International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

SECURE AND PRIVACY-PRESERVING WSNS IN SMART CITY VIA BLOCKCHAIN-ENABLED FEDERATED LEARNING AND HOMOMORPHIC ENCRYPTION

Gasim Alandjani
Computer Science and Engineering Dept., Yanbu Industrial College
Corresponding Author Email: gasim@rcjy.edu.sa

Abstract

The rapid exponential growth of WSN and IoT devices in smart cities has revolutionized urban services, including traffic management, environmental monitoring, and automation of major infrastructures. Voluminous data generated by heterogeneous sensors, however, present significant challenges to privacy, security, and resource efficiency. Traditional solutions have been plagued by high computation costs, vulnerabilities to malicious nodes, as well as a lack of adequate security for secretive information during model aggregation. In a quest to overcome such vulnerabilities, this work presents a novel framework for secure and private WSNs under blockchain-enabled federated learning as well as homomorphic encryption. In this proposed approach, two novelties are implemented, including Lightweight Local Model Training via Edge-MGTNet, utilizing MobileNet-V3, Tiny-GNN, Micro-TCN, edge pruning, as well as knowledge distillation for lightweight local intelligence extraction, and Encrypted Model Packaging via HashEnc-SparseNet, utilizing sparsified gradient encoding, Paillier homomorphic encryption, as well as hash-based integrity verification (HIV) for encrypting transmission of models with tamper-proof blockchain registration. Robust experiments on merged smart city IoT datasets reveal exemplary performance compared to prior solutions, with Accuracy = 99.22%, Precision = 98.32%, Sensitivity = 96.9%, and Specificity = 96.56%, indicating the effectiveness of such a framework for offering resilient, trustworthy, as well as private intelligence across heterogeneous WSN nodes.

Keywords: *Smart City-WSN, Federated Learning, Blockchain, Edge-MGTNet, HashEnc-SparseNet*

Introduction

As the smart city infrastructure continues to grow at a very high rate, Wireless Sensor Networks (WSNs) have become a vital backbone of the contemporary urban management system, enabling a large number of applications, including environmental monitoring, traffic control, energy optimization, waste management, and public safety [1][2]. These networks are heterogeneous IoT devices, such as sensors, edge nodes, gateways, and actuators, that continuously gather, process, and transmit large amounts of real-time data. The data include essential data like air quality, noise, vehicle density, energy use, and movement of citizens [3][4]. Although these systems allow the effective distribution of resources, predictive maintenance, and improved urban services, they also pose serious security and privacy concerns. Hackers can also interfere with the integrity and confidentiality of sensitive information through cyber threats like unauthorized access to data, eavesdropping, malicious data injection, and distributed denial-of-service attacks [5][6]. Besides, data breaches may interfere with essential services, undermine trust, and create potential threats to the safety of citizens, and strong, privacy-sensitive, and trustworthy WSN communication is essential to the sustainable operation of smart cities [7][8].

The conventional methods of WSN security are based on centralized processing or traditional encryption algorithms, which are usually insufficient in large-scale, distributed smart city applications [9][10]. FSGARH-L, BM-SDA, BS-SCRM, and FL-SCNN-Bi-LSTM are some of the techniques that offer partial solutions to the detection of anomalies, model aggregation, or privacy preservation. Nevertheless, these approaches face a number of challenges: they have a high computational cost, they are susceptible to malicious nodes, they lack privacy guarantees, and they are not flexible to heterogeneous IoT settings [11][12]. The current solutions do not strike a balance between the trade-off between model accuracy, energy efficiency, and data confidentiality, which has led to a dire need to develop more advanced solutions that are scalable and can run efficiently on resource-constrained edge networks [13].

Recent developments in federated learning, edge computing, homomorphic encryption, and blockchain technology are promising directions to overcome these issues [14]. Federated learning can be used to train models jointly on distributed devices without exchanging raw data, and homomorphic encryption can be used to perform computations on encrypted data, maintaining privacy [15]. Blockchain offers a verifiable and unchangeable record of model updates, which is trustworthy and accountable. The combination of these technologies can offer a solid platform of secure, privacy-conscious, and effective WSN functioning in smart city settings [16].

Based on these requirements, this paper presents a new model of safe and privacy-aware WSNs specific to smart cities. The suggested methodology takes advantage of the state-of-the-art edge-based lightweight learning and encrypted model packaging to optimize the local computation, improve privacy, and avoid attacks. Although the framework provides global model aggregation that is reliable and trustworthy in a distributed node, it also deals with the weaknesses of the current methods in accuracy, resilience, and scalability. The solution will be efficient in heterogeneous, resource-constrained IoT networks, and eventually enable safer, smarter, and more sustainable urban digital infrastructures.

The major contributions of this work include:

- To introduce Lightweight Local Model Training via Edge-MGTNet, integrating MobileNet-V3, Tiny-GNN, Micro-TCN, edge pruning, and knowledge distillation for efficient edge intelligence.
- To propose Encrypted Model Packaging via HashEnc-SparseNet, combining sparsified gradient encoding, Paillier homomorphic encryption, and hash-based integrity verification for secure model transmission.

The organization of this work: The literature review using the current methodology is found in Section 2. The proposed methodology, comprising the framework and key elements, is presented in Section 3. The result and comparison analysis are shown in Section 4. Finally, the work comes to a close in the section 5 with the conclusion.

2. Literature Review

To address the shortcomings of conventional machine learning approaches in dealing with imbalanced attack data, Gowdhaman and Dhanapal [17] have introduced an intrusion detection system based on deep neural networks (DNN) to wireless sensor networks. It used cross-correlation process to select the best features, and the refined features were used to build the DNN architecture. The experimental analysis showed that the proposed method was more effective in intrusion detection than classical models like SVM, Decision Tree, and Random Forest.

In 2024, Al-Fuhaidi *et al*. [18] introduced an anomaly-based intrusion detection system on WSNs based on mutual information (MI) to select features and the synthetic minority oversampling technique (SMOTE) to overcome the issue of class imbalance. Traffic analysis was done using different machine learning classifiers such as RF, DT, SVM and KNN. The model was implemented on the NSL-KDD dataset and it significantly increased the classification accuracy with a maximum improvement of 15 percent over the state-of-the-art algorithms.

In 2022, Hussain *et al*. [19] proposed an intrusion detection model that combines a hybrid Whale Optimization Algorithm-Artificial Bee Colony (WOA-ABC) to select features and a Convolutional Neural Network (CNN) to classify intrusion in WSNs. It concentrated on the detection of four key types of attacks, such as DoS, Probe, R2L, and U2R, with the help of the NSL-KDD dataset. The suggested approach performed better than the traditional approaches in terms of execution time, detection rate, accuracy and false alarm rate.

In 2025, Pichumani *et al*. [20] developed a secure data aggregation model called Federated Stochastic Gradient Averaging Ring Homomorphism-based Learning (FSGARH-L) to be used by WSNs. The model integrated federated learning with stochastic gradient averaging and ring homomorphic encryption to obtain privacy-preserving and efficient data aggregation. The results of the simulation proved that the method improved the ratio of packet delivery by 12 percent, decreased the transmission delay by 44 percent and increased the throughput by 37 percent over the current security models.

In 2023, Nouman *et al*. [21] applied the blockchain technology to base stations and cluster heads in WSNs to register nodes safely and identify malicious behavior by the Histogram Gradient Boost (HGB) classifier. The model used Interplanetary File System (IPFS) to store data in a decentralized manner and Verifiable Byzantine Fault Tolerance (VBFT) to achieve consensus. The performance and accuracy of the WSN-DS dataset were better, with VBFT performing 20-30% better than PoW and HGB performing better than other ML classifiers.

In 2025, Jain and Kumar [22] suggested a framework of intrusion detection and data protection in WSNs by using a Blockchain and Machine Learning-based Secure Data Aggregation (BM-SDA). The model combined a fast neural learning (FNL) algorithm to detect attacks in real-time, SMOTE to balance the data, and a feature selection method based on PSO. Authentication and safe storage of data were done through blockchain. The framework was 99.2 percent accurate and saved 35.8 percent of the transaction cost, which was very effective and scalable.

In 2024, Xiao *et al*. [23] designed a WSNs Secure Clustering Routing Method (BS-SCRM) based on blockchain technology and swarm intelligence to enhance network security and efficiency. The model reduced man-in-the-middle attacks through the use of an elite strategy-enhanced Whale Optimization Algorithm (WOA) to select cluster heads and blockchain to verify data integrity. Tests on simulations verified a 24-73 percent improvement in network lifetime and a 24-73 percent improvement in energy efficiency compared to current clustering methods.

In 2024, Jeyakumar *et al*. [24] proposed a Federated Learning-based intrusion detection system, which combines Stacked CNN, Bidirectional LSTM, and the African Vulture Optimization Algorithm (AVOA). The model, which is referred to as SCNN-Bi-LSTM-AVOA-FL, was trained on AVOA and performed well in attack detection without compromising data privacy. The assessments on the WSN-DS, CIC-IDS-2017, and WSN-BFSF datasets demonstrated a high level of accuracy and recall, which confirms its ability to improve the performance of IDS in federated settings.

In 2025, Devi *et al*. [25] suggested a Federated Learning-based Lightweight Intrusion Detection System (FL-LIDS) to be used in smart city applications by resource-constrained WSNs. Lightweight deep learning models such as hybrid CNN-LSTM models were optimized and trained jointly without sharing raw data, which guaranteed privacy. The FL-LIDS was tested on the TON-IoT dataset and demonstrated high detection and low latency of DDoS attacks, which provides an efficient and scalable cybersecurity solution to smart WSN environments.

In 2024, Bukhari *et al*. [26] introduced a Federated Learning-based intrusion detection model that uses Stacked Convolutional Neural Networks (SCNN) and Bidirectional Long Short-Term Memory (Bi-LSTM). The framework allowed the joint training of WSN nodes and maintained data privacy. The FL-SCNN-Bi-LSTM was tested on the WSN-DS and CIC-IDS-2017 datasets and obtained approximately 99.9% accuracy, precision, and recall, significantly lowering the false alarms and showing better results in detecting complex and previously unseen cyberattacks. Table 1 is the comparative analysis of the related works.

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 1: Comparison of Existing Techniques

Authors, Year	Framework / Technique	Objective	Significance	Limitations
Gowdhaman and Dhanapal, 2022	Deep Neural Network (DNN)-based Intrusion Detection using Cross-Correlation Feature Selection	• To enhance intrusion detection accuracy in WSNs by leveraging deep learning.	• DNN outperformed ML models (SVM, DT, RF). • Cross-correlation effectively reduced irrelevant features.	• No privacy-preserving mechanism. • Not evaluated under real-time constraints.
Al-Fuhaidi *et al*, 2024	MI + SMOTE-based ML Intrusion Detection (RF, DT, SVM, KNN)	• To design an anomaly-based IDS using feature selection (MI) and class balancing (SMOTE).	• Achieved up to 15% improvement in accuracy. • Effective handling of class imbalance.	• ML models lack adaptability to dynamic attacks. • No encryption or privacy layer.
Hussain *et al*, 2022	Hybrid WOA–ABC Feature Selection + CNN Classifier	• To detect four major WSN attack types (DoS, R2L, U2R, Probe).	• Improved detection accuracy and minimized false alarms. • Reduced execution time by 76.54%.	• High computational cost for large-scale WSNs. • Limited explainability of CNN decisions.
Pichumani *et al*, 2025	Federated Stochastic Gradient Averaging Ring Homomorphism-based Learning (FSGARH-L)	• To develop a secure, privacy-preserving data aggregation framework for heterogeneous WSN nodes.	• Enabled privacy-preserving model training. • Improved packet delivery by 12%, reduced delay by 44%.	• HE introduces computation latency. • Energy consumption increases for encryption.
Nouman *et al*, 2023	Blockchain + Histogram Gradient Boost (HGB) Classifier + VBFT Consensus + IPFS Storage	• To detect and revoke malicious WSN nodes using ML and blockchain.	• VBFT outperformed PoW by 20–30%. • HGB improved accuracy by up to 16% over others.	• High storage and communication cost in blockchain. • Limited scalability with increasing node count.
Jain and Kumar, 2025	Blockchain–Machine Learning Secure Data Aggregation (BM-SDA) with PSO, SMOTE, and FNL	• To detect intrusions and secure data aggregation via ML and blockchain integration.	• Achieved 99.2% accuracy and 35.8% lower transaction cost. • PSO optimized feature selection. • Blockchain provided authentication and traceability.	• FNL model lacks deep feature representation. • Blockchain integration adds latency.
Xiao *et al*, 2024	Blockchain-based Secure Clustering Routing Method (BS-SCRM) using Elite WOA	• To enhance secure routing and cluster head selection in WSNs. • To resist man-in-the-middle and data tampering attacks.	• Improved network lifetime by up to 73%. • Blockchain ensured data integrity.	• Energy overhead due to blockchain operations. • Implementation complexity in large-scale WSNs.
Jeyakumar *et al*, 2024	Federated Learning + SCNN–BiLSTM optimized via African Vulture Optimization Algorithm (AVOA)	• To enhance IDS accuracy while preserving privacy in distributed WSNs.	• FL maintained data privacy. • Robust detection of known and unknown attacks.	• Communication cost in FL remains high. • AVOA lead to slow convergence.
Devi *et al*, 2025	Federated Learning-based Lightweight IDS (FL-LIDS) using Hybrid CNN–LSTM	• To develop a lightweight, privacy-preserving intrusion detection framework for resource-limited WSNs.	• Reduced latency and resource usage. • Achieved high detection accuracy on TON-IoT dataset. • Scalable and privacy-aware architecture.	• Does not secure model updates (no encryption). • Vulnerable to model poisoning in FL.
Bukhari *et al*, 2024	Federated Learning-based SCNN–BiLSTM Model	• To detect sophisticated cyber threats in WSNs while preserving data	• Achieved 99.9% accuracy and recall. • Reduced false positives/negatives.	• Lacks blockchain or encryption layer for integrity.

Received: August 05, 2025

Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

		privacy through federated learning.	• Strong privacy preservation via FL.	• No energy-optimization strategy for WSN nodes.

3. Proposed Methodology

The proposed methodology integrates advanced techniques to enable secure, privacy-preserving, and efficient federated learning for smart city WSN environments. To begin with, a combination of SmartCity Cybersecurity IoT Dataset and Location Intelligence for Cybersecurity 2025 Dataset constitutes a single input for data acquisition. Raw data is subjected to Kalman Filter-based local staging coupled with time smoothing, arriving at temporally coherent, noise-suppressed inputs. Lightweight edge training of local models via Edge-MGTNet derives multi-modal features via MobileNet-V3, Tiny-GNN, and Micro-TCN, with subsequent edge pruning and knowledge distillation for local model optimization. Then local models are encrypted with encrypted model packaging via HashEnc-SparseNet, with sparsified gradient encoding, Paillier homomorphic encryption, as well as hash-based integrity verification. Blockchain registration and validation via Hyperledger Fabric and zk-SNARKs guarantee authenticity. Federated aggregation under encryption exploits secure aggregation, federated trimmed mean, and differential privacy (DP-SGD) for globally coherent model production. Lastly, trust and attack mitigation utilizes effective aggregation, Byzantine filtering, anomaly-based update rejection, and EWMA scoring, with resultant global model redeployment via secure multicast, guaranteeing confidentiality-preserving as well as trustworthy dissemination of intelligence across edge nodes in smart cities. Figure 1 represents the architecture of the proposed approach.

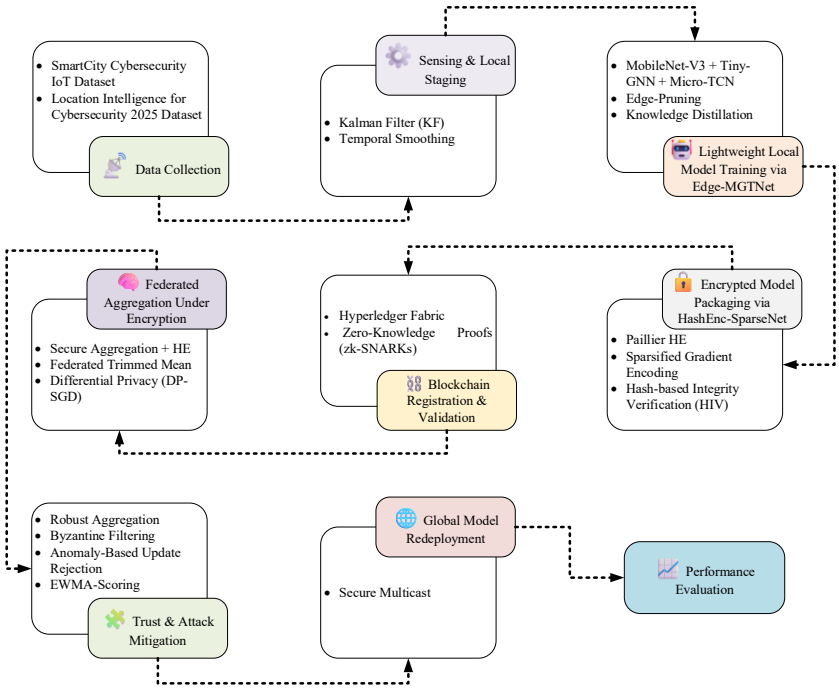

Figure 1: Overall Architecture of the Proposed Framework

3.1 Data Collection

This work uses two open-source datasets: the SmartCity Cybersecurity IoT Dataset (Zara2099, Kaggle) and the Location Intelligence for Cybersecurity 2025 Dataset (Wisam1985, Kaggle). The first captures communication behavior, energy usage, and threat indicators in smart city IoT and edge networks, while the second links cyberattack with geographic and environmental factors.

Both datasets are fused at the feature and temporal levels to form a unified smart city cybersecurity dataset for subsequent stages. The fusion is expressed as Eq. (1):

$$D_{fused} = \Phi(D_{IoT}, D_{Geo}) = \alpha f \cdot (D_{IoT} \oplus D_{Geo}) \quad (1)$$

Where, D_{IoT} and D_{Geo} denote the respective datasets, $\oplus$ indicates aligned feature concatenation, and $\Phi(\cdot)$ ensures temporal–spatial coherence.

3.2 Sensing and Local Staging

The fused dataset D_{fused} obtained above section serves as the input to the sensing and local staging phase. This stage enhances the temporal consistency, signal reliability, and data stability of smart city IoT streams before model training. It operates sequentially using two key processes: Kalman filtering followed by temporal smoothing.

3.2.1 Kalman Filtering

Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The Kalman Filter (KF) is applied first to eliminate sensor noise and prediction uncertainty in the multi-sensor data. It operates as a recursive estimator that predicts the next system state and updates it using observed measurements. The prediction and correction stages are represented as Eq. (2) to Eq. (5):

$$\hat{x}_{t|t-1} = A_t \hat{x}_{t-1|t-1} + B_t u_t \tag{2}$$

$$P_{t|t-1} = A_t P_{t-1|t-1} A_t^T + Q_t \tag{3}$$

$$K_t = P_{t|t-1} H_t^T \left(H_t P_{t|t-1} H_t^T + R_t \right)^{-1} \tag{4}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \left(z_t - H_t \hat{x}_{t|t-1} \right) \tag{5}$$

Where, $\hat{x}_{t|t-1}$ and $\hat{x}_{t|t}$ denote the predicted and updated state vectors, A_t and B_t represent transition and control matrices, Q_t and R_t provides process and observation noise covariances, K_t denotes the Kalman gain, and z_t signifies the observed measurement. This filtering process yields a denoised, dynamically stable signal set D_{KF}, effectively mitigating environmental and transmission-induced variations in smart city sensors.

3.2.2 Temporal Smoothing

The output D_{KF} from the Kalman Filter is then refined through temporal smoothing, which ensures continuity in rapidly changing sensor readings and reduces abrupt fluctuations due to transient disturbances. The smoothed data D_{TS} is computed as Eq. (6):

$$D_{TS}(t) = \beta i \cdot D_{KF}(t) + (1 - \beta i) \cdot D_{TS}(t-1) \tag{6}$$

Where, $\beta i \in [0,1]$ controls the smoothing intensity. Higher βi emphasizes recent values, while lower βi preserves long-term trends. This process generates temporally coherent and noise-suppressed sensor data, improving the stability of downstream model training. The final output of this phase D_{TS}, is forwarded to Lightweight Local Model Training for feature learning and local intelligence extraction.

3.3 Lightweight Local Model Training via Edge-MGTNet (Edge-efficient Mobile–GNN–TCN Network)

This phase introduces Edge-MGTNet, a novel edge-efficient hybrid model that integrates MobileNet-V3, Tiny-GNN, and Micro-TCN to enable intelligent local learning on resource-constrained wireless sensor and IoT nodes. The temporally smoothed data D_{TS} from Section 3.2 serves as input for this stage. Edge-MGTNet performs localized feature extraction, model compression, and knowledge transfer to generate lightweight yet expressive representations of sensor behavior. This design ensures low-latency, energy-efficient, and privacy-preserving model training at the network edge—forming the foundation for secure federated aggregation in subsequent stages. The general structure of the proposed Edge-MGTNet is depicted in Figure 2.

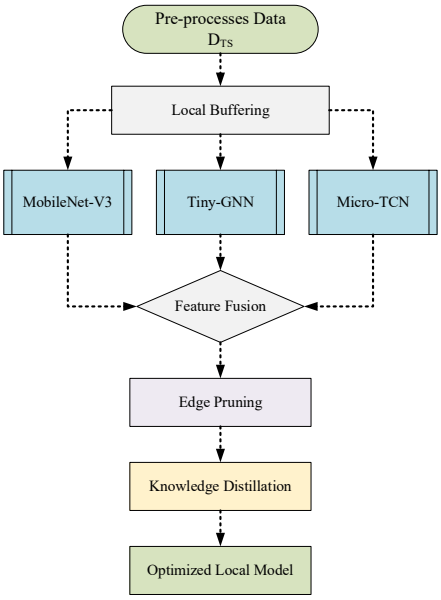

Figure 2: General Structure of the Proposed Edge-MGTNet

3.3.1 Local Data Reception and Mini-Batch Buffering

In this stage, each edge node in the smart city wireless sensor network receives its respective segment of the preprocessed dataset D_{TS} generated from the sensing and local staging phase. This partitioning ensures localized data ownership, thereby maintaining data privacy and reducing communication overhead. The dataset assigned to the i-th edge node is denoted as Eq. (7):

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$D_{TS}^{(i)} = \{(x_t^{(i)}, y_t^{(i)})\}_{t=1}^{T_i} \quad (7)$$

Where, $x_t^{(i)}$ represents the temporally smoothed sensor features, $y_t^{(i)}$denotes the corresponding labels or event indicators, and T_i indicates the number of time instances available for node i. Each node locally buffers this dataset into mini-batches to enable efficient gradient computation and energy-aware training. The partitioning function can be expressed as Eq. (8):

$$B^{(i)} = \{D_{TS}^{(i,1)}, D_{TS}^{(i,2)}, \dots, D_{TS}^{(i,K_i)}\} \quad (8)$$

Where, K_i represents the number of mini-batches for node i, dynamically chosen based on its memory and computational capacity.

This localized batching mechanism allows edge devices to perform incremental learning without overloading their limited resources. Moreover, by keeping all data $D_{TS}^{(i)}$ within the device, the approach inherently supports privacy preservation and data residency compliance, which are essential in smart city environments. The mini-batched dataset $B^{(i)}$serves as the direct input for hybrid lightweight feature learning.

3.3.2 Hybrid Lightweight Feature Learning (MobileNet-V3, Tiny-GNN, Micro-TCN)

In this phase, the mini-batched dataset $B^{(i)}$ serves as input to a parallel three-stage feature extraction pipeline, designed to capture spatial, topological, and temporal characteristics of the sensor/IoT data. The hybrid structure allows each edge node to efficiently encode multi-faceted information while maintaining low computational overhead.

1. MobileNet-V3: Spatial/Contextual Feature Encoding

MobileNet-V3 employs depthwise separable convolutions to extract local spatial patterns from each mini-batch $D_{TS}^{(i,k)}$. For a given input feature tensor $X \in \mathbb{R}^{H\times W\times C}$, the depthwise separable convolution operates as Eq. (9):

$$X' = PointwiseConv\big(DepthwiseConv(X)\big) \quad (9)$$

Here, the $DepthwiseConv$ filters each channel independently, while the $PointwiseConv$ fuses channels to produce compact, informative spatial embeddings F_s. MobileNet-V3 efficiently encodes contextual relations across sensors or spatially distributed IoT devices, reducing parameter count and computational cost which is defined as $F_s^{(i,k)} = f_{MobileNet-V3}\,(D_{TS}^{(i,k)})$.

2. Tiny-GNN: Topological Relationship Modeling

Tiny-GNN captures inter-node relationships by modeling the edge connections between neighboring sensors or IoT devices. Let the node feature matrix for mini-batch k be $Hm^{(0)} \in \mathbb{R}^{N\times d}$, and the adjacency matrix representing local sensor connectivity be $\mathcal{A}m \in \mathbb{R}^{N\times N}$. The graph convolution update is expressed as Eq. (10):

$$Hm^{(l+1)} = \sigma_{af}\big(\widetilde{Dm}^{-1/2}, \widetilde{\mathcal{A}m}\widetilde{Dm}^{-1/2} Hm^{(l)} Wm^{(l)}\big) \quad (10)$$

Where, $\widetilde{\mathcal{A}m} = \mathcal{A}m + I$ signifies the adjacency matrix with self-loops, $\widetilde{Dm}$ provides the degree matrix, $Wm^{(l)}$ refers the trainable weight matrix, and σ_{af} signifies the activation function. $Tiny\_GNN$ outputs topologically aware embeddings F_t that encode the relational dynamics of neighboring nodes, which is defined as Eq. (11):

$$F_t^{(i,k)} = f_{Tiny\_GNN}(Hm^{(0)}, \mathcal{A}m) \quad (11)$$

3. Micro-TCN: Temporal Dependency Encoding

Micro-TCN is a lightweight Temporal Convolutional Network that models sequential dependencies in sensor readings within each mini-batch. Given a temporally ordered sequence $x_{1:T}^{(i,k)}$, the Micro-TCN applies 1D causal convolutions with dilation to capture temporal patterns as Eq. (12):

$$F_\tau^{(i,k)} = ReLU(W_c * x_{1:T}^{(i,k)} + b_c) \quad (12)$$

Where, W_c and b_c are convolutional weights and biases, and $*$denotes the causal convolution operator. Micro-TCN preserves temporal causality and efficiently encodes short- and long-term dependencies with minimal parameters.

The three modules operate in parallel, producing complementary feature representations for each mini-batch as $\{F_s^{(i,k)}, F_t^{(i,k)}, F_\tau^{(i,k)}\}$. These outputs are forwarded to the feature fusion stage to generate a unified, low-dimensional embedding for lightweight local model training. This design enables each edge node to efficiently capture spatial, topological, and temporal information while remaining computationally feasible for WSN in smart city deployments.

3.3.3 Multi-Stream Feature Fusion

After obtaining parallel feature embeddings from MobileNet-V3 ($F_s^{(i,k)}$), Tiny-GNN ($F_t^{(i,k)}$), and Micro-TCN ($F_\tau^{(i,k)}$) for each mini-batch k at node i, a multi-stream feature fusion mechanism is applied to combine these heterogeneous representations into a unified, robust embedding suitable for lightweight local model training.

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Attention-Based Fusion

To effectively integrate spatial, topological, and temporal information, an attention-weighted fusion layer $\mathcal{AW}(\cdot)$is employed. The fused feature representation for node i is defined as Eq. (13):

$$F_{fusion}^{(i)} = \mathcal{AW}\left(F_s^{(i,k)}, F_t^{(i,k)}, F_\tau^{(i,k)}\right) \tag{13}$$

The attention mechanism computes importance weights for each feature stream, enabling the model to emphasize the most informative modality while suppressing less relevant signals, which is defined as Eq. (14), (15), and (16):

$$\alpha_s = \frac{\exp\left(W_s F_s^{(i,k)}\right)}{\exp\left(W_s F_s^{(i,k)}\right)+\exp\left(W_t F_t^{(i,k)}\right)+\exp\left(W_\tau F_\tau^{(i,k)}\right)}, \tag{14}$$

$$\alpha_t = \frac{\exp\left(W_t F_t^{(i,k)}\right)}{\exp\left(W_s F_s^{(i,k)}\right)+\exp\left(W_t F_t^{(i,k)}\right)+\exp\left(W_\tau F_\tau^{(i,k)}\right)}, \tag{15}$$

$$\alpha_\tau = \frac{\exp\left(W_\tau F_\tau^{(i,k)}\right)}{\exp\left(W_s F_s^{(i,k)}\right)+\exp\left(W_t F_t^{(i,k)}\right)+\exp\left(W_\tau F_\tau^{(i,k)}\right)} \tag{16}$$

Here, W_s, W_t, W_τ represents learnable parameters of the attention layer. The fused feature vector is then computed as Eq. (17):

$$F_{fusion}^{(i)} = \alpha_s F_s^{(i,k)} + \alpha_t F_t^{(i,k)} + \alpha_\tau F_\tau^{(i,k)} \tag{17}$$

This fused feature space has the benefits of enhanced robustness, improved data diversity and low-dimensional embedding. The resulting fused feature $F_{fusion}^{(i)}$ serves as the input for edge pruning and knowledge distillation in the subsequent compression and intelligence transfer phases.

3.3.4 Edge Pruning and Model Compression

Following multi-stream feature fusion, the fused feature representation $F_{fusion}^{(i)}$ is subjected to edge pruning and model compression to optimize the network structure for resource-constrained WSN edge devices. This step reduces computational complexity, memory usage, and energy consumption while preserving essential predictive performance.

Structural Optimization

The local model parameters at node i are represented as $\Theta^{(i)}$. To remove redundant or low-importance neurons and graph edges, a pruning function $\mathcal{PU}(\cdot)$ is applied as Eq. (18):

$$\Theta_{pruned}^{(i)} = \mathcal{PU}(\Theta^{(i)}, \lambda_{pt}) \tag{18}$$

Here, λ_{pt} denotes a pruning threshold, which filters out parameters with magnitude or contribution below this value. Specifically, parameters $\theta_j \in \Theta^{(i)}$ are removed if $|\ \theta_j\ | < \lambda_{pt}$.

The pruning operation effectively eliminates unnecessary computations in convolutional filters (MobileNet-V3), graph edges (Tiny-GNN), and temporal convolution kernels (Micro-TCN) while retaining critical pathways for accurate prediction.

The pruned model parameters $\Theta_{pruned}^{(i)}$ together with $F_{fusion}^{(i)}$ are then forwarded to the knowledge distillation phase for transferring intelligence from global teacher models to the local lightweight models.

3.3.5 Knowledge Distillation and Local Intelligence Output

After pruning, the optimized fused features $F_{fusion}^{(i)}$ and pruned model parameters $\Theta_{pruned}^{(i)}$ are forwarded to the knowledge distillation phase. This step transfers knowledge from a global teacher model to the local student model, ensuring that each edge node learns a lightweight yet high-performing approximation of the centralized intelligence.

Knowledge Distillation Process

The distillation loss L_{KD} combines the standard cross-entropy loss with a softened Kullback–Leibler (KL) divergence between teacher and student outputs, which is defined as Eq. (19):

$$L_{KD} = (1-\varrho)L_{CE}(y, \hat{y}^S) + \varrho TP^2 KL\left(\sigma_T(\hat{y}^T/TP), \sigma_S(\hat{y}^S/TP)\right) \tag{19}$$

Where, $\hat{y}^T$and $\hat{y}^S$ denotes the outputs (logits) of the teacher and student models, respectively. TP signifies the temperature parameter that softens the logits to emphasize relative probabilities. $\varrho \in [0,1]$ balances the contribution of the cross-entropy and distillation loss terms, and $\sigma_T(\cdot)$ and $\sigma_S(\cdot)$ are the softmax operations applied to the scaled teacher and student logits.

This combination allows the student model to not only learn from the true labels y but also mimic the dark knowledge encoded in the teacher's output distribution, improving generalization across unseen data.

The output of this stage is the optimized local intelligence vector, which is expressed as Eq. (20):

$$F_{local}^{*(i)} = StudentModel\left(F_{fusion}^{(i)}, \Theta_{pruned}^{(i)}\right) \tag{20}$$

The resulting output represents a compact, high-fidelity feature embedding that serves as the input for encrypted model packaging. It ensures consistency across distributed edge models while operating under resource constraints, thereby enabling robust and efficient federated aggregation. This completes the Edge-MGTNet local training pipeline, producing lightweight, energy-efficient, and intelligence-rich node-level representations for the smart city WSN.

3.4 Encrypted Model Packaging via HashEnc-SparseNet

To ensure confidentiality, integrity, and efficiency in federated communication within smart city WSNs, the proposed HashEnc-SparseNet framework packages local model updates through a three-stage sequential process. First, Sparsified Gradient Encoding compresses redundant parameters to minimize communication cost. Next, Paillier Homomorphic Encryption (HE) secures the encoded gradients against interception or inference. Finally, Hash-Based Integrity Verification (HIV) guarantees tamper-proof transmission by generating immutable hash signatures. This pipeline ensures lightweight, secure, and verifiable model exchange across distributed edge nodes.

3.4.1 Sparsified Gradient Encoding for Model Compression

In this stage, each edge node compresses its learned model updates to minimize communication and encryption overhead. The optimization process begins with the gradient vector derived from the local intelligence representation $F_{local}^{*(i)}$, defined as Eq. (21):

$$g^{(i)} = \nabla_{\Theta} L_{\mathrm{KD}}^{(i)},$$

Where, $g^{(i)}$ represents the gradient of the local model parameters $\Theta^{(i)}$ with respect to the knowledge distillation loss $L_{\mathrm{KD}}^{(i)}$.

To enhance efficiency, a sparsification operator $\mathcal{SO}(\cdot, \kappa)$is applied to retain only the top-κ significant gradients or those exceeding a predefined threshold. The sparsified gradient representation is expressed as Eq. (22):

$$g_{sparse}^{(i)} = \mathcal{SO}(g^{(i)}, \kappa) \tag{22}$$

Where, $\kappa \in (0,1]$ controls the sparsity ratio—smaller values yield higher compression but slightly reduce precision. This selective retention process effectively filters out redundant or low-impact parameters, substantially reducing transmission bandwidth and encryption workload. Despite compression, the essential structural fidelity of the model gradients is preserved, ensuring that critical learning information remains intact for subsequent secure encryption and aggregation. This step is crucial for maintaining computational scalability and energy efficiency across resource-constrained WSN nodes in smart city environments.

3.4.2 Homomorphic Encryption via Paillier Cryptosystem

Following gradient sparsification, the compressed vector $g_{sparse}^{(i)}$ is encrypted to ensure privacy-preserving communication and aggregation across the smart city network. The Paillier cryptosystem, an additive homomorphic encryption (HE) scheme, is employed for this purpose due to its efficiency and mathematical support for secure aggregation without revealing raw data.

The Paillier system defines a public key pair (n, g) and a private key ζ, where $n = pq$ denotes the product of two large primes p and q. Each component of the sparsified gradient vector $g_{sparse}^{(i,j)}$ is encrypted as Eq. (23):

$$E\left(g_{sparse}^{(i,j)}\right) = g^{g_{sparse}^{(i,j)}} r^{n} mod\ n^2 \tag{23}$$

Where, r denotes a random integer in $\mathbb{Z}_n^*$ ensuring semantic security. The resulting ciphertext $E\left(g_{sparse}^{(i,j)}\right)$ conceals both the magnitude and sign of the encoded gradient values. A key advantage of the Paillier scheme lies in its additive homomorphism, which allows secure aggregation of encrypted gradients without decryption as Eq. (24):

$$E\left(g_{sparse}^{(i)}\right) \otimes E\left(g_{sparse}^{(k)}\right) = E\left(g_{sparse}^{(i)} + g_{sparse}^{(k)}\right) \tag{24}$$

This property enables the federated server to perform model updates directly on ciphertexts, preserving data confidentiality throughout the aggregation phase.

Through this encryption step, the sparsified model gradients become mathematically protected, ensuring that even if transmission channels or intermediary nodes are compromised, no meaningful information about the model or local data can be inferred. The encrypted output $E\left(g_{sparse}^{(i)}\right)$ is then passed to the next stage for Hash-Based Integrity Verification, ensuring the immutability and authenticity of model updates before blockchain registration.

3.4.3 Hash-Based Integrity Verification and Secure Packaging

Once the sparsified gradients have securely encrypted through the Paillier cryptosystem, the next step ensures their authenticity, immutability, and resistance to tampering before blockchain registration. To achieve this, a Hash-Based Integrity Verification (HIV) mechanism is applied to the encrypted gradients.

Let $E\left(g_{sparse}^{(i)}\right)$ represent the encrypted gradient vector output from the previous stage. A cryptographic hash function $\hbar(\cdot)$, such as SHA-256, is applied to generate a unique integrity signature, which is defined as Eq. (25):

$$\mathcal{H}^{(i)} = \hbar\left(E\left(g_{sparse}^{(i)}\right)\right) \tag{25}$$

Here, $\mathcal{H}^{(i)}$ defined as a tamper-evident fingerprint for the encrypted model. Even a minor alteration in the ciphertext results in a completely different hash value, ensuring that any unauthorized modification can be instantly detected. This hash value $\mathcal{H}^{(i)}$ is then attached to its corresponding encrypted model to form the final secure encrypted package, which is defined as Eq. (26):

$$\mathfrak{P}^{(i)} = \left\{E\left(g_{sparse}^{(i)}\right), \mathcal{H}^{(i)}\right\} \tag{26}$$

The combined package $\mathfrak{P}^{(i)}$ is digitally signed and prepared for registration on the blockchain network. Once recorded, the blockchain ledger guarantees immutable storage and traceable verification, preventing any replay attacks, tampering, or injection of unauthorized updates.

This final step of the HashEnc-SparseNet process ensures a triple layer of protection—efficiency from sparsification, confidentiality from homomorphic encryption, and integrity from hash verification. The verified and sealed package $\mathfrak{P}^{(i)}$ is then securely forwarded to Blockchain Registration & Validation for decentralized trust establishment and cross-node validation within the smart city WSN ecosystem.

3.5 Blockchain Registration & Validation

In this stage, the securely packaged local model $\mathfrak{P}^{(i)} = \left\{E\left(g_{sparse}^{(i)}\right), \mathcal{H}^{(i)}\right\}$ undergoes decentralized registration and cryptographic validation before participating in federated aggregation. This phase ensures trust, traceability, and authenticity of model updates within the smart city's WSN ecosystem through Hyperledger Fabric and Zero-Knowledge Proofs (zk-SNARKs). The workflow proceeds sequentially, where blockchain registration precedes validation.

3.5.1 Decentralized Registration via Hyperledger Fabric

Each edge node transmits its secure package $\mathfrak{P}^{(i)}$ to a Hyperledger Fabric blockchain network, where it is logged as a transaction within a permissioned ledger. Hyperledger Fabric offers modular consensus and fine-grained access control, suitable for smart city infrastructures that require accountability without public exposure. Upon submission, the package is recorded as Eq. (27):

$$Tr^{(i)} = \text{Reg}(\mathfrak{P}^{(i)}, ts_i, \text{ID}_i) \tag{27}$$

Where, ts_i denotes the timestamp and ID_i represents the unique node identity. Each transaction $Tr^{(i)}$ is hashed and linked to the preceding block to ensure immutability as Eq. (28):

$$\mathcal{B}_k = Hash\left(\mathcal{B}_{k-1} \parallel Tr^{(i)}\right) \tag{28}$$

This registration guarantees that every model update is verifiable, time-stamped, and permanently anchored within the blockchain ledger. The resulting ledger entries are tamper-proof, providing non-repudiation and traceability of all contributing edge nodes.

3.5.2 Cryptographic Validation via Zero-Knowledge Proofs (zk-SNARKs)

After successful registration, each encrypted model undergoes validation through Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs). This ensures that nodes can prove the legitimacy of their model updates without revealing their underlying encrypted data. Let $\pi^{(i)}$ denote the proof generated by node i, which satisfies as Eq. (29):

$$Verify\left(vk, \pi^{(i)}, E\left(g_{sparse}^{(i)}\right)\right) = True \tag{29}$$

Where, vk represents the verification key. The zk-SNARK mechanism verifies that the encrypted model $E\left(g_{sparse}^{(i)}\right)$ corresponds to a correctly trained update, ensuring both integrity and privacy. This verification allows the blockchain network to accept only those updates that are mathematically consistent and free from adversarial tampering or poisoning attempts. Once the validation succeeds, the blockchain maintains a verified, immutable mapping, which is defined as Eq. (30):

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$Ledger = \left\{\mathrm{ID}_i, E\left(g_{sparse}^{(i)}\right), \mathcal{H}^{(i)}, \pi^{(i)}\right\} \tag{30}$$

Which represents all trusted encrypted local models across participating nodes. This verified output is then forwarded as the input to Federated Aggregation Under Encryption, where secure aggregation and differential privacy mechanisms combine the validated updates into a global model without ever decrypting individual node contributions.

3.6 Federated Aggregation Under Encryption

3.6.1 Secure Aggregation via Homomorphic Encryption (HE)

Once the blockchain validation in Section 3.5 is complete, each authenticated WSN node transmits its encrypted sparsified gradient package $E\left(g_{sparse}^{(i)}\right)$ to the global aggregator. Using the additive property of Paillier Homomorphic Encryption, the aggregator performs secure aggregation directly over ciphertexts without accessing any raw gradients. Let each node i hold an encrypted update as $\mathfrak{C}_i = E\left(g_{sparse}(i)\right)$. The aggregator computes the encrypted global sum as Eq. (31):

$$\mathfrak{C}_{agg} = \prod_{i=1}^{N} \mathfrak{C}_i \; mod \; \mathcal{p}_m{}^2 = E\left(\sum_{i=1}^{N} g_{sparse}(i)\right) \tag{31}$$

Where, $\mathcal{p}_m$ denotes the Paillier modulus. The decryption key is held only by the trusted coordinator, ensuring that intermediate values remain inaccessible throughout aggregation.

This ciphertext-level addition preserves privacy while enabling parallel aggregation across WSN clusters. The approach eliminates decryption overhead during model fusion, minimizes computation at edge nodes, and ensures complete confidentiality of transmitted model parameters. The resulting encrypted aggregate $E\left(g_{\mathrm{agg}}\right)$ is forwarded to the next phase—Federated Trimmed Mean—for robust outlier-resistant model averaging. The architecture of the federated learning utilized in this framework is represented in Figure 3.

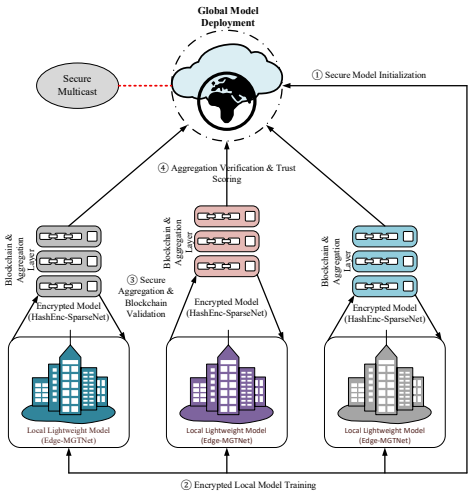

Figure 3: Blockchain-Based Federated Learning Architecture

3.6.2 Robust Model Integration via Federated Trimmed Mean

The encrypted aggregated model $E\left(g_{\mathrm{agg}}\right)$ obtained from the secure homomorphic aggregation phase is next processed through a Federated Trimmed Mean strategy to enhance robustness against outlier or Byzantine node updates. This step ensures that malicious or abnormal encrypted contributions do not distort the global model.

Each encrypted local update $E\left(g_{sparse}(i)\right)$ is first compared through a similarity check using encrypted distance metrics (homomorphic L2-norm estimations). Nodes whose updates deviate significantly from the median encrypted gradient distribution are flagged and excluded from aggregation. The remaining valid encrypted gradients are then used to compute a robust mean as Eq. (32):

$$E(g_{FTM}) = E\left(\frac{1}{N-2\alpha_{tr}N}\sum_{i=\alpha_{tr}N+1}^{N-\alpha_{tr}N} g_{sparse}(i)\right) \tag{32}$$

Where, $\alpha_{tr}(0 < \alpha_{tr} < 0.5)$ denotes the trimming ratio representing the fraction of extreme values removed from both ends of the distribution.

This robust aggregation mechanism maintains fairness across heterogeneous WSN nodes while mitigating the impact of adversarial or faulty devices. Since all operations remain within the encrypted domain, privacy is preserved throughout. The resulting $E(g_{FTM})$— the robustly aggregated encrypted model — is then forwarded to the final privacy enhancement stage, Differential Privacy (DP-SGD Noise Injection), to ensure formal privacy guarantees before global dissemination.

3.6.3 Differential Privacy Enforcement via DP-SGD

The robustly aggregated encrypted model $E(g_{FTM})$ from the previous stage is decrypted in a secure enclave and subjected to Differential Privacy Stochastic Gradient Descent (DP-SGD) to ensure node-level confidentiality before dissemination.

To limit information leakage, calibrated Gaussian noise is added to each aggregated gradient component, which is defined as Eq. (33):

$$\tilde{g}_{FTM} = g_{FTM} + \mathcal{N}(0, \sigma v^2) \tag{33}$$

Where, $\mathcal{N}(0, \sigma v^2)$ represents Gaussian noise with variance σv^2 tuned to the privacy budget (ε, δ). This process guarantees that any single node's contribution remains indistinguishable within the global model. The formal privacy bound is expressed as Eq. (34):

$$\mathcal{P}_B[\mathfrak{M}(\mathfrak{D}_1) \in \mathbb{S}] \leq e^{\varepsilon}\mathcal{P}_B[\mathfrak{M}(\mathfrak{D}_2) \in \mathbb{S}] + \delta \tag{34}$$

Where, $\mathfrak{D}_1$ and $\mathfrak{D}_2$ are neighboring datasets differing by one node, and $\mathfrak{M}$ denotes the DP mechanism. The resulting Differentially Private Global Model, denoted as $M_{DP} = DP\_SGD(g_{FTM})$, preserves global performance while enforcing rigorous privacy constraints. This securely privatized model is then forwarded to Trust & Attack Mitigation, where integrity validation and resilience mechanisms are applied before the final model broadcast across smart city WSN nodes.

3.7 Trust & Attack Mitigation

3.7.1 Multi-Layered Defense via Robust Aggregation and Byzantine Filtering

The Differentially Private Global Model M_{DP}, obtained from Section 3.6, serves as the input to the first defense layer that integrates Robust Aggregation and Byzantine Filtering. This stage aims to ensure the reliability and security of the model updates received from diverse and potentially compromised WSN nodes deployed in smart city environments.

In the Robust Aggregation phase, the system first collects all validated encrypted updates $\left\{M_{DP}^{(i)}\right\}_{i=1}^{N}$ and computes their mean while discarding statistical outliers that deviate significantly from the central tendency. A trimmed-mean strategy is applied as Eq. (35):

$$M_{RA} = \frac{1}{N-2\Sigma N}\sum_{i=\Sigma N+1}^{N-\Sigma N} M_{DP}^{(i)} \tag{35}$$

Where, Σ denotes the trimming ratio used to exclude the top and bottom ΣN anomalous updates based on their deviation magnitude. This step effectively mitigates extreme or inconsistent model contributions while retaining the essential distributional characteristics of legitimate updates.

Following this, Byzantine Filtering identifies and eliminates malicious or poisoned model updates by measuring their similarity to the global consensus. Each node's update is evaluated via cosine similarity as Eq. (36):

$$\mathfrak{S}_i = \frac{\langle M_{DP}^{(i)}, M_{RA}\rangle}{\|M_{DP}^{(i)}\|\|M_{RA}\|} \tag{36}$$

Where, a lower $\mathfrak{S}_i$ indicates higher deviation and potential Byzantine behavior. Nodes with similarity below a predefined trust threshold $\tau_{\mathfrak{B}}$ are filtered out as Eq. (37):

$$M_{BF} = \left\{M_{DP}^{(i)} | \mathfrak{S}_i \geq \tau_{\mathfrak{B}}\right\} \tag{37}$$

This ensures that only trustworthy, statistically consistent updates participate in subsequent training rounds. Through this layered defense, the system resists collusion, gradient poisoning, and update manipulation attacks while maintaining fairness and reliability across heterogeneous edge nodes. The output of this phase—denoted as M_{BF}—represents a Byzantine-resilient, statistically filtered model update set, which is securely forwarded to the next stage for anomaly-based rejection and trust reinforcement.

3.7.2 Adaptive Trust Reinforcement via Anomaly Rejection and EWMA-Scoring

The Byzantine-resilient model updates M_{BF} obtained from Section 3.7.1 are now passed to the second defense layer—Adaptive Trust Reinforcement, which combines Anomaly-Based Update Rejection and EWMA-Scoring to sustain long-term reliability and adaptiveness in smart city WSN environments.

In the Anomaly-Based Update Rejection phase, each participating node's update is monitored for abnormal behavior through deviation and residual analysis. The deviation score $\mathfrak{DS}_i$ of a node is computed as Eq. (38):

$$\mathfrak{DS}_i = \| M_{BF}^{(i)} - \overline{M_{BF}} \|_2 \tag{38}$$

Where, $\overline{M_{BF}}$ denotes the mean of the validated model updates. Nodes with $\mathfrak{DS}_i > \delta_{\mathfrak{A}}$, where $\delta_{\mathfrak{A}}$ denotes the adaptive anomaly threshold, are flagged as suspicious and removed from the aggregation pool. This filtering mechanism ensures that only nodes exhibiting stable and consistent update patterns are retained, thereby minimizing the influence of hidden poisoning or erratic updates.

Following anomaly rejection, EWMA (Exponentially Weighted Moving Average) Scoring dynamically adjusts trust levels based on the node's historical reliability. Each node i is assigned a trust score $\mathfrak{T}_i(\mathrm{t})$, updated iteratively as Eq. (39):

$$\mathfrak{T}_i(\mathrm{t}) = \varsigma \mathfrak{U}_i(\mathrm{t}) + (1-\varsigma)\mathfrak{T}_i(\mathrm{t}-1) \tag{39}$$

Where, $\mathfrak{U}_i(\mathrm{t})$ represents the recent reliability measure of node iderived from its successful participation and alignment with the global consensus, and $\varsigma \in [0,1]$ is the smoothing factor that controls responsiveness to new evidence. Nodes consistently providing valid, high-quality updates yield higher $\mathfrak{T}_i(\mathrm{t})$ values, while untrustworthy or anomalous participants experience exponential trust decay over time.

Finally, the system performs trust-weighted model reinforcement, computing the globally trusted model as Eq. (40):

$$M_{Trust} = \frac{\sum_{i=1}^{N} \mathfrak{T}_i(\mathrm{t})\, M_{BF}^{(i)}}{\sum_{i=1}^{N} \mathfrak{T}_i(\mathrm{t})} \tag{40}$$

This yields a Trust-Weighted, Attack-Free Global Model Update Set that not only resists adversarial disruptions but also promotes self-adaptive trust evolution among heterogeneous WSN nodes. The resulting secure and reliable model M_{Trust} is subsequently forwarded to Section 3.8 (Global Model Redeployment) for synchronization and broadcast across the distributed smart city network.

3.8 Global Model Redeployment

The Trust-Weighted Global Model M_{Trust}, obtained from Section 3.7, represents the final, integrity-assured, and privacy-preserving global intelligence synthesized from all participating WSN nodes. In this concluding stage, Secure Multicast Redeployment is performed to distribute the finalized model efficiently and safely across the entire smart city wireless sensor network.

Each verified node receives the updated global parameters through an end-to-end encrypted multicast channel, ensuring confidentiality and authenticity during transmission. The multicast employs cryptographically authenticated session keys $\mathcal{K}_{sess}$ derived from blockchain-based identity credentials, such that each communication round satisfies as Eq. (41):

$$M_{Global}^{(i)} = Dec_{\mathcal{K}_{sess}^{(i)}}(M_{Trust}) \tag{41}$$

Where, $Dec_{\mathcal{K}_{sess}^{(i)}}(\cdot)$ denotes decryption using node-specific secure keys managed via Hyperledger-based identity modules. This guarantees that only verified and trusted participants can receive and utilize the final model. Once deployed, each node locally synchronizes its parameters as $\Phi^{(i)} \leftarrow \varpi\Phi^{(i)} + (1-\varpi)M_{Trust}$, where $\varpi \in [0,1]$ controls adaptive blending between the node's previous state and the new global model, ensuring smooth convergence and avoiding abrupt parameter shifts in resource-constrained environments.

Through this secure redeployment cycle, every edge node in the smart city ecosystem gains access to a globally refined model that embodies collective intelligence, verified trust, and preserved privacy. This phase closes the end-to-end lifecycle of the Blockchain-Enabled Federated Learning Framework, wherein each stage—from local model training and encrypted transmission to privacy-preserving aggregation and trust-driven validation—collectively fortifies security, scalability, and reliability.

Ultimately, the redeployed global model serves as a continuously evolving intelligence layer, empowering smart city WSN nodes to make adaptive, decentralized, and resilient decisions while maintaining full compliance with data privacy and integrity standards.

4. Result and Discussion

4.1 Experimental Setup

The proposed smart city WSN framework has been implemented using Python, leveraging the fused dataset obtained from the SmartCity Cybersecurity IoT Dataset and Location Intelligence for Cybersecurity 2025. For performance evaluation, the proposed approach has been compared against existing state-of-the-art techniques, including FSGARH-L [20], BM-SDA [22], BS-SCRM [23], and FL-SCNN-Bi-LSTM [26]. Key performance metrics considered in this work for comparison include Accuracy (%), Precision (%), Sensitivity (%), Specificity (%), F1 Score (%), NPV (%), MCC (%), FPR (%), and FNR (%). Two comparative analyses have been provided: first, a performance comparison between the proposed method and existing techniques; and second, a K-fold comparison analysis to validate robustness and generalization. The evaluation has been visualized using three representative graphs: confusion matrix analysis (Figure 4), ROC curve comparison across models (Figure 5), and Accuracy/Loss versus Epochs trends (Figure 6).

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

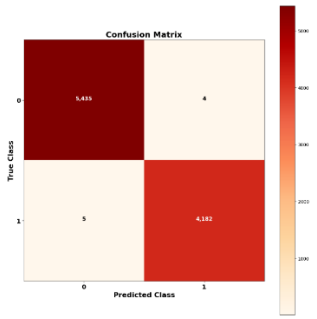

Figure 4: Confusion Matrix Analysis

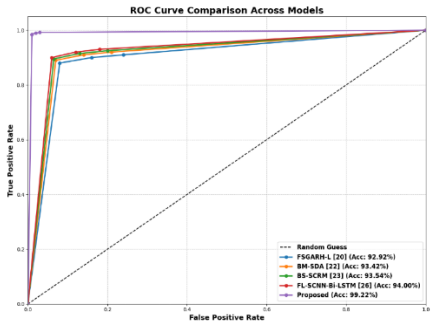

Figure 5: ROC Curve Comparison

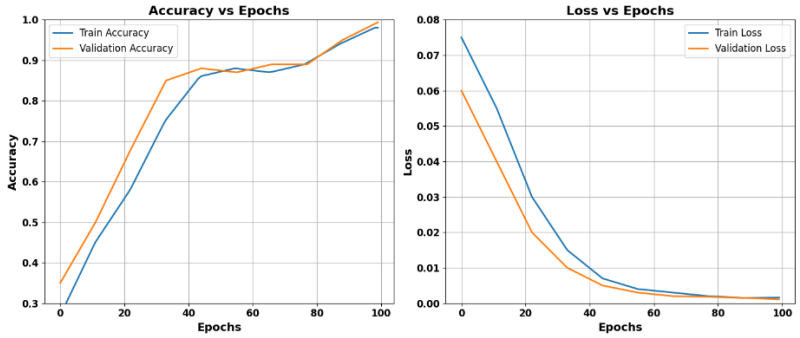

Figure 6: Accuracy/Loss vs. Epochs

4.2 Performance Comparison Analysis

The comparative performance of the proposed framework with the existing methods, such as FSGARH-L [20], BM-SDA [22], BS-SCRM [23], and FL-SCNN-Bi-LSTM [26] is summarized in Table 2 and Figure 7. The proposed method has a high Accuracy (99.22) as compared to the best available method, FL-SCNN-Bi-LSTM (94%). This improvement can be explained by Lightweight Local Model Training via Edge-MGTNet, which effectively incorporates MobileNet-V3 to encode spatial/contextual features, Tiny-GNN to encode topological features, Micro-TCN to encode temporal features, and Edge-Pruning and Knowledge Distillation. These modules are synergistic, enabling the model to obtain a wide range of high-fidelity representations at computational efficiency, which directly enhances classification accuracy in a heterogeneous WSN node.

Table 2: Comparative Performance Metrics of Proposed and Existing Methods

Model	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F1 Score (%)	NPV (%)	MCC (%)	FPR (%)	FNR (%)
FSGARH-L [20]	92.92	91.87	91.34	92.06	91.61	92.82	90.39	7.55	6.21
BM-SDA [22]	93.42	92.37	91.79	92.52	91.77	92.97	90.53	7.26	6.04
BS-SCRM [23]	93.54	92.48	91.92	92.63	91.89	93.06	90.64	6.19	5.85
FL-SCNN-Bi-LSTM [26]	94	92.56	92.24	92.77	92.46	93.53	91.13	6	5.23
Proposed	99.22	98.32	96.9	96.56	96.12	97.23	95.81	2.85	1.93

The proposed method is better in the context of Sensitivity (96.9%), compared to the previous methods (maximum 92.24% in FL-SCNN-Bi-LSTM), as it provides strong detection of the relevant anomalies and attacks. This is attributed to the Encrypted Model Packaging via HashEnc-SparseNet which maintains important gradient data using Paillier Homomorphic Encryption, sparsified gradient encoding, and hash-based integrity verification. Through safe and trustworthy updates of the model in the process of federated learning, the system prevents the loss of information and guarantees a high recall of legitimate events.

Equally, other metrics are also improved: Precision (98.32%), Specificity (96.56%), F1 Score (96.12%), NPV (97.23%), MCC (95.81%), FPR (2.85%), and FNR (1.93%), which proves that the proposed framework is both highly reliable and robust in the context of smart city WSN. These findings underscore the fact that the Edge-MGTNet-based local intelligence extraction and HashEnc-SparseNet-based secure packaging are critical in ensuring high performance in various evaluation aspects.

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

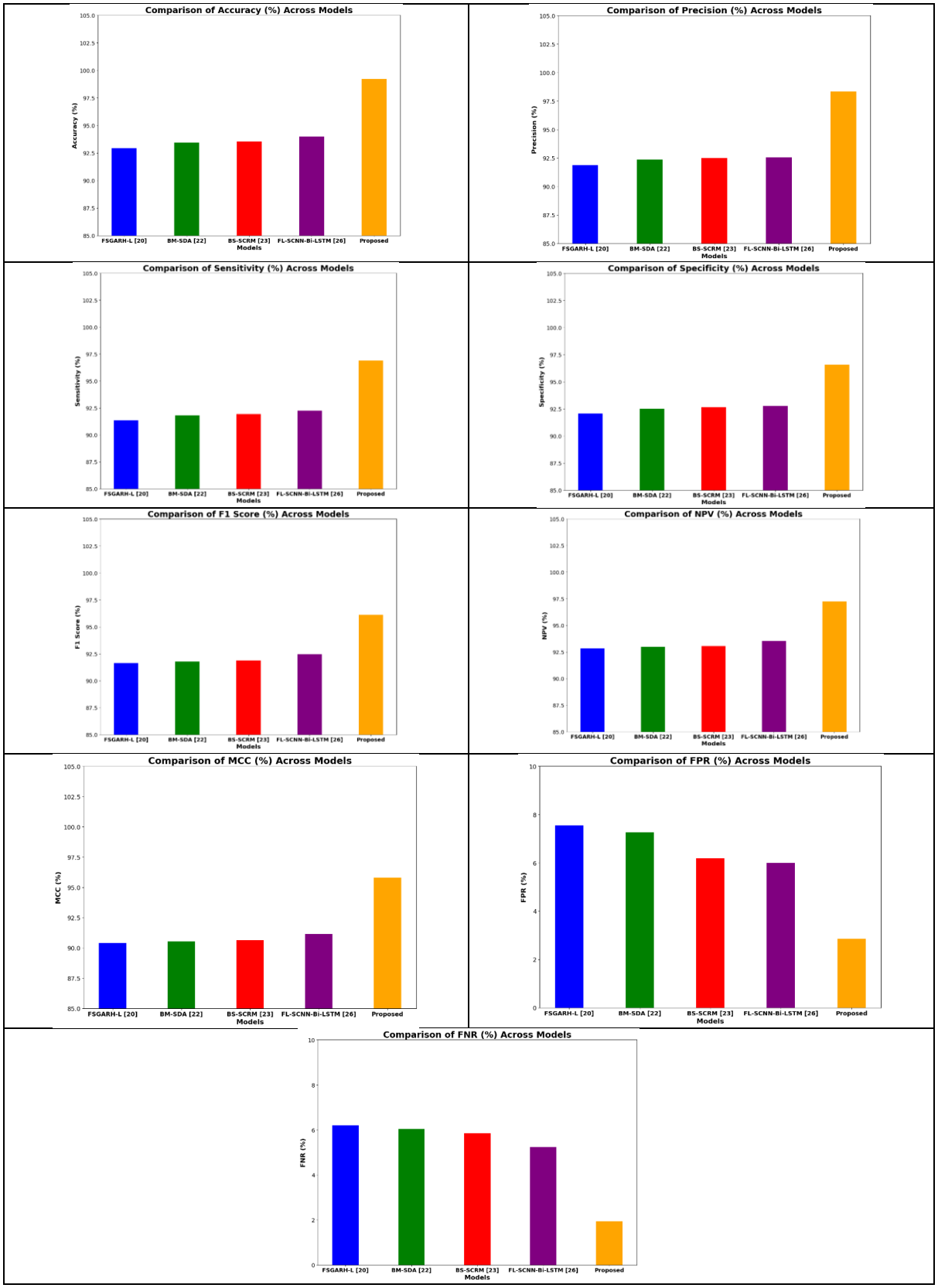

Figure 7: Graphical Comparison of Performance Metrics Across Proposed and Existing Techniques

4.3 K-Fold Comparison Analysis

In order to assess the strength and generalization ability of the suggested framework, a 5-fold cross-validation on the fused Dataset have conducted. Table 3 and Figure 8 provide the summary of the mean accuracy of the proposed approach versus the existing methods FSGARH-L [20], BM-SDA [22], BS-SCRM [23], and FL-SCNN-Bi-LSTM [26].

Received: August 05, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 3: K-Fold Cross-Validation Accuracy of Proposed and Existing Techniques

Model	k1 (%)	k2 (%)	k3 (%)	k4 (%)	k5 (%)	Mean Accuracy (%)
FSGARH-L [20]	92.8	92.85	92.9	92.88	92.87	92.92
BM-SDA [22]	93.35	93.4	93.45	93.42	93.41	93.42
BS-SCRM [23]	93.5	93.55	93.6	93.52	93.53	93.54
FL-SCNN-Bi-LSTM [26]	93.95	94	94.05	94.02	94.03	94
Proposed	98.95	99.05	99.15	99.1	99.2	**99.22**

The suggested framework is always performing better on all folds, and the mean accuracy of the proposed framework is 99.22, which is much higher than the highest existing approach, FL-SCNN-Bi-LSTM (94%). The variance of each fold (k1–k5) is low, which shows a stable and reliable model performance. Conversely, the current methods are less accurate and have minor variations between folds, which is indicative of sensitivity to heterogeneous data of IoT/WNS and be susceptible to noisy or anomalous data.

The superior k-fold performance of the proposed algorithm can be explained by its Edge-MGTNet local feature extraction, HashEnc-SparseNet encrypted model packaging, and federated aggregation with differential privacy that guarantee robust learning, secure gradient transmission, and privacy-preserving global model updates. This combination offers resistance to both statistical outliers and adversarial perturbations, which makes the system very appropriate to smart city WSN settings.

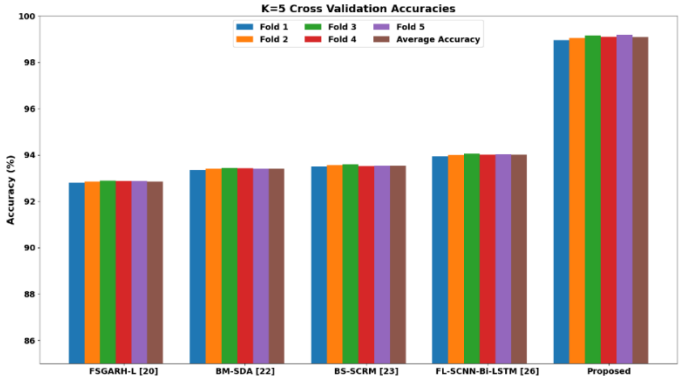

Figure 8: Graphical Comparison of K-Fold Accuracy Across Proposed and Existing Techniques

5. Conclusion

This work presented a novel methodology for ensuring secure, privacy-preserving, and efficient federated learning in smart city WSN settings. The methodology commenced with the integration of SmartCity Cybersecurity IoT Dataset and Location Intelligence for Cybersecurity 2025 Dataset to form a fused input, with subsequent sensing and local staging with Kalman Filter and temporal smoothing for producing noise-suppressed, temporally coherent data. Lightweight local model training via Edge-MGTNet extracted multi-modal features with MobileNet-V3, Tiny-GNN, and Micro-TCN, with edge pruning and knowledge distillation refining local models. Lightweight local models were compressed and secured with HashEnc-SparseNet, which utilizes sparsified gradient encoding, Paillier homomorphic encryption, as well as hash-based integrity authentication. Blockchain registration and validation with Hyperledger Fabric and zk-SNARKs guaranteed authenticity, with federated aggregation under encryption implementing secure aggregation, federated trimmed mean, as well as differential privacy (DP-SGD) to generate a globally coherent model. Attack mitigation with trust utilized robust aggregation, Byzantine filtering, anomaly-driven update reject, as well as EWMA scoring, and global model redeployment using secure multicast as the final step.

The work was coded in Python and experimental outcomes showed that it had high performance with an accuracy of 99.22, precision of 98.32 and F1-score of 96.12. This approach would greatly improve cybersecurity, privacy, and reliability of smart city WSNs, which would contribute to sustainable urban digital infrastructures. Future directions include dynamic adaptive learning in changing threat models and with heterogeneous 5G/6G-enabled IoT networks to enhance resilience and scalability further.

Data Availability

The datasets utilized in this work are publicly available. The SmartCity Cybersecurity IoT Dataset can be accessed at Kaggle, while the Location Intelligence for Cybersecurity 2025 dataset is available at Kaggle. These datasets provide heterogeneous IoT and cybersecurity information, supporting research on secure, privacy-preserving smart city WSNs.

Reference

[1]. Khalifeh, A., Darabkh, K. A., Khasawneh, A. M., Alqaisieh, I., Salameh, M., AlAbdala, A., ... & Rajendiran, K. (2021). Wireless sensor networks for smart cities: Network design, implementation and performance evaluation. *Electronics*, *10*(2), 218.

[2]. Aburukba, R., & El Fakih, K. (2025). Wireless Sensor Networks for Urban Development: A Study of Applications, Challenges, and Performance Metrics. *Smart Cities*, *8*(3), 89.

[3]. Mehmood, M. Y., Oad, A., Abrar, M., Munir, H. M., Hasan, S. F., Muqeet, H. A. U., & Golilarz, N. A. (2021). Edge computing for IoT-enabled smart grid. *Security and communication networks*, *2021*(1), 5524025.
[4]. Raj, E. F. I., Appadurai, M., Darwin, S., & Rani, E. F. I. (2022). Internet of things (IoT) for sustainable smart cities. In *Internet of things* (pp. 163-188). CRC Press.
[5]. Jabbar, W. A., Tiew, L. Y., & Shah, N. Y. A. (2024). Internet of things enabled parking management system using long range wide area network for smart city. *Internet of Things and Cyber-Physical Systems*, *4*, 82-98.
[6]. Park, S., & Lee, K. (2021). Improved mitigation of cyber threats in IIoT for smart cities: a new-era approach and scheme. *Sensors*, *21*(6), 1976.
[7]. Xie, Q., Li, K., Tan, X., Han, L., Tang, W., & Hu, B. (2021). A secure and privacy-preserving authentication protocol for wireless sensor networks in smart city. *EURASIP Journal on Wireless Communications and Networking*, *2021*(1), 119.
[8]. Khan, W., Usama, M., Khan, M. S., Saidani, O., Al Hamadi, H., Alnazzawi, N., ... & Ahmad, J. (2025). Enhancing security in 6G-enabled wireless sensor networks for smart cities: a multi-deep learning intrusion detection approach. *Frontiers in Sustainable Cities*, *7*, 1580006.
[9]. Jia, C., Ding, H., Zhang, C., & Zhang, X. (2021). Design of a dynamic key management plan for intelligent building energy management system based on wireless sensor network and blockchain technology. *Alexandria Engineering Journal*, *60*(1), 337-346.
[10]. Alam, T. (2024). Data privacy and security in autonomous connected vehicles in smart city environment. *Big Data and Cognitive Computing*, *8*(9), 95.
[11]. Su, L., & Lau, V. K. (2023). Accelerated federated learning over wireless fading channels with adaptive stochastic momentum. *IEEE Internet of Things Journal*, *11*(8), 14136-14152.
[12]. Kumar, K. S., Nair, S. A. H., Roy, D. G., Rajalingam, B., & Kumar, R. S. (2021). Security and privacy-aware artificial intrusion detection system using federated machine learning. *Computers & Electrical Engineering*, *96*, 107440.
[13]. Rehman, A., Abdullah, S., Fatima, M., Iqbal, M. W., Almarhabi, K. A., Ashraf, M. U., & Ali, S. (2022). Ensuring security and energy efficiency of wireless sensor network by using blockchain. *Applied Sciences*, *12*(21), 10794.
[14]. Hijazi, N. M., Aloqaily, M., Guizani, M., Ouni, B., & Karray, F. (2023). Secure federated learning with fully homomorphic encryption for iot communications. *IEEE Internet of Things Journal*, *11*(3), 4289-4300.
[15]. Deng, Y., Guo, B., & Chen, S. (2025). Privacy-Preserving Approach to Edge Federated Learning Based on Blockchain and Fully Homomorphic Encryption. *Electronics*, *14*(2), 361.
[16]. Choi, S., Patel, D., Zad Tootaghaj, D., Cao, L., Ahmed, F., & Sharma, P. (2024). FedNIC: enhancing privacy-preserving federated learning via homomorphic encryption offload on SmartNIC. *Frontiers in Computer Science*, *6*, 1465352.
[17]. Gowdhaman, V., & Dhanapal, R. (2022). An intrusion detection system for wireless sensor networks using deep neural network. *Soft Computing*, *26*(23), 13059-13067.
[18]. Al-Fuhaidi, B., Farae, Z., Al-Fahaidy, F., Nagi, G., Ghallab, A., & Alameri, A. (2024). Anomaly-Based Intrusion Detection System in Wireless Sensor Networks Using Machine Learning Algorithms. *Applied Computational Intelligence and Soft Computing*, *2024*(1), 2625922.
[19]. Hussain, K., Xia, Y., Onaizah, A. N., Manzoor, T., & Jalil, K. (2022). Hybrid of WOA-ABC and proposed CNN for intrusion detection system in wireless sensor networks. *Optik*, *271*, 170145.
[20]. Pichumani, S., Sundararajan, T. V. P., & Ramesh, S. M. (2025). Federated stochastic gradient averaging ring homomorphism based learning for secure data aggregation in WSN. *Scientific Reports*, *15*(1), 18590.
[21]. Nouman, M., Qasim, U., Nasir, H., Almasoud, A., Imran, M., & Javaid, N. (2023). Malicious node detection using machine learning and distributed data storage using blockchain in WSNs. *IEEe Access*, *11*, 6106-6121.
[22]. Jain, K., & Kumar, S. (2025). Integrating Blockchain and Machine Learning for Secure Data Aggregation in Wireless Sensor Networks. *International Journal of Communication Systems*, *38*(16), e70259.
[23]. Xiao, J., Li, C., Li, Z., & Zhou, J. (2024). BS-SCRM: a novel approach to secure wireless sensor networks via blockchain and swarm intelligence techniques. *Scientific Reports*, *14*(1), 9709.
[24]. Jeyakumar, S. R., Rahman, M. Z. U., Sinha, D. K., Kumar, P. R., Vimal, V., Singh, K. U., ... & Balajee, J. (2024). An Innovative Secure and Privacy-Preserving Federated Learning-Based Hybrid Deep Learning Model for Intrusion Detection in Internet-Enabled Wireless Sensor Networks. *IEEE Transactions on Consumer Electronics*, *71*(1), 273-280.
[25]. Devi, M., Nandal, P., & Sehrawat, H. (2025). Federated Learning-Enabled Lightweight Intrusion Detection System for Wireless Sensor Networks: A Cybersecurity Approach Against DDoS Attacks in Smart City Environments. *Intelligent Systems with Applications*, 200553.
[26]. Bukhari, S. M. S., Zafar, M. H., Abou Houran, M., Moosavi, S. K. R., Mansoor, M., Muaaz, M., & Sanfilippo, F. (2024). Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. *Ad Hoc Networks*, *155*, 103407.