completion rate comparison in Figure 18 demonstrates the efficacy of the Hybrid model, as it consistently remains between 0.93 and 0.96, while GAT and GCN barely reach 0.8 at their lowest.

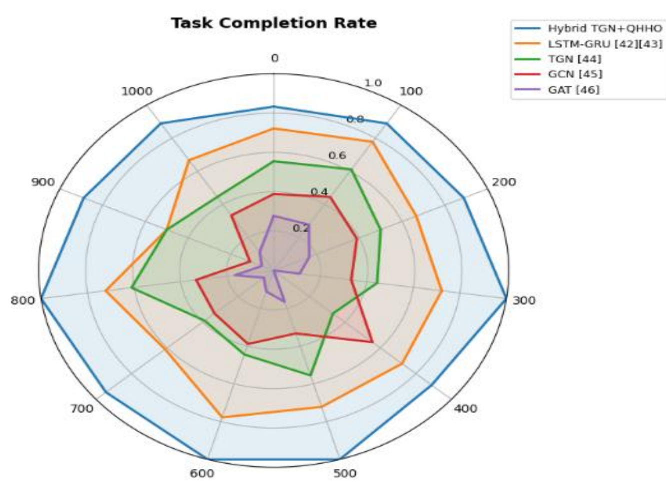

Figure 17 Radar Chart Showing the Task Completion Rate (TCR) Performance of Five Models Across Different Time Steps for 150 Nodes

5.2. Performance Metrics

Table 6 Performance Comparison for Different Models with 50,100 and 150 Nodes

Metric	Nodes	Hybrid TGN+QHHO	LSTM-GRU [42][43]	TGN [44]	GCN [45]	GAT [46]
MSE	50	0.0021	0.0036	0.0042	0.0051	0.0058
	100	0.0027	0.0041	0.0048	0.0056	0.0063
	150	0.0033	0.0047	0.0054	0.0062	0.0069
RM SE	50	0.0458	0.06	0.0648	0.0714	0.0762
	100	0.0519	0.064	0.0693	0.0748	0.0794
	150	0.0574	0.0686	0.0735	0.0787	0.083
MAE	50	0.0375	0.0498	0.0537	0.0582	0.0615
	100	0.0432	0.0541	0.0578	0.0621	0.0657
	150	0.0486	0.0583	0.062	0.0665	0.07
R^2	50	0.931	0.912	0.901	0.889	0.877
	100	0.918	0.899	0.887	0.874	0.862
	150	0.906	0.885	0.872	0.859	0.847

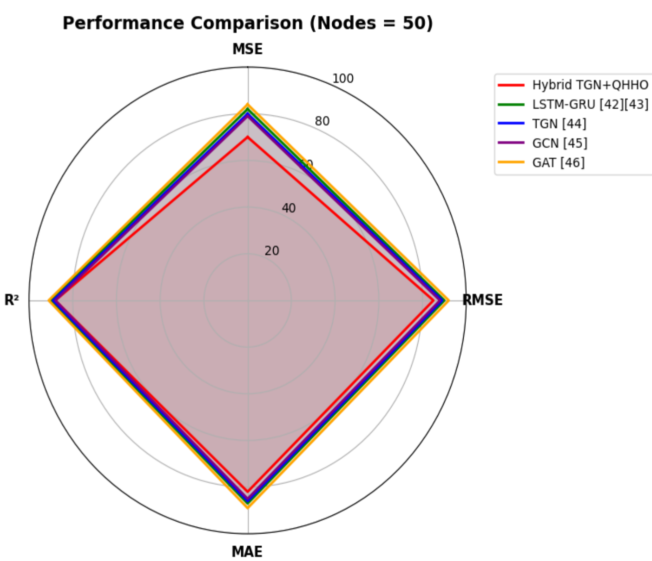

Figure 18 Performance Comparison with 50 Nodes

Table 6 and Figure 19, 20 and 21 demonstrate how the suggested Hybrid TGN-QHHO framework performs better than existing models like LSTM-GRU, TGN, GCN, and GAT in WSN-based IoT networks on many parameters. With much lower Mean Squared Error (MSE) values—0.0021, 0.0027, and 0.0033 for 50, 100, and 150 nodes, respectively—the Hybrid TGN-QHHO performs better than all other models. For instance, LSTM-GRU reports 0.0036, 0.0041, and 0.0047, whereas regular TGN lags farther behind with 0.0042, 0.0048, and 0.0054. Similarly, GCN and GAT have much higher MSE values, indicating less accuracy in tasks involving resource allocation and prediction. The Root Mean Squared Error (RMSE) findings further illustrate the durability of the Hybrid TGN-QHHO architecture, showing values of 0.0458, 0.0519, and 0.0574 with increasing node counts compared to 0.06, 0.064, and 0.0686 for LSTM-GRU. The continuously rising RMSE of the TGN, GCN, and GAT models—which reach up to 0.083 for GAT at 150 nodes—demonstrates the correctness of the recommended hybrid approach. With Mean Absolute Error (MAE) readings of 0.0375, 0.0432, and 0.0486 over a range of network sizes, the Hybrid TGN-QHHO maintains its dominance, while LSTM-GRU records 0.0498, 0.0541, and 0.0583. The higher MAE values of TGN, GCN, and GAT (up to 0.07), which show a greater deviation from real values, further bolster the hybrid model's believability. The R2 (Coefficient of Determination) values, which were 0.931, 0.918, and 0.906 for 50, 100, and 150 nodes, respectively, further illustrate the model's explanatory power. However, TGN, GCN, and GAT show declining performance; at 150 nodes, GAT's score drops to 0.847, while LSTM-GRU scores 0.912, 0.899, and 0.885. These results show that for trust-aware routing, secure communication, and

RESEARCH ARTICLE

efficient resource allocation in Internet of Things networks, the Hybrid TGN-QHHO architecture offers an extremely accurate, dependable, and efficient solution.

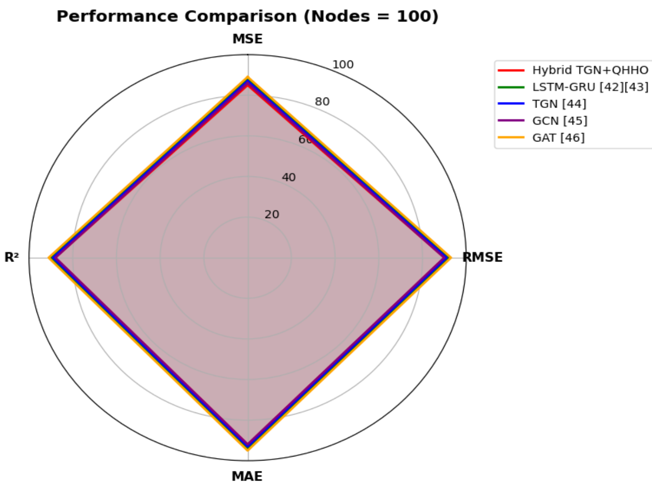

Figure 20 Performance Comparison with100 Nodes

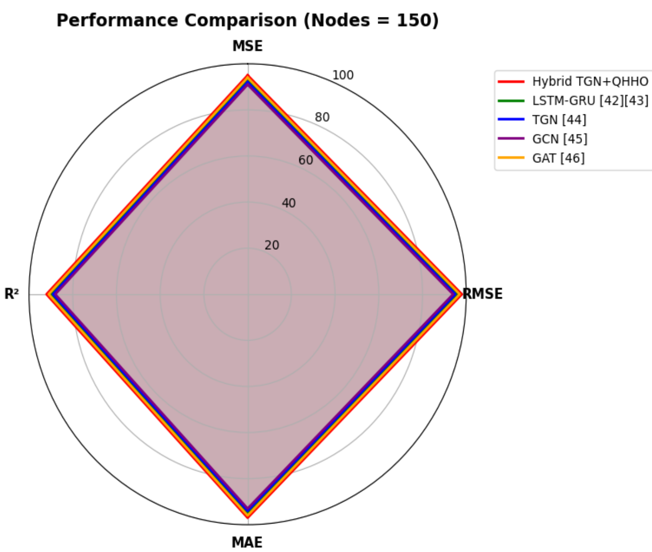

Figure 21 Performance Comparison with 150 Nodes

5.3. Discussion

The experimental evaluation demonstrates that the proposed Hybrid TGN+QHHO model consistently outperforms the baseline approaches (LSTM-GRU, TGN, GCN, and GAT) across wireless sensor network–based IoT scenarios with 50, 100, and 150 nodes. The hybrid framework is more energy-efficient than traditional models, with values ranging from 0.729 to 0.806. This means that it uses node energy more effectively. At the same time, the model greatly shortens the makespan, hitting a low of 0.375 in the 150-node setup. This shows that tasks are being scheduled better and waiting time is shorter when the network is more crowded.

The suggested model also maintains a higher throughput, reaching 0.877 in the 50-node network, while being stable as the number of nodes grows. Also, the time it takes to run is much shorter, going from 4.662 to 9.395 seconds for the 100-node case. This shows that the system converges quicker and makes better routing choices. The task completion rate (TCR) stays high, between 93% and 97%, showing that the suggested framework is strong enough to function in changing IoT settings.

Prediction accuracy and model stability are subjected to further validation through the implementation of error-based metrics. The model Hybrid TGN+QHHO displays the least mean squared error (MSE) value in the range of 0.0021–0.0033 along with a low RMSE (0.0458–0.0574) and MAE (0.0375–0.0486) computed for all node configurations. Moreover, the model secures the highest coefficient of determination (R^2) as its value ranges from 0.906 to 0.931 thereby outperforming all other models and also validating its superior capability of mastering the complex spatio-temporal network dynamics.

These quantitative results collectively indicate that the integration of TGN with Quantum-Inspired HHO, supported by a trust-aware mechanism, substantially enhances energy-efficient resource allocation, reliable routing, and overall network performance. The findings suggest that the proposed hybrid model is well suited for large-scale, dynamic IoT-based wireless sensor networks, where energy conservation, low latency, and high task reliability are critical. Future work may extend this framework to larger network scales and heterogeneous IoT environments to further assess its adaptability and real-world applicability.

6. CONCLUSION

This research shows the Hybrid TGN+QHHO model does a great job handling resources and routing in WSN-based IoT networks. When tested with networks of 50, 100, and 150 nodes, the model typically did better than older methods like LSTM-GRU, TGN, GCN, and GAT. It used energy well (0.729 to 0.806), which helps the network last longer and saves power. The model also finished tasks fast, hitting a minimum of 0.375 for 150 nodes. This means it's good for IoT apps where quick response times are key. It also showed throughput, hitting 0.877, and had short execution times (4.662 to 9.395), proving it can manage data transmission and processing in real-time. Stats confirm the model is accurate and stable, with low error rates (MSE: 0.0021–0.0033, RMSE: 0.0458–0.0574, and MAE: 0.0375–0.0486) and high R^2 scores (0.906–0.931). This means the model can predict things and be reliable in changing IoT setups. Combining Temporal Graph Networks (TGN) and Quantum-Inspired Harris Hawks (QHHO) really helps to improve secure routing, safe communication, and flexible resource handling. The results suggest the Hybrid TGN+QHHO model is a good

RESEARCH ARTICLE

bet for big, power-saving, and fast IoT setups. Future studies should check how well it scales in very crowded networks and how it adapts in real-time in different situations. Also, seeing how well it works with edge computing could make it even better for future IoT systems. This work helps in making smarter, tougher, and more power-conscious wireless sensor networks, which deals with main problems in IoT setups.

REFERENCES

[1] Chen, H., Huang, S., Zhang, D., Xiao, M., Skoglund, M., & Poor, H. V. (2022). Federated learning over wireless IoT networks with optimized communication and resources. IEEE Internet of Things Journal, 9(17), 16592–16605, DOI: 10.1109/JIOT.2022.3169048.

[2] Safara, F., Souri, A., Baker, T., Al Ridhawi, I., & Aloqaily, M. (2020). PriNergy: A priority-based energy-efficient routing method for IoT systems. The Journal of Supercomputing,76(11), 8609-8626. DOI: 10.1007/s11227-020-03145-4.

[3] Gulati, K., Boddu, R. S. K., Kapila, D., Bangare, S. L., Chandnani, N., & Saravanan, G. (2022). A review paper on wireless sensor network techniques in Internet of Things (IoT). Materials Today: Proceedings, 51, 161–165, DOI: 10.1016/j.matpr.2021.04.381.

[4] Wu,H.,Chen,X.,Shi,C.,Xiao,Y.,&Xu,M.(2012). An ACOA-AFSA fusion routing algorithm for underwater wireless sensor network. International Journal of Distributed Sensor Networks, 8(5), 920505.

[5] Ilyas, M., & Mahgoub, I. (2004). Handbook of sensor networks: Compact wireless and wired sensing systems. CRC Press.

[6] Islam, A., & Shin, S. Y. (2023). A digital twin-based drone-assisted secure data aggregation scheme with federated learning in artificial intelligence of things. IEEE Network, 37(2), 278–285.

[7] Akhtar, F., Ahmed, I., Youssef, A. F., Smaili, I. H., Ahmed, M. M. R., & El-Rifaie, A. M. (2025). Advance zonal rectangular low energy adaptive clustering hierarchy algorithm for optimal routing in wireless sensors network. PLoS One, 20(5), e0321938.

[8] Singh, O., Kushwaha, A. S., Sharma, C. R., & Kumar, L. (2025).Improving energy efficiency in scalable WSNs through IoT-driven approach. International Journal of Information Technology, 17(5), 2659–2669. DOI: 10.1007/s41870-025-02471-7.

[9] Ying, X. (2025). A study of energy-efficient routing protocols for industrial IoT WSNs combining GA and ACO algorithms. International Journal of High Speed Electronics and Systems, 2540718. DOI: 10.1142 /S01291 56425407181.

[10] Chander, B. (2021). ML and DL approaches for intelligent wireless sensor networks. In Machine Learning and Deep Learning Techniques in Wireless and Mobile Networking Systems (pp. 11–40). CRC Press.

[11] Q. You & B. Tang. (2021). Efficient task offloading using particle swarm optimization algorithm in edge computing for industrial Internet of Things. Journal of Cloud Computing, 10(1), 41. https://doi.org/10.1186/s13677-021-00247-8.

[12] Rani, S., Ahmed, S. H., & Rastogi, R. (2020). Dynamic clustering approach based on wireless sensor networks genetic algorithm for IoT applications. Wireless Networks, 26(4), 2307–2316. DOI: 10.1007/s11276-019-02083-7.

[13] Wang, B., Zhang, Z., Song, Y., Chen, M., & Liu, D. (2025). nPGSAO: A hybrid particle swarm optimization and genetic algorithm with niching technology for edge server placement. IEEE Internet of Things Journal, 12(12), 19370–19383. DOI: 10.1109/JIOT.2025.3541273.

[14] Karthikeyan, M., Manimegalai, D., & RajaGopal, K. (2024). Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection. Scientific Reports, 14(1), 231. DOI: 10.1038/s41598-023-50554-x.

[15] Aljebreen, M., Alohali, M. A., Saeed, M. K., Mohsen, H., Al Duhayyim, M., Abdelmageed, A. A., Drar, S., & Abdelbagi, S. (2023). Binary chimp optimization algorithm with ML based intrusion detection for secure IoT-assisted wireless sensor networks. Sensors, 23(8), 4073. DOI: 10.3390/s23084073.

[16] Sefati, S. S., Arasteh, B., Halunga, S., & Fratu, O. (2025). A comprehensive survey of cybersecurity techniques based on quality of service (QoS) on the Internet of Things (IoT). Springer (Communications in Computer and Information Science), 28(12). DOI: 10.1007/s10586-025-05449-z.

[17] Chen, T., Zhang, X., You, M., Zheng, G., & Lambotharan, S. (2022). A GNN-based supervised learning framework for resource allocation in wireless IoT networks. IEEE Internet of Things Journal, 9(3), 1712–1724. DOI: 10.1109/JIOT.2021.3091551.

[18] Hu, H., Fan, X., & Wang, C. (2024). Energy efficient clustering and routing protocol based on quantum particle swarm optimization and fuzzy logic for wireless sensor networks. Scientific Reports, 14(1), 18595. DOI: 10.1038/s41598-024-69360-0.

[19] Kumar, A., & Agrawal, K. K. (2023). Energy-efficient resource allocation and routing protocols for IoT-based WSN: A review. In Proceedings of the 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE) (pp. 363–369). DOI: 10.1109/IITCEE57236.2023.10091045.

[20] Ahmed, Q. W., Garg, S., Rai, A., Ramachandran, M., Jhanjhi, N. Z., Masud, M., & Baz, M. (2022). AI-based resource allocation techniques in wireless sensor Internet of Things networks in energy efficiency with data optimization. Electronics, 11(13), 2071. DOI: 10.3390/electronics11132071.

[21] Haseeb, K., Ud Din, I., Almogren, A., & Islam, N. (2020). An energy efficient and secure IoT-based WSN framework: An application to smart agriculture. Sensors, 20(7), 2081. DOI: 10.3390/s20072081.

[22] Fei, H., Jia, D., Zhang, B., Zhang, Y., Luo, T., & Zhou, J. (2024). A novel energy efficient QoS secure routing algorithm for WSNs. Scientific Reports, 14(1), 25969. DOI: 10.1038/s41598-024-77686-y.

[23] Narawade, V., & Kolekar, U. D. (2018). ACSRO: Adaptive cuckoo search based rate adjustment for optimized congestion avoidance and control in wireless sensor networks. Alexandria Engineering Journal, 57(1), 131–145. DOI: 10.1016/j.aej.2016.10.005.

[24] Xu, H., Liu, W., Li, L., Yao, D., & Ma, L. (2024). FSRW: Fuzzy logic-based whale optimization algorithm for trust-aware routing in IoT-based healthcare. Scientific Reports, 14(1), 16640. DOI: 10.1038/s41598-024-66392-4.

[25] Sajedi, S. N., Maadani, M., & Nesari Moghadam, M. (2022). F-LEACH: a fuzzy-based data aggregation scheme for healthcare IoT systems. Journal of Supercomputing, 78(1), 1030–1047. DOI: 10.1007/s11227-021-03890-6.

[26] Kaur, K., & Kang, S. (2024). A rank based secure M-WSN network using machine learning for efficient routing. International Journal of Intelligent Systems and Applications in Engineering, 12(4), 1381–1386.

[27] Antoniou, P., Pitsillides, A., Blackwell, T., Engelbrecht, A., & Michael, L. (2013). Congestion control in wireless sensor networks based on bird flocking behavior. Computer Networks, 57(5), 1167–1191. DOI: 10.1016/j.comnet.2012.12.008.

[28] Kalaikumar, K., & Baburaj, E. (2020). Fuzzy enabled congestion control by cross layer protocol utilizing OABC in WSN: Combining MAC, routing, non-similar clustering and efficient data delivery. Wireless Networks, 26(2), 1085–1103. DOI: 10.1007/s11276-018-1848-3.

[29] Gali, S., & Nidumolu, V. (2022). An intelligent trust sensing scheme with metaheuristic based secure routing protocol for Internet of Things. Cluster Computing, 25(3), 1779–1789. DOI: 10.1007/s10586-021-03473-3.

[30] Alipio, M., & Bures, M. (2025). A cache-aware congestion control mechanism using deep reinforcement learning for wireless sensor networks. Ad Hoc Networks, 166, 103678. DOI: 10.1016/j.adhoc.2024.103678.

[31] Mazloomi, N., Gholipour, M., & Zaretalab, A. (2024). Efficient fuzzy methodology for congestion control in wireless sensor networks. Journal of the Franklin Institute, 361(12), 107014. DOI: 10.1016/j.jfranklin.2024.107014.

[32] Khatami, S. S., Shoeibi, M., Salehi, R., & Kaveh, M. (2025). Energy-efficient and secure double RIS-aided wireless sensor networks: A QoS-aware fuzzy deep reinforcement learning approach. Journal of Sensor and Actuator Networks, 14(1), 18. DOI: 10.3390/jsan14010018.

[33] Udayaprasad, P. K., Shreyas, J., Srinidhi, N. N., Dilip Kumar, S. M., Dayananda, P., Askar, S. S., & Abouhawwash, M. (2024). Energy efficient optimized routing technique with distributed SDN-AI to large scale I-IoT networks. IEEE Access, 12, 2742–2759. DOI: 10.1109/ACCESS.2023.3346679.

[34] Sharma, A., Babbar, H., Rani, S., Sah, D. K., Sehar, S., & Gianini, G. (2023). MHSEER: a meta-heuristic secure and energy-efficient routing protocol for wireless sensor network-based industrial IoT. Energies, 16(10), 4198. DOI: 10.3390/en16104198.

[35] Karunkuzhali, D., Meenakshi, B., & Lingam, K. (2025). A QoS-aware routing approach for Internet of Things-enabled wireless sensor networks in smart cities. Multimedia Tools and Applications, 84(17), 17951–17977. DOI: 10.1007/s11042-024-18125-y.

[36] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems, 97, 849–872. DOI: 10.1016/j.future. 2019.02.028.

[37] Wang, H., Tang, J., & Pan, Q. (2024). MSI-HHO: Multi-strategy improved HHO algorithm for global optimization. Mathematics, 12(3), 0415. DOI: 10.3390/math12030415.

[38] Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., & Bronstein, M. (2020). Temporal Graph Networks for Deep Learning on Dynamic Graphs. arXiv preprint. DOI: 10.48550/arXiv.2006.10637.

[39] Shantal, M., Othman, Z., & Bakar, A. A. (2023). A novel approach for data feature weighting using correlation coefficients and min–max normalization. Symmetry, 15(12), 2185. DOI: 10.3390/sym15122185.

[40] Tahir, M., Abdullah, A., Udzir, N. I., & Kasmiran, K. A. (2025). A novel approach for handling missing data to enhance network intrusion detection system. Cyber Security and Applications, 3, 100063. DOI: 10.1016/j.csa.2024.100063.

[41] Kilaru, N. B., & Cheemakurthi, S. K. M. (2021). Techniques for feature engineering to improve ML model accuracy. NVEO-Natural Volatiles & Essential Oils Journal, 8(1), 194–200.

[42] Mohanty, S. N., Lydia, E. L., Singh, A. K., Al Otaibi, M. M. G., & Shankar, K. (2020). Deep learning with LSTM based distributed data mining model for energy efficient wireless sensor networks. Physical Communication, 40, 101097. DOI: 10.1016/j.phycom.2020.101097.

[43] Ramana, K., Revathi, A., Gayathri, A., Jhaveri, R. H., Narayana, C. V. L., & Kumar, B. N. (2022). WOGRU-IDS — An intelligent intrusion detection system for IoT assisted wireless sensor networks. Computer Communications, 196, 195–206. DOI: 10.1016/j.comcom.2022.10.001.

[44] Zhang, Q., Ye, M., & Deng, X. (2022). A novel anomaly detection method for multimodal WSN data flow via a dynamic graph neural network. Connection Science, 34(1), 1609–1637. DOI: 10.1080/09540091.2022.2078281.

[45] Kiamari, M., & Krishnamachari, B. (2022). Gcnscheduler: Scheduling distributed computing applications using graph convolutional networks. In Proceedings of the 1st International Workshop on Graph Neural Networking (pp. 13–17).

[46] Ahanger, A. S., Khan, S. M., Masoodi, F., & Salau, A. O. (2025). Advanced intrusion detection in Internet of Things using graph attention networks. Scientific Reports, 15(1), 9831. DOI: 10.1038/s41598-025-94624-8.

Authors

Mr. Anuj Kumar currently working as Assistant Professor, Ajay Kumar Garg Engineering College, Ghaziabad and pursuing PhD at Galgotias university. He has more than 18 years teaching experience in Academics and Industry. His areas of interest are Artificial Intelligence, Machine learning, IOT and Data Structure. He has published more than 15 research papers in different journals Springer, Scopus Indexed Journals and UGC Indexed journals and 6 papers in IEEE International Conferences Indexed by Scopus. He is an active member of IEEE. He is also reviewer committee member of different IEEE Conference.

Dr. Krishna Kant Agrawal currently working as Professor CSE & Associate Dean (Research) at Galgotias University, Greater Noida. He has completed his Ph.D. from Indian Institute of Information Technology, Allahabad (IIIT-A), has received the Degree of M. Tech. in Computer Science & Engineering from Birla Institute of Technology, Mesra, Ranchi. Dr. Agrawal is recipient of US Patent, Indian Patent Grant, German Patent Grant, South African Patent Grant and many other Patents filled and Design Patents Granted on his name. Dr. Agrawal is having more than 14 years of Leadership, Research and Teaching experience across multiple institutions and has published more than 62 Research papers in reputed International Refereed Journals/ SCI, Scopus Indexed Journals, Conferences Proceedings & Book Chapters. Dr. Agrawal is also a member of International Professional bodies like IEEE, IET-UK, CSTA-(ACM, USA), and ACM Professional Member.

How to cite this article:

Anuj Kumar, Krishna Kant Agrawal, "Hybrid TGN-QHHO Framework for Optimized Resource Allocation, Trust-Aware Routing, and Secure Communication in WSN-Based IoT Networks", International Journal of Computer Networks and Applications (IJCNA), 13(1), PP: 105-130, 2026, DOI: 10.22247/ijcna/2026/06.