JJEE

Volume 12 | Number 2 | June 2026 | Pages 345-371

Jordan Journal of Electrical Engineering

ISSN (print): 2409-9600, ISSN (online): 2409-9619

Homepage: jjee.ttu.edu.jo

An SDN-Oriented Fuzzy CNN-LSTM Framework for Intrusion Detection in Wireless Sensor Networks

Zaid J. Al-araji[1*], Balqees Talal Hasan[2], Abdulmajeed Ahmed[3], Huthaifa Luay[4], Hussein M. Farhood[5]

[1, 3, 4] Department of Computer Network and Internet, College of Information Technology, Ninevah University, Mosul, Iarq

E-mail: zaid.jasim@uoninevah.edu.iq

[2] Department of Artificial Intelligence, College of Information Technology, Ninevah University, Mosul, Iarq

[5] Department of Computer Engineering, College of Engineering, Mustansiriyah University, Baghdad, Iraq

Received: Dec 12, 2025 *Revised: Feb 06, 2026* *Accepted: Mar 05, 2026* *Available online: Jun 15, 2026*

***Abstract*—** Wireless Sensor Networks (WSNs) are used in smart and mission-critical applications because their wireless technology enables remote monitoring. The distributed design of WSNs combined with their limited computing power creates a situation where they become highly susceptible to advanced cyber threats. The research presents an SDN-oriented hybrid intrusion detection system which uses Fuzzy CNN-LSTM architecture at network border points to solve these technical problems. The proposed model uses a convolutional neural network (CNN) to extract spatial features and a long short-term memory (LSTM) network to model temporal behavior while its fuzzy inference layer improves prediction accuracy by handling softmax output uncertainty to decrease false alarm rates. Edge nodes conduct sensor node traffic data processing and analysis, while SDN controllers provide centralized network monitoring and defense capabilities through flow control and traffic redirection and distribution balance operations. The framework uses the WSN-DS and WSNBFSF datasets to test its performance under binary classification between normal and attack modes. The proposed method achieved a detection accuracy of 99.67% and a precision value of 98.33% and a recall rate of 98.16% and an MCC value of 98.07% when tested with WSN-DS. Our model achieved 99.2% accuracy and 97.75% precision and 96.9% recall and 96.85% MCC on the WSNBFSF dataset while outperforming standalone CNN and LSTM models which maintained low inference latency for real-time deployment across both datasets. The results demonstrate that deep learning combined with fuzzy uncertainty reasoning and SDN control creates a sustainable solution to detect intrusions across resource-limited WSN environments.

***Keywords*—** Attack detection, WSN, Fuzzy inference system, SDN, Deep learning.

1. INTRODUCTION

The foundation of smart cities is Wireless Sensor Networks (WSNs), which constitute the very genesis of sustainable energy structures that are being developed today [1]. The WSN system consists of multiple lightweight sensor nodes which operate throughout the designated area to gather and transmit information. Sensors first send data to their nearby nodes before transmitting it to the central sink node according to their operational design [2]. Information propagation schemes under these restricted conditions face three main challenges which include energy constraints, data collision problems and severe security vulnerabilities.

The research area aims to improve network performance through different network designs and energy-saving protocols and optimization methods. The process of clustering involves designating particular nodes to function as cluster heads which connect sensor nodes to the sink while achieving energy savings and reducing network congestion and attack

* Corresponding author

DOI: https://doi.org/ 10.5455/jjee.204-1765542864

impacts [3]. The selection of cluster heads requires both energy efficiency and node trustworthiness as critical factors. The nodes that hold the highest power become the primary targets for attacks so they should not assume leadership positions because of their vulnerability to compromise [4].

The combination of load balancers with clustered interactions enables better traffic distribution which leads to longer operational periods for network nodes. The network load distribution across various clusters through multiple path connections prevents both energy exhaustion and system bottlenecks according to Tawfeek et al., (2025) [5]. The system possesses various operational limitations which exist as its fundamental characteristics. The system faces multiple challenges because its nodes have low programmability while their hardware design lacks flexibility and it cannot detect hidden traffic patterns and it remains vulnerable to advanced attack techniques.

The introduction of Software-Defined Networking (SDN) into wireless sensor network environments addresses existing system limitations. SDN develops a new network architecture through its control plane and data plane separation which enables centralized network visibility and network settings control and policy enforcement capabilities [6]. The SDN system enables flexible network operations which can adapt to changing conditions, but it also suffers from two main problems which include security weaknesses against flow-based attacks and difficulties in managing high-volume network traffic. The researchers propose to combine Machine Learning (ML) methods with SDN to achieve smarter network traffic control and better anomaly detection capabilities [7].

Fuzzy logic functions as a practical solution which delivers a reasoning framework to handle uncertain situations while enabling advanced reasoning processes needed to tackle high-uncertainty situations and cases with restricted labeled data which typically act as constraints for machine learning models [8].

The process of detecting intrusions in Wireless Sensor Networks encounters major difficulties because of the fundamental limitations present in those particular environments. First, WSN traffic exhibits strong temporal dependencies and spatial correlations which single-paradigm learning approaches find difficult to model. Second, intrusion detection models experience high false-positive rates when they process noisy and imbalanced and ambiguous traffic patterns which lead to excessive energy use and network performance decline. Existing IDS solutions function as passive classifiers because they do not integrate with their systems yet, which prevents them from executing real-time defense operations and adapting to new attack patterns. Centralized detection systems cause two major problems for WSN deployments in areas with limited resources because they introduce detection delays and they cannot handle increases in system demands.

The study proposes an advanced attack detection system which operates effectively under heavy network loads while protecting WSNs from security threats. The system combines deep learning fuzzy logic edge computing and SDN principles to build its advanced detection framework. The main contributions of this work are:

- A hybrid CNN–LSTM detection model tailored for WSN traffic, where CNN extracts spatial feature patterns, and LSTM captures time-dependent attack behaviour.
- This framework, which reduces dependence on centralised infrastructure, allows real-time data preprocessing and light anomaly detection to take place at the edge nodes to optimise energy efficiency and improve responsiveness.

- The system leverages SDN to orchestrate dynamic routing and centralised control, enabling adaptive traffic management, threat containment, and seamless policy updates.

The novelty of this work, in comparison with existing intrusion detection approaches for Wireless Sensor Networks, can be summarized as follows:

The first research study which combines fuzzy decision refinement with CNN–LSTM for WSN IDS implementation shows that existing CNN–LSTM models for intrusion detection require this research to introduce a new fuzzy inference layer which handles uncertainty in probabilistic outputs, a feature which previous WSN IDS studies did not include. The proposed framework integrates intrusion detection capabilities into the SDN control plane, which allows security personnel to execute immediate threat response measures through flow isolation and rerouting and load-aware defense mechanisms. The new system design for intrusion detection systems now uses edge network points for monitoring to achieve better results through reduced system delays and decreased data transfer needs while keeping existing SDN control functions which previous WSN IDS systems failed to investigate together with their respective monitoring functions. Empirically validated uncertainty-aware design: The fuzzy decision thresholds are empirically justified through probability distribution analysis and sensitivity evaluation, ensuring principled uncertainty handling rather than heuristic rule definition.

The rest of the dissertation proceeds accordingly. Section 2 reviews existing approaches to intrusion detection in WSNs, including deep learning, fuzzy systems, SDN-based solutions, and related security mechanisms. Section 3 introduces the proposed hybrid model of the Fuzzy CNN-LSTM along with the edge-SDN architecture and workflow in data preprocessing. In Section 4, the experimental setup, evaluation metrics, and performance results from the WSN-DS and WSNBFSF datasets are described. Section 5 will discuss the findings and present strengths and weaknesses with respect to the proposed method and the baseline models. Finally, Section 6 will summarize the study and propose future research directions.

2. RELATED WORK

Indeed, the IDSs would be extremely popular going into the future, particularly in countries where the internet of things has become a common phenomenon as opposed to the WSNs on their own, considering all the peculiar problems that resource and heterogeneity constraints pose. Studies have taken a dimension toward some of the recent intelligent techniques by their introduction in the form of deep learning, fuzzy logic, and SDN regarding enhancing accuracy and reducing the computational overhead.

Reference [9] describes a brand-new hybrid deep learning model that ties CNNs to LSTM but also has a self-attention paradigm incorporated to highlight the most informative input features. CNN, representing spatial features extraction, and LSTM, representing temporal sequence modeling. This model is called Attention-CNN-LSTM. The performance of this model has been evaluated over the NSL-KDD and Bot-IoT datasets, where it achieved an accuracy ranging from 94.8% to 97.5% and a substantial improvement on MCC and F1-score. An innovative Stacked Convolutional Neural Network and Bidirectional LSTM (SCNN-Bi-LSTM) model for intrusion detection in WSNs is being proposed by [10], using Federated Learning (FL) to enhance intrusion detection performance while ensuring privacy. The first federated SCNN-Bi-LSTM model works by allowing multiple sensor nodes to train collaboratively on a central global model without disclosing their private data, thus addressing privacy concerns. [11] Proposed advanced intrusion detection systems based on hybrid

machine learning (AIDS-HML) in wireless sensor networks to identify and classify attacks. Hybrid machine learning classifiers identify threats in wireless sensor networks. [12] Proposes a Secure Automatic Two-level Intrusion Detection System (SATIDS) based on an improved LSTM network. The proposed system differentiates between attack and benign traffic, identifies the attack category, and determines the sub-attack type with high performance. [13] Leverage Genetic Algorithm (GA) with a correlation coefficient as a fitness function for feature selection. Additionally, mutual information (MI) is applied for feature ranking to measure their dependency on the target variable. The selected optimal features were used to train a hybrid DNN model to uncover attacks in IoT networks. [14] employs machine learning techniques, notably the Gaussian Naive Bayes (GNB) and Stochastic Gradient Descent (SGD) algorithms. The effectiveness of recommendation systems is improved with the introduction of context awareness. To reduce the burden on the computer, the authors first perform principal component analysis and singular value decomposition on the raw traffic data. [15] proposes an intelligent hybrid model that leverages machine learning and artificial intelligence to enhance the security of WSNs by identifying and preventing cyberattacks. Feature extraction with K-means clustering model enhanced information gain (KMC-IG) is based on the combined effect of SVD and PCA along with the other feature dimensionality-reduction techniques. [16] combined Hidden Markov Models with Gaussian Mixture Models and dimensionality reduction techniques, reaching classification rates for anomaly detection of up to 94.55%. [17] Using the NS3.37 simulator, he addressed Wormhole attacks using SVMs and Deep Neural Networks, with significant improvements over various performance metrics. Detecting DoS attacks was the aim of the work [18], proposing and testing a lightweight decision tree algorithm on an enhanced WSN-DS dataset achieving 99.5% accuracy and reduced processing time compared to other algorithms such as XGBoost, Random Forest, and k-nearest neighbours. [18] describes how a highly accurate anomaly detection system based on Python has been developed using the NSL-KDD dataset, employing various feature selection and data balancing techniques and using algorithms such as Random Forest, Decision Tree, SVM, and KNN. Further, an innovative hybrid deep learning model, integrating convolutional neural networks and long short-term memory for healthcare-related WSNs, was proposed by [19] to tackle some of the challenges, such as slow processing and low attack detection rates. This model employed a modified Huber independent component analysis method for dimensionality reduction, which was validated using the NS2.34 network simulator. [20] Employ DL techniques to detect and classify several DoS attacks. The goal of this research is to experiment with several DL-based algorithms to develop an efficient, lightweight, and accurate approach for detecting DoS attacks in WSNs. [21] Proposes an approach called Genetic Sacrificial Whale Optimisation (GSWO) to address the limitations of conventional methods. GSWO combines a genetic algorithm (GA) and a whale optimisation algorithm (WOA), with a new three-population division strategy and a proposed conditional inherited choice (CIC) to overcome premature convergence in WOA.

Despite significant progress in intrusion detection for Wireless Sensor Networks using deep learning, fuzzy logic, and SDN-based techniques, several critical gaps remain in the current body of knowledge. Most existing approaches focus on single-paradigm models or limited hybrid combinations that primarily operate as passive classifiers without system-level integration. The system requires uncertainty-based decision improvement methods together

with real-time resource conflicts solutions which the current system cannot deliver. Table 1 provides a summary of the related methods included in this study.

Table 1. Related work.

Ref.	Proposed	Method	Strength	Weaknesses
[9]	Introduces a novel hybrid deep learning model named Attention-CNN-LSTM, designed to enhance intrusion detection systems (IDS).	CNN-LSTM	Significant improvement over standalone models across all metrics.	Only 10 training epochs may not fully capture the training dynamics of large-scale IDS models.
[10]	Hybrid LSTM-GRU deep learning architecture	LSTM-GRU	Improve the performance	LSTM-GRU and Bayesian optimisation increase the complexity of training.
[11]	Detect malicious nodes and anomalies in a hierarchical WSN structure using hybrid machine learning techniques.	Combining multiple ML algorithms	Improve the accuracy	There is no scalability testing on large-scale networks
[12]	Develop a system that can automatically detect cyber threats in these increasingly integrated network infrastructures.	LSTM	Improve the accuracy and detection rate	It still requires enhancements in terms of cross-dataset generalisation
[13]	Proposes a hybrid deep learning framework for detecting cyber attacks in IoT networks, leveraging SDN for enhanced network management.	hybrid DNN	The proposed model demonstrates impressive accuracy and detection time	The execution time is still high
[14]	Combining feature selection algorithms with machine learning techniques, current state-of-the-art intrusion detection systems.	Gaussian Nave Bayes and Stochastic Gradient Descent	Improve the performance metrics	There is no scalability testing on large-scale networks
[15]	Intelligent hybrid model	K-means clustering, DNN	Improve the performance metrics	There is no scalability testing on large-scale networks
[16]	Enhancing routing security	Hidden Markov Model (HMM), Gaussian Mixture Model (GMM), dimensionality reduction	High detection rate (94.55%), robust against noise	Limited scalability, sensitive to parameters
[17]	Wormhole attack detection in WSN/IoT	SVM and DNN	Effective against wormhole attacks, validated in NS3.37	High computational cost, may not generalise
[22]	DoS attack detection in WSNs	Decision Tree	High accuracy (99.5%), lightweight	Focused only on DoS, limited to WSN-DS
[18]	Anomaly-based IDS in WSN	Random Forest, Decision Tree, SVM, KNN	Flexible, high accuracy	Feature engineering complexity
[19]	IDS for healthcare WSN	CNN + LSTM + MHICA	Robust detection, low processing delay	Specific to healthcare, complexity
[20]	Efficient and lightweight (single-layer) IDSs	CNN, RNN, DNN, CNN+RNN	High accuracy and lightweight	The size of the training data is large
[21]	Enhancing intrusion detection in WSNs	GSWO	Tested the proposed work with different datasets	The complexity of the GSWO algorithm and the CatBoost model could pose challenges

The detection performance assessment in previous research studies was conducted under offline conditions because actual deployment requirements which include latency and energy consumption and system flexibility to handle new attack patterns were not taken into account. The CNN–LSTM architecture provides effective detection capability, but its performance suffers because it generates unstable predictions when faced with noisy and borderline traffic conditions which results in more false positives and false negatives. The SDN-based security frameworks establish centralized control as their primary security mechanism while they fail to establish proper connection between their detection intelligence systems and control operations.

The existing research progresses knowledge through its introduction of an integrated edge security framework which uses spatial-temporal deep learning and fuzzy uncertainty reasoning together with SDN-based network control for intrusion detection. The study establishes a connection between precise attack detection and operational network defense by implementing a Fuzzy CNN-LSTM model at network edges which works together with an SDN controller to achieve adaptive threat response. This integrated design enables robust, low-latency, and adaptive intrusion detection, thereby extending existing research from isolated detection models toward deployable, system-level security solutions for modern WSNs.

3. PROPOSED WORK

This paper presents a new multilayer approach for real-time attack detection within wireless sensor networks that takes advantage of edge computing, deep learning, fuzzy logic, and software-defined networking. It aims to provide a robust and intelligent framework for the detection of advanced cyber threats in the WSN environment while compensating for the deficiencies in latency, energy, and scalability, as illustrated in Fig. 1.

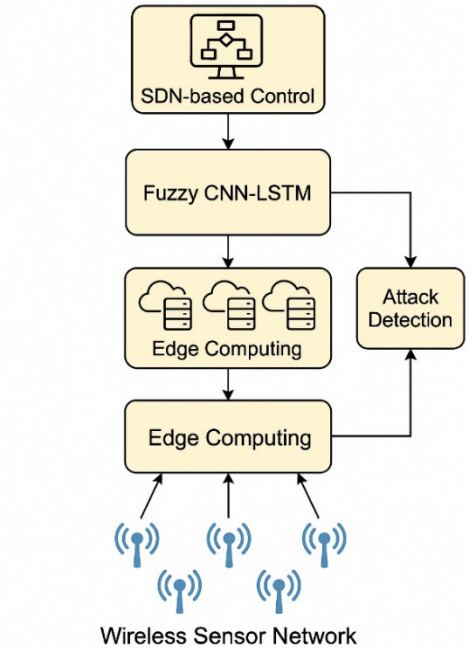

Fig. 1: Proposed work.

The bottom level of sensing consists of a network in which sensor nodes are widely distributed to measure environmental as well as operational parameters with respect to their location within the environment. Because of the limited processing and energy resources in

these nodes, they are likely to become hosts for different kinds of cyberattacks. When data undergo normal transmission, they are taken up to the edge computing units, which are located at strategic points near sensor clusters, to relieve these nodes of processing their findings. Instead, those edge nodes are going to serve as intermediaries in the processes by providing real-time filtering, feature extraction, and first anomaly detection during run time. It creates reduced volumes of information traffic toward the centric systems to enable faster responses to threats at the local level and significantly decrease the network latency.

The detection system is based on a hybrid Fuzzy CNN-LSTM deep learning architecture that provides edge detection. The CNN model extracts spatial features from traffic flow and, in addition, captures local patterns and relationships among input features. Next, the LSTM input will model temporal dependencies and behaviours, along with time-based attack signatures, through the following sequence. Decision-making under uncertainty occurs at the level of fuzzy logic after the LSTM output, using a fuzzy inference system for uncertain or borderline predictions. This will then serve as a complement to a much finer classification model based on linguistic rules and membership functions. The model thus aims to actively accommodate noisy or imprecise data along with improving its overall robustness to the evolving attack patterns.

Once classified, if the edge node detects an anomaly or malicious behaviour at the node or sensor level, it will trigger an alert message to the control plane based on the SDN. The SDN controller is characterised as the central orchestration component and maintains a dynamic global view of the network under the management of data flows. Once an alert is generated, the system may either isolate or reroute affected traffic, or it may apply mitigation policies such as blocking or rate-limiting based on predefined security policies. The SDN layer enables Programmability which allows networks to swiftly adapt to new security threats while performing real-time load balancing across edge resources to sustain consistent service quality (QoS) delivery.

The framework begins its operation by processing data from sensor nodes which it transmits to edge processor nodes. The edge layer performs data preprocessing after which it sends the cleaned data to the hybrid deep-learning model for processing. The system will initiate network modifications through SDN controller actions which will implement designated security measures based on detection results. The system uses a modular design to achieve maximum detection accuracy which requires minimal data transmission while enabling the system to handle different types of WSN-based attacks.

It has addressed the intrinsic security challenges within WSNs by harnessing speed and locality from edge computing, deep learning modeling, fuzzy systems for uncertainty handling, and the inherent flexibility of SDN. Thus, providing end to end integrated scalable and intelligent solutions for next-generation applications based on WSN technologies, including smart agriculture, military surveillance operations, and even industrial IoT systems.

3.1. Data Preprocessing

For the model developed, the hybrid Fuzzy CNN-LSTM, experimentation and evaluation were conducted using the WSN-DS and WSNBFSF datasets which happens to be freely available in the public for use by anyone interested in researching on intrusion detection in WSNs. WSN-DS was specifically designed to suit the needs of researching such techniques in the WSN environment while also containing various attack types such as blackhole,

grayhole, flooding, scheduling (TDMA), and normal traffic instances, while WSNBFSF dataset contains flooding, selective forwarding, blackhole, sinkhole-related and normal traffic instances.

3.1.1. Data Import and Initial Inspection

The input data is composed of time series captured and scripted in CSV format and analyzed through the powerful processing techniques presented by the pandas library in Python. On first perusal analysis of the dataset, it showed a mixture of numerical features (for instance, packet arrival time, energy consumption, hop count) and a categorical class label. In fact, there was also an assessment of the class distribution to check for possible imbalances and consider the proportional representation of each attack type during training.

3.1.2. Handling Missing and Duplicate Records

A situation in which there are missing, or corrupted values is a common scenario in large-scale simulation logs. Excessively missing-field records were removed, while missing values that were isolated in numerical fields were imputed with the meaning of the corresponding feature. Eliminating duplicate records was also enforced to reduce redundancy and mitigate any chances of bias during model training.

3.1.3. Label Encoding

The initially string-classified class attribute was converted into integer-encoded labels by the use of Label Encoder followed by one-hot encoding to modify the labels to be compatible with multilabel classification with the softmax output of the deep learning classifier.

3.1.4. Feature Normalisation

By means of min-max normalisation, all numerical features were scaled to the range [0, 1]. This preprocessing step ensures that features with larger numerical ranges will not put an undue emphasis on model training and will support faster convergence in gradient-based optimisation.

3.1.5. Sequence Construction for Temporal Modelling

This structure was needed as LSTM requires a sequential form of input. The datasets were therefore framed into fixed-length time windows using the sliding window method. Each sample comprises a sequence of 10 consecutive flow records, decided through empirical observation, to give the model an opportunity to capture the temporal patterns related to attack propagation or anomalous behaviour.

3.1.6. Feature Selection

Feature selection is a critical preprocessing step in intrusion detection for Wireless Sensor Networks due to the inherent constraints on computational power, memory, and energy consumption. High-dimensional input data may introduce redundant or irrelevant features which create additional computational demands while decreasing training speed and leading to overfitting, which results in lower generalization performance. The researchers

used systematic feature selection methods to find a set of features which maintained their ability to distinguish between classes while simplifying model development.

I. Initial Feature Set Analysis

The two datasets, WSN-DS and WSNBFSF provide multiple numerical features which encompass traffic patterns and routing operations and energy consumption patterns to depict how sensor nodes communicate with each other. The features provide multiple network activity measurements which include metrics for packet transmission and node energy usage and hop count and flow duration and timing behavior. All numerical features were kept in their original state to prevent early information loss while enabling objective evaluation of redundancy and relevance assessment.

II. Correlation-Based Redundancy Elimination

To identify redundant features, a correlation analysis was conducted using the Pearson correlation coefficient. Pearson's coefficient measures the strength of linear dependence between pairs of numerical features and is defined as:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (1)$$

The presence of highly correlated feature pairs results in duplicate information which disrupts learning stability without providing any advantage to predictive accuracy. In this study, feature pairs with an absolute correlation coefficient greater than 0.95 ($|r| > 0.95$) were identified as strong correlation candidates. The researchers eliminated one feature from each pair because the feature had less importance for detecting intrusion activities and its value for understanding WSN traffic behavior was more limited.

The threshold value of 0.95 was selected to strike a balance between redundancy elimination and information preservation. The lower thresholds removed features that captured different traffic patterns, while the higher thresholds produced only small reductions in dimensionality. The choice of this method follows established standards which network security and machine learning research commonly use

III. Low-Variance Feature Filtering

The study examined features with extremely low variance across samples together with correlation analysis. Low-variance features provide minimal value for discrimination because they maintain their stable characteristics throughout all traffic classes. The system required removal of such features because they would create training problems through both noise amplification and extra computational work.

IV. Domain-Aware Feature Retention

Apart from statistical criteria, knowledge about WSN was put in to reach the concluding set of features. Features that describe:

- traffic intensity and packet flow behavior,
- routing dynamics and hop-based communication,
- node energy usage and transmission timing, were prioritized, as these characteristics are known to be highly sensitive to attacks such as blackhole, grayhole, flooding, and TDMA-based scheduling attacks. This is to preserve the spatial patterns, captured by the convolutional neural network layers, along with the temporal dependency that would have been modeled through memory cells like those of the LSTM.

V. Final Feature Set and Impact

The original feature set was reduced to 18 representative features after applying three procedures which included correlation-based redundancy removal and low-variance filtering and domain-aware selection.

The features were first normalized through min-max scaling and later arranged into time-windowed sequences for the purpose of temporal modeling. This dimensionality reduction significantly reduced the computational complexity of the proposed Fuzzy CNN–LSTM model while maintaining high detection accuracy, as evidenced by the experimental results.

The reduced feature space also contributed to faster inference time and improved suitability for deployment in edge-based and resource-constrained WSN environments.

3.1.7. Dataset Partitioning and Balancing

After processing, this dataset was randomly shuffled and split into training (70%), validation (15%), and testing (15%) datasets. Because of the imbalance between the standard and attack samples, with rare attacks such as scheduling (TDMA) and grayhole being disproportionately underrepresented, the Synthetic Minority Over-sampling Technique (SMOTE) was applied only to the training dataset in order to achieve balanced class representation and enhance the generalisation capacity of the model.

3.2. SDN Reaction and Load Balancing

It is at the SDN layer that one finds the main control plane, which orchestrates intelligent and real-time responses to the apparent security threats.

An alert from the edge computing layer would mean that the traffic pattern may have been classified as malicious by the Fuzzy CNN-LSTM model. Consequently, the SDN controller authorizes the dynamic mitigation workflow to contain the threat and restore stability to the network.

3.2.1. Threat Mitigation via Flow Control

The SDN controller operates on a logically centralized view of the global network topology, allowing it to rapidly pinpoint affected nodes, links, or flows.

The controller starts to check the alert's source and destination identifiers after he receives the alert notification. He uses active flow tables to find matches before he decides which path to classify as compromised.

A set of predefined security policies is then enforced. These include:

- Flow Isolation: Immediate deletion of malicious flow entries from forwarding devices to prevent further propagation of the attack.
- Dynamic Blocklisting: Temporarily or permanently blocking the source node or its MAC/IP address based on the severity and frequency of detected attacks.
- Traffic Redirection: In cases where isolation is not feasible (e.g., critical nodes), the controller redirects traffic through secure, redundant paths to maintain network availability.

The controller uses OpenFlow commands to control the operation of SDN-enabled switches and virtual testbeds which include Mininet to perform those actions.

3.2.2. Adaptive Load Balancing

The SDN controller monitors network performance by tracking bandwidth and flow density and node load in addition to its security response duties. The network traffic distribution process relies on these metrics which help in adaptive load balancing to avoid performance declines that result from both node overuse and targeted attacks.

Key strategies include:

- Path Optimisation: Selecting alternative routes with lower congestion using Dijkstra-based or heuristic routing algorithms.
- Flow Reassignment: Dynamically reassigning active flows from overburdened nodes to underutilised parts of the network.
- Edge Task Reallocation: In edge-enabled environments, the SDN controller can offload computational tasks from overloaded edge nodes to neighbouring nodes with higher available resources.

These strategies not only improve the network's resilience under attack conditions but also enhance the quality of service (QoS) for legitimate traffic.

3.2.3. Logging and Forensic Support

All SDN-triggered mitigation and load-balancing actions are logged in a central log database. Timestamp, affected node ID, rule modifications, flow redirection decisions, and model confidence scores are all included in the logs. Logs can be used in post-incident forensic analysis, compliance auditing, and retraining the detection model with updated patterns.

3.3. Fuzzy CNN-LSTM Classifier Design

The detection engine proposed here is based on a hybrid Fuzzy CNN-LSTM architecture that integrates the spatial feature extraction capacity of CNNs, the temporal pattern recognition of LSTMs, and the uncertainty handling underpinnings of fuzzy logic. Such an integration is proposed to tackle the issues encountered in anomaly detection in WSNs, where data are often considered sparse, noisy, and ambiguous.

Each input sample is a multivariate time series represented as a matrix. $X \in R^{T \times F}$, where $T = 10$ Is the number of time steps and $F = 18$ is the number of preselected features after preprocessing. These samples are derived using a sliding window over sequential flow records to preserve temporal integrity in the network behaviour.

3.3.1. CNN-LSTM Architecture

The first stage of the model is a 1D CNN that applies convolutional filters to capture local spatial dependencies among the features. Let $x_t \in R^F$ Be the feature vector at the time step. tThen the output of a convolutional filter $w \in R^{k \times F}$ Is computed as:

$$h_t = ReLU\ (w * x_{t:t+k+1} + b) \quad (2)$$

where $*$ denotes convolution, k Is the kernel size, b The bias term is used, and ReLU is the activation function. A max-pooling layer follows to down-sample the feature maps and emphasise the most prominent features.

The output of the CNN is then passed to the LSTM layer, which learns temporal correlations in the sequence:

$$h_t^{LSTM} = LSTM(h_{t-1}, x_t) \quad (3)$$

The LSTM maintains internal memory through gated operations (input, forget, and output gates), allowing it to remember long-term dependencies that are essential for attack pattern recognition.

The final output of the LSTM is fed to a dense layer with softmax activation to generate preliminary class probabilities:

$$P(y = c|x) = \frac{e^{z_c}}{\sum_{j=1}^{C} e^{z_j}} \quad for\ c \in \{1, 2, \dots, C\} \tag{4}$$

where C Is the number of classes and z_j Is the output of the last dense layer before activation.

3.3.2. Fuzzy Inference Layer

To handle uncertain predictions, particularly when softmax probabilities are close across multiple classes, we introduce a fuzzy decision layer. The fuzzy logic module interprets the softmax output vector. $P = [p_1, p_2, \dots, p_C]$Using trapezoidal membership functions that define the degree to which each class probability belongs to linguistic categories: Low, Medium, and High.

The membership functions are defined as follows:

$$\mu_{Low}(p) = \begin{cases} 1 & p \le a \\ \frac{b-p}{b-a} & a < p < b \\ 0 & p \ge b \end{cases}$$

$$\mu_{Mid}(p) = \begin{cases} 0 & p \le a\ or\ p \ge d \\ \frac{p-a}{b-a} & a < p \le d \\ 1 & b < p \le c \\ \frac{d-p}{d-c} & c < p < d \end{cases} \tag{5}$$

$$\mu_{High}(p) = \begin{cases} 0 & p \le c \\ \frac{p-c}{d-c} & c < p < d \\ 1 & p \ge d \end{cases}$$

with typical parameter settings:

Low: $a = 0.0, b = 0.4$

Medium: $a = 0.3, b = 0.5, c = 0.7, d = 0.9$

High: $c = 0.7, d = 1.0$

This logic allows the classifier to treat ambiguous cases conservatively, thus reducing false positives or misclassifications under uncertainty. The pseudocode for the proposed Fuzzy CNN-LSTM is as in Algorithm 1.

It is a classifier based on the Fuzzy CNN-LSTM, which logically produces a sequence of processing layers contributing certain functionality to the entire detection pipeline. The initial task of the model is to apply 1D convolutional filters to the time-windowed input features to extract important spatial maps at each timestep. Max pooling downsamples feature dimensions and improves computational efficiency. The flattening step then leads into an LSTM layer that models the time-related behaviour of the traffic. In the end, a dropout is applied to the LSTM output to reduce overfitting. Using a probability distribution over the set of possible classes, the dense output layer is implemented using a softmax. These fuzzy memberships then evaluate probabilities between classes. Thus, an entity is said to be 'normal'

when no attacks or suspicious behaviour are detected. Where one or more classes have high membership values (greater than 0.6), that class is taken as the predicted label; otherwise, if any suspicious (non-normal) classes exhibit medium membership exceeding 0.6, it is an attack. When this fails, that instance is labelled normal. It improves robustness, as rule-based fuzzy refinement pays dividends in scenarios involving uncertain or ambiguous traffic patterns for which softmax outputs would lead to misclassification.

Algorithm 1: Fuzzy CNN-LSTM Classifier

Input:
X: Time-windowed flow sequence of shape $[T \times F]$
Output:
y_pred: Predicted attack class

1: CNN_out ← Apply Conv1D to X with filters=64, kernel_size=3, activation=ReLU
2: CNN_out ← Apply MaxPooling1D to reduce dimensionality
3: CNN_out ← Flatten output to form a feature vector
4: LSTM_out ← Feed CNN_out to LSTM (units=100)
5: LSTM_out ← Apply Dropout with rate=0.2
6: P ← Dense(LSTM_out) → Softmax activation
// $P = [p_1, p_2, \ldots, p_C]$Probabilities for each class
7: for each class c in {1, ..., C} do
8: μ_high[c] ← High_Membership(P[c])
9: μ_med[c] ← Medium_Membership(P[c])
10: end for
11: if max(μ_high) > 0.6 then
12: y_pred ← argmax(μ_high)
13: else if max(μ_med) > 0.6 and argmax(μ_med) ≠ normal then
14: y_pred ← argmax(μ_med)
15: else
16: y_pred ← normal
17: end if
18: return y_pred

3.3.3. Fuzzy Layers

The thresholds for the fuzzy membership functions (Low: 0.0-0.3, Medium: 0.2-0.7, High: ≥0.6) were selected based on empirical probability distributions from the CNN-LSTM baseline model. Specifically, softmax outputs for correctly classified samples showed high confidence values clustered above 0.75-0.85, while misclassified or borderline instances commonly produced probabilities between 0.35-0.55. These observations motivated the use of overlapping trapezoidal ranges, enabling the fuzzy layer to capture uncertainty in the mid-range region where the CNN-LSTM model was most unstable. The selected thresholds were therefore aligned with actual model behaviour rather than arbitrary selection and were configured to avoid over-penalising classes with naturally lower confidence ranges.

A sensitivity analysis was performed to evaluate the effect of different membership ranges on classification performance. Three configurations were tested: (i) tighter thresholds (High ≥ 0.7), (ii) baseline thresholds used in this work (High ≥ 0.6, Medium 0.2–0.7), and (iii) more permissive thresholds (High ≥ 0.5). Results showed that tighter thresholds reduced false

positives by 1.4% but increased false negatives by 2.1%, negatively affecting Recall. In comparison, with more permissive thresholds causing 3.8% more false positives and a lesser Precision, suitable balance between performance and execution time was struck for the baseline configuration, producing the best values for the F1 score and MCC with negligible (<2%) degradation in execution time.

3.4. Evaluation Metrics

Several standard classification metrics have been used to qualitatively assess the empirical performance of the hybrid fuzzy C-NN-LSTM classifier designed for attack detection in varying WSN environments. All these metrics, derived from the confusion matrix, provide subtle assessment of the model's performance on grounds of accuracy, reliability, and robustness under conditions significantly influenced by imbalance and uncertainty, which are characteristics commonly prevalent in intrusion-detection scenarios.

3.4.1. Accuracy

Accuracy refers to the general proportion of correctly classified instances into all instance predictions. Its definition is:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

However, the fact that accuracy is a good global indicator does not suggest that it shall always be appropriate, especially in class-imbalanced using the attack sample and the greatly overwhelming number of normal traffic samples.

3.4.2. Precision

Precision means that the relative ratio of correct classified attacks to all predicted instances describes how the model can avoid false positives. Hence, the equation is as follows:

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

The measure precision tells how many false alarms the model generates, and generally, this is required in WSNs, where unnecessary actions incur a loss in the limited resources availability.

3.4.3. Recall

Measure of actual attacks detected by model is called recall or true positive rate:

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

High recall is crucial for security scenarios, where a material failure can cause devastating damage to the entire network because of an attack overlooked by the system as if it were a false negative.

3.4.4. F1-Score

The F1-score represents the harmonic mean of precision and recall, offering a balanced evaluation for the accuracy of the classifier in the presence of both false positives and false negatives:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (9)$$

It is useful in cases with unequal class distribution.

3.4.5. Matthews Correlation Coefficient (MCC)

There are a few more balanced algorithms used in the evaluation of imbalanced datasets; therefore, we compute the MCC as:

$$MCC = \frac{(TP\ .TN)-(FP.FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (10)$$

The value of the MCC will represent a correlation: -1-perfect discordance, 0- random performance, and +1-perfect prediction.

3.4.6. Execution Time

Overall classification accuracy as well as total execution time is worth a measure for evaluation criteria for real-time deployment scenarios in resource-constrained WSNs. The measurements were made in seconds and compared across the various deep learning models along with and without fuzzy logic incorporated in them.

3.5. Security Attacks

The proposed model was evaluated using the WSN-DS dataset, which contains several common and critical wireless sensor network attacks. The hybrid Fuzzy CNN-LSTM model demonstrates high detection performance against the following attack categories:

- Blackhole Attack: Malicious nodes absorb all incoming packets, preventing forwarding and disrupting routing.
- Grayhole Attack: Selective dropping of packets, making detection more difficult than blackhole attacks.
- Flooding Attack: Generation of excessive traffic to exhaust network resources and cause denial of service.
- TDMA (Scheduling) Attack: Manipulation of transmission schedules, causing collisions, energy waste, and routing instability.
- Normal Traffic Classification: The model correctly learns normal WSN communication behaviour, ensuring low false positives.

Although the results were primarily reported using a binary classification scheme (Normal vs Attack), the training and evaluation process utilised traffic behaviours across all the above attack types. The fuzzy refinement layer further enhances the model's ability to distinguish borderline or ambiguous attack patterns, resulting in fewer false positives and false negatives across these categories.

The assessment of framework generalizability beyond the specific dataset was conducted through additional testing with the WSNBFSF dataset. The WSNBFSF dataset functions as a Wireless Sensor Network intrusion detection dataset which uses different traffic patterns and attack distribution methods than the WSN-DS dataset. The system includes sensor communication features which demonstrate network activity during normal operation and malicious behavior to create an additional evaluation environment.

WSNBFSF enables researchers to test their method across different datasets because it introduces various traffic patterns and attack behaviors which differ from the fixed simulation settings used in WSN-DS. The WSNBFSF test evaluates model performance by showing its ability to handle different WSN conditions while demonstrating that spatial-temporal models do not become tailored to specific training data.

The WSNBFSF dataset includes normal sensor traffic as well as several representative WSN attack categories, such as:

- Flooding attacks, targeting bandwidth and energy exhaustion,
- Selective forwarding attacks, where compromised nodes drop or alter packets,
- Blackhole attacks, disrupting routing paths by advertising false routing information,
- Sinkhole-related behaviors, which attract traffic toward malicious nodes for interception or disruption.

These attack categories reflect realistic threats commonly observed in practical WSN deployments and introduce variability in traffic patterns and malicious behavior.

4. EXPERIMENT AND RESULTS

4.1. Experimental Setup and Hyperparameter Configuration

All experiments were conducted on a workstation equipped with an Intel® Core™ i7 CPU running at 3.40 GHz, 16 GB of RAM, and a 64-bit Windows operating system. No GPU acceleration was used, in order to reflect realistic edge and resource-constrained deployment scenarios.

The proposed Fuzzy CNN-LSTM model was trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 32. The training was performed for 5-30 epochs depending on the experiment, and categorical cross-entropy was used as the loss function. All model components including the CNN, LSTM, and fuzzy inference layers were tuned to balance detection accuracy and computational cost suitable for WSN edge deployment. A complete list of hyperparameters is provided in Table 2.

4.2. Training and Testing Performance

This part chronicles the empirical evaluation of the proposed Fuzzy CNN-LSTM model toward binary intrusion detection (Normal vs. Attack) on the Wireless Sensor Network-Dataset (WSN-DS and WSNBFSF datasets). The foremost purpose of this investigation was to ascertain the model's ability to achieve high detection accuracy at the expense of some false positives to enhance its practicality in real-time WSN applications. Much of the experimentation was done for training and testing purposes of the integrated model, while emphasis was laid conspicuously on time tasks and detailed analysis of classification performance.

Different epochs with a batch size of 32 were used to train the model, as depicted in Fig. 2, showing that accuracy improved across epochs, converging at above 99% by all epochs in WSN-DS. This means that the model is learning the spatial and temporal patterns in network traffic data effectively, while using WSNBFSF, the model achieved more than 98% accuracy across epochs.

Execution time becomes a real parameter in WSNs, considering that energy and computing resources are limited. As can be seen in Fig. 3 and Fig. 4, the training duration for each epoch was maintained within reasonable levels and, therefore, the time taken in the testing (inference) phase also fell in a short time window. This would mean that the model is very applicable to be deployed in a real-time or near-real-time edge environment.

Table 2. Hyperparameters Used in the Proposed Fuzzy CNN-LSTM Model

Component	Hyperparameter	Value
Training Configuration	Optimizer	Adam
	Learning Rate	0.001
	Batch Size	32
	Epochs	5–30
	Loss Function	Categorical Cross-Entropy
	Validation Split	15%
CNN Module	Number of Filters	64
	Kernel Size	3
	Activation	ReLU
	Pooling	MaxPooling1D
	Padding	Same
LSTM Module	LSTM Units	100
	Dropout	0.2
	Recurrent Dropout	Not used
	Return Sequences	False
Dense Layer	Activation	Softmax
	Number of Output Classes	2 (Normal, Attack)
Fuzzy Module	Membership Levels	Low, Medium, High
	Membership Function Type	Trapezoidal
	High Threshold	0.6
	Medium Range	0.2–0.7
	Low Range	0.0–0.3
Data Processing	Sliding Window Size	10 timesteps
	Feature Count	18 selected features
	Normalization	Min–Max Scaling
SMOTE Balancing	Oversampling Method	SMOTE
	Applied To	Training set only

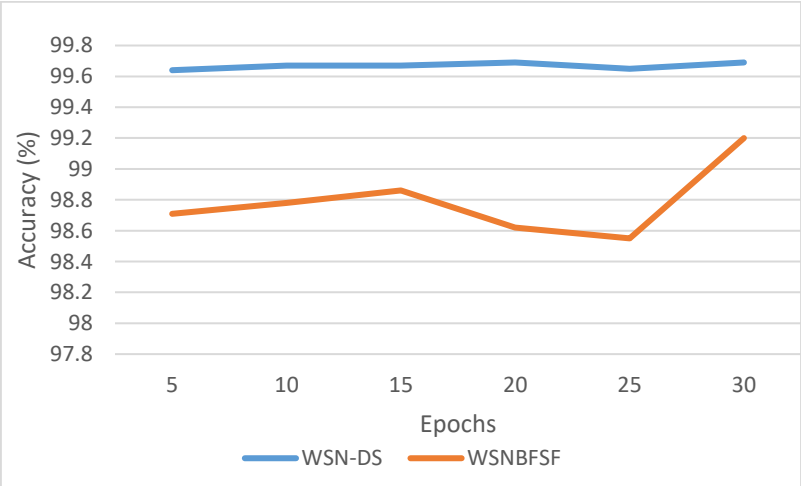

Fig. 2. Fuzzy CNN-LSTM accuracy

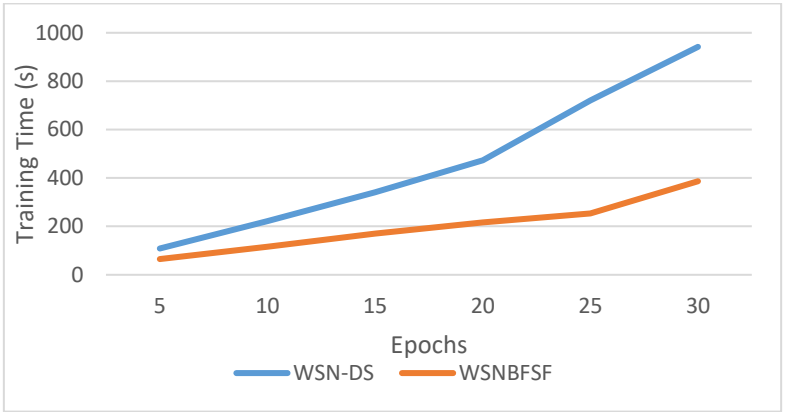

Fig. 3. Fuzzy CNN-LSTM training time.

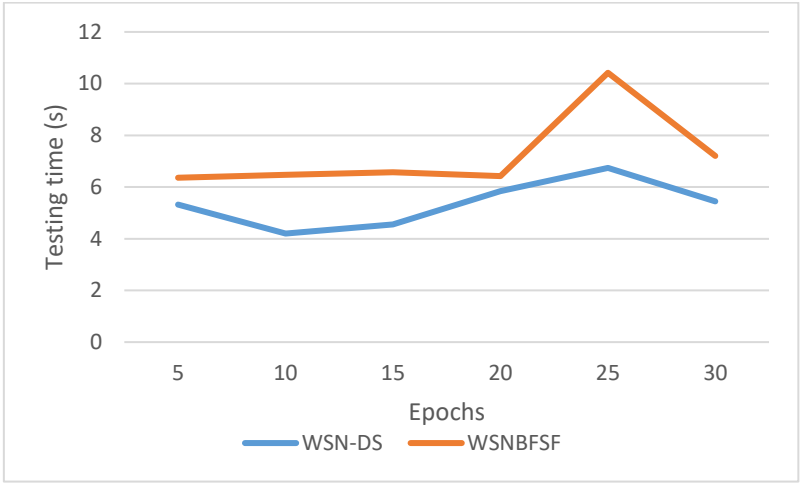

Fig. 4. Fuzzy CNN-LSTM testing time.

Quantitative evaluation of the model was carried out using the standard classification metrics such as Recall, Precision, F1-Score, Matthews Correlation Coefficient (MCC), and components of confusion matrix (TP, TN, FP, FN). Table 3 summarizes the results across the different epochs.

Table 3. Performance summary of WSN-DS.

Epochs	Recall	Precision	F1 Score	MCC	TP	TN	FP	FN
5	98.48	97.68	98.08	97.88	6862	67802	163	106
10	98.52	97.93	98.23	98.04	6865	67820	145	103
15	98.16	98.33	98.25	98.07	6840	67849	116	128
20	98.38	98.24	98.31	98.13	6855	67842	123	113
25	98.31	97.89	98.1	97.9	6850	67817	148	118
30	98.06	98.56	98.31	98.14	6833	67865	100	135

The evidence here demonstrates the robustness and generalization capability of the model, obtaining high F1 score (~98.3%) and MCCs (>98%) across the epochs. Importantly, both false positive and false negative rates remained consistently low, reinforcing the trustworthiness of the classifier.

The experiment was conducted five times using independent random seeds to obtain mean and standard deviation results for the key performance metrics. The proposed model had an average detection accuracy of 99.63% with an SD of 0.11 over these five runs, indicating

an extremely stable performance. The mean F1 score was 98.22% with a standard deviation of (±0.14) and the mean MCC was 98.04% with a standard deviation of (±0.17) which shows that the predictive performance of the model remained consistent across testing.

To statistically validate the model comparison using paired t-tests with per-run F1-scores of the proposed model, standalone CNN and LSTM baselines have been compared. Improvements yielded results that were statistically significant at the level of 0.05 ($p < 0.01$ for CNN, $p < 0.05$ for LSTM), thus showing that such improvements are not likely due to mere chance coincidence. Moreover, both false positive and false negative rates less than 2% relative variation showed fairly low dispersion among runs, which supports the robustness of the fuzzy refinement layer for stabilizing results of classifications under different initialization conditions.

Moreover, training and inference time didn't significantly fluctuate, with mean testing time being 4.55 sec (±0.09). This steadiness indicates that model deployment cost is not sensitive to stochastic initializations for real-time deployment in an edge environment very important by dependable sources.

Table 4 summarizes the performance of the proposed Fuzzy CNN–LSTM framework on the WSNBFSF dataset across different training epochs. The results indicate that the proposed model maintains strong and stable detection performance on a WSN dataset with traffic characteristics distinct from WSN-DS, demonstrating good generalization capability.

Table 4: Performance summary of WSNBFSF dataset

Epochs	Recall	Precision	F1 Score	MCC	TP	TN	FP	FN
5	96.68	93.31	94.96	94.24	7590	54027	544	261
10	96.90	93.77	95.30	94.62	7704	53959	512	247
15	96.30	94.70	95.5	94.85	7561	54148	423	290
20	95.28	94.98	95.13	94.32	8433	53125	446	418
25	95.38	94.10	94.73	93.90	8156	53359	512	395
30	96.88	97.75	97.31	96.85	8962	52965	206	289

The model demonstrates its capacity to identify most attack instances through its sustained high recall values which range from 95.28% to 96.90% across all testing periods. The model achieves its optimal precision of 97.75% after 30 training epochs because increased training results in more accurate pattern recognition. The F1-score maintains its high level throughout all testing periods, reaching its highest point of 97.31% because the system effectively balances its precision and recalls performance.

The MCC values between 93.90% and 96.85% confirm that the proposed framework maintains its reliability when dealing with class-imbalanced situations. The confusion matrix analysis shows that increasing training epochs results in fewer false positive mistakes while keeping high true positive counts, which demonstrates better classification stability.

The Fuzzy CNN–LSTM framework demonstrates its ability to generalize to the WSNBFSF dataset through its detection performance which remained strong under different WSN traffic patterns and attack scenarios.

4.3. Comparison Results

The proposed security model establishes its superiority over existing state-of-the-art algorithms through the results shown in Table 5. The model achieves superior attack detection results when compared to all other existing models. The performance evaluation was conducted using the WSN-DS dataset as testing material. The results show that fuzzy CNN-LSTM delivers superior performance results through its better FN results and shorter testing time and increased accuracy.

Table 5. Comparison with state-of-the-art algorithms

Metrics	Fuzzy CNN-LSTM	CNN	LSTM	LSTM-CNN	BiLSTM	CNN–GRU
Precision	98.33	89.27	90.84	92.87	89.34	95.45
Recall	98.16	87.2	84.04	91.06	92.39	92.47
F1	98.25	88.23	87.31	91.96	90.84	93.93
MCC	98.07	87	86.03	91.15	89.93	93.32
TP	6840	6251	6367	6345	6253	6628
TN	67849	67014	66899	67478	67419	67449
FP	116	751	642	487	746	316
FN	128	917	1209	623	515	540
Testing time	4.55	7.65	6.94	5.97	5.32	4.86

The proposed Fuzzy CNN-LSTM framework demonstrates better accuracy performance against baseline models on the WSNBFSF dataset according to the results shown in Fig. 5. The visual comparison clearly shows that the proposed approach consistently outperforms standalone CNN, LSTM, CNN-GRU, BiLSTM, and CNN–LSTM architectures.

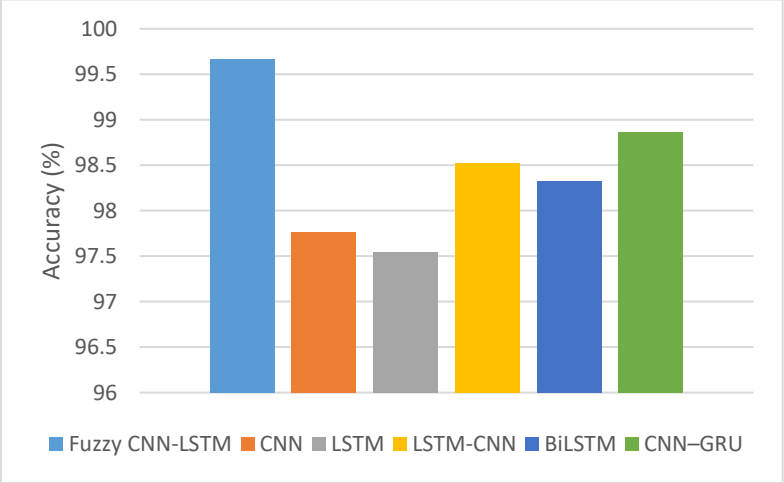

Fig. 5. Comparison with different algorithms.

The accuracy improvement demonstrates the effectiveness of integrating spatial feature extraction, temporal modeling, and fuzzy uncertainty refinement within a unified framework. The Fuzzy CNN–LSTM model achieves the best accuracy results because it outperforms the hybrid deep learning models CNN–LSTM and BiLSTM which attain competitive performance. The visual evidence confirms the generalization ability of the proposed framework on the WSNBFSF dataset because it supports the numerical results which Table 4 displays.

The researchers conducted an ablation study to measure the impact of each main component of their framework by testing various architectural elements which they added to their system to see how it affected model performance. The researchers used separate CNN models and LSTM models to study how spatial features and temporal features would affect their results. The researchers used the CNN–LSTM model to test how spatial–temporal modeling functions together with its uncertainty handling system. The researchers tested the complete Fuzzy CNN–LSTM system to measure how much the fuzzy inference layer contributes to the system.

The study shows that using CNN and LSTM systems together results in better performance than using each system separately because spatial-temporal feature learning needs to be done through joint learning. The fuzzy inference layer improves detection results by creating better predictions from uncertain cases and stopping wrong identifications, which demonstrates how each system component works in the framework.

The model combines Fuzzy CNN-LSTM technology with SDN setup to achieve better accuracy and system strength and deployment capabilities than existing systems. The essential outcomes and system characteristics from Table 6 provide a basis for conducting a comparative analysis of the results. Notably, our approach presents CNN and LSTM constrained by a fuzzy logic inference layer to handle uncertainty in predictions, all in an SDN-oriented architecture catering to active network defense. Such a comprehensive design fills gap areas identified in previous works, improving detection performance while allowing for real-time mitigation and dynamic WSN deployment. As that table below indicates, this proposed Fuzzy-CNN LSTM surpasses the previous works. It achieves approximately 99.67% accuracy on WSN-DS, which is higher than the CNN-based model (94.99% for [23], 98% for [24], 98.79% for [20] and 98.25% for [21]), with our technique's most important benefit being marked improvement in both precision and recall.

Table 6. Comparison with related work.

Ref.	Accuracy	Precision	Recall	F1-Score
Our Proposed	**99.67%**	98.33%	**98.16%**	**98.25%**
Nguyen et al. [21]	98.25%	93.62%	88.85%	90.93%
Salmi & Oughdir [20]	98.79%	94.86%	92.97%	93.72%
Baniata et al. [24]	98.00%	**98.42%**	97.91%	97.91%
Hussain et al. [23]	94.99%	96.21%	94.95%	94.50%

The proposed Fuzzy CNN-LSTM underperforms as earlier works with respect to detection accuracy and maintenance of low latency, as indicated in the previous pages. From the above data, it is evident that the proposed Fuzzy CNN-LSTM model has shown comparatively better traction for the WSN intrusion detection. It integrates the best of previously proposed direct algorithms, optimized ensemble model's high accuracy, deep network's adaptive learning, and federated model's low alarm rates, all into one single framework. Qualitatively, this is a robust and deployable solution that handles temporal dynamics and uncertain decisions and integrates seamlessly with an SDN-based network for automated threat response. The combination of WSNs which achieves high detection rates with its actual deployment capability provides a major achievement which surpasses all existing research work. Our proposed system creates an integrated IDS solution through its

combination of deep learning and fuzzy logic and SDN models which solve the system limits present in earlier IDS solutions.

4.4. Discussion

The study results demonstrate that the Fuzzy CNN-LSTM framework successfully detects WSN attacks because of its operational efficiency and strong defense capabilities against attacks. The combined system that uses CNNs with LSTM layers and fuzzy inference system produced strong results across multiple testing methods which included accuracy, precision, recall, F1-score, and MCC evaluation metrics.

The model showed its best performance during all testing periods when it achieved recall values between 98.06% and 98.52% because it correctly identified most actual attacks while making very few false negative errors. This security measure holds vital importance for WSN environments which need protection against potential threats. The high recall results demonstrate that LSTMs successfully capture time-dependent patterns in network traffic which allows the system to identify different attack patterns occurring at different times.

Precision values were also maintained high, particularly at epochs 15 and 30, attaining levels of 98.56%. This reflects a low false-positive rate and substantiates the contribution of the fuzzy logic layer, which refines class probability outputs by linguistic rules to handle uncertain cases. The fuzzy system thus minimised overconfidence in borderline decisions and thus reduced unnecessary alerts, which is critical in resource-constrained networks where false alarms entail unwarranted expenditure of energy and bandwidth.

F1 Score above 98% across all the epochs-at all times in all of the test cases demonstrated the model's performance in terms of the ability to classifying errors. In addition, the Matthew correlation coefficient values, which hovered around 98%, are indicative of the model's reliability under class-imbalanced conditions since the MCC considers all four outcomes of a confusion matrix part: true positive, true negative, false negative, and false positive.

In addition to the classification performance, the system showed the ability to compute with less effort. There were always small training and testing times as demonstrated in the model accuracy and latency figures, contributing to the applicability of the proposed model for building latency-sensitive applications, such as real-time decision-making and surveillance in WSNs. Integration of the fuzzy decision unit did not add heavy computational cost to further validate the system efficiency.

Infusion of fuzzy logic alongside the CNN-LSTM framework greatly contributes to performance. The fuzzy inference basements are refinements in decisions relying on uncertainties inherent in the outputs and contextual features generated by the neural network. The system demonstrates improved decision-making abilities through its enhanced capabilities to handle uncertain situations which would normally disrupt CNN and LSTM performance. The hybrid systems achieve superior results in distinguishing between normal and abnormal activities because they showed better accuracy results through their reduction of both false positives and false negatives. The system achieves better model understanding and flexible model development to handle the complex and uncertain aspects of actual WSN data.

SDN-based mitigation and flow isolation and load balancing procedures have been implemented at the architectural level in this work; however their effects on network latency

and control overhead and QoS have not been experimentally measured yet and will be studied in upcoming research.

4.4.1. Industrial Significance and Practical Implications

The proposed SDN-oriented Fuzzy CNN–LSTM intrusion detection framework has strong industrial relevance because it achieves three key requirements for resource-limited environments through its deployment capability and adaptability and its ability to operate in real time. Security breaches in Wireless Sensor Networks create critical financial losses and production downtime and safety hazards for industrial systems which depend on these networks to maintain continuous monitoring and control and automation operations.

Sensor networks in industrial Internet of Things (IIoT) settings which include smart manufacturing and process automation must achieve strict latency and reliability standards for their large-scale sensor network deployments. The proposed detection model uses edge-based deployment to reduce communication requirements while enabling quick local decision-making, which decreases the need for cloud systems and provides immediate responses to cyber threats.

The proposed framework achieves compatibility with contemporary industrial network management practices through its implementation of SDN technology. The SDN control system enables central policy enforcement, real-time traffic rerouting, and quick compromised node isolation, which industrial networks require for their operational security. The proposed method provides security functions through closed-loop operations because its detection system activates automated threat response when it identifies an intrusion.

The implementation of fuzzy uncertainty reasoning improves system trustworthiness in industrial applications because sensor data in these environments typically presents issues because of noise and missing information and environmental disturbances. The framework achieves its goal of decreasing unnecessary interventions by reducing false alarms and decision-making errors which would have disturbed essential industrial activities in applications that require high safety and mission reliability.

The CPU-only computation without GPU support verifies that industrial edge devices and embedded controllers can run the proposed framework according to deployment requirements. The solution enables implementation in smart grids industrial monitoring systems pipeline surveillance operations and smart agriculture because these applications require security systems that need minimal resources yet deliver strong protection.

The proposed strategy improves industrial cybersecurity through its adaptive and scalable intrusion detection solution which organizations can implement in practice to protect their systems from attacks while maintaining security for their WSN-based industrial networks.

4.4.2. Computational Complexity Analysis

The computational complexity of the proposed SDN-oriented Fuzzy CNN–LSTM framework is primarily determined by the convolutional and recurrent components, while the fuzzy inference and SDN control logic introduce only negligible overhead. For a given input sample represented as a multivariate time window of length T with F features, the one-dimensional convolutional layer employing K filters of kernel size s incurs a time complexity

of $O(T \cdot F \cdot K \cdot s)$. This operation captures spatial correlations among traffic features and is followed by lightweight pooling operations whose computational cost is comparatively minor.

The dominant computational cost arises from the LSTM module, which models temporal dependencies across the sequence. For an LSTM with H hidden units receiving input of dimensionality D(corresponding to the CNN feature representation), the time complexity of the recurrent computations across the time window is $O(T \cdot H(H + D))$.

Given that $H$and $D$are significantly larger than the number of output classes, this term dominates the overall inference cost of the detection model. The subsequent fully connected and softmax layers contribute only $O(H \cdot C)$operations, where $C$is the number of classes, and thus do not significantly affect overall complexity.

The fuzzy inference layer operates on the softmax output probabilities using a small number of predefined membership functions and rule evaluations. Since the number of classes and linguistic membership levels is fixed and small, the fuzzy refinement introduces constant-time overhead, $O(C)$, and does not increase the asymptotic computational complexity of the framework.

Similarly, SDN-based mitigation actions, such as flow rule updates and rerouting, are event-driven and executed only upon detection of anomalies, making their computational impact independent of the learning model's inference complexity.

Overall, the inference-time complexity of the proposed framework per sample can be expressed as $O(T \cdot F \cdot K \cdot s + T \cdot H(H + D))$, which in practice is dominated by the LSTM component. Training complexity scales linearly with the number of samples and training epochs but follows the same asymptotic behavior as inference, with an additional constant factor due to backpropagation.

This analysis confirms that the proposed framework maintains manageable computational cost while achieving high detection performance, making it suitable for deployment in edge-based and resource-constrained Wireless Sensor Network environments.

4.4.3. Limitation

While the suggested hybrid fuzzy CNN-LSTM-based attack detection framework shows some promising results in terms of detection accuracy and computational efficiency, certain limitations should be kept in mind:

1. Limited Feature Interpretation:

Fuzzy logic does help to build confidence in such decisions, but the system is still a largely black-box system, which also means low transparency and trustworthiness for important applications.

2. Computational Constraints on Edge Devices:

Trained mainly in an edge-processing framework, still the combination CNN + LSTM + fuzzy logic consumes moderate resources. The deployment of such models in very resource-constrained IoT or WSN nodes may still remain a challenge, especially in remote or battery-operated environments.

3. Scalability of Fuzzy Logic Integration:

The fuzzy inference layer is contingent upon a rule system that has been defined for restricted input features. Increasing the input features or the complexity of the network often makes it difficult to scale the fuzzy rule base.

5. CONCLUSION

Wireless Sensor Networks (WSNs) continue to find ever-widening applications in mission-critical infrastructures, yet they remain susceptible to sophisticated multi-stage cyberattacks aimed at exploiting structural limitations such as energy, bandwidth, and coordination. This paper has presented the concept of a hybrid intrusion detection framework that integrates CNN-based spatial encoding, LSTM-based temporal modeling, fuzzy uncertainty reasoning, and SDN-oriented mitigation. The comparative experimental results produced on the WSN-DS and WSNBFSF datasets suggested significant improvements in terms of accuracy, robustness, and inference efficiency by such cross-layer integration vis-a-vis their traditional deep-model counterparts.

The study results show that their findings extend beyond meeting performance requirements. The results reduced decision-making false-positive rates and attack detection false-negative rates using uncertainty-aware decision refinement in network security testing which endangers traditional deep learning systems. The SDN control plane detection model creates a security system that combines active intrusion detection with its essential function of identifying breaches. Future IDS research needs to focus on closed-loop control systems which use detection to initiate real-time responses while maintaining strict limits on both latency and energy consumption. Third, the proposed model requires cloud-level computing resources according to existing knowledge because it needs advanced intrusion detection capabilities. Decentralized privacy-preserving defense architecture will use this system to protect large-scale IoT ecosystems from security threats which depend on cloud services because these services create risks related to latency, data concentration, and policy violations.

The existing research has achieved progress, yet it still needs to solve multiple remaining challenges. The fuzzy inference layer delivers practical results, yet its added complexity will not perform effectively when handling multiple high-dimensional feature spaces and advanced attack classification systems. The knowledge representation system lacks online learning capabilities which limits its ability to adapt when adversarial attack patterns change. The process of filling these gaps needs lightweight representation learning together with continuous adaptation and automated rule generation to execute its function.

Thus, some promising future directions are: (i) ultra-lightweight hybrid models designed especially for low-power microcontroller-class implementations; (ii) full online as well as federated adaptation under evolving threats without requiring centralized data aggregation; (iii) development of multi-agent cooperation among distributed edge nodes; and (iv) measuring trade-offs at a network level between security performance, control overhead, and quality-of-service under actual SDN deployments. Moreover, it is an important but difficult step in operational deployment to extend the model to multi-class attack taxonomies with guarantees on adversarial robustness.

Thus, the study has proven that the integration of deep sequential modeling, uncertainty-aware reasoning and network programmability into one suite will form a promising trail for next generation intrusion detection based on wireless sensor networks. The findings support a paradigm of security mechanisms that are not only traditionally accurate, but are also adaptive, distributed, and integrated within the control fabric of the network- a necessity under emerging IoT and cyber-physical systems embattled in constant uncertainty and adversarial pressure.

Acknowledgement: The authors would like to thank Ninevah University for supporting this article.

REFERENCES

[1] A. John, I. Bin Isnin, S. Madni, M. Faheem, ''Intrusion detection in cluster-based wireless sensor networks: Current issues, opportunities and future research directions,'' *IET Wireless Sensor Systems,* vol. 14, pp. 293–332, 2024, doi:10.1049/wss2.12100.

[2] O. Abdulkareem, R. Kontham, F. Mahmood, ''Collaborative intrusion detection system to identify joint attacks in routing protocol for low-power and lossy networks routing protocol on the internet of everything,'' *Mesopotamian Journal of Cybersecurity*, vol. 4, pp. 251–277, 2024, doi: 10.58496/MJCS/2024/026.

[3] B. Zhu, E. Bedeer, H. Nguyen, R. Barton, J. Henry, ''Improved soft-k-means clustering algorithm for balancing energy consumption in wireless sensor networks,'' *IEEE Internet Things Journal,* vol. 8, pp. 4868–4881, 2024, doi: 10.1109/JIOT.2020.3031272.

[4] X. Yan, C. Huang, J. Gan, X. Wu, ''Game theory-based energy-efficient clustering algorithm for wireless sensor networks,'' *Sensors*, vol. 22, p. 476, 2022, doi: 10.3390/s22020478.

[5] M. Tawfeek, I. Alrashdi, M. Alruwaili, L. Jamel, G. Elhady, H. Elwahsh, ''Improving energy efficiency and routing reliability in wireless sensor networks using modified ant colony optimization,'' *Journal on Wireless Communications and Networking*, vol. 22, 2025, doi: 10.1186/s13638-025-02449-w.

[6] A. Narwaria, A. Mazumdar, ''Software-defined wireless sensor network: A comprehensive survey,'' *Journal of Network and Computer Applications*, vol. 215, p. 103636, 2023, doi: 10.1016/j.jnca.2023.103636.

[7] F. Jurado-Lasso, L. Marchegiani, J. Jurado, A. Abu-Mahfouz, X. Fafoutis, ''A survey on machine learning software-defined wireless sensor networks (ml-SDWSNS): current status and major challenges,'' *IEEE Access*, vol. 10, pp. 23560–23592, 2022, doi: 10.1109/ACCESS.2022.3153521.

[8] P. Rao, A. Lipare, D. Edla, S. Parne, ''An energy-efficient routing algorithm for WSNs using fuzzy logic,'' *Sensors*, vol. 23, p. 8074, 2023, doi: 10.3390/s23198074.

[9] A. Alashjaee, ''Deep learning for network security: an Attention-CNN-LSTM model for accurate intrusion detection,'' *Science Reports*, vol. 15, p. 21856, 2025, doi: 10.1038/s41598-025-07706-y.

[10] S. Bukhari, M. Zafar, M. Abou Houran, S. Moosavi, M. Mansoor, M. Muaaz, F. Sanfilippo, ''Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability,'' *Ad Hoc Networks*, vol. 155, p. 103407, 2024, doi: 10.1016/j.adhoc.2024.103407.

[11] G. Gebremariam, J. Panda, S. Indu, ''Design of advanced intrusion detection systems based on hybrid machine learning techniques in hierarchically wireless sensor networks,'' *Connection Science*, vol. 35, p. 2246703, 2023, doi: 10.1080/09540091.2023.2246703.

[12] R. Elsayed, R. Hamada, M. Abdalla, S. Elsaid, ''Securing IoT and SDN systems using deep-learning based automatic intrusion detection,'' *Ain Shams Engineering Journal*, vol. 14 p. 102211, 2023, doi: 10.1016/j.asej.2023.102211.

[13] A. Behera, K. Sahoo, T. Mishra, M. Bhuyan, ''A combination learning framework to uncover cyber attacks in IoT networks,'' *Internet of Things*, vol. 28, p. 101395, 2024, doi: 10.1016/j.iot.2024.101395.

[14] H. Saleh, H. Marouane, A. Fakhfakh, ''Stochastic gradient descent intrusions detection for wireless sensor network attack detection system using machine learning,'' *IEEE Access*, vol. 12, pp. 3825–3836, 2024, doi: 10.1109/ACCESS.2023.3349248.

[15] M. Behiry, M. Aly, ''Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods,'' *Journal of Big Data*, vol. 11, 2024, doi: 10.1186/s40537-023-00870-w.

[16] H. Satori, et al, ''Machine learning attack detection based-on stochastic classifier methods for enhancing of routing security in wireless sensor networks,'' *Ad Hoc Networks*, vol. 163, p. 103581, 2024, doi: 10.1016/j.adhoc.2024.103581.

[17] A. Alshehri, ''Wormhole attack detection and mitigation model for Internet of Things and WSN using machine learning,'' *PeerJ - Computer Science Journal*, vol. 10 p. e2257, 2024, doi: 10.7717/peerj-cs.2257.

[18] B. Al-Fuhaidi, Z. Farae, F. Al-Fahaidy, G. Nagi, A. Ghallab, A. Alameri, ''Anomaly-based intrusion detection system in wireless sensor networks using machine learning algorithms,'' *Applied Computational Intelligence and Soft Computing*, vol. 2024, p. 2625922, 2024, doi: 10.1155/2024/2625922.

[19] C. Subasini, S. Karuppiah, A. Sheeba, S. Padmakala, ''Developing an attack detection framework for wireless sensor network-based healthcare applications using hybrid convolutional neural network,'' *Transactions on Emerging Telecommunications Technologies*, vol. 32, p. e4336, 2024, doi: 10.1002/ett.4336.

[20] S. Salmi, L. Oughdir, ''Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network,'' Journal of Big Data, vol. 10, p. 17, 2023, doi: 10.1186/s40537-023-00692-w.

[21] T. Nguyen, H. Vo, M. Yoo, ''Enhancing intrusion detection in wireless sensor networks using a GSWO-CatBoost approach,'' *Sensors,* vol. 24, p. 3339, 2024, doi: 10.3390/s24113339.

[22] M. Elsadig, ''Detection of denial-of-service attack in wireless sensor networks: A lightweight machine learning approach,'' *IEEE Access*, vol. 11, pp. 83537–83552, 2023, doi: 10.1109/ACCESS.2023.3303113.

[23] O. Hussain, Z. Chen, H. Zhu, ''Secure Net: A Hybrid CNN-LSTM-based Intrusion detection system for securing IoT networks,'' 4th International Conference on Computer, Artificial Intelligence and Control Engineering, 2025, doi: 10.1145/3727648.3727736.

[24] L. Baniata, A. ALDabbas, J. Atwan, H. Alahmer, B. Elmasri, C. Bunterngchit, ''A dual-attention cnn–gcn–bilstm framework for intelligent intrusion detection in wireless sensor networks,'' *Future Internet*, vol. 18, 2025, doi: 10.3390/fi18010005.