A Dual-Attention CNN–GCN–BiLSTM Framework for Intelligent Intrusion Detection in Wireless Sensor Networks

Laith H. Baniata *, Ashraf ALDabbas, Jaffar M. Atwan, Hussein Alahmer, Basil Elmasri, Chayut Bunterngchit *

Posted Date: 19 November 2025

doi: 10.20944/preprints202511.1423.v1

Keywords: wireless sensor networks (WSN); intrusion detection system (IDS); deep learning; multiscale convolution; graph convolutional networks; attention mechanism; bidirectional lstm; wsn-ds dataset; cybersecurit; edge computing

 doi:10.20944/preprints202511.1423.v1

Article

A Dual-Attention CNN–GCN–BiLSTM Framework for Intelligent Intrusion Detection in Wireless Sensor Networks

Laith H. Baniata [1,*], Ashraf ALDabbas [2], Jaffar M. Atwan [2,3], Hussein Alahmer [1], Basil Elmasri [2] and Chayut Bunterngchit [4,*]

[1] Department of Autonomous Systems, Faculty of Artificial Intelligence, Al-Balqa Applied University, Al-Salt 19117, Jordan
[2] Intelligent Systems Department, Faculty of Artificial Intelligence, Al-Balqa Applied University, Al-Salt 19117, Jordan
[3] Department of Computer Science, Faculty of Information Technology, Applied Science Private University, Amman 11937, Jordan
[4] Division of Industrial and Logistics Engineering Technology, Faculty of Engineering and Technology, King Mongkut's University of Technology North Bangkok, Rayong Campus, Rayong 21120, Thailand
* Correspondence: laith.baniata@bau.edu.jo (L.H.B.); chayut.b@eat.kmutnb.ac.th (C.B.)

Abstract

Wireless sensor networks (WSNs) are increasingly being used in mission-critical infrastructures. In such applications, they are evaluated on the risk of cyber intrusions that can target the already constrained resources. The traditional intrusion detection systems (IDS) in WSNs are based on machine learning techniques. Such models fail to capture the nonlinear, temporal, and topological dependencies across the network nodes. Consequently, they cause degradation in the detection accuracy and poor adaptability against evolving threats. To overcome these limitations, this study introduced a hybrid deep learning-based IDS that integrated multi-scale convolutional feature extraction, dual-stage attention fusion, and graph convolutional reasoning. In addition, bidirectional long short-term memory components are embedded into the unified framework. The proposed architecture captures the hierarchical spatial-temporal correlations in the traffic patterns. This allows making a precise discrimination between the normal and attack behaviors across several intrusion classes. The model has been evaluated on the benchmarking public available dataset and found to attain a higher classification capability in the multiclass scenarios. The model has further been found to outperform the conventional models focusing on the IDS frameworks. In addition, the proposed design is aimed at retaining suitable computational efficiency, which is suitable for edge and distributed deployments. This makes it an effective solution for the next-generation WSN cybersecurity. The overall findings have focused on combining topology-aware learning with multi-branch attention mechanisms for offering a balanced trade-off between interpretability, accuracy, and deployment efficiency for the resource-constrained WSN networks.

Keywords: wireless sensor networks (WSN); intrusion detection system (IDS); deep learning; multi-scale convolution; graph convolutional networks; attention mechanism; bidirectional lstm; wsn-ds dataset; cybersecurit; edge computing

1. Introduction

Wireless sensor networks (WSNs) are an essential part of the internet of things that are increasingly becoming common in critical infrastructures, environmental monitoring, healthcare, and industrial control systems [1–5]. These networks have revolutionized the sensing and communication frameworks. Nevertheless, the distributed and energy-constrained nature of these networks makes them susceptible to cyber intrusions [6–8]. Some of the commonly found attacks involve denial of service (DoS), jamming, sinkhole, blackhole, and selective forwarding. Such attacks lead to a compromise in the data integrity, network reliability, and real-time decision-making. Thus, the development of

 doi:10.20944/preprints202511.1423.v1

efficient and intelligent intrusion detection systems (IDS) is essential for protecting the WSNs while ensuring their energy efficiency and scalability [6–8].

The traditional WSN IDS framework predominantly relies on statistical analysis and rule-based systems. The classical machine learning (ML) algorithms like random forest (RF), support vector machine, k-nearest neighbors (kNN), and decision trees (DT) offer interpretability and low computational overhead [9–11], yet their performance reduces drastically under the dynamic topologies and non-linear attack patterns which are commonly found in real-world deployments. Some of the recent advancements towards deep learning based IDS have explored architectures like convolutional neural networks (CNNs), long short-term Memory (LSTM), and a hybrid combination of CNN-LSTMs [9–11]. In addition, autoencoders and graph neural networks have also been explored for capturing hierarchical and temporal dependencies in the traffic behavior. While offering higher accuracies, these methods exhibit certain limitations: (i) high energy and memory footprints unsuited for low-power sensor nodes, (ii) poor generalization to unseen attack types and evolving network conditions, and (iii) lack of interpretability and privacy preservation in distributed scenarios. Such gaps have been targeted by several studies in recent times.

Houda et al. [12] offered a collaborative federated learning framework that makes use of a secure aggregation protocol for the detection of jamming (including contrast, random, reactive, and deceptive). The study attained an accuracy of ≈99%, yet the focus has been limited to jamming behaviors while further assuming federated learning connectivity. The broader multi-attack generalization and on-device energy and communication costs have not been incorporated in the proposed design. Jeyakumar et al. [13] proposed a hybrid stacked CNN-bidirectional LSTM (BiLSTM). The model has been tuned by an African vulture optimization algorithm and trained using federated learning. The communication and aggregation overheads, along with the robustness of federated learning under, have not been fully quantified in the study. In addition, the interpretability and node-level resource constraints are found to be partially addressed. In Zhou et al. [14], a tabular to image transform has been incorporated by using transfer learning. The article involves MobileNet and Xception with black kite algorithm for the hyperparameter search and ensemble. The model has been found to offer a heavy computational footprint with a lack of explainability and uncertainty for real-world deployments.

Halbouni et al. [15] adopted CNN and LSTM for spatial and temporal features for the classification using three datasets. The models offered limited treatment of the class imbalance and WSN energy/latency constraints. Vinayakumar et al. [16] focused on a systematic deep learning (DNN) benchmarking compared to classical ML using multiple intrusion datasets. The early deep benchmarking datasets were found to lack WSN-specific topology and context modeling for interpretability. The false positive control was found under multi-attack scenarios. Birahim et al. [17] adopted particle swarm optimization (PSO) for the feature and hyperparameter search using an ensemble (RF/DT/kNN) and class imbalance handled using SMOTE-Tomek. The study failed to model the network-structure awareness and temporal dependencies. In addition scalability of the models was not fully addressed. Hakami et al. [18] presented a pipeline having SMOTE for the balancing of Pearson correlation for the feature selection. While using the WSN-DS dataset, the studies offered fair performance yet failed to address the privacy-preserving learning along with computational resource requirements. Alzahrani et al. [19] presented a ConvLSTM model offering benchmarking performance in several datasets. However, it was tailored for UAV networks having poor transferability towards resource-constrained environments. Similarly, a Red Kite Optimization framework [20] was proposed that focused on average ensemble, LCWOA, and hyperparameter tuning. The model relies on heuristic feature selection and moving ensembles without offering temporal and graph modeling. Atitallah et al. [21] Jiang et al. [21], and Saleh et al. [22], collectively enhanced the detection accuracy using fuzzy-graph attention, meta-heuristic optimization, and SGD-based learning. Yet their real-time adaptability and scalability are limited for making lightweight deployments.

A summary of these articles has been presented in Table 1

 doi:10.20944/preprints202511.1423.v1

Table 1. Summary of reviewed intrusion detection models on WSN-DS and related datasets.

Article	Methodology / Model	Dataset	Key Limitation / Gap
[12]	Federated learning with secure aggregation for jamming attack detection	WSN-DS (Jamming classes)	Limited to jamming attacks; lacks multi-attack scalability
[13]	Hybrid SCNN–BiLSTM optimized via african vulture optimization under a federated learning setup	WSN-DS, CIC-IDS2017	Low communication efficiency in FL; lacks interpretability
[14]	Transfer learning with MobileNet/VGG19 ensemble optimized by Black Kite Algorithm	ToN-IoT, Edge-IIoTset, WSN-DS	High computational load; poor real-time adaptability
[15]	CNN–LSTM hybrid model integrating spatial-temporal dependencies	WSN-DS (Binary and Multi-class)	Class imbalance and explainability not addressed
[16]	DNN benchmarked against classical ML baselines	KDDCup'99, NSL-KDD, WSN-DS	No WSN-specific topology modeling; high false positives
[17]	PSO-based feature selection with RF, DT, and kNN ensemble plus LIME/SHAP explanations	WSN-DS (Binary)	No temporal or spatial dependency modeling
[18]	SMOTE-based balancing and PCC feature selection for ML/DL comparison	WSN-DS, UNSW-NB15, CIC-IDS2017	No topology-aware or energy-efficient design
[19]	ConvLSTM for spatial-temporal intrusion detection in IoD networks	WSN-DS, NSL-KDD, Drone dataset	Limited to UAV context; weak transferability to WSNs
[20]	Red Kite Optimization with average ensemble fusion and LCWOA tuning	WSN-DS (Binary)	No adaptive temporal modeling; lacks robustness to evolving threats
[21]	Fuzzy graph attention network for relational uncertainty learning	Edge-IIoTSet, CIC-Malmem, WSN-DS	Computationally expensive; unsuitable for constrained WSNs
[22]	SGD-based optimization for lightweight ML classifiers in WSN intrusion detection	WSN-DS (Binary)	Simplistic linear models; limited scalability for dense WSNs
[23]	Improved arithmetic optimization algorithm integrated with XGBoost	WSN-DS (Binary)	Static learning; lacks adaptive or online retraining

The studies reviewed above help in highlighting substantial progress in the IDS detection approaches; however, they have several challenges. The federated and optimization-based models, including Federated SCNN-BiLSTM and AVOA, along with Privacy-Preserving FL for jamming, have

 doi:10.20944/preprints202511.1423.v1

improved distributed learning, yet they face communication and synchronization bottlenecks. The optimization-driven framework involving CBCTL-IDS, RKOA, AEID, and ST-IAOA-XGBoost has shown high detection accuracies. Yet they remain computationally heavy on the node deployments. The explainable artificial intelligence (XAI) techniques, including PSO-Ensemble and LIME/SHAP, have enhanced the interpretability, yet they lack real-time adaptability and spatial-temporal reasoning. The fuzzy graph attention networks capture the topological relations, yet suffer from high complexity and limited scalability in the constrained WSN environments. Overall, the following research gaps have been identified:

- **Deployment Realism:** Existing models overlook computational and energy limitations of distributed sensor nodes.
- **Temporal–Spatial Dependency:** Most IDS fail to jointly model both the temporal evolution of attacks and spatial correlations among nodes.
- **Dynamic Adaptation:** Static training prevents adaptation to changing traffic distributions and novel intrusions.
- **Interpretability and Fusion:** Few works integrate multi-level feature fusion or interpretable decision mechanisms within hybrid deep architectures.

To overcome these limitations, the proposed framework introduces a multi-branch hybrid deep learning framework optimized for the IDS in WSNs. The model integrated multi-scale CNN blocks, attention-based fusion layers, graph convolutional network (GCN) operations, and BiLSTM for effectively capturing the multi-resolution, spatial, and temporal dependencies in the WSN-DS traffic. The multi-scale CNN layers have been used to extract the hierarchical frequency-temporal patterns. The attention fusion dynamically re-weights the salient spatial-temporal features. The GCN components help in encoding the inter-node topological relationships, and the BiLSTM layers model the bidirectional temporal correlations. These help in enhancing the detection of subtle and evolving attack behaviors. Finally, the dense layers and softmax classifiers produce probabilistic intrusion classifications. Compared to previous models, the proposed framework is lightweight, modular, and optimized for both the centralized and distributed detection schemes. It helps in offering improved generalization towards unseen intrusions and maintains computational efficiency in real-time edge deployments. Overall, the study contributes as follows:

1. It integrates multi-scale CNN, attention fusion, GCN, and BiLSTM to capture comprehensive spatio-temporal dynamics of WSN traffic.
2. The model learns hierarchical and context-aware embeddings that improve separability between normal and anomalous traffic. This is attained through multi-branch feature extraction and adaptive attention weighting.
3. Introduces advanced preprocessing and normalization steps to ensure stability.

The rest of the article has been structured as follows. Section II offers a detailed methodology framework incorporating preprocessing, model, and evaluation details. Section III provides results along with a critical analysis of the findings and validation against the benchmarking models. Section IV concludes the study along with a potential future roadmap.

2. Materials and Methods

2.1. Dataset Description

The study employed the WSN-DS dataset for carrying out the experimentation and evaluations. The WSN-DS dataset, developed by Almomani *et al.* [24], is specifically designed for the detection of Denial-of-Service (DoS) attacks and consists of 374,661 records, with approximately 9% labeled as DoS incidents. This dataset was constructed using the *LEACH* protocol, a widely adopted hierarchical routing protocol in Wireless Sensor Networks (WSNs), and encompasses both normal network behavior and four distinct types of DoS attacks: Grayhole, Blackhole, TDMA, and Flooding. Data collection was performed using *Network Simulator 2 (NS-2)*, and the resulting traces were processed to extract

 doi:10.20944/preprints202511.1423.v1

18 relevant features. Due to its comprehensive structure and labeled attack scenarios, the WSN-DS dataset serves as a valuable benchmark for researchers developing intrusion detection strategies and enhancing the security of WSNs [25].

The dataset has been treated as a benchmark corpus for the IDS in the WSNs. Each of the records in the data comprises 18 continuous-valued attributes that represent traffic, energy, and protocol-level indicators. These are followed by a categorical class label $y \in \{C_1, C_2, \ldots, C_5\}$ corresponding to different attack or normal states. The data attributes are represented as follows:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^{18},\ y_i \in \{1, \ldots, 5\},\ i = 1, \ldots, N\} \tag{1}$$

where N denotes the total number of samples. The dataset was partitioned into training and testing subsets with an 80:20 ratio:

$$\mathcal{D}_{train} \cup \mathcal{D}_{test} = \mathcal{D}, \quad \mathcal{D}_{train} \cap \mathcal{D}_{test} = \emptyset \tag{2}$$

Algorithm 1 Proposed Intrusion Detection Framework

Require: Dataset $\mathcal{D}$, learning rate η, epochs E, batch size B
Ensure: $\Theta^\star$
Split $\mathcal{D} \to (\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}})$
Apply Min–Max normalization; rank features by χ^2
Reshape inputs to $\mathbf{X} \in \mathbb{R}^{k \times 1}$
for $e = 1$ to E **do**
  Forward $\rightarrow$**:**
  Apply 1D convolutions with kernel sizes $\{3, 5\}$ to $\mathbf{X}$ to obtain $\mathbf{C}_i$
  Concatenate $[\mathbf{C}_i]_i$, apply batch normalization and dropout $\to \tilde{\mathbf{C}}$
  Compute spatial attention $\mathbf{A}_s$ on $\tilde{\mathbf{C}}$, temporal attention $\mathbf{A}_t$ on $\mathbf{A}_s$
  Build graph features $\mathbf{H}$ via GCN($\mathbf{A}_s$, $\mathbf{A}_t$); obtain $\mathbf{h}$ via BiLSTM($\mathbf{H}$)
  Compute logits $\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}\mathbf{h} + \mathbf{b})$
  Backward $\Leftarrow$**:**
  Compute gradients $\nabla_\Theta \mathcal{L}$ and update $\Theta \leftarrow \Theta - \eta \nabla_\Theta \mathcal{L}$ (Adam)
  Validate on $\mathcal{D}_{\text{val}}$ and save best checkpoint
end for
Test on $\mathcal{D}_{\text{test}}$; report accuracy, precision, recall, and F1

The following features are the part of dataset as described in Table 2:

 doi:10.20944/preprints202511.1423.v1

Table 2. Feature description of the WSN-DS dataset.

Feature symbol	Description	Feature symbol	Description
id	A unique identifier assigned to each sensor node; distinguishes nodes across rounds and stages.	Time	Current simulation time of the node representing its temporal position in the network.
Is_CH	Binary flag indicating whether a node is a cluster head (1) or a normal node (0).	who_CH	Identifier of the cluster head associated with the node in the current round.
Dist_To_CH	Distance between the node and its respective cluster head, calculated per round.	ADV_S	Number of advertise messages broadcast by cluster heads to surrounding nodes.
ADV_R	Number of advertise messages received by a node from nearby cluster heads.	JOIN_S	Number of join request messages sent by nodes to cluster heads for cluster formation.
JOIN_R	Number of join request messages received by cluster heads from their member nodes.	SCH_S	Number of TDMA schedule broadcast messages sent by cluster heads to nodes.
SCH_R	Number of TDMA schedule messages received from cluster heads by the nodes.	Rank	The order or rank of a node within the TDMA schedule during communication.
DATA_S	Number of data packets sent from a sensor node to its cluster head.	DATA_R	Number of data packets received by the cluster head from its sensor nodes.
Data_Sent_To_BS	Number of data packets transmitted from the cluster head to the base station.	dist_CH_To_BS	Distance between the cluster head and the base station used for energy computation.
send_code	Cluster sending code identifying the transmitting node within its cluster.	Expanded_Energy	Amount of energy consumed by the node during the previous communication round.
Attack_type	Target variable representing the attack category with five classes: Blackhole, Grayhole, Flooding, TDMA, and Normal.	–	–

2.2. Design Framework

The proposed framework entails multi-scale convolutional filters, attention-based fusions, and BiLSTM for the extraction of spatio-temporal dependencies in the WSN traffic. The overall structure of the proposed IDS framework has been presented in Algorithm 1.

 doi:10.20944/preprints202511.1423.v1

2.3. Data Preprocessing

To allow numerical stability and optimal convergence, the features have been subjected to normalization by using mix-max scaling as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \in [0, 1] \tag{3}$$

Feature selection was achieved using the χ^2 statistical test. For each feature f_j, the relevance score was computed as:

$$\chi^2(f_j) = \sum_{i=1}^{m} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{4}$$

where O_{ij} and E_{ij} denote observed and expected frequencies across class distributions. The top-16 features with highest χ^2 scores were retained:

$$\mathbf{X}' = \text{SelectKBest}_{\chi^2, k=16}(\mathbf{X}) \tag{5}$$

Additionally, label encoding was applied to the target variable, which was categorical in nature and therefore unsuitable for direct use in ML algorithms. Label encoding is a common preprocessing technique that transforms categorical labels into numerical representations, making them more suitable for algorithmic processing—particularly when the target classes are limited and discrete. The WSN-DS dataset includes five target categories: *Flooding*, *TDMA*, *Grayhole*, *Blackhole*, and *Normal*. Each class was assigned a unique numerical identifier, as summarized in Table 3, to ensure compatibility with supervised learning models.

Table 3. Label encoding used in the proposed method.

Class	Label
Blackhole	0
Flooding	1
Grayhole	2
Normal	3
TDMA	4

2.4. Feature Engineering

Feature transformation for temporal learning was performed via 3D reshaping:

$$\mathbf{X}'' \in \mathbb{R}^{n \times k \times 1}, \quad \text{where } k = 16 \tag{6}$$

This embedding enables convolutional and recurrent layers to explore the localized dependencies.

2.5. Model Design

The proposed hybrid framework $\mathcal{M}(\Theta)$ constitutes a multi-branch. It is a hierarchically coupled architecture designed to capture spatio–temporal–spectral correlations. Θ denotes the complete set of learnable parameters of the model $\mathcal{M}$, including convolutional kernels, recurrent weights, normalization matrices, and bias terms. The model comprises four principal computational entities: a *Multi-Scale Convolutional Block* for local feature extraction, an *Attention Fusion Layer* for dynamic context weighting, a *Graph Convolutional Module* for structural regularization, and a *Bidirectional LSTM with Contextual Attention* for temporal propagation modeling. These integrated modules help in collectively representing the learnable tensors that involve convolutional kernels, recurrent weights, normalization matrices, and bias offsets. Overall, the model design has been depicted in Figure 1.

 doi:10.20944/preprints202511.1423.v1

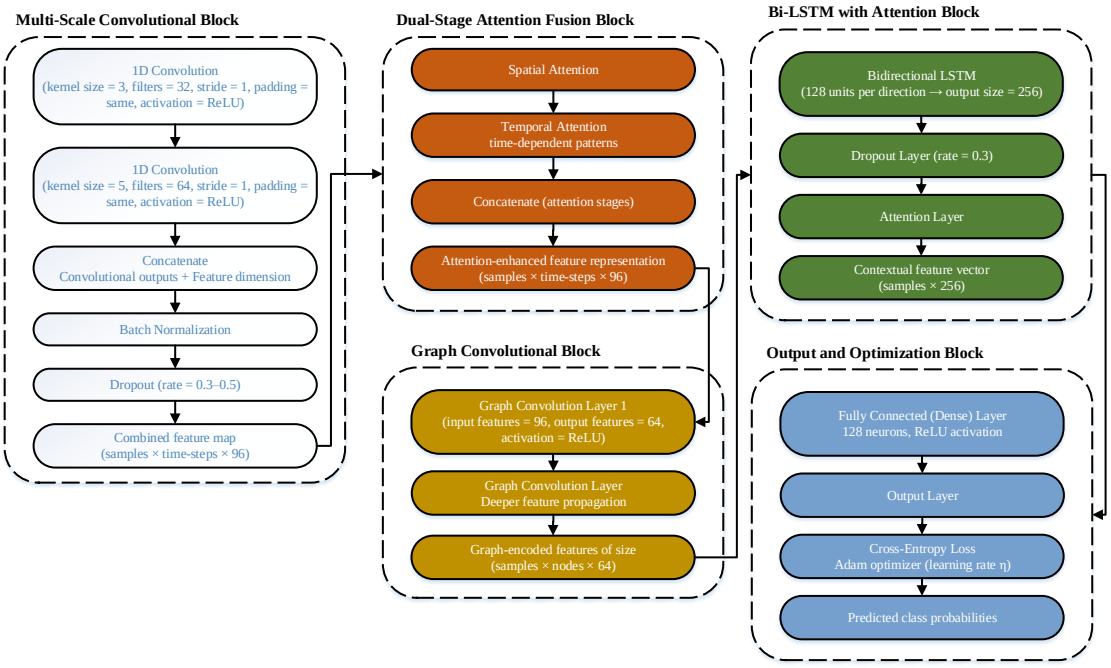

Figure 1. Model architecture of the proposed framework

2.5.1. Multi-Scale Convolutional Block

Given a normalized sequence tensor $\mathbf{X} \in \mathbb{R}^{k\times 1}$ that represents the compacted feature manifold. The multi-scale convolutional extractor performs convolutions at multiple receptive field scales for capturing heterogeneous spatial dependencies:

$$\mathbf{F}_1 = \sigma(\mathcal{N}_1(\mathbf{W}_1 *_3 \mathbf{X} + \mathbf{b}_1)), \tag{7}$$

$$\mathbf{F}_2 = \sigma(\mathcal{N}_2(\mathbf{W}_2 *_5 \mathbf{X} + \mathbf{b}_2)), \tag{8}$$

$$\mathbf{F}_3 = \sigma(\mathcal{N}_3(\mathbf{W}_3 *_7 \mathbf{X} + \mathbf{b}_3)), \tag{9}$$

where $*_n$ denotes 1-D convolution with kernel size n, $\mathcal{N}_i(\cdot)$ indicates batch normalization, and $\sigma(\cdot)$ is the ReLU activation. The multi-scale responses are concatenated into a composite feature tensor:

$$\mathbf{F}_{ms} = [\mathbf{F}_1 \,\|\, \mathbf{F}_2 \,\|\, \mathbf{F}_3] \in \mathbb{R}^{k\times d_{ms}}, \tag{10}$$

where d_{ms} denotes the total concatenated dimensionality. A dropout mapping $\mathcal{D}_p$ with stochastic rate $p = 0.3$ is subsequently applied to prevent co-adaptation:

$$\tilde{\mathbf{F}}_{ms} = \mathcal{D}_p(\mathbf{F}_{ms}). \tag{11}$$

This operation enforces robustness to local perturbations. In addition, it preserves gradient stability across convolutional depths.

2.5.2. Dual-Stage Attention Fusion

The fused features $\tilde{\mathbf{F}}_{ms}$ are passed into a dual-stage attention mechanism. These have been designed to disentangle spatial and temporal significance within the feature domain. Let $\mathbf{Q}_s, \mathbf{K}_s, \mathbf{V}_s \in \mathbb{R}^{T\times d_k}$ represent the query, key, and value embeddings for the spatial attention subspace:

$$\mathbf{A}_s = \text{softmax}\left(\frac{\mathbf{Q}_s\mathbf{K}_s^T}{\sqrt{d_k}} + \mathbf{M}_s\right)\mathbf{V}_s, \tag{12}$$

 doi:10.20944/preprints202511.1423.v1

where $\mathbf{M}_s$ is a learned bias mask regulating sparsity across nodes. The temporal refinement stage analogously computes:

$$\mathbf{A}_t = \text{softmax}\left(\frac{(\mathbf{Q}_t\mathbf{K}_t^T)\mathbf{W}_\tau + \mathbf{B}_\tau}{\sqrt{d_k}}\right)\mathbf{V}_t, \tag{13}$$

where $\mathbf{W}_\tau$ introduces a learnable transformation capturing cross-time contextual drift. The joint fused representation is then expressed as:

$$\mathbf{A}_{fused} = \alpha\mathbf{A}_s + (1-\alpha)\mathbf{A}_t + \lambda\,(\mathbf{A}_s \odot \mathbf{A}_t), \tag{14}$$

where α and λ are trainable coupling coefficients, and $\odot$ denotes element-wise interaction. This composite fusion reinforces both spatially localized and temporally evolving intrusion cues.

2.5.3. Graph Convolutional Regularization

To embed topological priors of the WSN, an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is constructed. This encodes communication reachability among sensor nodes. The spectral graph convolution for layer l is formulated as:

$$\mathbf{H}^{(l+1)} = \xi\Big(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)} + \gamma\mathbf{H}^{(l)}\Big), \tag{15}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ ensures self-loops, $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ is the degree matrix, $\xi(\cdot)$ is a nonlinear mapping (ReLU), and γ is a residual stability factor.

This operation integrates both direct communication correlations and latent dependencies that are inferred using the higher-order neighborhoods.

2.5.4. Bidirectional LSTM with Contextual Attention

To capture temporal recurrency and bidirectional dependencies, the model employs forward and backward LSTMs defined as:

$$\overrightarrow{\mathbf{h}}_t = f_{\text{LSTM}}\Big(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}; \Theta_f\Big), \tag{16}$$

$$\overleftarrow{\mathbf{h}}_t = f_{\text{LSTM}}\Big(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}; \Theta_b\Big), \tag{17}$$

yielding the contextual embedding. Here, Θ_f and Θ_b represent the sets of learnable parameters (weights and biases) for the forward and backward LSTM networks, respectively.

$$\mathbf{H}_t = [\overrightarrow{\mathbf{h}}_t \,\|\, \overleftarrow{\mathbf{h}}_t]. \tag{18}$$

An adaptive attention mechanism refines $\mathbf{H}_t$ into a context-weighted summary vector:

$$\mathbf{h}_{att} = \sum_{t=1}^{T}\alpha_t\mathbf{H}_t, \quad \alpha_t = \frac{\exp(\mathbf{u}_t^\top\mathbf{w}_a)}{\sum_{i=1}^{T}\exp(\mathbf{u}_i^\top\mathbf{w}_a)}, \quad \mathbf{u}_t = \tanh(\mathbf{W}_u\mathbf{H}_t + \mathbf{b}_u), \tag{19}$$

where $\mathbf{w}_a$ serves as the attention query vector, optimizing temporal salience through soft alignment.

2.5.5. Hierarchical Aggregation and Output Projection

The contextual embedding $\mathbf{h}_{att}$ is aggregated through a hierarchical fusion of global average and maximum pooling:

$$\mathbf{z} = \beta_1 \cdot \frac{1}{T}\sum_{t=1}^{T}\mathbf{h}_{att,t} + \beta_2 \cdot \max_t(\mathbf{h}_{att,t}), \tag{20}$$

where β_1 and β_2 are learned weighting scalars enforcing balanced statistical and extremal emphasis. The resultant descriptor $\mathbf{z}$ traverses two nonlinear dense transformations under L_2 regularization:

$$\mathbf{z}' = \phi(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \rho\,\phi(\mathbf{W}_2\mathbf{z} + \mathbf{b}_2), \tag{21}$$

 doi:10.20944/preprints202511.1423.v1

where ϕ denotes ReLU activation and ρ acts as a dense fusion coefficient. Finally, the class posterior distribution over intrusion categories is modeled as:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_o \mathbf{z}' + \mathbf{b}_o), \tag{22}$$

with optimization governed by categorical cross-entropy loss:

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log \hat{y}_{i,c}. \tag{23}$$

This hierarchical deep representation ensures that $\mathcal{M}(\Theta)$ can capture localized transient anomalies. It further encodes persistent cross-node intrusion dynamics characteristic of WSN attack behavior.

2.6. Evaluation and Simulation

The model was implemented in TensorFlow 2.15 and trained on a single NVIDIA GPU with CUDA acceleration. Hyperparameters and simulation settings are summarized in Table 4.

Table 4. Simulation parameters used in the proposed intrusion detection framework.

Parameter	Value
Learning rate	1×10^{-4}
Batch size	128
Epochs	30
Optimizer	Adam
Regularization	$L_2(\lambda = 0.001)$
Dropout rate	0.25–0.30
Feature dimension	16
Hidden units (BiLSTM)	64 per direction

2.6.1. Evaluation Metrics

Performance evaluaiton has been carried out using Accuracy (Acc), Precision (P), Recall (R), and F1-score (F_1):

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{24}$$

$$P = \frac{TP}{TP + FP} \tag{25}$$

$$R = \frac{TP}{TP + FN} \tag{26}$$

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{27}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Confusion matrices and learning curves were used to further assess classification stability and convergence.

3. Results

The performance analysis of the proposed IDS in WSN has been carried out under multi-tier analysis. The system has been implemented using Python and TensorFlow while using Keras backends. The dataset has been divided into three sub-sets (training, validation, and testing). The model has been trained for 30 epochs using the Adam optimizer with a learning rate of $\eta = 10^{-3}$. It has been categorized by using a categorical cross-entropy loss function.

 doi:10.20944/preprints202511.1423.v1

3.1. Training and Validation Performance

The model convergence behavior has been depicted in Fig. 2. These depict the evolution of both the accuracy and loss across epochs. The training accuracy has been found to consistently increase with the increasing epochs. The stabilization is approximately 98% after epoch 10. This indicated that a fast convergence has been attained along with strong generalization. The validation accuracy also follows the same trajectory. This suggests that the overfitting has been successfully mitigated through the inclusion of dropout and batch normalization layers.

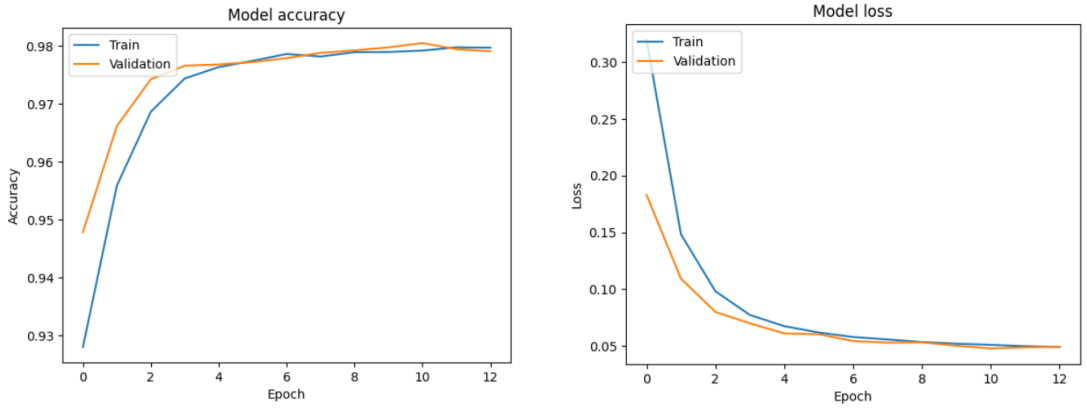

Figure 2. The model convergence behavior: (a) Training and validation accuracy curves; (b) Training and validation loss convergence.

The model's stability can be characterized by the loss differential:

$$\Delta\mathcal{L}(t) = |\mathcal{L}_{train}(t) - \mathcal{L}_{val}(t)|, \tag{28}$$

This asymptotically approaches zero as $t \to T_{final}$ and leads to confirming the convergence without oscillation and divergence. This results from the adaptive gradient dynamics, which are inherent in the Adam optimizer, represented as follows:

$$\Theta_{t+1} = \Theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t, \tag{29}$$

where $\hat{m}_t$ and $\hat{v}_t$ denote bias-corrected first and second moment estimates, respectively.

3.2. Confusion Matrix Analysis

To further analyze the classification performance of the model across the multiple attack classes, outcomes have been presented in the form of a confusion matrix. The proposed CNN-Attention-BiLSTM hybrid model has been found to exhibit strong diagonal dominance as presented in Fig. 3. The model has been found to attain accurate multi-class discrimination. The normal and DoS categories in particular have attained a near-perfect classification. This led to minimal confusion between normal network behavior and four distinct types of DoS attacks: Grayhole, Blackhole, TDMA, and Flooding.

The overall precision (P), recall (R), and F_1-score were computed as:

$$P = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)} = 0.9842, \tag{30}$$

$$R = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)} = 0.9791, \tag{31}$$

$$F_1 = \frac{2PR}{P + R} = 0.9791, \tag{32}$$

which collectively demonstrate the superior discriminative capacity of the model.

 doi:10.20944/preprints202511.1423.v1

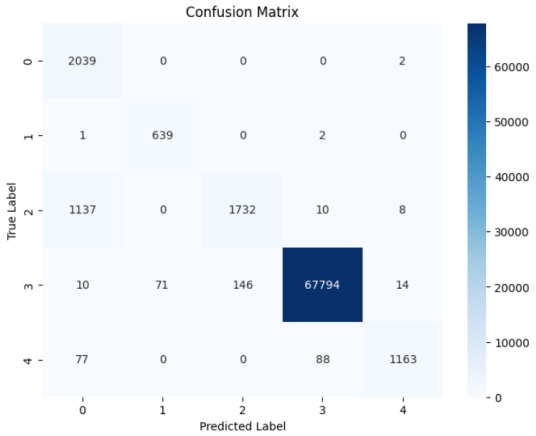

Figure 3. Confusion matrix of the proposed intrusion detection model.

3.3. Model Interpretability and Structural Visualization

The network architecture has been depicted in Fig. 4. It entails a multiscale convolutional feature extraction with multi-head attention fusion and BiLSTM encoding for the temporal dependency modeling. The total number of trainable parameters is approximately 96,735, with only 256 non-trainable parameters. This ensured a lightweight deployment within constrained WSN environments.

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 16, 1)	0	-
conv1d (Conv1D)	(None, 16, 32)	128	input_layer[0][0]
conv1d_1 (Conv1D)	(None, 16, 32)	192	input_layer[0][0]
concatenate (Concatenate)	(None, 16, 64)	0	conv1d[0][0], conv1d_1[0][0]
batch_normalization (BatchNormalizatio…	(None, 16, 64)	256	concatenate[0][0]
dropout (Dropout)	(None, 16, 64)	0	batch_normalizat…
attention (Attention)	(None, 16, 64)	0	dropout[0][0], dropout[0][0]
attention_1 (Attention)	(None, 16, 64)	0	attention[0][0], attention[0][0]
add (Add)	(None, 16, 64)	0	attention[0][0], attention_1[0][0]
conv1d_2 (Conv1D)	(None, 16, 64)	12,352	add[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 16, 64)	256	conv1d_2[0][0]
dropout_1 (Dropout)	(None, 16, 64)	0	batch_normalizat…
bidirectional (Bidirectional)	(None, 16, 128)	66,048	dropout_1[0][0]
attention_2 (Attention)	(None, 16, 128)	0	bidirectional[0]… bidirectional[0]…
global_average_poo… (GlobalAveragePool…	(None, 128)	0	attention_2[0][0]
dense (Dense)	(None, 90)	11,610	global_average_p…
dropout_2 (Dropout)	(None, 90)	0	dense[0][0]
dense_1 (Dense)	(None, 64)	5,824	dropout_2[0][0]
dropout_3 (Dropout)	(None, 64)	0	dense_1[0][0]
dense_2 (Dense)	(None, 5)	325	dropout_3[0][0]

Total params: 96,991 (378.87 KB)
Trainable params: 96,735 (377.87 KB)
Non-trainable params: 256 (1.00 KB)

Figure 4. Architectural summary of the proposed model.

 doi:10.20944/preprints202511.1423.v1

3.4. Learning Dynamics and Validation Logs

Figure 5 presents the epoch-wise training logs. These depict a consistent improvement in accuracy and reduction in loss. Each epoch's validation loss ($\mathcal{L}_{val}$) was evaluated, and the best-performing model was checkpointed according to:

$$\mathcal{L}_{val}^{best} = \min_{t}\{\mathcal{L}_{val}(t)\}. \tag{33}$$

```
2341/2342 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step - accuracy: 0.9774 - loss: 0.0622
Epoch 6: val_loss improved from 0.06090 to 0.06022, saving model to model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend usi
2342/2342 ━━━━━━━━━━━━━━━━━━━━ 189s 68ms/step - accuracy: 0.9774 - loss: 0.0622 - val_accuracy: 0.9772 - val_loss: 0.0602
Epoch 7/30
2341/2342 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step - accuracy: 0.9783 - loss: 0.0585
Epoch 7: val_loss improved from 0.06022 to 0.05417, saving model to model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend usi
2342/2342 ━━━━━━━━━━━━━━━━━━━━ 199s 66ms/step - accuracy: 0.9783 - loss: 0.0585 - val_accuracy: 0.9779 - val_loss: 0.0542
Epoch 8/30
2341/2342 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step - accuracy: 0.9784 - loss: 0.0560
Epoch 8: val_loss improved from 0.05417 to 0.05272, saving model to model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend usi
2342/2342 ━━━━━━━━━━━━━━━━━━━━ 164s 70ms/step - accuracy: 0.9784 - loss: 0.0560 - val_accuracy: 0.9788 - val_loss: 0.0527
Epoch 9/30
2341/2342 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step - accuracy: 0.9792 - loss: 0.0535
Epoch 9: val_loss did not improve from 0.05272
2342/2342 ━━━━━━━━━━━━━━━━━━━━ 194s 67ms/step - accuracy: 0.9792 - loss: 0.0535 - val_accuracy: 0.9792 - val_loss: 0.0530
Epoch 10/30
2342/2342 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step - accuracy: 0.9789 - loss: 0.0522
Epoch 10: val_loss improved from 0.05272 to 0.05008, saving model to model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend usi
2342/2342 ━━━━━━━━━━━━━━━━━━━━ 201s 66ms/step - accuracy: 0.9789 - loss: 0.0522 - val_accuracy: 0.9798 - val_loss: 0.0501
Epoch 11/30
2341/2342 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step - accuracy: 0.9796 - loss: 0.0506
Epoch 11: val_loss improved from 0.05008 to 0.04767, saving model to model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend usi
2342/2342 ━━━━━━━━━━━━━━━━━━━━ 205s 67ms/step - accuracy: 0.9796 - loss: 0.0506 - val_accuracy: 0.9805 - val_loss: 0.0477
Epoch 12/30
2341/2342 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step - accuracy: 0.9801 - loss: 0.0491
Epoch 12: val_loss did not improve from 0.04767
2342/2342 ━━━━━━━━━━━━━━━━━━━━ 214s 72ms/step - accuracy: 0.9801 - loss: 0.0491 - val_accuracy: 0.9794 - val_loss: 0.0487
Epoch 13/30
2341/2342 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - accuracy: 0.9796 - loss: 0.0491
Epoch 13: val_loss did not improve from 0.04767
2342/2342 ━━━━━━━━━━━━━━━━━━━━ 205s 74ms/step - accuracy: 0.9796 - loss: 0.0491 - val_accuracy: 0.9791 - val_loss: 0.0492
```

Figure 5. Epoch-wise model training logs showing validation checkpoints.

3.5. Comparative Evaluation

The proposed model has been validated against the benchmark models and classifiers that involve CNN, CNN+ recurrent neural network (RNN), and naïve nayes. The summary of the comparison has been presented in Table 5. The model has been found to attain an overall accuracy of 98.0%. The accuracy is higher compared to the conventional approaches by up to 2.2%. In addition, the inclusion of the attention mechanism further improved the interpretability. This allows offering insights into the neuron activation relevance during detection.

Table 5. Summary of comparative results for intrusion detection performance.

Model	XAI	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	No	97.00	83.60	82.60	82.00
CNN + RNN	No	97.04	98.79	96.48	96.86
Naive Bayes	No	95.82	96.80	95.40	–
Proposed Model	**Yes**	**98.00**	**98.42**	**97.91**	**97.91**

The CNN model recorded 97.0% accuracy, demonstrating the ability of convolutional layers to extract important spatial features. However, it presented lower precision and recall of 83.60% and 82.60% respectively, indicating difficulty in identifying several attack types and resulting in a relatively high false positive rate. The CNN + RNN gained better recall of 96.48%, and F1-score with 96.86% due to the ability of learn patterns over time, while the overall accuracy was similar to the CNN model. This means that using only simple time modeling is not enough to express complex and nonlinear behaviors observed in WSNs.

On the other hand, the Naïve Bayes classifier demonstrated the lowest accuracy at 95.82% compared to other baseline models. This finding shows the limitation to cope with non-linear, and high-dimensional feature interactions which appear in intrusion data.

 doi:10.20944/preprints202511.1423.v1

The proposed model outperforms baseline models across all metrics, reaching 98.42% precision, 97.91% recall, and a 97.91% F1-score. This indicates the model is more effective at detecting attacks while minimizing false positives, which matters in real-time WSN uses. The dual-attention mechanism assists the model in targeting key features, boosting both robustness and stability.

Unlike other baseline models, the proposed model offers interpretability based on the XAI feature. Attention layers reveal the spatial or temporal regions that most strongly influence classification decisions. Interpreting decisions helps network administrators to understand the model's reasoning and identify vulnerable parts of the network, unlike black-box models that only provide prediction results. The proposed approach provides both interpretable reasoning and high-performance detection, which is beneficial for WSN security monitoring in the real world.

3.6. Discussion

Overall, the proposed framework has been found to outperform the baseline models in terms of accuracy and interpretability. In addition, the contextual attention mechanisms have allowed improved discrimination between the overlapping attack signatures. The mathematical baseline for this improvement is related to non-linear fusion of multi-scale convolutional and temporal embeddings:

$$\mathbf{z}_{final} = \phi(\mathbf{W}_a[\mathbf{F}_{ms} \oplus \mathbf{h}_{bi}] + \mathbf{b}_a), \tag{34}$$

where $\oplus$ denotes concatenation and ϕ represents a non-linear mapping. The end-to-end architecture thus achieves robust generalization, scalability, and real-time inference capability. This makes it suitable for deployment in resource-constrained WSN environments.

The integration of multi-scale convolutional features and bidirectional temporal embeddings enables the framework to create a coherent latent space that simultaneously captures both local spatial patterns and temporal dynamics of WSN traffic. This combination of features from both domains enhances the model's ability to detect subtle differences in behavior between normal and malicious traffic. The linear transformation, characterized by W_a and the bias term b_a, maps these features into a more differentiated subspace, while the non-linear activation function ϕ facilitates higher-order interactions among features, allowing the model to capture complex and non-linear attack behaviors common in WSN environments. The proposed approach not only improves the representative power of the learned embeddings but also enhances the differentiation between classes within the latent space, as demonstrated by the improved clustering of attack categories during the evaluation process.

Compared with conventional CNN or LSTM-based models, the proposed hybrid framework offers a significant advantage in handling the complex characteristics of WSNs. Traditional CNNs can extract local spatial features but often struggle with irregular topologies of WSN. LSTM models excel at capturing temporal patterns but overlook the spatial relationships among network nodes. Introducing the GCN addresses these limitations by learning the connections between nodes and how anomalies propagate across the network, which is crucial for identifying distributed or coordinated attacks. Furthermore, the BiLSTM component of the model improves temporal learning by analyzing data in both forward and backward directions. Additionally, the attention fusion mechanism emphasizes the most significant spatial, structural, and temporal features. As a result, the proposed framework provides reliable and precise intrusion detection, decreases false alarms, and adapts effectively to changing network conditions.

4. Conclusion

The research presents an advanced hybrid deep learning framework for intrusion detection in the WSNs. The model integrated *multi-scale convolutional blocks*, *attention fusion layers*, *graph convolutional reasoning*, and *BiLSTM* components. The proposed framework helped in effectively capturing both spatial and temporal dependencies in sensor network traffic. The evaluation of the model has been carried out on the WSN-DS dataset. The model has been found to attain superior detection capability and has shown the ability to distinguish against diverse attacks, including Grayhole, Blackhole, TDMA,

 doi:10.20944/preprints202511.1423.v1

and Flooding. The study not only attained high detection accuracy but also maintained lightweight computational complexity that is suitable for real-time WSN environments. Compared to the existing baseline models, the proposed model attained enhanced generalization, reduced false alarms, and improved feature interpretability through its attention-driven design. Overall, the model bridged the gap between a high-performance IDS framework and practical WSN deployment. These offer scalable, energy-efficient, and topology-aware detection solutions. In the future, the following can be explored: (i) adaptive federated implementations for decentralized WSN nodes, (ii) self-evolving detection modules leveraging online learning to handle emerging attack patterns, and (iii) explainable visual analytics to strengthen trust and interpretability in mission-critical applications.

References

1. Puccinelli, D.; Haenggi, M. Wireless sensor networks: applications and challenges of ubiquitous sensing. *IEEE Circuits and Systems Magazine* **2005**, *5*, 19–31. https://doi.org/10.1109/MCAS.2005.1507522.
2. Borges, L.M.; Velez, F.J.; Lebres, A.S. Survey on the Characterization and Classification of Wireless Sensor Network Applications. *IEEE Communications Surveys & Tutorials* **2014**, *16*, 1860–1890. https://doi.org/10.1109/COMST.2014.2320073.
3. Bunterngchit, C.; Pornchaivivat, S.; Bunterngchit, Y. Productivity Improvement by Retrofit Concept in Auto Parts Factories. In Proceedings of the 2019 8th International Conference on Industrial Technology and Management (ICITM), 2019, pp. 122–126. https://doi.org/10.1109/ICITM.2019.8710655.
4. Othman, M.F.; Shazali, K. Wireless Sensor Network Applications: A Study in Environment Monitoring System. *Procedia Engineering* **2012**, *41*, 1204–1210. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), https://doi.org/10.1016/j.proeng.2012.07.302.
5. Bunterngchit, C.; Baniata, L.H.; Baniata, M.H.; ALDabbas, A.; Khair, M.A.; Chearanai, T.; Kang, S. GACL-Net: Hybrid Deep Learning Framework for Accurate Motor Imagery Classification in Stroke Rehabilitation. *Computers, Materials & Continua* **2025**, *83*, 517–536. https://doi.org/10.32604/cmc.2025.060368.
6. Chhaya, L.; Sharma, P.; Bhagwatikar, G.; Kumar, A. Wireless Sensor Network Based Smart Grid Communications: Cyber Attacks, Intrusion Detection System and Topology Control. *Electronics* **2017**, *6*. https://doi.org/10.3390/electronics6010005.
7. Prodanović, R.; Rančić, D.; Vulić, I.; Zorić, N.; Bogićević, D.; Ostojić, G.; Sarang, S.; Stankovski, S. Wireless Sensor Network in Agriculture: Model of Cyber Security. *Sensors* **2020**, *20*. https://doi.org/10.3390/s20236747.
8. Dritsas, E.; Trigka, M. A Survey on Cybersecurity in IoT. *Future Internet* **2025**, *17*. https://doi.org/10.3390/fi17010030.
9. Thapa, N.; Liu, Z.; KC, D.B.; Gokaraju, B.; Roy, K. Comparison of Machine Learning and Deep Learning Models for Network Intrusion Detection Systems. *Future Internet* **2020**, *12*. https://doi.org/10.3390/fi12100167.
10. Biermann, E.; Cloete, E.; Venter, L. A comparison of Intrusion Detection systems. *Computers & Security* **2001**, *20*, 676–683. https://doi.org/10.1016/S0167-4048(01)00806-9.
11. Abdulganiyu, O.H.; Ait Tchakoucht, T.; Saheed, Y.K. A systematic literature review for network intrusion detection system (IDS). *International journal of information security* **2023**, *22*, 1125–1162. https://doi.org/10.1007/s10207-023-00682-2.
12. Houda, Z.A.E.; Naboulsi, D.; Kaddoum, G. A Privacy-Preserving Collaborative Jamming Attacks Detection Framework Using Federated Learning. *IEEE Internet of Things Journal* **2024**, *11*, 12153–12164. https://doi.org/10.1109/JIOT.2023.3333870.
13. Jeyakumar, S.R.; Rahman, M.Z.U.; Sinha, D.K.; Kumar, P.R.; Vimal, V.; Singh, K.U.; Syamsundararao, T.; Kumar, J.N.V.R.S.; Balajee, J. 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* **2025**, *71*, 273–280. https://doi.org/10.1109/TCE.2024.3442015.
14. Zhou, H.; Zou, H.; Zhou, P.; Shen, Y.; Li, D.; Li, W. CBCTL-IDS: A Transfer Learning-Based Intrusion Detection System Optimized With the Black Kite Algorithm for IoT-Enabled Smart Agriculture. *IEEE Access* **2025**, *13*, 46601–46615. https://doi.org/10.1109/ACCESS.2025.3550800.
15. Halbouni, A.; Gunawan, T.S.; Habaebi, M.H.; Halbouni, M.; Kartiwi, M.; Ahmad, R. CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System. *IEEE Access* **2022**, *10*, 99837–99849. https://doi.org/10.1109/ACCESS.2022.3206425.

 doi:10.20944/preprints202511.1423.v1

16. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334.
17. Birahim, S.A.; Paul, A.; Rahman, F.; Islam, Y.; Roy, T.; Asif Hasan, M.; Haque, F.; Chowdhury, M.E.H. Intrusion Detection for Wireless Sensor Network Using Particle Swarm Optimization Based Explainable Ensemble Machine Learning Approach. *IEEE Access* **2025**, *13*, 13711–13730. https://doi.org/10.1109/ACCESS.2025.3528341.
18. Hakami, H.; Faheem, M.; Bashir Ahmad, M. Machine Learning Techniques for Enhanced Intrusion Detection in IoT Security. *IEEE Access* **2025**, *13*, 31140–31158. https://doi.org/10.1109/ACCESS.2025.3542227.
19. Alzahrani, A. Novel Approach for Intrusion Detection Attacks on Small Drones Using ConvLSTM Model. *IEEE Access* **2024**, *12*, 149238–149253. https://doi.org/10.1109/ACCESS.2024.3471806.
20. Alruwaili, F.F.; Asiri, M.M.; Alrayes, F.S.; Aljameel, S.S.; Salama, A.S.; Hilal, A.M. Red Kite Optimization Algorithm With Average Ensemble Model for Intrusion Detection for Secure IoT. *IEEE Access* **2023**, *11*, 131749–131758. https://doi.org/10.1109/ACCESS.2023.3335124.
21. Atitallah, S.B.; Driss, M.; Boulila, W.; Koubaa, A. Securing Industrial IoT Environments: A Fuzzy Graph Attention Network for Robust Intrusion Detection. *IEEE Open Journal of the Computer Society* **2025**, *6*, 1065–1076. https://doi.org/10.1109/OJCS.2025.3587486.
22. Saleh, H.M.; Marouane, H.; Fakhfakh, A. Stochastic Gradient Descent Intrusions Detection for Wireless Sensor Network Attack Detection System Using Machine Learning. *IEEE Access* **2024**, *12*, 3825–3836. https://doi.org/10.1109/ACCESS.2023.3349248.
23. Jiang, L.; Gu, H.; Xie, L.; Yang, H.; Na, Z. ST-IAOA-XGBoost: An Efficient Data-Balanced Intrusion Detection Method for WSN. *IEEE Sensors Journal* **2025**, *25*, 1768–1783. https://doi.org/10.1109/JSEN.2024.3489623.
24. Almomani, I.; Al-Kasasbeh, B.; AL-Akhras, M. WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks. *Journal of Sensors* **2016**, *2016*, 4731953. https://doi.org/https://doi.org/10.1155/2016/4731953.
25. Marriwala, N.; Rathee, P. An approach to increase the wireless sensor network lifetime. In Proceedings of the 2012 World Congress on Information and Communication Technologies, 2012, pp. 495–499. https://doi.org/10.1109/WICT.2012.6409128.