International Information and Engineering Technology Association
Advancing the World of Information and Engineering

International Journal of Safety and Security Engineering
Vol. 15, No. 11, November, 2025, pp. 2297-2310
Journal homepage: http://iieta.org/journals/ijsse

Intrusion Detection for Flooding-Based Denial-of-Service Attacks in Wireless Sensor Networks Using a Long Short-Term Memory Deep Learning Model

Abeer Hussein Abdulrasool*, Ekhlas Kadhum Hamza, Ahmed Mudheher Hasan

Control and Systems Engineering College, University of Technology, Baghdad 10066, Iraq

Corresponding Author Email: cse.23.08@grad.uotechnology.edu.iq

https://doi.org/10.18280/ijsse.151110

Received: 23 October 2025
Revised: 24 November 2025
Accepted: 27 November 2025
Available online: 30 November 2025

Keywords:
wireless sensor networks, deep learning algorithms, cyber security, intrusion detection system, long short-term memory

ABSTRACT

The need for deep learning (DL) approaches as effective and practical models for attaining security in wireless sensor networks (WSNs) has grown with the rise of artificial intelligence applications. With the capacity to identify threats and guarantee data integrity, DL models improve security efficacy and lower the total risks of different assaults. Intelligent detection and protection systems are essential for the security of information transfer since wireless computer networks are vulnerable to several incursions, including malware, intrusion flows, and security flaws. In order to identify and stop distributed denial of service (DDoS) assaults, this study will categorize and analyze data transferred across the virtual computer network using a DL approach known as long short-term memory (LSTM). In this study, a deep learning (LSTM) algorithm model has been employed for a virtual cloud WSN and proposed to check security using the UNSW-NB15 dataset and detect/stop the DDoS cyber-attacks flood type. The proposed LSTM deep learning model has been designed to analyze and classify the flood of the transmitted dataset inside the WSN by training the internal weights and adjusting their parameter variations. According to the simulation results, a high training efficiency was recorded, reaching 99.96% with a very low error rate of 0.04% in training the proposed LSTM model according to the employed dataset.

1. INTRODUCTION

The variety of cyber-attacks, their use of modern and unique programming, and the growth of the extent of electronic breaches by computer and cloud network programmers have made monitoring cyber-attacks a crucial issue. Distributed denial of service (DDoS) assaults, however, can be addressed in two ways. The fundamental process involves offering a directly equivalent token, and the final method manifests as tokens. Direct assaults target any flaw in the design of the information system that might result in harm or even the termination of service. Additionally, hostile attacks look for various components that are linked to other components of the system to attack and skew the information flow and content. To prevent external and internal intrusions and to protect the data and information of institutions that use the Internet and communications networks, there is a growing need to develop better security schemes for various real-world applications. Efficient algorithms that collaborate with wireless sensor networks (WSNs) to offer highly dependable cyber-attack detection and prevention using a variety of techniques are suggested to meet these fundamental needs. Evaluation algorithms and control algorithms are really the two main algorithms for cyber-security aggregation and cloud computing planning [1-3]. WSNs' assessment and control algorithms are made to accomplish "utilitarian goals, such as closed-loop security objectives. Theoretically, obtaining insulation from malevolent attacks on the electronic system is the main security objective.

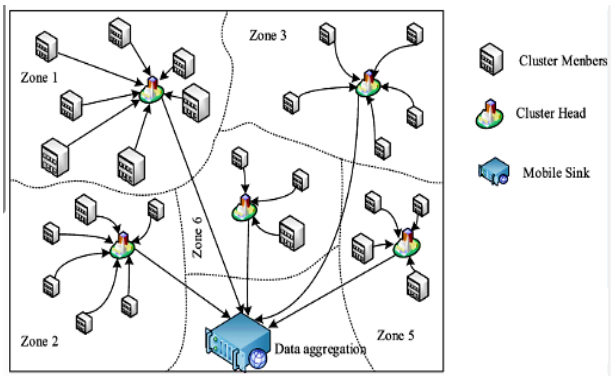


Figure 1. Schematic diagram of the wireless sensor networks (WSNs) construction [3-5]

However, handling all variables and emergency situations is also necessary to reach the degree of security. Additionally, when tags and data are gathered from sensors of different computer network units, including private sensitive data, security methods must be applied to guarantee the data's validity. Figure 1 shows a schematic diagram of the WSNs construction [3-5]. In order to convey and present theoretical ideas, as well as to exhibit and plan, WSNs organize exercises and real-world activities with outstanding projects and stages.

They also combine examination procedures. As basic advancements, it necessitates outstanding connection activities amongst PCs, networks, and genuine frameworks. Since the product is remembered for its equipment and its standard, innovation is dependent on a variety of trains, such as embedded frameworks, PCs, communications, and so forth. The main element involves more than simply calculations, such as intelligent transportation frameworks, logical devices, and automobiles, as well as medical equipment. Concerned experts are presently showing a great deal of interest in the WSNs initiative [6]. Energy, transport, rainfall, and healthcare systems are just a few of the contemporary sectors that are seeing notable advancements in modernization and monitoring. Concern for management, effectiveness, and appropriateness has led to an attempt to regulate data security. This calls for rigorous study in evaluating and integrating real-world cyber systems, cyber-attack detection, and full compatibility of the systems, information, communications, and computing developed in reality.

1.1 Types of cyber-attack techniques

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 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. Figure 2 displays the effect of the denial of service (DoS) attack [6].

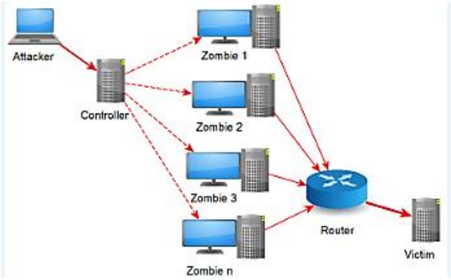


Figure 2. The effect of the denial of service (DoS) attack [6]

Instead of employing a single server, which is currently unpersuasive given the notable advancements in Web usage over the past ten years, attackers leverage these vulnerable hosts to launch an attack. Furthermore, a single server assault may be successfully identified. Before launching an attack, an attacker takes control of many Internet-connected PCs; this type of PC motor is known as an overseer, and it places these PCs in a precarious position. The attacker then installs malicious software, tools, and other hacking techniques to take advantage of the PCs' flaws and weaknesses and use them to enforce orders. Also, Table 1 summarizes the crucial types of cyber-attack techniques [7, 8].

Table 1. Summary of common cyber-attack techniques [7, 8]

Type of Attack	Description
Phishing	Fraudulent emails or messages trick users into revealing sensitive data.
Spear Phishing	A targeted form of phishing aimed at specific individuals or organizations.
Ransom Ware	Malware that encrypts data and demands ransom to unlock it.
Denial of Service (DoS)	Overloads systems or networks, making services unavailable to legitimate users.
Distributed Denial of Service (DDoS)	Similar to DoS, but attacks are launched from multiple compromised systems.
Man-in-the-Middle (MitM)	Attacker intercepts and possibly alters communication between two parties.
SQL Injection	Exploits vulnerabilities in applications to access or manipulate databases.
Cross-Site Scripting (XSS)	Injects malicious scripts into trusted websites viewed by other users.
Malware	Malicious software (viruses, worms, trojans) is designed to harm or exploit.
Zero-Day Exploit	Attacks that target vulnerabilities unknown to the software vendor.
Credential Stuffing	Using stolen credentials to gain unauthorized access to user accounts.
Brute Force Attack	Repeatedly tries different passwords or keys to gain access.
Social Engineering	Manipulating individuals into divulging confidential information.
Drive-by Download	Malware is automatically downloaded when visiting compromised websites.
Insider Threats	Attacks initiated by employees or trusted individuals within the organization.

Table 1 above illustrates the most important modern cyber-attack methods and techniques discovered to date. In fact, the types of cyber-attacks and breaches are impossible to list, as they are constantly evolving. Attackers, network, and internet hackers update their attack techniques and hacking programs whenever a defense method is introduced, making the topic a subject of scientific research and study.

1.2 Types of defense techniques

Modern applications, including industrial automation, healthcare, and environmental monitoring, depend on WSNs. However, because of their decentralized architecture and resource limitations, WSNs are susceptible to assaults such as data injection, Sybil attacks, and jamming. A multi-layered defense strategy is necessary to protect these networks. It combines detection tools like Intrusion Detection System (IDS), trust management, and anomaly monitoring to spot malicious activity in real time, with prevention strategies like encryption, authentication, and secure routing protocols to prevent unwanted access and tampering. By emphasizing resource economy and flexibility, these methods tackle the particular WSN constraints of limited energy, scalability, and dynamic topologies. Their breadth goes beyond protecting secrecy and data integrity to guarantee network availability and dependability, allowing WSNs to function dependably in challenging conditions. Strong defensive tactics are essential to maintaining the expanding use of WSNs in mission-critical systems as cyber threats change, underscoring their function as a pillar of safe Internet of Things (IoT) ecosystems [9, 10]. The cyber protection strategies employed in WSNs to identify and stop threats are compiled in Table 2.

Table 2. Demonstration of the cyber protection strategies employed in wireless sensor networks (WSNs) to identify and stop various threats

Technique	Type (Detection / Prevention)	Description	Examples
Encryption	Prevention	Secures data confidentiality by converting data into unreadable formats.	AES, RSA, TinySec.
Authentication Protocols	Prevention	Ensures only authorized nodes join the network.	Digital certificates, HMAC, Two-factor authentication.
Intrusion Detection Systems (IDS)	Detection	Monitors network traffic/node behavior for anomalies.	Anomaly-based IDS, Signature-based IDS.
Jamming Detection/Mitigation	Both	Detects and counters radio jamming attacks.	Frequency hopping, Spread Spectrum (DSSS), Energy-aware routing.
Secure Routing Protocols	Prevention	Protects routing paths from manipulation (e.g., sinkhole, selective forwarding).	SPINS, LEAP, INSENS.
Trust Management Systems	Detection	Evaluates node trustworthiness to identify malicious actors.	Reputation-based systems, Bayesian trust models.
Physical Tamper Detection	Prevention	Detects physical tampering of sensor nodes.	Tamper-proof packaging, self-destruct mechanisms.
Key Management	Prevention	Secures cryptographic key distribution and rotation.	LEAP, Random key pre-distribution, Periodic key updates.
Sybil Attack Detection	Detection	Identifies nodes using multiple fake identities.	Resource testing, Radio fingerprinting, Neighbor monitoring.
Data Redundancy Checks	Detection	Validates data integrity through consistency analysis.	Majority voting, Checksums, Hash-based verification.
Firmware/Software Updates	Prevention	Ensures nodes run updated, patched software to fix vulnerabilities.	Secure OTA updates, Secure boot mechanisms.
Energy Monitoring	Detection	Detects abnormal energy consumption (e.g., battery-draining attacks).	Power usage profiling, Threshold-based alerts.
Secure Time Synchronization	Prevention	Prevents time-based attacks by securing synchronization protocols.	TESLA, Reference Broadcast Synchronization (RBS).
Clone/Replication Detection	Detection	Identifies duplicate malicious nodes.	Location-based attestation, Unique hardware IDs.

Table 3. A summary table of the main deep learning neural network (DLNN) kinds [11-13]

Type	Construction	Description
Convolutional Neural Network (CNN)	Convolutional layers, pooling layers, and fully connected layers.	Processes grid-like data (e.g., images) via filters to detect spatial patterns (edges, textures).
Recurrent Neural Network (RNN)	Basic recurrent layers, sequential input/output.	Handles sequential data but struggles with long-term dependencies due to vanishing gradients.
Long Short-Term Memory (LSTM)	Memory cells with input, forget, and output gates; recurrent connections.	Solves RNN limitations by retaining long-term dependencies via gated memory cells.
Generative Adversarial Network (GAN)	Generator and discriminator networks trained adversarial.	Generates synthetic data (images, audio) by competing networks to improve realism.
Transformer	Self-attention mechanisms, encoder-decoder architecture	Processes sequences in parallel for NLP tasks (translation, summarization) via attention weights.
Auto-Encoder	Encoder (compression) and decoder (reconstruction) networks.	Reduces data dimensionality for tasks like anomaly detection or feature learning.
Multilayer Perceptron (MLP)	Fully connected layers, input-output mapping.	Basic feed forward network for classification/regression on tabular or simple structured data.

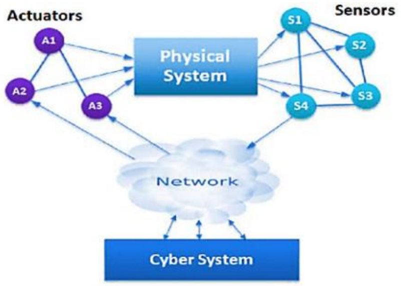


Figure 3. General construction of the wireless sensor networks (WSNs) against cyber-physical system (CPS) [14, 15]

It is important to remember that prevention strategies aim to thwart attacks before they start. Ongoing or previous assaults are identified using detection techniques for mitigation. Due to resource limitations, hybrid approaches—such as IDS and secure routing—are also often used in WSNs. Furthermore, as WSNs frequently run on a small amount of power, memory, and processing power, resource efficiency is essential.

In any case, while some CSS kinds are already in use, the growing popularity of remotely implanted sensors and actuators is spawning some new uses in fields including clinical instruments, driverless cars, intelligent designs, and enhancing the capabilities of already-existing products. Figure 3 presents the general construction of the WSNs against the cyber-physical system (CPS) [14, 15].

1.3 Deep learning neural network

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 TensorFlow, 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. The back-propagation and optimization algorithms (such as Adam and Stochastic Gradient Descent (SGD)) are used in training in order to reduce loss functions. GANs are used for data creation, CNNs for image processing, 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. A summary table of the main DLNN kinds is shown in Table 3 [11-13].

We will describe the LSTM technique, a specialized RNN variant for long-term dependencies, and the RNN techniques used in processing sequential time series data in order to comprehend some of the most significant deep learning (DL) approaches. Figure 4 demonstrates the construction of the RNN model [13, 16].

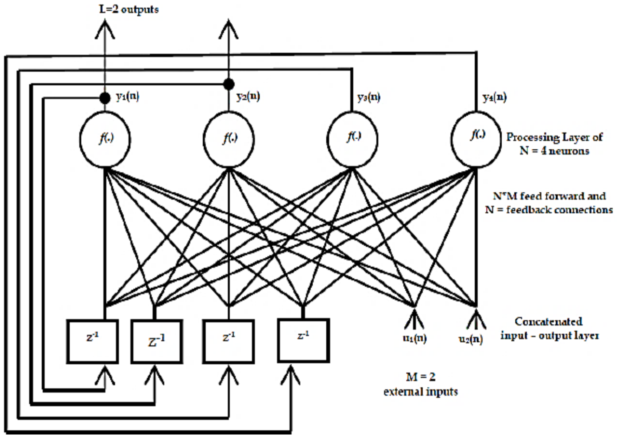

L=2 outputs
y₁(n)
y₂(n)
y₃(n)
y₄(n)
f(.)
f(.)
f(.)
f(.)
Processing Layer of N = 4 neurons
N*M feed forward and N = feedback connections
z⁻¹
z⁻¹
z⁻¹
z⁻¹
u₁(n)
u₂(n)
Concatenated input - output layer
M = 2 external inputs

Figure 4. Schematic diagram of the recurrent neural network (RNN) [13]

As seen in Figure 4, 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. The RNNs might handle posting successions of varying dimensions with display physical kinetics since they have internal storage. Figure 4 might be seen as an example of a simple RNN architecture. One may see that the RNN algorithm design presented in Figure 4 has common sections that use the aperture $X_{t-w,\, t-1}$ 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, utilizing the inputted grouping x_{t-1} of the repeated entity o_{t-2} and the enactment capacity as $tanh$ [13, 16]:

$$\begin{gathered} x'_t = \sigma(W_{x'}, o_{t-1} + b_{x'}), \\ o_{t-1} = \tanh\,(W_{o.x_{t-1}} + U_o . o_{t-2} + b_h) \end{gathered} \qquad (1)$$

where, the network's components are denoted by $W_{x'}$, W_o, U_o, and b. As previously described, repetition occurs when the network performs recall and remembering operations for the information learnt from training data using the prior findings as they were entered. Figure 5 shows the LSTM deep learning structure [16].

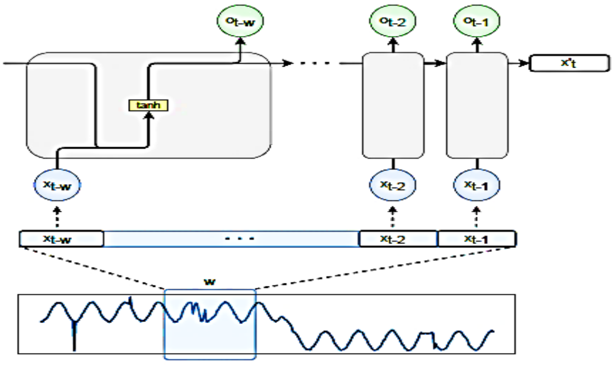
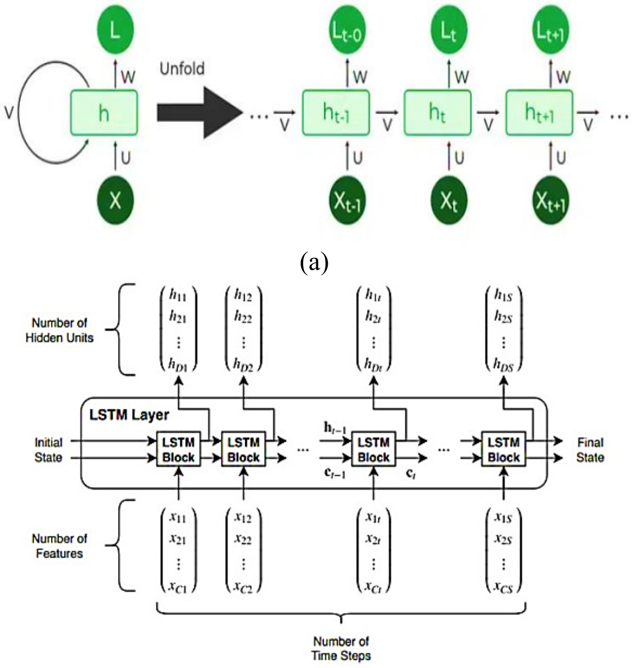


Figure 5. Construction of a deep learning long short-term memory (LSTM) structure [16]

The long-term and short-term assumption network is really trained here. Figure 6 illustrates the construction of the RNN deep learning algorithm [13, 16-20].



(a)

(b)

Figure 6. The internal structure of the recurrent neural network (RNN) deep learning main model [18]

DLNNs revolutionize artificial intelligence by enabling complex data modelling using layered architectures. Their ability to acquire hierarchical features and their adaptability across domains—from language to vision—make them indispensable for modern AI applications.

The detailed structure of the LSTM deep learning algorithm will be discussed and examined in this project, the proposed model of the study. The LSTM deep learning algorithm will consist of several layers, including a learning, input, hidden, weighted, and output layers.

The flow of time samples X with C features (channels) of length S through an LSTM layer is seen in this graph. The output (sometimes referred to as the hidden state) and cell state at time step t are indicated in the diagram by the symbols ht and ct, respectively. The first LSTM block computes the first output and the updated cell state using the network's starting state and the sequence's first-time step. The block utilizes the network's current state (c_{t-1}, h_{t-1}) at time step *t*. The subsequent step in the sequence is to compute the output and the updated cell state ct. The cell state and the concealed state, also referred to as the output state, make up the layer state. The outputs of the LSTM layer for that time step are contained in the hidden state at time step t. Knowledge gained from earlier time steps is stored in the cell state. The layer adds or subtracts data from the cell state at each time step. Through gateways, the layer manages these changes. The cell state and concealed state of the layer are managed by the following elements. Table 4 presents the cell and hidden states of the layer control components.

Table 4. The cell and hidden states of the layer control components [18]

Components	Justification
Entered Gate (i)	Measure the control in updating the cell state
Forgetting Gate (f)	Measure the control in the reset (forget) cell state
Cell Candidate (g)	Summing information for the cell state
Output Gate (o)	cell state control level summed to the hidden state

Also, the forgetting, updating strategy inside the deep learning (LSTM) structure is shown in Figure 7 [20].

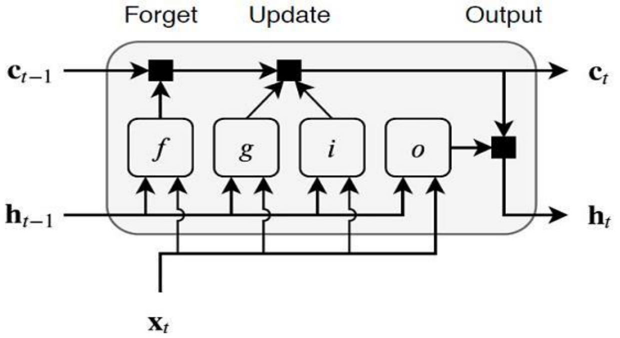


Figure 7. The forgetting, updating strategy inside the deep learning (LSTM) structure [20]

Thus, the learnable weights of the LSTM layer are the input weights W (InputWeights), the recurrent weights R (RecurrentWeights), and the bias b (Bias). The matrices W, R, and b are sequences of input weights, frequency weights, and bias of each component, respectively. These matrices are linked as follows [18-22]:

$$W = \begin{bmatrix} W_i \\ W_f \\ W_g \\ W_o \end{bmatrix}, R = \begin{bmatrix} R_i \\ R_f \\ R_g \\ R_o \end{bmatrix}, b = \begin{bmatrix} b_i \\ b_f \\ b_g \\ b_o \end{bmatrix} \tag{2}$$

Such that*, i, f, g*, and *o* denote the entrance gate, forget gate, cell candidate, and resulting gate, respectively. The cell state at time step t is represented by:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{3}$$

where, ⊙ indicates the Hadamard product (vector-wise multiplication). Also, the hidden state at time step *t* is represented by:

$$h_t = o_t \odot \sigma_c(c_t) \tag{4}$$

where, σ_c indicates the state activation function. By default, the LSTM layer function employs the hyperbolic tangent (*tanh*) function to evaluate the state activation function. Because deep neural networks can simulate complicated data using multi-layered structures, they are transforming artificial intelligence. They are essential for contemporary AI applications because of their versatility across domains—from language to vision—and their capacity to pick up hierarchical information.

2. LITERATURE REVIEW

The topics of WSN cloud security and 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 with such themes will be compiled and examined in this section. We will provide a fair scientific backdrop in this evaluation to help identify the study's difficulties and issues. In the field of intrusion detection, Ferrag et al. [15] examined DL approaches in 2020, specifically RNN, CNN, Deep Boltzmann Machine (DBM), Deep NN (DNN), RBM, DBN, and deep auto-encoder (DAE). The approaches are analyzed using two datasets: CSE-CIC-IDS2018 and Bot-IoT. According to the study's findings, the CNN and DAE methods get the highest accuracy levels, 97% and 98%, respectively, and the highest performance levels across both datasets. Mahdavifar and Ghorbani [11] developed another evaluation of DL missions in the foundation and programming security domains in 2020. They also examined the projects that employed DL structures for grouping, interruption placement, virus detection, and site disfigurement recognition. This study is expanded upon in 2020 by Subashini et al. [16], who reviewed ML and DL algorithms. Botnet and malware detection in the software area is mentioned in passing, but their primary focus is on intrusion and anomaly detection in the infrastructure sector. Cyber-attacks pertaining to software and infrastructure are also covered in 2020 by Berman et al. [17] and Hitaj et al. [18]. The two assessments look at the AE, CNN, RNN, and GAN topologies used for recovering from cyber-attacks. They focus on assaults like malware, botnets, and network intrusions. The unreliability of low-processing devices in IoT contexts is highlighted by Singla and others. In 2020, Hemdan and Manjaiah [19] examined studies on the use of big data analytics for cyber-attack detection and prevention. They examine the use of DL and Big Data Analytics in Social Networks, Cloud Computing, and IoT to foresee contemporary assaults. Wickramasinghe et al. [20] provide a quick overview of the DL techniques employed in security implementations in 2020. The researchers look on organization strategies to enhance the generalizability of DL-based security systems. They specifically examine the DL schemes that are utilized to identify software infections and attacks with unusual recovery in the infrastructure industry. Harnessing artificial intelligence capabilities to improve cyber-security for IoT and CPS-related cyber-security challenges was explored by Godala and Kumar [23].

They talk about the effects of different cyber security assaults on the application stack and networking. They begin by providing a summary of security solutions that do not employ DL, discussing their shortcomings, and then going on to discuss how contemporary DL findings may improve cyber-security. Actually, IoT and CPS-focused software and network infrastructure are examples of DL-based solutions. Belarbi et al. [24], employed federated DL technology, with pre-training and aggregation techniques. Contributions: Introduced a federated learning-based IDS tailored for IoT networks, emphasizing data privacy and scalability. The study utilized the TON-IoT dataset to simulate realistic conditions and compared federated models against centralized counterparts. Limitations: Performance degradation observed due to data heterogeneity; reliance on pre-training to mitigate this issue. Gueriani et al. [25] suggested a Hybrid CNN-LSTM deep learning model. The contributions were made by developing an IDS combining CNNs and LSTM networks to capture spatial and temporal features in IoT traffic. Achieved high accuracy using the CICIoT2023 and CICIDS2017 datasets. The limitations are restricted to the potential challenges in real-time deployment due to computational complexity. Shen et al. [26] proposed Federated Learning technology with Ensemble Knowledge Distillation (FLEKD). The study contributions proposed FLEKD to address data heterogeneity in IoT networks, enhancing intrusion detection performance without compromising data privacy. Demonstrated improved detection rates on the CICIDS2019 dataset. On the other hand, drawbacks show increased system complexity and potential communication overhead due to ensemble methods. Gowdhaman and Dhanapal [27] recommended a ResNet-Inception DL strategy integrated with Support Vector Machine (SVM). The research contributions presented a hybrid IDS leveraging DL for feature extraction and SVM for classification, achieving 99.46% accuracy on the NSL-KDD dataset. The study gaps include limited evaluation to a single dataset; generalizability to other datasets remains untested. Shi and Li [28] implemented the Artificial Neural Network (ANN) technique optimized with Particle Swarm Optimization (PSO). With an emphasis on privacy protection, the study aids in the creation of an IDS for WSNs and the use of PSO to enhance ANN performance. The absence of evaluation on a variety of datasets and the neglect of scalability and adaptation to various network circumstances are among the limitations. The Wiley Online Library [25, 27, 29-31].

3. PROBLEM STATEMENT (CHALLENGES)

By analyzing the literature review and researchers' contributions over the past three years, we can analyze and classify the latest approaches and techniques used to address threats and cyber-attacks on WSNs, and the summary is shown in Table 5. We can analyze, classify, and summarize these techniques according to data details, contributions, and benefits, while taking into account the constraints, challenges, and obstacles outlined. WSNs are being used more and more in crucial settings where dependable and secure communication is essential, such as the military, healthcare, and industrial automation. However, WSNs are extremely susceptible to a variety of security risks, including data tampering, eavesdropping, spoofing, and DoS attacks, because of their resource-constrained nature, which includes limited processing power, energy, and memory. Adaptive and scalable security is sometimes lacking from traditional cryptographic and rule-based techniques, especially when it comes to unknown and changing attack patterns. Additionally, real-time threat identification is a constant issue because of WSNs' dispersed nature and changeable topology. Therefore, creating strong, intelligent, and low-latency security methods for WSNs is still an ongoing research challenge.

Table 5. The most recent published studies' summary

Year	Authors	Technology Employed	Contributions	Limitations
2023	Belarbi et al. [24]	Federated DL	Privacy-preserving IDS for IoT using federated learning	Performance affected by data heterogeneity
2024	Gueriani et al. [25]	CNN-LSTM Hybrid Model	High-accuracy IDS capturing spatial-temporal features	Computational complexity for real-time deployment
2024	Shen et al. [26]	Federated Learning with Knowledge Distillation	Enhanced IDS performance addressing data heterogeneity	Increased system complexity and communication overhead
2024	Gowdhaman and Dhanapal [27]	ResNet-Inception DL + SVM	High-accuracy hybrid IDS on NSL-KDD dataset	Limited evaluation on a single dataset
2022	Shi and Li [28]	ANN with PSO	Privacy-focused IDS for WSNs	Limited dataset evaluation and scalability concerns

Note: DL = Deep Learning; CNN = Convolutional Neural Network; LSTM = Long Short-Term Memory; SVM = Support Vector Machine; ANN = Artificial Neural Network; PSO = Particle Swarm Optimization; IDS = Intrusion Detection System; IoT = Internet of Things; WSNs = Wireless Sensor Networks.

4. NOVELTY AND CONTRIBUTIONS

Investigating and evaluating how well DL approaches can improve the security of WSNs is the main goal of this project. The study's specific goal is to develop and assess DL-based models for the detection and categorization of different network assaults in WSNs, including CNNs, RNNs, and LSTM networks. To maintain high accuracy and low false positive rates, the study will concentrate on creating models that can function well within the resource limitations of WSN nodes. To help create intelligent, self-learning security frameworks, the study will also investigate how interpretable and flexible these models are in dynamic WSN situations. Thus, by analyzing the literature review, Table 6 provides a summary of the novelty and contributions of the proposed research.

Table 6. Summary of novelty and contributions of the proposed research from analyzing the literature review

Aspect	Proposed Research Feature	Novelty Relative to Existing Works	Practical Contribution to Wireless Sensor Network (WSN) Security
Problem Formulation	DoS detection as time-series reconstruction via LSTM regression.	Most prior work uses packet/flow classification with static features.	Enables direct recovery of clean sensor data and signal-level attack rejection.
WSN Modeling	Explicit random WSN with 50 nodes, energy per node, TCP/IP-like traffic.	Many studies use abstract datasets without explicit WSN topology.	Provides a reproducible lab test-bed for WSN cyber-physical simulations.
Attack Modeling	MATLAB-generated flood-like oscillatory DoS streams mixed with normal data.	Floods are usually modeled only as labeled records, not as physical signals.	Captures the temporal shape of attacks and their impact on node resources.
DL Architecture	Lightweight, 1-layer LSTM (200 units) + dense + dropout + Mean Squared Error (MSE) loss.	Simpler and more deployable than deep hybrids (ResNet, CNN-LSTM).	Suitable for resource-aware implementations and rapid prototyping.
Performance	Accuracy ≈99.95%, prediction efficiency ≈99.96%, error rate 0.04%.	Comparable or superior to recent IDS studies using heavier models.	Validates that compact LSTM designs can match state-of-the-art IDS accuracy.
Implementation Pathway	End-to-end MATLAB 2020b workflow with WSN, attack generation, and DL.	Most literature uses Python/TensorFlow on generic datasets.	Gives an accessible tool-chain for control/WSN engineers using MATLAB.

5. METHODOLOGY

This section will describe and explain, with the use of DL, the crucial design stages for putting into practice the suggested IDS security model in order to detect and stop cyber-attacks and 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 discuss IDS, which uses an AI (PC-based intelligence) algorithm to detect cyber-attacks and identify DDoS network incursion executions. In this study, MATLAB m-file scripts will be used to equip and simulate the cyber-security architecture details. The proposed cyber-security architecture includes the following incoming units: 1) WSN 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 ensuring that it is free from attack flows or any malicious software. More precisely, the flood assault used in this study might be classified as a Flooding DoS assault, which sends a large number of requests or traffic in an attempt to deplete network or node resources (such as bandwidth, CPU, or battery). Justification: An attacker may use a vibration flood assault in WSNs by continuously triggering sensor readings (from vibration sensors, for example), which would cause the network to analyze and send an excessive amount of data, much of it useless. This causes service disruption by consuming power, blocking communication channels, and sometimes delaying or preventing proper sensing activities. It also affects potential subcategories by overloading the network layer and hindering data movement. Another security issue is the event flooding problem, which results from continuously triggering sensor events, such as false vibration signals. Consequently, energy-draining attacks are considered a subcategory of resource exhaustion attacks. The flowchart in Figure 8 provides a summary of how the suggested security architecture for the WSN operates.

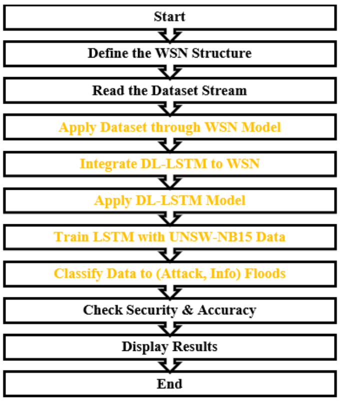

Start
Define the WSN Structure
Read the Dataset Stream
Apply Dataset through WSN Model
Integrate DL-LSTM to WSN
Apply DL-LSTM Model
Train LSTM with UNSW-NB15 Data
Classify Data to (Attack, Info) Floods
Check Security & Accuracy
Display Results
End

Figure 8. Flow chart of the suggested wireless sensor networks (WSNs) security model methodology

Looking at Figure 9, one could notice that the suggested model starts by specifying the architecture and structure of the WSN by varying the number of nodes and connections among them to establish the network's size. After that, the dataset flow is read, and the cyber-attack flows (floods) are added. Following attack verification, a deep learning DL-LSTM approach is used to identify and stop assaults. 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 Figure 9.

By introducing (N = 10000 tests) to be distinguished to train the 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 DL 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 through the ReLU layer. To prepare and 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. Also, the dataset type employed in this study is loaded and examined according to the settings shown in Table 7. Lastly, Table 8 displays the remaining variables of the suggested model together with the settings and control parameters of the suggested LSTM method.

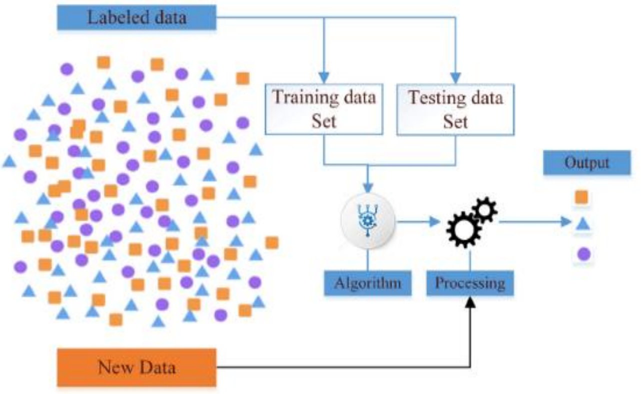


Figure 9. The operation of the proposed deep learning (LSTM) algorithm in detecting and prevention cyber-attacks floods

Table 7. The dataset type employed in this study is loaded and examined according to the settings

Dataset Name	Description	Samples Number	Source2
UNSW-NB15	Contains DDoS and other attack traffic, labeled for intrusion detection	10^4	GitHub (https://github.com/talhatk/UNSW-NB15)

Table 8. The design specifications of the suggested wireless sensor network (WSN) model

Unit Type	Specifications			
WSN	WSN Nodes Number N = 50	Data Length / Node Stream L = 100	Network Connection Random	Nodes Energy 1 Joules
Data Stream	Samples Number Ns = 10^4	Data Type TCP/IP	Data Distribution Gaussian-Like Clean Data	Capacity 5 Volts
Attack Flood	Samples Number Ns = 10^4	Attack Type Denial of Service (DoS)	Attack Distribution Flood-Like Oscillatory Attack	Capacity 1 Volts
Deep Learning Long Short-Term Memory	Fully Connected Layer Neurons 50	Hidden Neurons No. 200	Input/ Output Layer Type Sequence-Input Regression-Output	Max Epoch Counts 20-50

6. RESULTS AND DISCUSSION

To stop and secure wireless network invasions, the suggested technological model is used in this section to identify malware and cyber-attacks. Artificial intelligence (deep learning) techniques and software simulations utilizing the MatLab2020b program are used to verify whether security criteria are met. Additionally, text logs are emulated using MATLAB library functions, which are also used to model distributed DDoS attack flows. With the use of artificial intelligence techniques and a DL recurrent neural network (LSTM) algorithm, this model illustrates how DDoS assaults affect a TCP/IP dataset on a wireless cloud network and how to manage and identify such intrusions. The proposed model for detecting and securing WSNs 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. Figure 10 shows the implementation diagram of the proposed WSN architecture design employed to achieve the work environment.

Moreover, the design details of the proposed LSTM deep learning model might be demonstrated in Figure 11.

By observing Figure 11, we notice that the implementation plan of the virtual 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. Next, Figure 12 displays the distribution of the (TCP/IP) dataset stream through the WSN nodes.

By viewing Figure 12, it could be noted that the flow of data forms for information between the wireless network contracts 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 Figure 13.

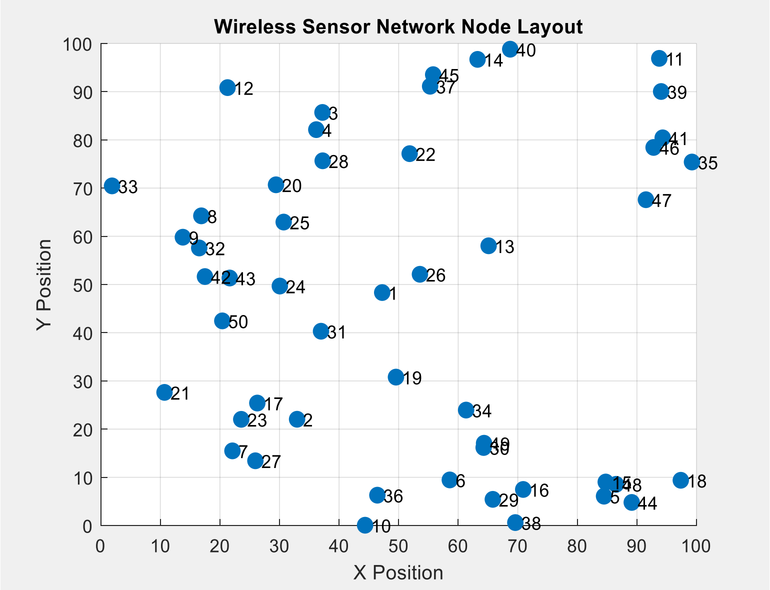
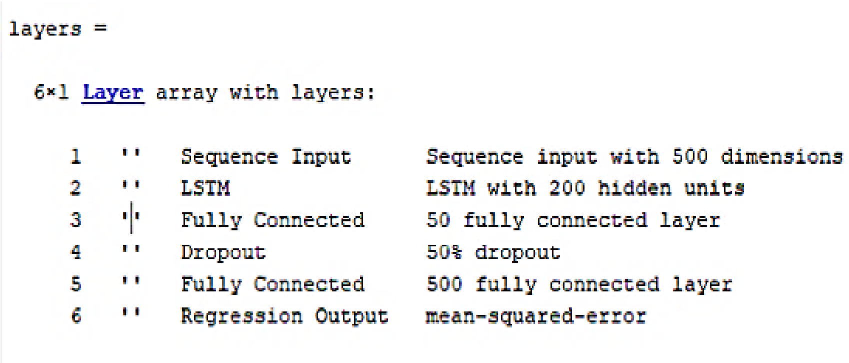


Figure 10. The implementation diagram of the proposed wireless sensor network (WSN) architecture design is employed to achieve the work environment



```
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 layer
     6   ''   Regression Output     mean-squared-error
```

Figure 11. Design details of the proposed long short-term memory (LSTM) deep learning model

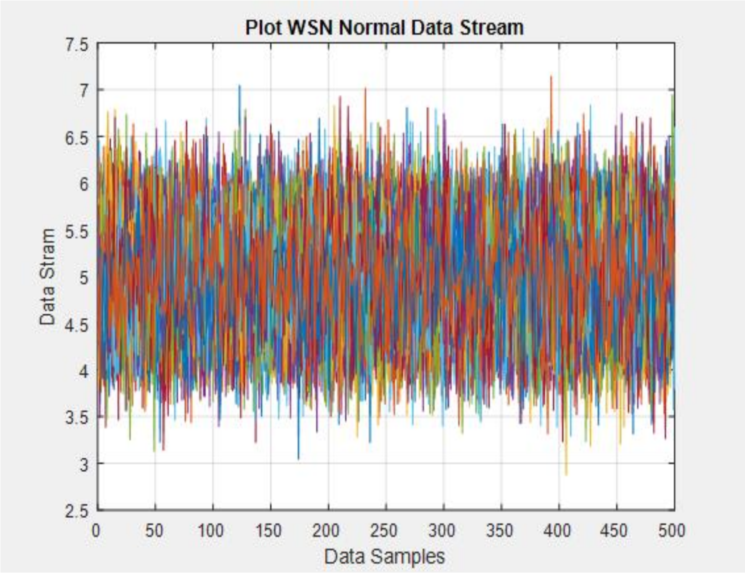
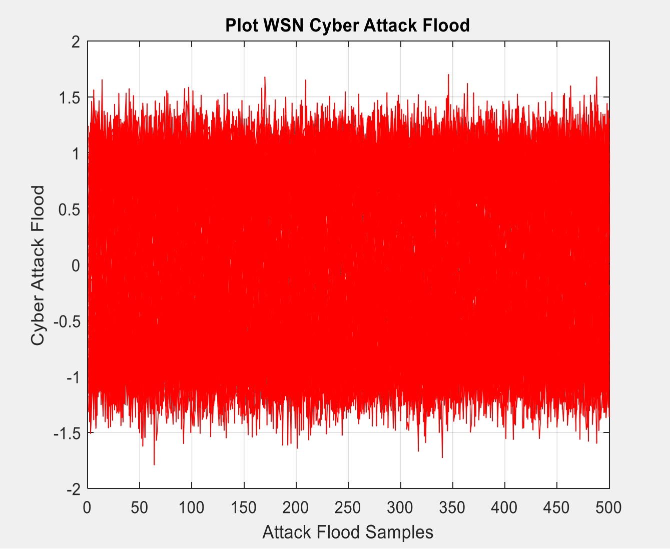


Figure 12. The distribution of the (TCP/IP) dataset stream through the wireless sensor network (WSN) nodes



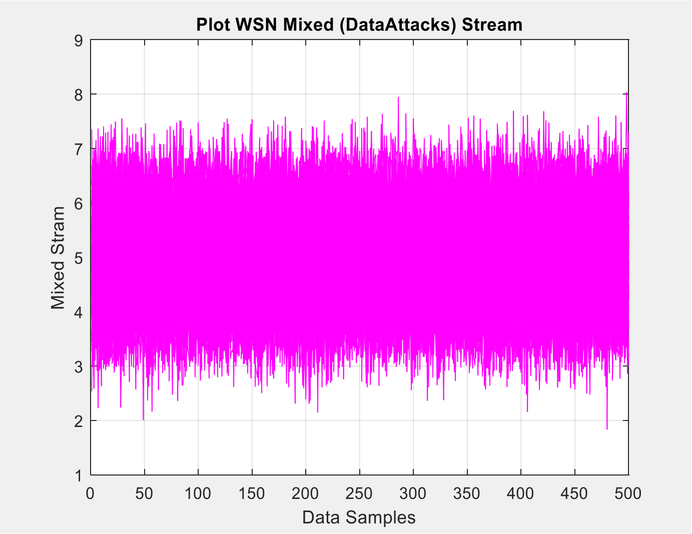
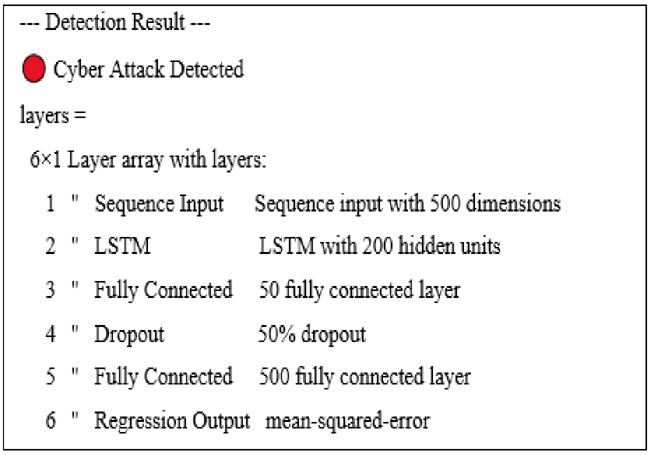


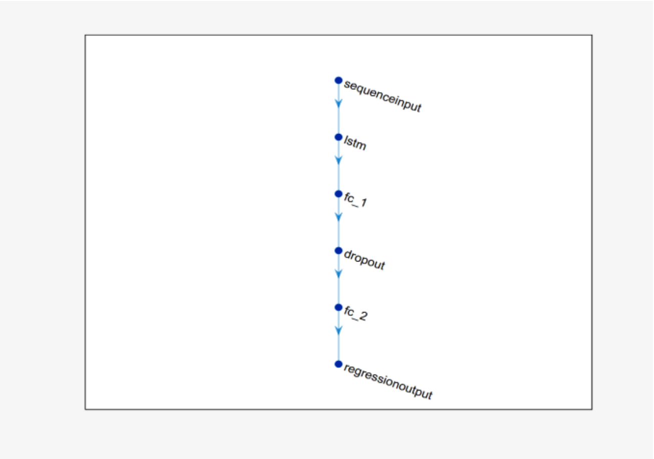
Figure 13. The generated cyber-attacks flood the samples

Figure 14. Resulting mixed, corrupted stream from adding the denial of service (DoS) attack flood to the data stream samples



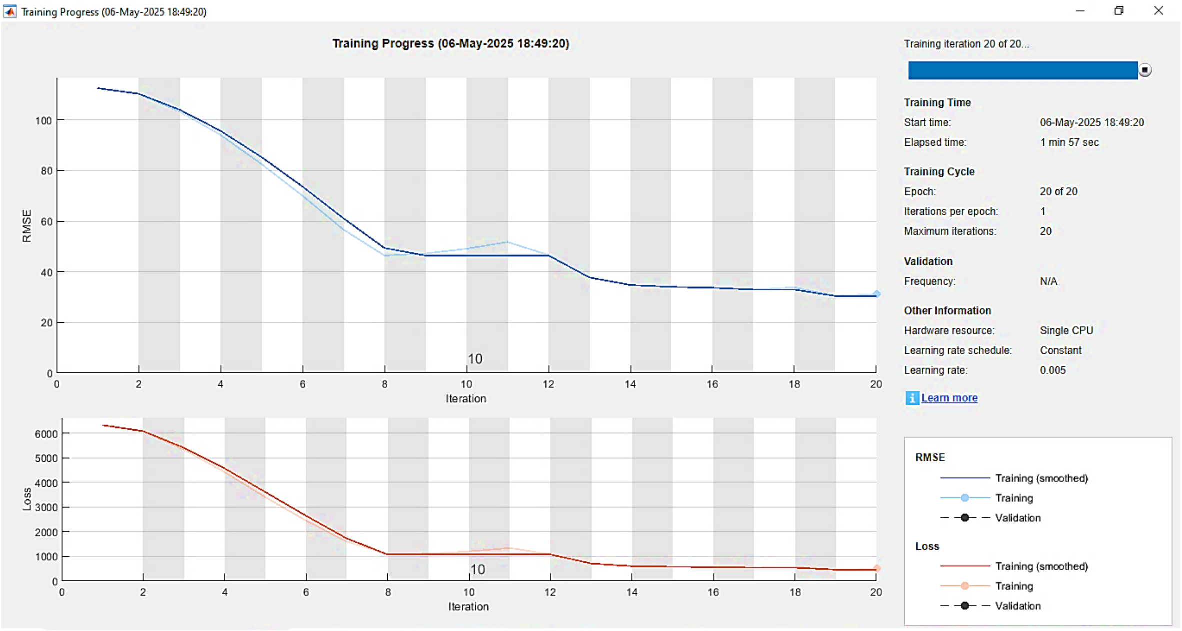
```
--- Detection Result ---
● Cyber Attack Detected
layers =
 6×1 Layer array with layers:
   1  ''  Sequence Input      Sequence input with 500 dimensions
   2  ''  LSTM                LSTM with 200 hidden units
   3  ''  Fully Connected     50 fully connected layer
   4  ''  Dropout             50%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

(a)



(b)

Figure 15. The flow diagram of the proposed deep learning long short-term memory (LSTM) algorithm layers structure through applying it to detect and reject the effect of the distributed denial of service (DDoS) cyber-attack floods: (a) MATLAB command prompt description; (b) Layers graph view



Training Progress (06-May-2025 18:49:20)

Training iteration 20 of 20...

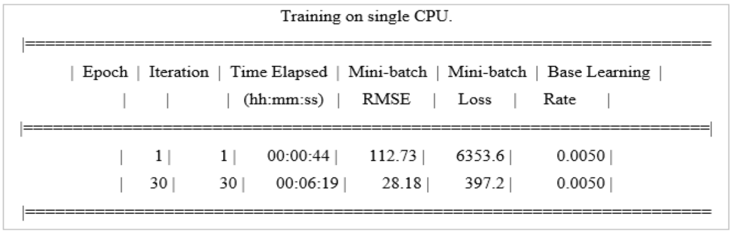
Training Time
Start time: 06-May-2025 18:49:20
Elapsed time: 1 min 57 sec

Training Cycle
Epoch: 20 of 20
Iterations per epoch: 1
Maximum iterations: 20

Validation
Frequency: N/A

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.005

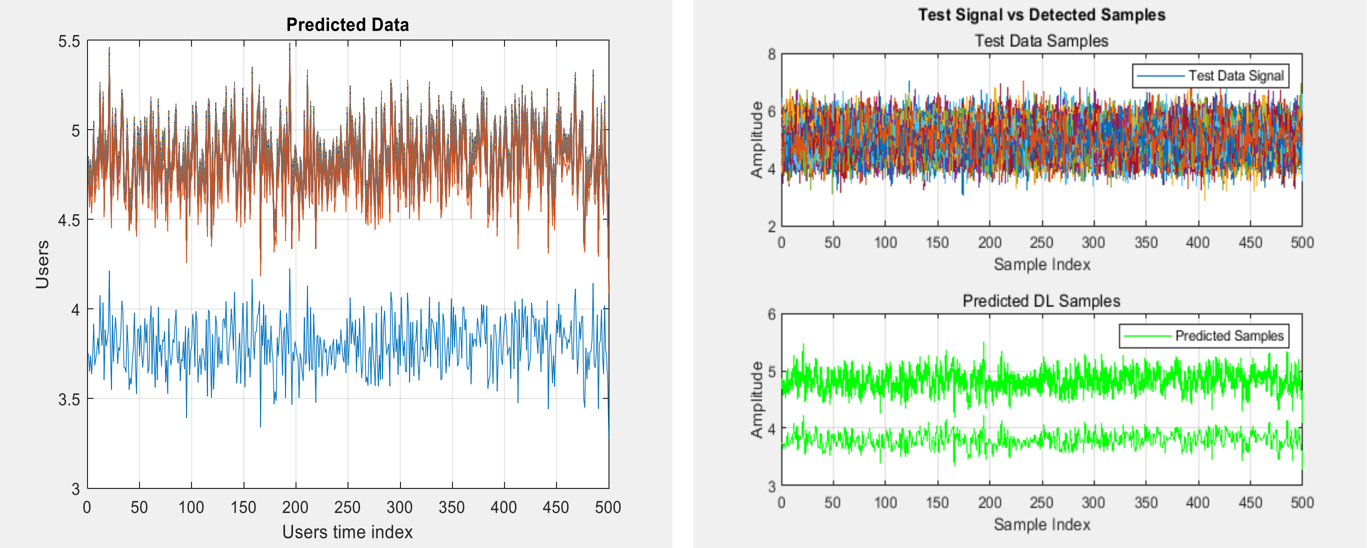
(a)



```
Training on single CPU.
|=========================================================================
| Epoch | Iteration | Time Elapsed | Mini-batch | Mini-batch | Base Learning |
|       |           | (hh:mm:ss)   |    RMSE    |    Loss    |     Rate      |
|=========================================================================|
|     1 |         1 |     00:00:44 |     112.73 |     6353.6 |        0.0050 |
|    30 |        30 |     00:06:19 |      28.18 |      397.2 |        0.0050 |
|=========================================================================
```

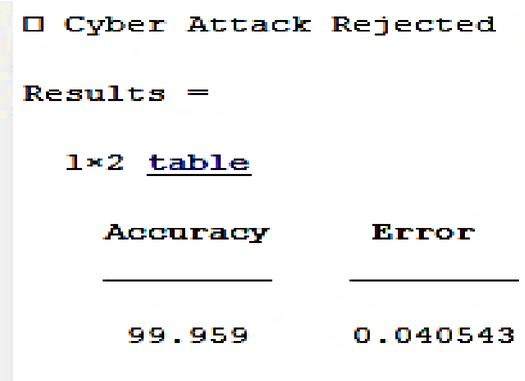
(b)

Figure 16. The training progress diagram of the deep learning algorithm operation as the corrupted mixed flows of network data enters to its internal layer: (a) Training progress curves; (b) Training progress table summary



(a) (b)

Figure 17. The results of the predicted samples at the output of the trained deep learning (LSTM) algorithm: (a) Long short-term memory (LSTM) predicted samples; (b) Comparison with original sent data samples



```
□ Cyber Attack Rejected

Results =

  1×2 table

    Accuracy      Error
    ________    ________

     99.959     0.040543
```

Figure 18. The efficiency and the error rate result for the prediction indicators of the proposed deep learning long short-term memory (LSTM) algorithm

As we might observe from Figure 13, 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 Figure 14. By reviewing Figure 14 above, the signal content resulting from inserting the flows of the service attack models into the original data models can be seen. Where 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 Figure 15.

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 (Figure 15(a)). Also, "the same Figure 15(a) shows the details of the contents of the DL 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, Figure 15(b) displays the same proposed DL algorithm structure in graph view.

Next, the training progress diagram of the DL 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 Figure 16.

As shown in Figure 16, 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. Where 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 Figure 17.

By observing Figure 17, we can see the expected sample results at the output of the trained DL algorithm (LSTM), which were highly consistent with the original data samples of the information sent through the WSN. 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. Figure 18 displays the efficiency results and the error rate for the prediction indicators of the proposed DL LSTM algorithm.

When looking at Figure 18, it can be observed that the training efficiency of the proposed DL algorithm in blocking the attack packets and extracting the original data samples was tested in the same quantity of data packets for the group N = 1000. A high training efficiency was recorded, reaching 99.96% with a very low error rate of 0.04%, as shown in the figure, that indicating the proposed model's ability to eliminate harmful DoS attack floods with successful rejection. Also, the performance measures obtained using the proposed model in detecting and blocking DDoS attacks for WSNs might be shown in Table 9.

Table 9. The performance measures obtained using the proposed model in detecting and blocking distributed denial of service (DDoS) attacks for wireless sensor networks (WSNs)

Measure	Value (%)
Training Accuracy (LSTM Model)	99.95
Prediction Efficiency	99.96
Error Rate	0.04
Number of WSN Nodes	50

Table 10. A comparative analysis might be achieved with recent deep learning-based wireless sensor networks (WSN) intrusion detection system articles

Study (Year, Authors)	Method / Technology	Dataset	Accuracy (%)	Key Limitation
Our Proposed Model	LSTM DL (MATLAB Cloud)	**UNSW-NB15**	99.95	Evaluated on a single dataset
Gueriani et al. [25]	Hybrid CNN-LSTM DL	CICIoT2023	High (> 99)	Real-time complexity
Gowdhaman and Dhanapal [27]	ResNet-Inception + SVM	NSL-KDD	99.46	Limited dataset generalizability

Moreover, a comparative analysis might be achieved with recent DL-based WSN IDS articles as presented in Table 10.

The study's comparative summary shows that our suggested model, which achieves a very high training accuracy of 99.95% and prediction efficiency of 99.96%, reveals a highly successful LSTM-based DL strategy for detecting DoS assaults in WSNs. These outcomes are better than or on par with more recent, sophisticated techniques like the ResNet-Inception SVM hybrid model by Gowdhaman and Dhanapal [27], which accomplished 99.46% accuracy but had a limited evaluation scope, and the Hybrid CNN-LSTM by Gueriani et al. [25], which likewise reports high accuracy on benchmark IoT datasets but faces real-world deployment challenges. The attached study's great attack rejection efficiency and reliable performance in MATLAB-based WSN simulations are its main advantages. The LSTM approach in the attached paper stands out for its usefulness, low mistake rates, and flexibility in unique wireless network situations, even if all of the studies highlight the promise of deep learning for IDS.

7. CONCLUSIONS

DL-based network models are crucial for security tasks like malware detection, intrusion detection, anomaly recognition, and node scanning. DL offers helpful solutions for neural network models in safeguarding data from breaches and attacks in WSNs. The efficacy of security improves, and the overall risks of cyber-attacks are avoided while distinguishing threats and reducing manual scanning. In order to identify data security and stop DDoS intrusions, a DL technique (LSTM) was suggested for the cloud sensor network in this study. These assaults disrupt services and have an impact on the integrity and quality of data while it is being transmitted. The DL model's effectiveness in thwarting various attacks in the cloud sensor network is determined by the network architecture as well as the computations for its layers and parameters. With a training and detection accuracy of 99.95% and an error rate of 0.04, the simulation validates the capacity to identify and stop denial-of-service attacks and to train the DL model using the standard NSW2019 dataset. The quality and effectiveness of the simulation results in fulfilling network security requirements were demonstrated by comparing them with the findings of recent research of a similar nature. The number of layers, computing complexity, the kind and amount of the dataset, and the type of malicious attacks all influence the suggested method and are seen as significant potential obstacles and limitations for additional study.

REFERENCES

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

[2] Beltran, A., Das, S. (2020). Particle swarm optimization with reducing boundaries (PSO-RB) for maximum power point tracking of partially shaded PV arrays. In 2020 47th IEEE Photovoltaic Specialists Conference (PVSC), Calgary, AB, Canada, pp. 2040-2043. https://doi.org/10.1109/PVSC45281.2020.9300516

[3] Hashim, S.A., Hamza, E.K, Kamal, N.N. (2024). Analyzing dynamic source routing protocol behavior in MANETs. Ingenierie des Systemes d'Information, 29(6): 2357-65. https://doi.org/10.18280/isi.290623

[4] Wei, W., Chen, N., Liu, Z., Zuo, M. (2020). Disturbance rejection control for a wastewater treatment process by a learning approach. Measurement and Control, 53(9-10): 1-10. https://doi.org/10.1177/0020294020909953

[5] Mustafa, Z., Hamza, E.K. (2025). Enhancing indoor positioning accuracy using a hybrid Li-Fi/Wi-Fi system with deep learning support. Eng Technol Appl Sci Res, 15(2): 21575-21585. https://doi.org/10.48084/etasr.10249

[6] Azlan, N.Z., Kamarudzaman, N. (2021). Soft pneumatic exoskeleton for wrist and thumb rehabilitation. International Journal of Robotics and Control Systems, 1(4): 440-452. https://doi.org/10.31763/ijrcs.v1i4.447

[7] Hamza, E.K., Ibraheem, E.K., Abou-loukh, S.J. (2025). Improvement and analysis of polar codes based on new radio-deep learning. International Journal of Intelligent Engineering & Systems, 18(3): 322-337. https://doi.org/10.22266/ijies2025.0430.22

[8] Hamza, E.K., Mohammed, L.A., Hasan, A.M. (2024). Dynamic parameter adjustment in ant colony optimization for energy efficiency in wireless sensor network. Journal of Engineering Science and Technology, 19(6): 2225-2249. https://jestec.taylors.edu.my/Vol%2019%20Issue%206%20December%202024/19_6_11.pdf.

[9] Alawad, N.A., Humaidi, A.J., Al-Araji, A.S. (2022). Improved active disturbance rejection control for the knee joint motion model. Mathematical Modelling of Engineering Problems, 9(2): 477-483. https://doi.org/10.18280/mmep.090225

[10] Aljuboury, A., Hameed, A., Ajel, A., Humaidi, A., Alkhayyat, A., Mhdawi, A. (2022). Robust adaptive control of knee exoskeleton-assistant system based on nonlinear disturbance observer. Actuators, 11(3): 78. https://doi.org/10.3390/act11030078

[11] Mahdavifar, S., Ghorbani, A. (2020). Application of deep learning to cybersecurity: A survey. Neurocomputing, 347: 149-176. https://doi.org/10.1016/j.neucom.2019.12.099

[12] Wani, A.R., Rana, Q., Saxena, U., Pandey, N. (2020). Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques. In 2020 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, pp. 870-875. https://doi.org/10.1109/AICAI.2019.8701238

[13] Gill, S.S., Buyya, R. (2020). SECURE: Self-protection approach in cloud resource management. IEEE Cloud Computing, 5(1): 60-72. https://doi.org/10.1109/MCC.2018.011791715

[14] Ibrahem, R., Hamza, E. (2024). The comparative analysis between Distance DEEC and IoT DEEC based on network lifetime and energy consumption. International Journal of Intelligent Engineering & Systems, 17(5): 142-157. https://doi.org/10.22266/ijies2024.1031.13

[15] Ferrag, M.A., Maglaras, L., Moschoyiannis, S., Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50. https://doi.org/10.1016/j.jisa.2019.102419

[16] Subashini, Krishnaveni, M., Dhivyaprabha, T.T., Shanmugavalli, R. (2020). Review on intelligent algorithms for cyber security. In Handbook of Research on Machine and Deep Learning Applications for Cyber Security, IGI Global, pp. 1-22. https://doi.org/10.4018/978-1-5225-9611-0.ch001

[17] Berman, D.S., Buczak, A.L., Chavis, J.S., Corbett, C.L. (2020). A survey of deep learning methods for cyber security. Information, 10(4): 122-157. https://doi.org/10.3390/info10040122

[18] Hitaj, B., Ateniese, G., and Perez-Cruz, F. (2017). Deep models under the GAN: Information leakage from collaborative deep learning. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 603-618. https://doi.org/10.1145/3133956.3134012

[19] Hemdan, E.E.D., Manjaiah, D.H. (2018). Digital investigation of cybercrimes based on big data analytics using deep learning. In Deep Learning Innovations and Their Convergence with Big Data, pp. 79-101. https://doi.org/10.4018/978-1-5225-3015-2.ch005

[20] Wickramasinghe, C.S., Marino, D.L., Amarasinghe, K., Manic, M. (2020). Generalization of deep learning for cyber-physical system security: A survey. In Proceedings of the 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, pp. 745-751. https://doi.org/10.1109/IECON.2018.8591773

[21] Goyal, A., Mishra, S., Chaurasiya, V.K. (2023). Intrusion detection in wireless sensor networks using deep learning. In 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-6. https://doi.org/10.1109/INCET57972.2023.10170318

[22] Zeadally, S., Adi, E., Baig, Z., Khan, I.A. (2020). Harnessing artificial intelligence capabilities to improve cybersecurity. IEEE Access, 8: 23817-23837. https://doi.org/10.1109/ACCESS.2020.2978832

[23] Godala, S., Kumar, M.S. (2023). Intrusion detection by stacked deep ensemble model with entropy and correlation feature set. International Journal of Intelligent Systems and Applications in Engineering, 11(1): 1-9. https://doi.org/10.22266/ijies2023.01123

[24] Belarbi, O., Spyridopoulos, T., Anthi, E., Mavromatis, I., Carnelli, P., Khan, A. (2023). Federated deep learning for intrusion detection in IoT networks. In GLOBECOM 2023-2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, pp. 237-242. https://doi.org/10.1109/GLOBECOM54140.2023.10437860

[25] Gueriani, A., Kheddar, H., Mazari, A.C. (2024). Enhancing IoT security with CNN and LSTM-based intrusion detection systems. In 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), EL OUED, Algeria, pp. 1-7. https://doi.org/10.1109/PAIS62114.2024.10541178

[26] Shen, J., Yang, W., Chu, Z., Fan, J., Niyato, D., Lam, K.Y. (2024). Effective intrusion detection in heterogeneous Internet-of-Things networks via ensemble knowledge distillation-based federated learning. In ICC 2024-IEEE International Conference on Communications, Denver, CO, USA, pp. 2034-2039. https://doi.org/10.1109/ICC51166.2024.10622262

[27] Gowdhaman, V., Dhanapal, R. (2024). Hybrid deep learning-based intrusion detection system for wireless sensor network. International Journal of Vehicle Information and Communication Systems, 9(3): 239-255. https://doi.org/10.1504/IJVICS.2024.139627

[28] Shi, L., Li, K. (2022). Privacy protection and intrusion detection system of wireless sensor network based on artificial neural network. Computational Intelligence and Neuroscience, 2022(1): 1795454. https://doi.org/10.1155/2022/1795454

[29] Yaras, S., Dener, M. (2024). IoT-based intrusion detection system using new hybrid deep learning algorithm. Electronics, 13(6): 1053. https://doi.org/10.3390/electronics13061053

[30] Gupta, B.B., Chui, K.T., Gaurav, A., Arya, V. (2023). Deep learning based cyber attack detection in 6G wireless networks. In 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, pp. 1-5. https://doi.org/10.1109/VTC2023-Fall60731.2023.10333795

[31] Pant, P., Kumar, A., Vashishtha, L.K., Dash, S., Ray, N.K., Sahu, S.K. (2024). A comparative study of deep learning techniques for network intrusion detection. In 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC), Bhubaneswar, India, pp. 722-727. https://doi.org/10.1109/ESIC60604.2024.10481540

NOMENCLATURE

MATLAB	Matrix Laboratory (proprietary software name)
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NSL-KDD	Network Security Laboratory – Knowledge Discovery in Databases (benchmark dataset)
PSO	Particle Swarm Optimization
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RSA	Rivest–Shamir–Adleman (cryptosystem)
RBS	Reference Broadcast Synchronization (in time synchronization for WSNs)
SGD	Stochastic Gradient Descent
SPINS	Secure Protocols for Sensor Networks
SVM	Support Vector Machine
TCP/IP	Transmission Control Protocol/Internet Protocol
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TinySec	Tiny Security (lightweight link-layer security protocol for WSNs)
TON-IoT	Towards Next-generation IoT (dataset from UNSW Canberra)
UNSW-NB15	University of New South Wales – Network-Based 2015 (dataset name)
WSN(s)	Wireless Sensor Network(s)