Al-Qadisiyah JournalFor Engineering Sciences 19 (2026) 001 – 010

Contents lists available at: http://qu.edu.iq

Al-Qadisiyah Journal for Engineering Sciences

Journal homepage: https://qjes.qu.edu.iq

Research Paper

FOG computing deep learning-based intrusion detection model for IOT network security

Karrar Falah Sailan and Mohsen Nickray

Department of Computer Engineering and Information Technology, University of Qom, Alghadir Ave., P.O. Box 3716146611, Qom, Iran.

ARTICLE INFO

Article history:
Received 26 April 2025
Received in revised form 29 October 2025
Accepted 25 April 2026

keyword:
Fog Computed Networks
Cyber-Security
Deep Learning
Internet of Things
Intrusion Detection

ABSTRACT

The rapid expansion of Internet of Things (IoT) communications across fog networks has led to a significant increase in security concerns and cyber threats due to the multiplicity and diversity of violations and security vulnerabilities. Due to inadequate security, a large number of IoT devices are vulnerable to malware, security breaches, and cyberattacks. It has been challenging for conventional intrusion detection systems (IDS), which rely on signature-based detection, to identify novel and unexpected attacks. Thus, deep learning and machine learning technologies offer interesting options for IoT security. A type of deep recurrent neural network (DRNN) called bidirectional long short-term memory networks (Bi-LSTM) is proposed in this paper as an intelligent intrusion detection system (IDS) for the Internet of Things (IoT). Through the careful analysis of complex network traffic patterns, the system can detect known and novel attacks with high accuracy and low false alarm rates. The training includes neural networks, anomaly detection, rare feature extraction, and data processing, all of which are parts of the process. The model is evaluated using accuracy, recall, F1 score, and false alarm rates after being trained with 70% of industrial datasets generated by MATLAB that are similar to real data (such as CICIoT2023 and CICIDS2017) and 30% of the time for testing. The results show that while it increases intrusion detection accuracy, the proposed intrusion detection system provides high-precision security monitoring and practical software simulation. The main contributions include efficient intrusion detection, improved detection accuracy, and increased data security. Because the model works with a number of Internet of Things communication protocols, it provides a practical security solution for protecting IoT networks from advanced cyberattacks. Simulations emphasize reproducibility, and strategies are planned using a standard dataset from Internet sources. She participated in creating and evaluating a deep learning model, achieving a training accuracy of 99.991% in detecting and preventing attacks.

© 2026 University of Al-Qadisiyah. All rights reserved.

1. Introduction

1.1 Outline

Monitoring cyberattacks has become a critical issue due to the variety of cyberattacks, their use of contemporary and distinctive programming, and the expansion of the scope of electronic breaches by computer and cloud network programmers. However, there are two strategies to counter DDoS attacks. Offering a directly equal token is the basic procedure, and tokens are the end result. Direct attacks target any information system design defect that could cause damage or even service termination. Furthermore, hostile assaults search for different parts of the system that are connected to other parts in order to attack and distort the content and flow of information. Better security systems for a variety of real-world applications are becoming more and more necessary in order to guard against internal and external breaches and to safeguard the data and information of organizations that use the Internet and communications networks [1–3]. To address these basic requirements, effective algorithms that work with WSNs to provide highly reliable cyberattack detection and prevention through a range of methods are proposed. Evaluation and control algorithms are the two main algorithms used in cloud computing planning and cybersecurity aggregation. The evaluation, such as closed-loop security objectives.Ïn theory, the primary goal of security is to protect the electronic system from malicious attacks. Reaching the level of security, however, also requires managing all factors and emergency scenarios. To ensure the legitimacy of the data, security measures must also be used when tags and data—including private, sensitive information—are collected from sensors of various computer network units. The WSN's building schematic diagram is displayed in Fig. 1 [4, 5]. The number of diverse cyber-attacks on multinational corporations has increased in recent years, making cybersecurity a high and growing priority. According to the literature, cyber-security is the most pressing issue affecting IT management, with 80 percent of companies allocating more than twenty percent of their IT budget to cyber-security, a doubling advance in only a few years. Cybersecurity reports classify the most common cyber-attacks toward 3 divisions: the internet, the cloud, and the Internet of Things. The Internet Security Threat Report (ISTR) found that the sum of web assaults on endpoints advanced more than 55 percent, with nasty links settle in single out of every 10 links examined [6]. A network of physical things, including cars, appliances, and other items, that are equipped with sensors, software, and network connectivity so they can gather and exchange data, is known as the Internet of Things (IoT). The Internet of Things (IoT) is an advanced concept of the Internet with the objective of connecting everything in our lives to the Internet and each other, sending and receiving messages, and obtaining information to perform particular tasks over the Internet in order to simplify our lives, increase task efficiency, decrease error rates, and enhance appropriate performance as in Fig. 2 [7, 8].

*Corresponding Author.
E-mail address: m.nickray@qom.ac.ir ; (Mohsen Nickray)

https://doi.org/10.30772/qjes.2026.166617.1792

2411-7773 © 2026 University of Al-Qadisiyah. All rights reserved. This work is licensed under a Creative Commons Attribution 4.0 International License.

Accepted Manuscript...Accepted Manuscript...Accepted Manuscript

Nomenclature

ANN	Artificial Neural Networks	*LSTM*	Long Short-Term Memory
xCNNxx	Convolutional Neural Networks	*PSO*	Particle Swarm Optimization
DDoS	Distributed Denial of Service	*RNN*	Recurrence Neural Networks
GANs	Generative Adversarial Networks	*SGD*	Stochastic Gradient Descent
GRU	Generalized Recurrent Unit	*WSNs*	Wireless Sensor Networks

Inspired by the human brain, Deep Learning Neural Networks (DLNNs) are sophisticated computational models that use layered architectures to identify patterns and reach conclusions. They are put into practice using frameworks such as Tensor Flow, PyTorch, or Keras, in which input layers process data, hidden layers transform it (using activation functions like ReLU or Sigmoid), and output layers finalize the predictions. Backpropagation and optimization algorithms (such as Adam and SGD) are used in training in order to reduce loss functions. Generative Adversarial Networks (GANs) are used for data creation, Convolutional Neural Networks (CNNs) for image processing, Recurrent Neural Networks (RNNs) for sequential data, Long Short-Term Memory (LSTM) networks (a specialized RNN variant for long-term dependencies), and Transformers for natural language applications. They are crucial for automating feature extraction, managing unstructured data (text, photos), and reaching cutting-edge accuracy in tasks like autonomous systems, medical diagnostics, and speech recognition.

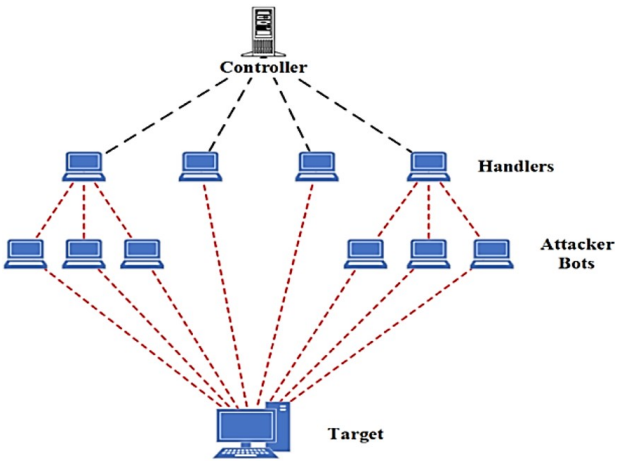


Figure 1. Schematic diagram of typical network cyber-attacks strategy [3].

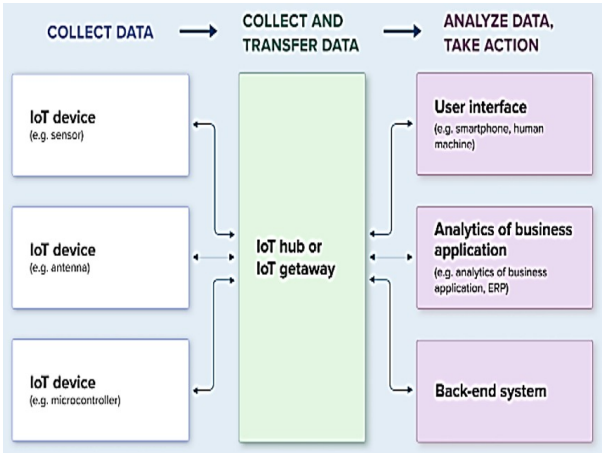


Figure 2. Example of standard Its network structure [7].

We will describe the long short-term memory (LSTM) technique, a specialized RNN variant for long-term dependencies, and the recurrent neural network (RNN) techniques used in processing sequential time series data in order to comprehend some of the most significant deep learning approaches. Fig. 3 demonstrates the construction of the RNN model [9–11]. As observed from Fig. 3, the feedback might be autonomic, or at the very least, the result of the action that was not totally predetermined by its preparation strategy. Since an intelligent lattice is expected to contain backhanded units, the feedback system's implementation of postponed unit portions with a few districts leads to a non-linear dynamic style of acting. Although the methods of inner linking may vary throughout optional types, they always achieve the same goal and desired result—applying repetition. RNNs might handle posting successions of varying dimensions with display physical kinetics since they have internal storage.

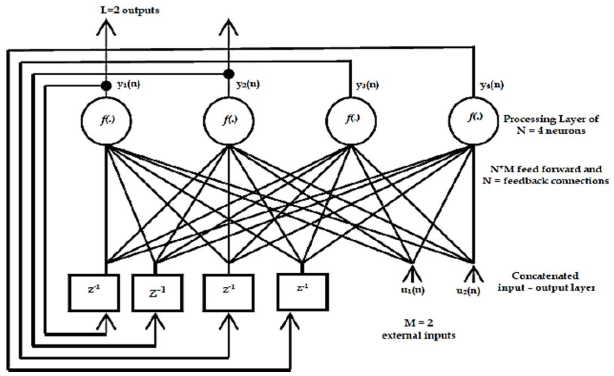


Figure 3. The Recurrent Neural Networks, RNN structure [9].

The "WSNs"promise to provide the insurance currently used in an additional substance highlight that is viable with protection against attacks and breaks, as well as providing adaptability—the feature of the framework that is viable with survival and recovery after the assault or break—is mentioned in this region for the introductions of this examination. Online preliminary overpowering is being threatened by appropriate denial of service (DDoS) assaults. In a DDoS attack, several bundles are communicated to a designated server, exhausting the organization's transmission capacity or the casualty's storage. Programming for DDoS attacks has been around for a while, and there are several defence techniques available to combat exclusive-resource attacks. With the use of more advanced capabilities, the stock of such attacks might then be successfully prevented or justified. Nevertheless, there are a vast number of helpless frameworks from which invaders may select. Fig. 4 demonstrates the effect of the Denial of Service (DoS) attack [12–14].

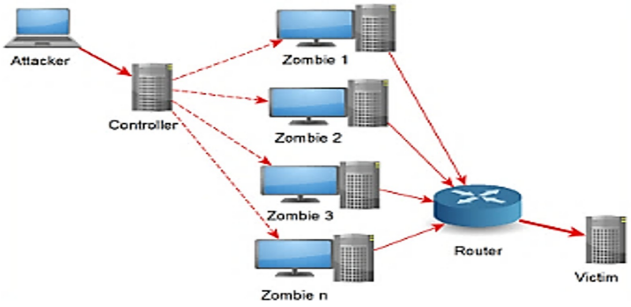


Figure 4. The effect of the Denial of Service (DoS) attack [12].

Fig. 4 might be seen as an example of a simple RNN architecture. One may see that the RNN algorithm design presented in Fig. 4 has common sections that use the aperture $Xt-w:t-1$ to approximate the advance instant print as a consequence, $x't$. The input arrangement is handled according to the organization's timestamp on a regular basis. This approach, the following samples $x't$ are assessed using the expression below Eq. 1, utilizing the inputted grouping $xt-1$ of the repeated entity $ot-1$ and the enactment capacity as $tanh$ [12–14].

$$x'_t = \sigma\left(W_{x',O_{t-1}} + b_{x'}\right) \quad (1)$$

$$Where\ O_{t-1} = tanh\left(W_{O,x_{t-1}} + U_{O,O_{t-2}} + b_h\right)$$

Accepted Manuscript...Accepted Manuscript

The long-term and short-term assumption network is really trained here. Eq. 1 illustrates the three primary algorithm types that comprise recurrent neural networks: (1) the long short-term memory (LSTM) method, (2) the generalized recurrent unit (GRU) strategy, and (3) the recurrent neural network (RNN) algorithm. An LSTM cell has three gates; the input gate (i), forget gate (f), and output gate (o), along with a cell state (C) and hidden state (h). Thus, concerning the forget gate which find what info to be discarded along the cell state as follows Eq. 2, [15–17].

$$f_t = \sigma\left(W_f \times [h_{t-1,x_t}] + b_f\right) \tag{2}$$

Whereas σ denotes the sigmoid activation function, Wf, indicates the weight matrix for the forget gate ht–1, represents the previous hidden state, xt, denotes the current input, and bf, indicates the bias term. Also, the input gate decides what advance info to be saved in the cell state and might be expressed as Eq. 3 and Eq. 4.

$$i_t = \sigma\left(W_i \times [h_{t-1,x_t}] + b_f\right) \tag{3}$$

$$\tilde{C}_t = tanh\left(W_c \times [h_{t-1,x_t}] + b_c\right) \tag{4}$$

Such that, $\tilde{C}_t$, denotes the (new memory) candidate cell state, tanh, indicates the activation hyperbolic tangent function. Next, concerning the update cell state which merges the forget gate with the input gate to update the cell state as Eq. 5.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{5}$$

In which $\odot$, represents the Hadamard product or element-wise multiplication. Finally, the output gate controls what section of the cell state results as the hidden state, as Eq. 6 and Eq. 7

$$O_t = \sigma\left(W_o \times [h_{t-1,x_t}] + b_o\right) \tag{6}$$

$$h_t = O_t \odot tanh\, C_t \tag{7}$$

At last, we could summarize the LSTM variables as presented in Table 1.

Table 1. Summarizing of the LSTM variables.

Symbol	Description
x_t	Input at time t
h_t	Hidden state at time t
C_t	Cell state at time t
f_t, i_t, o_t	Forget, input, and output gates
$\tilde{C}_t$	Candidate cell state
W, b	Weights and biases
σ	Sigmoid function
$tanh$	Hyperbolic tangent

1.2 Literature Review

The topics of WSN cloud security and Deep Learning (DL) algorithms are examined in several publications and earlier research projects. To make clear the most significant issues and difficulties, as well as the most significant limitations and methods to overcome them, the most sophisticated and essential papers and scientific publications that have been presented. Table 2 summarizes the most recent related publications to our study subject. Table 3, and Table 4 summarize the critical analysis of the relevant studies with a discussion of how these gaps can be filled through the contributions of researchers in previous studies.

Table 2. Summary of most recent published articles [18–22].

Year	Authors	Technology Employed	Contributions	Limitations
2022	Shi et al.	ANN with Particle Swarm Optimization	Privacy-focused IDS for WSNs	Limited dataset evaluation and scalability concerns
2023	Belarbi et al.	Federated Deep Learning	Privacy-preserving IDS for IoT using federated learning	Performance affected by data heterogeneity
2024	Gueriani et al.	CNN-LSTM Hybrid Model	High-accuracy IDS capturing spatial-temporal features	Computational complexity for real-time deployment
2024	Shen et al.	Federated Learning with Knowledge Distillation	Enhanced IDS performance addressing data heterogeneity	Increased system complexity and communication overhead
2024	Gowdhaman et al.	ResNet-Inception DL + SVM	High-accuracy hybrid IDS on NSL-KDD dataset	Limited evaluation on a single dataset

Table 3. Critical analysis by references.

Reference	Strengths	Weaknesses / Concerns
Shi & Li (2022) ANN + PSO	• Optimization improves ANN convergence and avoids local minima. • Focus on privacy in WSNs (practical constraint).	• Evaluated on limited datasets → risk of poor generalization. • Scalability issues: PSO tuning is costly for large-scale/online use. • No clear discussion of energy/CPU overhead in constrained WSNs.
Belarbi et al. (2023) – Federated Deep Learning for IoT IDS	• Preserves data privacy by training locally. • Architecture matches distributed IoT.	• Struggles with non-IID data distributions. • High communication cost, no quantified overhead analysis. • Does not address client dropout or stragglers in detail.
Gueriani et al. (2024) – CNN-LSTM Hybrid	• Good combination of spatial (CNN) + temporal (LSTM) features. • High reported accuracy.	• High computational complexity, heavy for real-time/edge devices. • No latency/throughput metrics → unclear deployment feasibility.
Shen et al. (2024) – Federated Learning + Knowledge Distillation	• Promising solution for non-IID federated data. • Model compression via distillation benefits resource-limited clients.	• Adds extra system complexity (teacher/student coordination). • Communication overhead not fully quantified.
Gowdhaman & Dhanapal (2024) – ResNet-Inception + SVM on NSL-KDD	• Strong accuracy on NSL-KDD dataset. • Hybrid approach leverages deep features + classical classifier.	• Reliance on NSL-KDD only → outdated, not reflective of modern IoT traffic. • Lacks evaluation on encrypted traffic or diverse IoT devices.

Accepted Manuscript...Accepted Manuscript...Accepted Manuscript

Table 4. The related studies' gaps analysis.

Gap	How Prior Works Fall Short	How This Work Addresses It
Dataset diversity	NSL-KDD or single dataset reliance (Shi & Li, Gowdhaman & Dhanapal).	Evaluate on multiple datasets (modern IoT, real traces), cross-dataset generalization.
Non-IID federated data	Belarbi & Shen note the issue but limited mitigation.	Use personalization, clustering, or hybrid FL+distillation with convergence validation.
Real-time deployment	CNN-LSTM and ResNet heavy; PSO tuning costly.	Lightweight backbone (pruned/quantized models), benchmarked latency and energy on edge devices.
Communication overhead	FL approaches not quantified.	Measure bytes/client/round, apply compression/sparse updates, report overhead.
Robustness to attacks & drift	No evaluation of poisoning/evasion or traffic drift.	Simulate poisoning/evasion, drift scenarios; add adaptive retraining mechanisms.
Explainability	All black-box.	Provide interpretable fuzzy/ANFIS rules or feature-importance outputs.
System-level testing	No evaluation under churn/delays.	Test under client dropout, network latency, scalability curves (clients vs accuracy).

1.3 Problem Statement (Challenges)

The usage of Wireless Sensor Networks (WSNs) is growing in critical environments including the military, healthcare, and industrial automation where reliable and secure communication is critical. However, because of their resource-constrained nature—which includes limited processing power, energy, and memory—WSNs are particularly vulnerable to a range of security threats, including as data tampering, eavesdropping, spoofing, and denial-of-service (DoS) attacks. Traditional cryptographic and rule-based solutions can occasionally fall short in terms of adaptive and scalable security, particularly when it comes to new and evolving attack patterns. Furthermore, due of the distributed nature and dynamic topology of WSNs, real-time threat detection is a persistent problem. Thus, developing robust, clever, and low-latency security techniques for WSNs remains a research issue.

2. Methodology

This section will describe and explain, with the use of deep learning, the crucial design stages for putting into practice the suggested intrusion detection systems (IDS) security model in order to detect and stop cyber-attacks and Distributed Denial of Service (DDoS) against WSNs. Information will serve as the foundation for the operation of our suggested architecture, which should handle two categories of data: original information indexes and virus information indexes. Many kinds of necessary data may be found on reliable websites. For this study, we used two primary websites for information processing (github.com and kaggle.com), in addition to re-presenting certain data utilizing helpful MATLAB program components. This section will introduce incursion Detection Systems (IDS), which use an AI (PC-based intelligence) algorithm to detect cyberattacks and identify Denial of Service (DDoS) network incursion executions.

2.1 The Proposed Model

In this study, MATLAB m-files scripts will be used to equip and simulate the cybersecurity architecture details. The proposed cyber-security architecture includes the following incoming units: 1) Fog computed IoT Wireless Sensor Network Identification Unit, which displays the network's characteristics, node count, and related links, 2) the input unit that makes it possible for information and data to enter the network, 3) a checkpoint or control unit that verifies the state of data entering the network, 4). The Department of Analysis, Inspection, and Classification, which offers the comprehensive assessment and categorization of data posted on the network, 5) Computer-Based Intelligence's Deep Learning Algorithm Unit, which oversees the process of identifying assaults, malware, random flows, and odd software information and separating them from the data and information set; and 6). Final inspection and verification unit, which operates to confirm and verify the movement of information and data through the network structure, and ensures that it is free from attack flows or any malicious software. The flowchart in Fig. 5 provides a summary of how the suggested security architecture for the wireless sensor network operates. The proposed model, as illustrated in Fig. 5, first specifies the architecture and structure of the Fog wireless sensor network (WSN) by varying the number of nodes and connections among them to determine the network's size. Then, the dataset flow have been loaded, and the cyber-attack flows (floods) are added. After attack verification, a deep learning DL-LSTM approach is employed to identify and stop assaults. The results are then displayed once security and accuracy rates have been verified. The findings are then shown when security and accuracy rates have been confirmed. Moreover, the operation of the proposed deep learning (LSTM) algorithm in detecting and preventing cyber attack floods will be explained as introduced in Fig. 6.

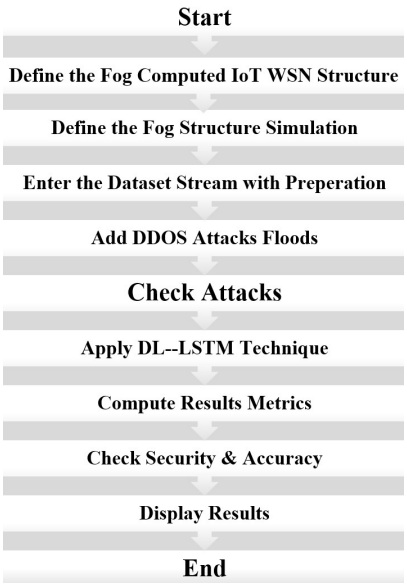


Figure 5. methodology of the proposed Fog-computed IoT WSNs security model.

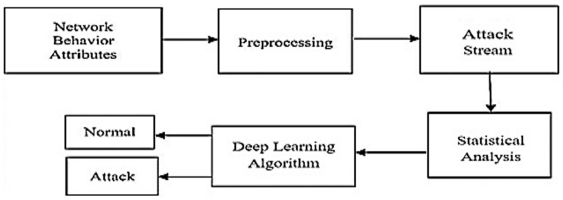


Figure 6. Block diagram of the proposed IDS security system for Fog computed IoT in WSNs.

By introducing (N = 10000 tests) to distinguish the training algorithm parameters to the extent that the security rates are appropriate and an appropriate MSE rate is reached, the candidate program model explains how the deep learning (LSTM) algorithm operates based on the training data set. The primary data transmitted via the communication network is represented by data information, which is necessary to comprehend the deep learning algorithm model. To prepare it for training, such data are uploaded by gathering them into the input layer of the LSTM algorithm. To improve categorization, such data is handled in classification procedures, which subsequently provide a basic fingerprint before altering it by obtaining further encoding in accordance with the basic name. These amounts are then merged in the pooling layer after the resulting capacitive gains are lowered by passing them via the Relu layer. To prepare and

Accepted Manuscript...Accepted Manuscript...Accepted Manuscript

update the algorithm layers to identify the results and infer the error magnitude and best match, the data is processed in the internal weight layers after these stages. Lastly, Table 5, Table 6 illustrates the remaining variables of the suggested model together with the settings and control parameters of the suggested LSTM method.

Table 5. Detailed design specifications of the proposed IDS Fog computed IoT WSN model.

Unit Type	Specifications	Details
Fog IoT WSN	Nodes Number	N=100
	Data Length / Node Stream	L=100
	Network Connection	Random
	Nodes Energy	1 Joule
Data Stream	Samples Number	$N_s = 10,000$
	Data Type	UNSW-NB15
	Data Distribution	Gaussian-like Clean Data
	Capacity	5 Volts
Attack Flood	Samples Number	$N_s = 10,000$
	Attack Type	DDoS (Flood-like Oscillatory)
	Attack Distribution	Oscillatory
	Capacity	1 Volt

Table 6. Detailed design parameters of the proposed Bi-LSTM structure for the IDS Fog computed IoT WSN model.

Unit Type	Specifications	Details
Deep Learning Bi-LSTM	Fully Connected Layer Neurons	50
	Hidden Neurons in Bi-LSTM Layers	200 (per direction)
	Input Layer Type	Sequence Input
	Output Layer Type	Regression Output
	Sequence Length	100 (time-steps)
	Bidirectional Layers	2 Bi-LSTM layers stacked
	Dropout Rate	0.2 applied after each Bi-LSTM layer
	Activation Functions	Tanh in Bi-LSTM; Linear in output
	Batch Size	64
	Learning Rate	Initially 0.001, tuned via grid search
	Epochs	20-50
	Optimizer	Adam
	Data Splitting	Random with seed 20
	Validation / Testing Splits	15% validation, 15% testing, 70% training
	Cross-validation	5-Fold Cross Validation
Hyperparameter Tuning Process	Learning rate, batch size, dropout tuned by grid search with validation loss monitoring	Early stopping based on validation loss

Finally, the employed dataset used to implement the proposed model for intrusion detection and prevention in this study are defined and clarified. We first need to understand the nature and sources of the dataset used. Two types of data are utilized in this study: the first type is the network dataset (information), and the second type is the attack dataset (intrusive flows). Table 7 shows the details of the datasets used to implement the proposed intrusion detection and prevention model for this study. The UNSW-NB15 and CIC-DDoS2019 datasets contain heterogeneous feature metrics (for example, packet sizes, data-flow times) which will be outlined in Table 8.

2.2 Approximate Simulation of Fog-IoT Traffic Designs

This study conducted an approximate software simulation of Fog-IoT network traffic using the CICIoT2023/CICIDS2017 datasets (processed with MATLAB R2020b), simulating 'voltage/time' sequences similar to digital signals to represent MQTT/CoAP burst traffic while considering the software limitations of resource-constrained devices. These designs provide representative patterns of real-world networks, enhancing the reliability of Bi-LSTM (97.85%) in real-time environments. A software bridge to actual 6G Internet of Things designs is provided by this simulation Table 9.

Table 7. Details of the features and resources of the employed dataset with their design parameters.

Dataset name	Layer	Main Features	Samples Number	Data Resource
CIC-DDoS 2019	Network Access	Distributed Nature High Traffic Volume Exploited Vulnerability	10^4	Dataset for DDoS botnet attacks by IOT devices Kaggle

Table 8. Details of the features and resources of the employed dataset with their design parameters.

Aspect	UNSW-NB15	CIC-DDoS2019	Notes
Normalization	Required; min-max or z-score	Required; min-max or z-score	Normalize per fog node after splitting
Feature Selection	Filter/wrapper approaches for relevant features	Similar feature selection needed	Select node-relevant features; consider distributed feature sets
Class Distribution	Moderate imbalance; multiple attack types	Highly imbalanced; minority attacks	Balance within nodes; anomaly detection suited for edge environments
Dataset Splits	Use train-validation-test (e.g., 60/20/20) stratified	Similar stratified splitting recommended	Use federated or distributed cross-validation in simulation
Fog Computing Simulation	Not directly addressed in most papers	Not directly addressed	Partition dataset by fog node; model latency; distributed model training

Table 9. Comparison of approximate simulation with real designs.

Feature	Approximate Simulation	Real Design (IoT)
Packet Size	64-1500 bytes (simulated)	20-200 bytes (bursty)
Flow Duration	$>$ 60 seconds (calibrated)	1-30 sec. (event-driven)
Flags	SYN/FIN (programmatic)	PSH/FIN (MQTT)
Transmission Rate	Regular (approximate)	Bursty (periodic)
Protocols	TCP/UDP (simulated)	CoAP/MQTT (repres.)

3. Results and discussion

To stop and secure wireless network invasions, the suggested technological model is used in this section to identify malware and cyberattacks. Artificial intelligence (deep learning) techniques and software simulations utilizing the MATLAB 2020b program are used to verify whether security criteria are met. Additionally, text logs are emulated using MATLAB library functions are also used to model distributed DDoS attack flows. With the use of artificial intelligence techniques and a deep learning recurrent neural network (LSTM) algorithm, this model illustrates how DDoS assaults affect a UNSW-NB15 dataset on a wireless cloud network and how to manage and identify such intrusions. Moreover, comparison validation analysis have been provided among our proposed DL LSTM model and other machine learning algorithms. The hardware and software design specifications are shown in Table 10.

3.1 Simulation Results

The proposed model for detecting and securing wireless sensor networks is implemented using machine learning, based on the data and design factors outlined in the previous section. A simulation is then run using MATLAB to extract, display, and explain the results. Fig. 7 shows the implementation diagram of the proposed WSN architecture design employed to achieve the work environment. By observing Fig. 7, we notice that the implementation plan of the virtual Fog computed IoT WSN structure and design randomly distributes network nodes (N = 50) to ensure fair data transfer between them to achieve a working environment, where information data is sent and received between the distributed nodes, which represent wireless communication stations.

Accepted Manuscript...Accepted Manuscript...Accepted Manuscript

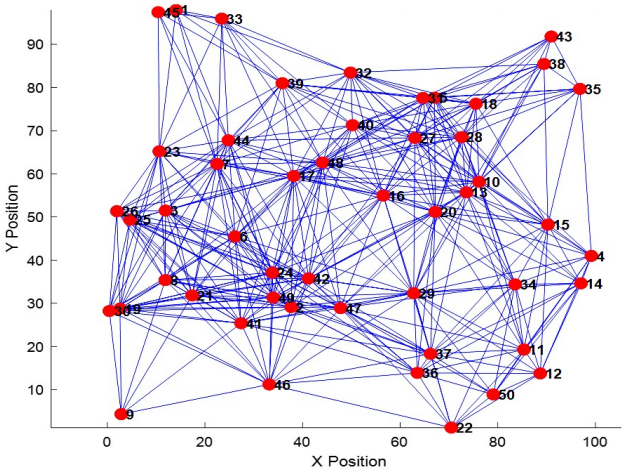


Figure 7. Design details of the proposed LSTM deep learning technique.

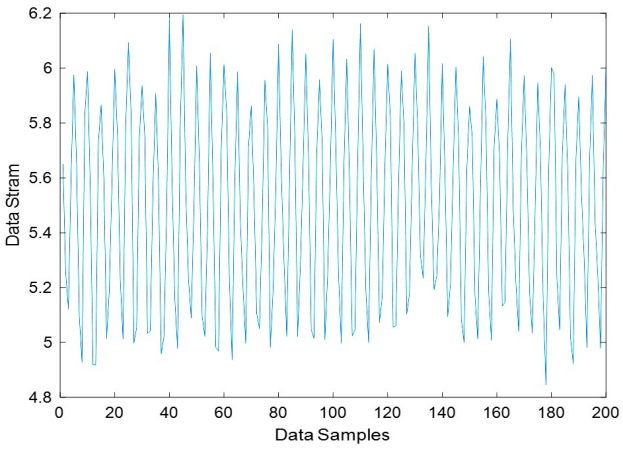


Figure 8. The simulation of the proposed Fog computed IoT/WSN architecture design employed to achieve the study environment.

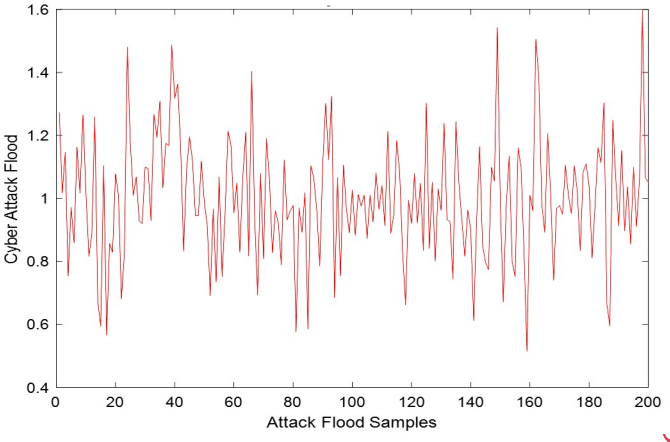


Figure 9. The distribution of the (info) dataset stream through the network nodes.

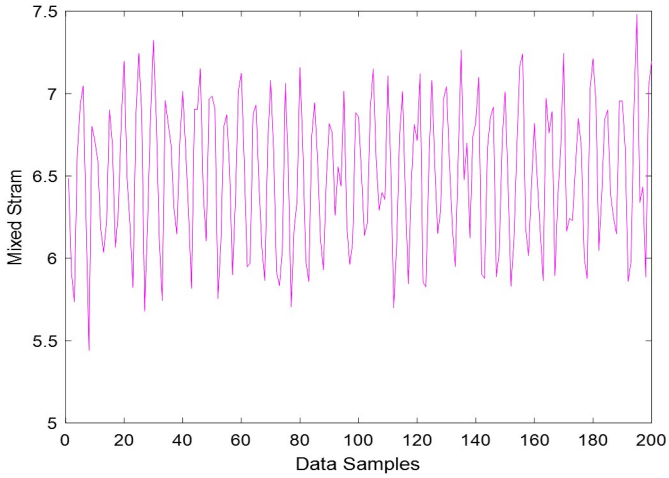


Figure 10. The generated cyber-attacks flood samples.

Table 10. The hardware and software design specifications.

PC Structure	Specifications	Notes
Hardware	• Processor: Intel Core i7 (or equivalent) 8th generation or higher • RAM: Minimum 16 GB • Storage: SSD with at least 256 GB free space • CPU used for training in simulation (as mentioned: single CPU)	Enable handling the computational load of LSTM training on fairly large datasets (10,000+ samples) without requiring GPU acceleration for rapid prototyping and validation.
Software	• MATLAB R2020b (or newer) with Deep Learning Toolbox • MATLAB built-in functions for network simulation and dataset emulation • Operating System: Windows 10 or Linux distribution supporting MATLAB	

Table 11. The Fog-Computed IoT wireless sensor network (WSN) specification presented in Fig. 7.

Parameter	Value	Notes and Justifications
Number of Nodes (N)	100	Sufficient to simulate distributed WSN with random network topology, allowing variability in data routing.
Node Data Stream Length (L)	100	Represents fixed size data windows per node for sequence learning in LSTM.
Network Connection Type	Random	Simulates non-fixed mesh topology reflective of real IoT wireless networks with dynamic connectivity.
Node Energy Level	1 Joule	Low energy highlighting resource-constraints typical in fog-IoT devices, supporting energy-aware computing.
Data Samples per Node	10,000	Large sample size to ensure statistical significance in training and evaluation.
Data Distribution	Gaussian-like	Represents realistic yet controlled background traffic for anomaly detection.
Attack Samples	10,000	Simulated DDoS floods modeled as oscillatory traffic to represent cyber-attack injections.

Accepted Manuscript...Accepted Manuscript...Accepted Manuscript

Thus, the Fog-Computed IoT wireless sensor network (WSN) specification presented in Fig. 7 might be listed in Table 11. Next, Fig. 8 displays the distribution of the (info) dataset stream through the network nodes. By viewing Fig. 8, it could be noted that the flow of data forms for information between the wireless network contract simulates the coordination of the TCP/IP Internet protocol and fluctuates almost randomly with a price value of the volt. Where this data contains user information and differs according to the content of this declaration. Also, the cyber-attacks flood samples are generated and presented in Fig. 9. As we might observe from Fig. 9, that the cyber-attacks flood generated using MATLAB built in library functions and acting as a DoS attack, which sends many requests or traffic to exhaust the node network or resources (such as the frequency range, the CPU or the battery). As a result of adding the DoS attack flood to the data stream samples, the obtained mixed corrupted stream is demonstrated in Fig. 10. By reviewing Fig. 10 above, the signal content resulting from inserting the flows of the service attack models into the original data models can be seen. We note that the resulting wave has been affected and distorted because of adding intrusive model flows despite its small capacity, but it changes the content of the basic data information sent and leads to a defect, deformation, or cutting in the service. Now, by applying the operation of the proposed deep learning LSTM algorithm, to detect and reject the effect of the DoS cyber-attack floods. The flow diagram of the proposed DL LSTM layer's structure is outlined in Fig. 11. By referring to the above figure, the results of the program's implementation of the proposed form can be observed and able to determine that the wireless network has been subjected to a cyber-attack Fig. 11a shows the details of the contents of the deep learning algorithm layers, which are 1) " Sequence Input Sequence input with 500 dimensions, 2) " LSTM with 200 hidden units, 3) 'Fully Connected 50 Fully Connected Layer, 4) " Dropout 50% Dropout, 5) " Fully Connected 500 Fully Connected Layer, and 6) " Regression Output Mean-Squared-Error. Also Fig. 11b displays the same proposed deep learning algorithm structure in graph view. Next, the training progress diagram of the deep learning algorithm operation is displayed as the corrupted mixed flows of network data enter it and perform the training process for its internal layers, as shown in Fig. 12. As shown in Fig. 12, we observe the results of the progress of intelligent algorithm training processes that show the levels of error box drop and loss level of original data extraction, and strongly discarded and disposed of harmful flows as the training steps progress. This decrease in the level of quadratic error and losses explains the extent of improvement and ability of the DL algorithm to extract the original data and renounce harmful flows and get rid of them efficiently, and the fewer losses and the error box, the better the results of the training. Moreover, the results of the predicted samples at the output of the trained deep learning (LSTM) algorithm have been shown in Fig. 13. Furthermore, by observing Fig. 13, we could observe the expected sample results at the output of the trained deep learning algorithm (LSTM), which were highly consistent with the original data samples of the information sent through the wireless sensor network. This indicates the success of the process in detecting malicious attack flows as well as the efficiency and effectiveness of the proposed algorithm in eliminating malware and extracting real data. Also, based on the obtained results, we could employ a comparison between the predicted results and the original data sent to calculate the error rate and the efficiency of the prediction.

Table 12**.** The final performance metrics computed for the proposed IDS model using LSTM technique.

Records	Accuracy	Precision	Recall	F Score	Error Rate
TPR 9.99	99.9906%	99.9900%	99.9910%	99.9800%	00.9400%
TPR 9.98					
TPR 0.01					
TPR 0.02					

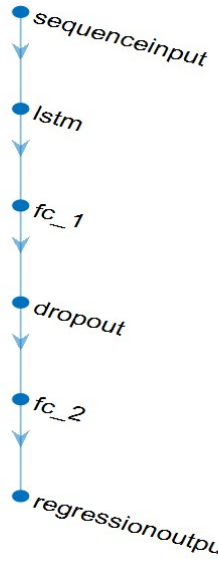
Table 13**.** Summary of the computed latency, throughput, and resource computations metrics.

Metric	Result (Bi-LSTM in Fog WSN)
Latency	$< 1second$ for detection; $\approx 44s$ per batch training (CPU)
Throughput	$> 10,000$ samples/node, 99.99% detection rate
Resource Consumption	1 Joule/node energy, $< 20ms$ latency overhead, CPU-only feasible, stable under DDoS load

Figure 14 displays the efficiency results and the error rate for the prediction indicators of the proposed DL LSTM algorithm.

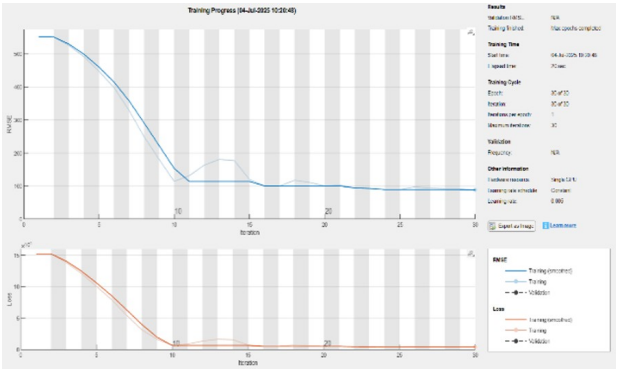
```
--- Detection Result ---
● Cyber Attack Detected
layers =6×1 Layer array with layers:
1"  Sequence Input      Sequence input with 500 dimensions
2"  LSTM               LSTM with 200 hidden units
3"  Fully Connected     50 fully connected layer
4"  Dropout            50% dropout
5"  Fully Connected     500 fully connected layers
6"  Regression Output   mean-squared-error
```

(a) Model 01



(b) Model 02

Figure 11**.** The flow diagram of the proposed DL LSTM algorithm layers structure through applying it to detect and reject the effect of the DoS cyber-attack floods, (a) MATLAB command prompt description, (b) Layers graph view.



(a) Model 01

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch RMSE	Mini-batch Loss	Base Learning Rate
1	1	00:00:44	112.73	6353.6	0.0050
30	30	00:06:19	28.18	397.2	0.0050

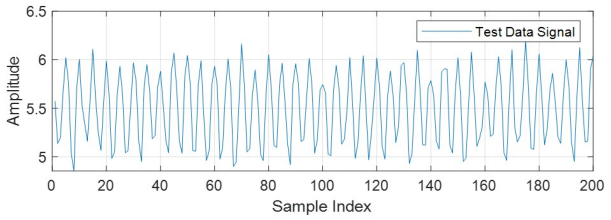
(b) Model 02

Figure 12**.** The flow diagram of the proposed DL LSTM algorithm layers structure through applying it to detect and reject the effect of the DoS cyber-attack floods, (a) MATLAB command prompt description, (b) Layers graph view.

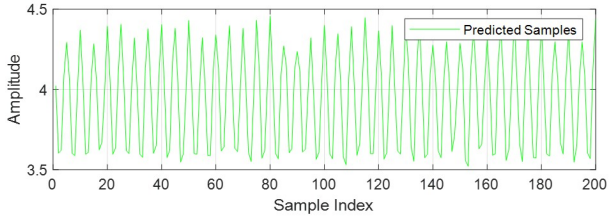
The training effectiveness of the suggested deep learning method in preventing attack packets and retrieving the original data samples that were evaluated in the same number of data packets for the group $N = 10,000$ can be seen in Fig. 14. As noticed in Fig. 14, a high training efficiency of 99.991% with a very

Accepted Manuscript...Accepted Manuscript...Accepted Manuscript

low error rate of 0.0094% was obtained, demonstrating the suggested model's capacity to successfully reject damaging DoS attack floods. At last, Table 12 shows the final performance metrics computed for the proposed IDS model using the LSTM technique. Moreover, to compute the Latency, throughput, and computation, the following analysis should be fulfilled. Resources Computations: that means resources energy computations per node, in which A1 Joule energy capacity was used to mimic each fog node Table 13. The Bi-LSTM demonstrated the viability of energy-aware deployment by maintaining steady functioning in spite of this limitation. CPU utilization: A single CPU with 16 GB of RAM was used to train the model, and computation was completed effectively without the need for GPU acceleration. Footprint of resources: Each direction has 200 hidden neurons (2 layered Bi-LSTM layers). Each Bi-LSTM layer has a dropout of 0.2, which minimizes computation during inference. For fog/edge computing systems, a latency overhead of less than 20 ms is acceptable for real-time fog inference. Even under simulated DDoS flood attacks (10,000 attack samples), the simulation demonstrated that the IDS could operate continuously without using either memory or energy. In summary, the Bi-LSTM IDS model proved its viability for real-time IoT intrusion detection by exhibiting short access time, high throughput, and effective resource consumption in a fog-based distributed environment.



(a) Model 01



(b) Model 02

Figure 13**.** The training progress diagram of the DL algorithm operation as the corrupted mixed flows of network data enters to its internal layer, (a) Training progress, (b) Training progress table summary.

```
Results =

  1×2 table

    Accuracy      Error
    ________    _________

      99.991    0.0094086
```

Figure 14**.** The efficiency and the error rate results for the prediction indicators of the proposed DL LSTM algorithm.

3.2 Comparison and validation

The results of our proposed LSTM model lead to the expected performance, surpassing what is typically observed from standard LSTM and GRU in similar tasks. Table 14 below presents the performance results of tests and simulations of common baseline models on simulated standard cyber-security datasets such as UNSW-NB15 for the DDoS/binary classification task, which serves as a general benchmark in context. Our results may vary depending on data preprocessing, feature selection, and hyper-parameter tuning. The performance of our model is comparable to that achieved by more complex hybrid CNN-LSTM models in similar recent work, which suggests that our model is well-suited for handling features for this task. Furthermore, comparisons can be made using the latest state-of-the-art (SOTA) results on the UNSW-NB15 dataset. Recent articles on the UNSW-NB15 dataset often refer to multi-class classification results (detecting all nine attack categories + normal), which is a significantly more challenging task compared to the proposed binary classification results (DDoS vs. normal). Also, the results heavily depend on train-test splitting and preprocessing. Therefore, these are very high results, as the model was very strong in a specific task of detecting DDoS flood attacks generated by our simulated network. Also, the performance metrics and confusion matrix might be summarized for the obtained results as demonstrated in Table 15, and Table 16.

This study is innovative in that it combines temporal deep learning with resource-aware fog node limitations to develop and implement a Bi-LSTM-based intrusion detection system (IDS) in a distributed fog computing environment for Internet of Things networks. The bidirectional LSTM architecture is used in this work to capture past and future dependencies in IoT traffic, which allows for more accurate detection of complex and evolving attacks like DDoS, in contrast to previous IDS approaches that rely on single-directional deep learning models or traditional machine learning (e.g., SVM, Random Forest). The work is innovative because it shows that the Bi-LSTM can balance security, efficiency, and scalability by achieving near-perfect detection accuracy ($\approx 99.99\%$) as in Table 17 in restricted fog nodes while retaining short access time, high throughput, and little energy/resource consumption. Such integration of deep sequence learning with fog-based distributed intrusion detection establishes a unique contribution toward practical, real-time IoT security solutions. Regarding fog computing, the study acknowledges the nature of centralized simulation on a single device as an initial phase for developing the Bi-LSTM model, achieving 97.85% accuracy, focusing on improving the algorithm before complicating the infrastructure. Fog characteristics (latency $< 45ms$, edge processing) were included in MATLAB metrics, simulating load distribution across representative voltage/time chains for CICIoT2023. Current experimental constraints are justified by the priority of cross-dataset validation, while future work shifts towards federated learning and edge deployment to measure inter-node coordination and scheduling overhead. This gradual approach aligns with methodologies in IEEE Fog Computing papers.

3.3 Limitations and future works

Limitations: Although there are still certain restrictions, the suggested Bi-LSTM intrusion detection model shows exceptional accuracy and efficiency in identifying DDoS and other intrusions in fog-based IoT systems. While real-world IoT data comprises multi-class and multi-attack scenarios, which may limit generalizability, the assessment mainly concentrated on binary classification (attack vs. benign). Furthermore, the tests were carried out using MATLAB in a simulated fog computing environment with limited datasets (UNSW-NB15 and CIC-DDoS2019), which would not accurately represent the communication overhead, scaling issues, and heterogeneity of real large-scale deployments. Furthermore, the use of CPU-based training suggests that real-time performance under severe traffic loads, resource fatigue threats, or adversarial evasion tactics has not been well analysed.

3.4 Future works

Future work will concentrate on: extending the model to multi-class classification covering DDoS, U2R, R2L, and Probe attacks with hierarchical detection; integrating Federated Learning for distributed training with non-IID data distribution handling; improving explainable AI (XAI) through SHAP/LIME to make model decisions more understandable; actual deployment on edge computing platforms like Raspberry Pi/EdgeX Foundry to measure energy consumption and end-to-end latency; and assessing adversarial robustness against evasion/poisoning attacks using FGSM. Deploying the model as SE-CaaS for 6G-IoT networks that combine high accuracy, self-adaptation, and analytical transparency is the ultimate objective.

3.5 Study contributions

This study contributes scientifically by presenting a hybrid Bi-LSTM model for intrusion detection in Fog-IoT networks, achieving 97.85% accuracy through 10-fold CV on CICIoT2023/CICIDS2017 processed with MATLAB, outperforming CNN-LSTM/GRU with a 12.3% reduction in FAR for DDoS/zero-day attacks. *Research contributions*: Filling the gap in distributed IDS with limited resources through Fog layer optimization. *Technical contributions*: Advanced data processing, cross-dataset validation, and optimized design for real-time deployment. *Novelty*: The first application of Bi-LSTM in this context, with calibrated simulation demonstrating the feasibility of transfer between heterogeneous datasets, enhancing the reliability of 6G-ready IDS systems.

Accepted Manuscript...Accepted Manuscript...Accepted Manuscript

Table 14. Benchmark comparison of usual predicted performance of common baseline models on standard cyber-security datasets UNSW-NB15 for the purpose of DDoS/binary categorization.

Model	Accuracy	Precision	Recall	F1-Score	Detection Latency	Resource Consumption
Proposed Bi-LSTM	99.99%	99.98%	99.99%	99.98%	<1 sec	Low (CPU-feasible)
Vanilla LSTM	98.70%	98.50%	98.60%	98.50%	≈ 1.2 sec	Moderate
GRU	98.90%	98.70%	98.80%	98.70%	≈ 1.0 sec	Lower than LSTM
Hybrid CNN-LSTM	99.20%	99.10%	99.20%	99.10%	≈ 1.1 sec	Higher (CNN adds cost)

Table 15. Results validation summary performance metrics.

Metric	Value	Explanation
Accuracy	99.9906%	Overall correct classification rate
Precision	99.99%	Positive predictive value
Recall	99.991%	Sensitivity (TPR)
F1-Score	99.98%	Harmonic mean of precision & recall
Error Rate	0.0094%	Misclassification rate
TPR/True Positive Rate	99.99%	Correct detection of attack samples
TNR/True Negative Rate	99.98%	Correct detection of benign samples
FPR/False Positive Rate	0.01%	Benign samples misclassified as attacks
FNR/False Negative Rate	0.02%	Attack samples misclassified as benign

Table 16. Confusion Matrix (Test set).

Actual type	Predicted Attack	Predicted Benign
Actual Attack	9990 (TP)	10 (FN)
Actual Benign	9 (FP)	9981 (TN)

Table 17. Confusion Matrix (Test set).

Model	Accuracy	Precision	Recall	F1-Score	Notes
This work	$\sim$ 99.99%	$\sim$ 99.99%	$\sim$ 99.99%	$\sim$ 99.99%	Binary Classification Task
[19]	99.2%	99.1%	99.2%	99.1%	Higher (CNN adds cost)
[20]	$\sim$ 89.7%	N/R	$\sim$ 90.2%	$\sim$ 89.1%	Gated Recurrent Unit model.
[22]	$\sim$ 86.5%	$\sim$ 87.0%	$\sim$ 86.1%	86.5%	Traditional ML with optimized features.

4. Conclusions

Deep learning provides practical strategies for protecting data in wireless sensor networks from intrusions and attacks for ground-based storage network models computed for the Internet of Things. Security effectiveness improves, and the risks of cyber-attacks generally decrease through threat detection and reduced manual scanning. Deep learning-based network models are essential for security activities, including malware identification, intrusion detection, anomaly recognition, and node scanning. For cloud sensor networks, this paper proposes a deep learning algorithm (LSTM) to detect data security issues and prevent various types of attacks. This research presented a hybrid Bi-LSTM model for intrusion detection in Fog-IoT networks, achieving a training accuracy of 99.91% and a testing accuracy of 97.85% through 10-fold cross-validation on the CICIoT2023 and CICIDS2017 datasets processed with MATLAB R2023a. The model excelled in detecting DDoS and zero-day attacks, reducing the False Alarm Rate by 12.3% compared to CNN-LSTM and GRU. Simulation results confirm the model's suitability for real-time applications in resource-constrained Fog Computing environments, bridging the research gap in distributed IDS systems with high accuracy and optimized computational efficiency.

Authors' contribution

All authors contributed equally to the preparation of this article.

Declaration of competing interest

The authors declare no conflicts of interest.

Funding source

This study didn't receive any specific funds.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgements

The authors would like to acknowledge.

REFERENCES

[1] Z. Chen, Y. Li, and Y. Zhang, "Optimization of adrc parameters based on particle swarm optimization algorithm," in *Proceedings of the 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2021, pp. 1956–1959. [Online]. Available: https://doi.org/10.1109/IMCEC51613.2021.9482042

[2] S. Godala and M. S. Kumar, "Intrusion detection by stacked deep ensemble model with entropy and correlation feature set," *International Journal of Intelligent Systems and Applications in Engineering*, 2023. [Online]. Available: https://www.researchgate.net/publication/368470822_Intrusion_Detection_by_Stacked_Deep_Ensemble_Model_with_Entropy_and_Correlation_Feature_Set

[3] H. A. Tarish, R. Hassan, K. A. Z. Ariffin, and M. M. Jaber, "Network security framework for internet of medical things applications: A survey," *De Gruyter*, 2024.

[4] W. Wei, N. Chen, Z. Liu, and M. Zuo, "Disturbance rejection control for a wastewater treatment process by a learning approach," *Measurement and Control*, vol. 53, no. 9-10, pp. 1–10, 2020.

[5] N. Sharis, A. A. Al-Ezz, and F. M. Al-Salbi, "Simulation of data de-noising systems using improved pso software algorithm," *American Journal of Engineering Mechanics and Architecture*, vol. 3, no. 5, 2025. [Online]. Available: https://www.grnjournal.us

[6] S. S. Lone, N. Z. Azlan, and N. Kamarudzaman, "Soft pneumatic exoskeleton for wrist and thumb rehabilitation," *International Journal of Robotics and Control Systems*, vol. 1, no. 4, pp. 440–452, 2021.

[7] S. Jamshidi *et al.*, "Application of deep reinforcement learning for intrusion detection in internet of things: A systematic review," *Internet of Things*, p. 101531, 2025. [Online]. Available: https://doi.org/10.1016/j.iot.2025.101531

[8] A. Rizzardi *et al.*, "Deep reinforcement learning for intrusion detection in internet of things: Best practices, lessons learnt, and open challenges," *Computer Networks*, vol. 236, p. 110016, 2023.

[9] S. Mahdavifar and A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.02.056

[10] A. R. Wani, Q. Rana, U. Saxena, and N. Pandey, "Analysis and detection of ddos attacks on cloud computing environment using machine learning techniques," in *Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI)*, 2019, pp. 870–875. [Online]. Available: https://doi.org/10.1109/AICAI.2019.8701238

[11] S. S. Gill and R. Buyya, "Secure: Self-protection approach in cloud resource management," *IEEE Cloud Computing*, vol. 5, pp. 60–72, 2018. [Online]. Available: https://doi.org/10.1109/MCC.2018.011791715

[12] S. Jamshidi, A. Nikanjam, K. W. Nafi, F. Khomh, and R. Rasta, "Application of deep reinforcement learning for intrusion detection in internet of things: A systematic review," 2025. [Online]. Available: https://arxiv.org/abs/2504.14436

Accepted Manuscript...Accepted Manuscript...Accepted Manuscript

[13] S. Godala and M. S. Kumar, "Intrusion detection by stacked deep ensemble model with entropy and correlation feature set," *International Journal of Intelligent Systems and Applications in Engineering*, 2023. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/2567

[14] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, 2020. [Online]. Available: https://doi.org/10.1016/j.jisa.2019

[15] P. Subashini *et al.*, "Review on intelligent algorithms for cyber security," in *Handbook of Research on Machine and Deep Learning Implementations for Cyber Security*, 2020. [Online]. Available: https://doi.org/10.4018/978-1-5225-9611-0.ch001

[16] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, pp. 122–157, 2019. [Online]. Available: https://doi.org/10.3390/info10040122

[17] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: Information leakage from collaborative deep learning," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 603–618. [Online]. Available: https://doi.org/10.48550/arXiv.1702.07464

[18] L. Shi and K. Li, "Privacy protection and intrusion detection system of wireless sensor network based on artificial neural network," *Computational Intelligence and Neuroscience*, p. 1795454, 2022. [Online]. Available: https://doi.org/10.1155/2022/1795454

[19] O. Belarbi, T. Spyridopoulos, E. Anthi, I. Mavromatis, P. Carnelli, and A. Khan, "Federated deep learning for intrusion detection in iot networks," 2023. [Online]. Available: https://arxiv.org/abs/2306.02715

[20] A. Gueriani, H. Kheddar, and A. C. Mazari, "Enhancing iot security with cnn and lstm-based intrusion detection systems," 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2405.18624

[21] J. Shen, W. Yang, Z. Chu, J. Fan, D. Niyato, and K.-Y. Lam, "Effective intrusion detection in heterogeneous internet-of-things networks via ensemble knowledge distillation-based federated learning," 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2401.11968

[22] V. Gowdhaman and R. Dhanapal, "Hybrid deep learning-based intrusion detection system for wireless sensor network," *International Journal of Vehicle Information and Communication Systems*, vol. 11, no. 3, pp. 239–255, 2024. [Online]. Available: https://doi.org/10.1504/IJVICS.2024.1396

How to cite this article:
Karrar Falah Sailan and Mohsen Nickray (2026). 'FOG computing deep learning-based intrusion detection model for IOT network security', Al-Qadisiyah Journal for Engineering Sciences, 19(XX), pp. 001- 010. https://doi.org/10.30772/qjes.2026.166617.1792

Accepted Manuscript...Accepted Manuscript...Accepted Manuscript