Ain Shams Engineering Journal xxx (xxxx) xxx

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: www.sciencedirect.com

Full Length Article

Dos attack detection using fuzzy temporal deep long Short-Term memory algorithm in wireless sensor network

P. Sathishkumar *, A. Gnanabaskaran, M. Saradha, R. Gopinath

Department of Computer Science and Engineering, K.S Rangasamy College of Technology, Tamilnadu, India

ARTICLE INFO

Keywords:
Wireless sensor network (WSN)
Denial of service (DoS)
Intrusion detection system (IDS)
Crow search algorithm
Deep LSTM algorithm

ABSTRACT

Wireless sensor networks (WSNs) are becoming more prevalent for data collection and monitoring. Compared to other sensing methods, each sensor node is typically tiny, inexpensive, and simple to implement. Consequently, a wide range of applications and industries utilize WSNs extensively. WSNs, on the other hand, are vulnerable to various security attacks and threats. Because they require more resources (including power, storage, processing power, and bandwidth) to develop defenses, it is necessary to have an efficient Intrusion Detection System (IDS) to detect these attacks and ensure security, even with these constraints. Traditional IDSs are losing effectiveness due to the increasing intelligence, frequency, and sophistication of malicious attacks. One of the most common attacks threatening WSNs is denial of service (DOS). To overcome these challenges, this research work proposes a novel fuzzy model incorporating the Deep Long Short-Term Memory (LSTM) algorithm for intrusion detection in WSNs. The proposed algorithm uses temporal constraints and fuzzy rules for weight fitting in decision-making, as well as neural networks in deep LSTM classifiers. The proposed method consists of (i) a pre-processing stage for preparing the data for further evaluation, (ii) a dynamic feature selection stage to select the most adaptable and efficient features and reduce processing time, and (iii) a detection stage to identify Denial-of-Service (DoS) attacks. It introduces the Crow Search Algorithm (CSA) for feature selection, aimed at optimizing feature efficiency. The Fuzzy Logic with LSTM model introduces a unique methodology where multiple sensor nodes collaboratively train a central global model while protecting private data and addressing privacy concerns effectively. This approach enables the model to detect sophisticated and previously unknown cyber threats by analyzing local and temporal correlations within network patterns, specifically tailored for identifying various types of DoS attacks. The model utilizes specialized KDDCup99 and NSL-KDD datasets. The suggested FL-LSTM approach achieved the highest accuracy (99.58%), the highest precision (98.42%), the highest recall (98.45%), and the highest f-score (98.36%). Compared to the conventional algorithm, the proposed FL-LSTM outperforms.

1. Introduction

Wireless sensor networks have been introduced in many applications recently and are essential to research. A wireless sensor network comprises small, low-power sensor devices distributed spatially with radio transceivers to detect and collect data in various environments. Unlike wired networks, wireless sensor networks can adapt to harsh environments. [1,2]. Wireless multimedia sensor networks are frequently the most recent development in WSNs. It gathers scalar metrics from the sensors used, which have the ability to obtain and process media for streaming. This information includes images, audio, and video streams gathered from sensor nodes (SNs). Various fields, including remote monitoring, home automation, and environmental monitoring, have subsequently studied WSNs [3].

WSN collects information and sends it between one of the base station (BS) receiving nodes, which requires limited functionality, operation without supervision, and random deployment. This leaves the SN vulnerable to further attacks and security breaches in hostile environments, such as enemy territory [4–6]. Depending on the implementation method, network attacks may include DoS attacks, replay attacks, deception attacks, and others. One of the most common and serious threats to WSNs is a DoS attack, which seeks to disable and disrupt the services they provide. Unlike other types of attacks, DoS attacks are deliberate attempts to prevent the network from functioning properly,

* Corresponding author.
E-mail addresses: psvsathishphd@gmail.com (P. Sathishkumar), gnanabaskarana@ksrct.ac.in (A. Gnanabaskaran), saradha@ksrct.ac.in (M. Saradha), gopinath@ksrct.ac.in (R. Gopinath).

https://doi.org/10.1016/j.asej.2024.103052
Received 24 September 2023; Received in revised form 28 June 2024; Accepted 1 September 2024

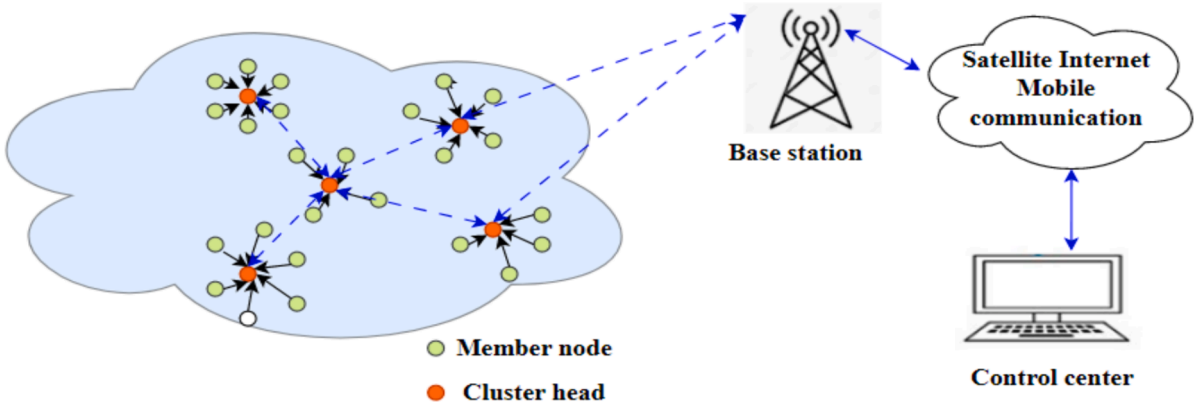

Fig. 1. WSN logical structure diagram.

resulting in lower system performance. WSN nodes typically have limited computational capabilities, memory, and energy resources, making them vulnerable to attacks that deplete these finite resources. Attackers can exploit the shared wireless medium in WSNs for jamming attacks, disrupting communication between nodes and with base stations. Moreover, the decentralized nature of WSNs complicates the detection and mitigation of DoS attacks, as attackers can simultaneously target multiple nodes or exploit vulnerabilities across the network.

The changes in network topologies that happen quickly because of node mobility or environmental factors make things more complicated by making routing protocols and communication paths less reliable during attacks. Furthermore, real-time applications that rely on timely data transmission, such as environmental monitoring or healthcare systems, are particularly vulnerable to the disruptive effects of DoS attacks. Such attacks have the potential to disrupt critical operations and decision-making processes. Thus, protecting WSNs from DoS attacks necessitates tailored security measures that address these issues while preserving the network's efficiency and operational capabilities. In WSNs, IDSs play an important role in fortifying defenses against various security threats. IDS systems closely monitor network traffic and node behavior, looking for anomalies or suspicious activity that could indicate a security breach. By quickly identifying and notifying operators of potential threats, IDS enables timely responses to mitigate risks, protect sensitive data, and ensure network reliability and integrity. The introduction of deep learning-based IDS algorithms designed specifically for WSNs significantly increases the network's ability to detect and respond to security threats with greater effectiveness.

Several researchers have proposed various IDS to assist in the detection of these security attacks. The literature suggests several methods for detecting DoS attacks, but most of these studies concentrate on identifying patterns and types of attacks, rather than the tools used to carry them out.

Artificial intelligence (AI) introduces ML algorithms, significantly benefiting WSNs' adaptability [7]. The selection of advanced AI techniques for wireless sensor networks has become increasingly stringent over the last decade [8]. Artificial intelligence focuses on biologically inspired models, including fuzzy systems, neural networks (NN), and evolutionary techniques. Fig. 1 depicts a typical WSN logical structure diagram. We commonly use techniques like K-Nearest Neighbors (KNN), DT, and NN to detect DoS attacks on WSNs. The algorithm used depends on the specific requirements and type of DoS attack. Employing diverse and large data sets and suitable evaluation parameters can overcome the challenge of ensuring the accuracy of ML models.

Overall, the DL-based approach offers a promising solution for detecting DoS attacks on WSNs while not interfering with standard device functionality. The work concentrates on the need for a practical DL-based classification approach to defend against DoS attacks. This work proposes a novel feature selection technique called CSA for efficient feature selection, as well as a unique classification algorithm known as the deep LSTM algorithm for robust classification. The algorithm employs fuzzy rules and time constraints to adjust neural network weights and make decisions in deep LSTM classifiers. The goal is to use the fewest number of attributes while accurately classifying both normal and attack traffic.

This work's main contributions are as follows:

- The technique is divided into the following modules: pre-processing, feature selection, and classification, and it uses fuzzy with deep LSTM algorithms to classify network traffic on the NSL-KDD and KDDCup 99 datasets.
- The crow search algorithm technique is proposed for dynamically selecting attributes to identify and prevent DoS attacks on WSNs, thereby reducing the number of features.
- The algorithm adjusts the neural network weights and deep LSTM classifier decisions using fuzzy rules and temporal constraints.
- Analysis of the suggested method uses five different classifiers: intelligent decision tree, DNN, RNN, CNN, and LSTM.
- The deep LSTM classifier was identified as the optimum performing classifier, with 99.76 % accuracy.
- Providing a reliable approach for identifying and avoiding DoS attacks in WSNs is essential for ensuring the security of these interconnected networks.
- Significant advantages of the suggested Intrusion Detection System and novel feature selection and classification algorithms, such as lower false positive rates, network latency, and power utilization in WSNs.

We organize the remainder of the work as follows: Section II examines recent literary works. Section III discusses the fundamental concept of the proposed method. Section IV presents and analyzes the experiment's results in order to evaluate the proposed strategy. Section V concludes with an overview of the key findings and recommendations for future research.

2. Literature survey

Salim Salmi et al. (2022) demonstrated a CNN-LSTM system for detecting and categorizing DoS attacks. This study uses a computer-generated data set derived from wireless sensor network detection systems. To simulate the WSN environment, we use an NS-2 network simulator with the LEACH protocol. We tested the suggested intrusion detection model over ten training epochs and found that it was precise (0.894), accurate (0.889), and recall (0.894), with scores ranging from 0 to 1 [4].

Ramesh et al. (2020) proposed an optimized DNN algorithm for identifying DoS in WSN. They employ an adaptive particle swarm

optimization (PSO) algorithm to choose the mandatory metrics. The proposed DNN-based DoS detection method outperforms all test cases. The neural network's multilayer neurons aid the proposed method's detection accuracy and low time consumption. The experimental results are based on the assumption that optimization techniques enhance the efficiency of the learning process [9].

M. Premkumar et al. (2020) presented a unique DoS diagnosis system using a defense mechanism based on DL (DLDM). This document introduces new algorithms for detecting DoS attacks, including jamming, bootstrapping, flooding, and exhaustion. The module selects the DL process based on network efficiency. Additionally, the module can pass the information to a predictor, which adheres to a schedule and executes time-consuming and computationally intensive tasks. The suggested framework can do a good job of finding DoS attacks more quickly and effectively, as shown by the results of experiments and performance tests of the MAS and DLDM methods [10].

Francesco Flammini et al. (2023) presented research on wireless communication techniques for detecting and preventing attacks and malicious nodes. The proposed model identifies and terminates hostile nodes from the network. The performance parameters utilized for evaluation are attack prevention, detection accuracy, security, computation time, and network overhead. We use various evaluation metrics to evaluate and compare the effectiveness of the model to more traditional methods. Furthermore, collaborative detection makes every node much more responsible for its actions [11].

Mohammad Al-Nayim et al. (2020) analyzed the current state of DDoS detection technology to find the best AI-based detection systems to locate DDoS attacks. This review followed the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) statement. Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) are the most widely utilized AI-based algorithms for detecting DDoS on WSNs, according to 15 of the 983 who met the inclusion criteria. The DDoS attack detection system in WSN, using AI technology, performs admirably [12]. S. Ramesh Kumar et al. (2023) created a deep CNN with state-action relational mapping to detect DDOS attackers on smart farming WSNs. This method examines the neighbors' information and determines that the malicious node is present in the system. In terms of TLDQN's performance, the transfer rate is 95 % at 90 data bits per second (bps), the DDOS detection accuracy is 89 %, and the multipath analysis is 95 % [13].

Jiabin Li et al. (2020) suggested a way to find DDoS attacks that has three main parts: sliding time windows to speed up entropy calculations, one-way filters to find DDoS attacks early on instead of after a block, and a quintile bias checking algorithm to get the best results for finding DDoS. Finally, this results in efficient real-time analysis to detect IoT distributed denial-of-service attacks as soon as possible. QuinDC also has accurate performance and low latency, making it suitable for use in networks with real-time requirements, such as intrusion prevention networks in IoT systems [14]. Zhipeng Liu et al. (2020) proposed various ML algorithms to detect anomalies in IoT network intrusion datasets efficiently. A zero-day attack does not identify the attack signature, but subsequent network activity deviates from standard traffic patterns, allowing the IDS to detect and respond to anomalies. The proposed preliminary experimental results using all of the dataset's features are auspicious, expanding the proposed work from binary to multiclass classification [15].

Suman Sankar et al. (2017) proposed SoftThings, a secure SDN-based IoT architecture, to detect abnormal behaviors and attacks early and mitigate them appropriately. SDN controllers use ML to monitor and learn from changes in IoT devices over time. Experiments in the Mininet simulator aid in quickly identifying attacks on IoT devices and initiating appropriate mitigation procedures. The experimental findings demonstrate that the suggested technique is capable of (i) accurately detecting and recovering from attacks, and (ii) quickly mitigating attacks within seconds [16]. McDermott C.D. et al. (2018) suggested a model for detecting malware activity in consumer IoT devices and systems. The development of an RNN detection method based on bidirectional LSTM (BLSTM-RNN) is a new application of deep learning. We compare the developed detection method with an LSTM-RNN to detect four attack vectors used by the Mirai botnet, and assess the accuracy and loss. Despite adding epoch overhead and increasing processing time, the paper shows that the two-way approach is an asymptotically better model over time [17].

Table 1
A comparison of existing models for detecting DoS attacks in WSN.

Reference number	Technique	Advantages	Disadvantages
[14]	Entropy-based algorithm	Defending against DDoS attacks	Server is still responding to malicious attacks.
[15]	KNN algorithm and XGBoost	KNN accuracy was 99 % and XGBoost accuracy was 97 %.	The suggested approach is not recommended for enormous data sets.
[16]	SVM	Appropriate for low computational complexity environments.	A simulated approach does not replicate the properties of actual devices.
[17]	A bidirectional LSTM	Detect DoS attacks accurately.	A simulated approach is possible, and detection of various attacks is impossible.
[18]	Software Defined WirelessNetworks (SDN) and cloud	Excellent detection rate.	Failure proclivity if the input is ambiguous there is an issue in the resource-constrained device.
[19]	Deep Learning	It had a 100 % detection rate.	There was no mitigation strategy developed, and it was not scalable.
[20]	Using unsupervised DLwith deep AE	Detect malware activity	It is challenging to capture some normal device behavior.
[19]	Implementation of a distributed DL method at the network interface (for example, the fog layer)	Solution for detecting zero-day attacks.	A signature-based system's command processing is expensive and incapable of dealing with the most recent threats.
[23]	CNN	Achieved 99 % accuracy	It has an impact on the model's learning.
[24]	Wi-Fi deauthentication attackdetection scheme	Updated hardware with built-in encrypted management frame functionality can protect against such attacks.	Only DNS, ARP, and spoofing attacks were investigated.
[25]	Deep learning	The accuracy for binary traffic detection is 99.95 %, and the accuracy for multiclass traffic detection is 99.92 %.	Real-time analysis was not carried out.
[26]	least squares SVM	Improve log security, reduce bandwidth consumption, and relieve resource exhaustion.	Packet loss and throughput are not measured.

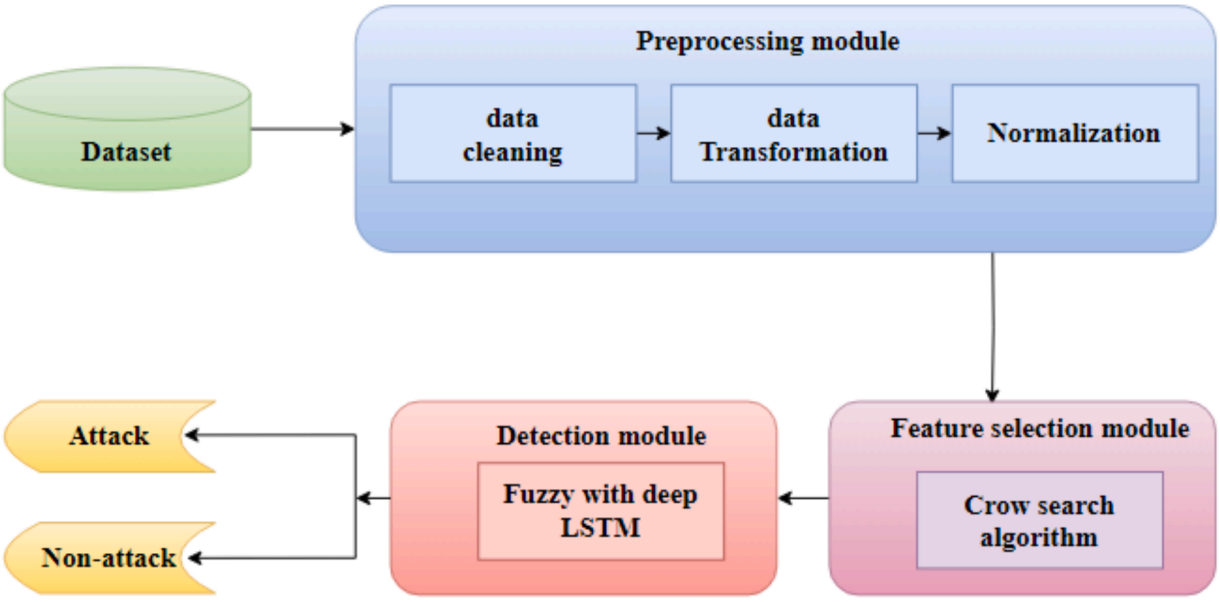

Fig. 2. Proposed Dos attack detection system.

Pradeep Kumar et al. (2018) developed a security attack mitigation framework with a highly programmable monitoring network to detect attacks. It has a flexible control architecture that allows for the rapid definition of attacks and specific party reactions. We compare the performance of the proposed architecture with existing models for various performance metrics. The evaluation outcomes show that the suggested architectural model is efficient in dealing with the security challenges of the new network paradigm [18]. Diro et al. (2018) presented the implementation of a distributed DL technique to detect zero-day attacks. They use simple deep feed-forward networks to make granular decisions that traditional ML methods cannot. They tested their proposed method using 1-vs-n coding on the NSL-KDD dataset, achieving more than 99 % accuracy. However, they tested the model on peripheral devices and found no model performance limitations [19].

Yair Meidan et al. (2018) described and tested a new network-based intrusion detection model that uses deep auto-encoders to spot strange network traffic coming from IoT networks that have been hacked. The suggested method used nine IoT devices and two well-known IoT-based botnets, BASHLITE and Mirai, to evaluate the proposed model. The simulation outcomes demonstrate that the proposed model can accurately and instantly identify attacks launched from infected IoT networks that are part of a botnet [20]. Abebe Abeshu et al. (2018) suggested a new cyber security model, DL, for social IoT attack detection. The study compares the performance of deep models with traditional ML methods, and compares attack detection with centralized detection networks. Simulation outcomes show that the distributed attack detection network outperforms centralized detection networks based on DL models. Deep models have also been more effective than shallow models at detecting attacks [19]. Table 1 depicts a summary of the existing models for detecting network DoS attacks.

Many research fields widely use LSTM networks due to their capacity to model sequential data and capture long-term dependencies. Aytaç Altan et al. (2022) came up with a new way to predict the price of crude oil that uses LSTM, technical indicators like trend, volatility, and momentum, and the chaotic Henry gas solubility optimisation (CHGSO) method. The authors derive the model's features from trend, momentum, and volatility technical indicators. The results show that the proposed prediction model can effectively handle the chaotic and nonlinear dynamics observed in both WTI and Brent crude oil prices [21]. Aytaç Altan et al. (2023) [10] presented a resilient artificial intelligence (AI)-driven model that addresses nonlinear dynamics with low computational complexity and high classification accuracy. Using the recurrent neural network-long short-term memory (RNN-LSTM) architecture, this model sorts fundus images into groups based on the stages of DR disease. It focuses on choosing the smallest feature sets needed to keep the accuracy of the groups. The results show that the suggested model handles nonlinear dynamics in colour fundus images well while using very few computing resources. This makes it very useful for diagnosing all stages of DR disease [22].

2.1. *Problem statement*

The limitation of the above-mentioned cutting-edge ML and DL methods is that they select all features in the data set for classification. On the other hand, we must filter out unimportant data set features that add space and time complexity. As a result, when classifying normal and attack traffic, an efficient classification technique is required to select the fewest number of variables to utilize limited system resources. ML classification algorithms have emerged as a viable option for thwarting service attacks as an additional technique. However, due to the need for proper and comprehensive evaluations of these methods, determining their valid contribution to improving DoS attack detection on WSNs takes much work. This study investigates a technique that considers fewer features while employing efficient and intelligent DL DoS detection techniques to achieve the same or even optimum accuracy.

3. Proposed methodology

Despite careful network infrastructure design, each WSN layer is vulnerable to many DoS attacks. If the attacker's intrusion level is minimal, it may be impossible to identify him. Due to the sequence of diverse behaviors of malicious nodes, a DoS attack leads to the unauthorized usage of the network system. To improve the detection of DoS assaults, this work offers a deep learning-based categorization technique to improve DoS assault detection. There are three stages in the system. It is a critical module in the proposed framework for detecting attacks utilizing the sub-modules illustrated in Fig. 2. In real-time, the detection module dynamically acquires the parameters representing the attack nodes via the communication module. If the process is mobile and active, the detection module sets a flag for each sub-module and sends the relevant data to the transmission module. Each module is in charge of a distinct type of attack, and if the transmission module raises the

Table 2
Training and testing connection records from NSL-KDD and KDDCup 99 datasets.

Attack category	Description	Data intsances			
		NSL-KDD		KDDCup99	
		Train	Test	Train	Test
Normal	Normal connection records	67,343	9,710	97,278	60,593
DoS	The attacker intends to deplete network resources	45,927	7,458	391,458	229,853
Probe	Acquiring detailed statistics of the system and network configuration information	11.656	2.422	4,107	4,166
R2L	Unauthorized accessibility from a remote system	995	2,887	1,126	16,189
U2R	Acquiring root or super-user rights on a specific machine	52	67	52	228
Total		125,973	22,544	494,021	311,029

alarm, the countermeasure unit begins blocking DoS attempts on the network.

3.1. Dataset

The dataset is utilized to determine if the method can detect attacks accurately. The dataset's quality ultimately determines the accuracy of any Network IDS (NIDS). This section looks at two datasets: KDD Cup'99 and NSL-KDD. Its features are detailed in below.

3.1.1. KDD Cup'99 data set

DARPA generated the KDD'99 dataset in 1999 by [27] using network traffic from the 1998 dataset. Each network connection contains 41 preprocessing functions. The KDD'99 dataset contains four types of functions: basic functions (#1 to #9), content functions (#10 to #22), time-based flow functions (#23 to #31), and host flow-based functions (#32–41). KDD'99 has 4,898,430 records, making it larger than other datasets. DoS, R2L (unauthorized access from a remote computer), U2R (unauthorized root access), and probing are the basic types of assaults. Multiple data mining techniques were used in the KDD'99 dataset to identify network traffic intrusions. This dataset is primarily utilized to develop IDSs. According to a statistical study, the KDD dataset has two significant flaws impacting system performance. The most severe issue with the KDD dataset is the large number of duplicate records.

3.1.2. NSL-KDD

The NSL-KDD intrusion dataset is a condensed version of the KDDCup 99 intrusion dataset [28]. The NSL-KDD protects ML techniques from bias. Compared to the KDDCup 99 dataset, this may be sufficient for abuse detection. The representation of real-time network traffic profile attributes also plays a role. Table 2 displays the complete data for NSL-KDD.

3.2. Data preprocessing module

The KDD Cup99 and NSL-KDD datasets are combined before being divided. The shuffling process separates the dataset into a training and testing dataset, allocating 80 % of the split for training and 20 % for testing. One proposed method is data preprocessing (cleaning, transformation, and normalization). The proposed method's initial step is to preprocess the dataset to generate refined data. The research conducted two critical steps while preprocessing: cleaning and converting the cleansed data to numeric values (0, 1, 2, 3, 4). The preprocessing module initially cleans the dataset, eliminating spaces, nulls, and duplicates. We then train and test the suggested model on preprocessed information to evaluate its effectiveness in identifying DoS attack patterns. After cleaning the dataset, it is normalized to a standard scale of 0 to 1. We label benign samples as 0, and DoS samples as 1. Generally, we can classify packets based on the wealth of information they contain. These characteristics aid in distinguishing between regular and DoS attacks [29].

3.3. Feature selection module

Feature selection deletes irrelevant and worthless information to improve the classifier's learnability and predictability. Random Forest is one of the most extensively utilized ML approaches for feature selection. They are beneficial because they provide predictive results with minimal overfitting, solid performance, and are easy to understand. From the tree, one can easily infer the correlation between each variable and its comprehension. We use the suggested Crow search optimization technique to optimize feature selection given a reference dataset.

3.3.1. Crow search algorithm

In 2016, Askarzadeh [30] presented the Crow Search Algorithm method as a *meta*-heuristic algorithm. The algorithm was primarily motivated by the crow's search mechanism for hidden food. Crows are thought to be among the most intelligent birds. Their brains are more significant than their bodies. Crows are astute. The brain of a Crow is slightly smaller than that of a human. They were also self-aware and able to recall faces during the mirror test. When they notice a threatening crow, they use sophisticated communication to warn the other crows. They may also store and recall food for several months. They earn the nickname "thieves" due to their tendency to steal food from other birds. They predict the thief's actions based on their own experience as a thief. They are pretty careful. When crows report a robbery, they flee into hiding. It enables people to avoid becoming future victims of any kind. The four fundamental principles of CSA are as follows:

- Crows dwell in the form of a flock.
- Crows remember the locations of food hiding spots.
- Crows are particularly wary of thieving.
- They defend their caches with a chance of success.

3.3.2. Mathematical model of CSA

Assume the population of crows is the total amount of dimensions and $y^{j,t}$ denotes the current location of the j_{th} crow in the search space at iteration t, where j = 1, 2,… M. t_{Max} gives the greatest possible number of iterations. Every crow has a memory that allows it to recall where it hides. The concealed location of crow's j at iteration t is $N^{j,t}$. $N^{j,t}$ represents the optimum position gained by Crow j thus far. Crow j wants to find out where raven z is hiding while iterating over t. In this example, two scenarios are possible:

Case 1: Crow z is unaware that Crow j is following it. Crow j approaches Crow Z's hiding spot. Crow j's updated position is defined as follows:

$$y^{j,t+1} = y^{j,t+1} + R_j \times fl^{j,t} \times (N^{z,t} - y^{j,t}) \quad (1)$$

where fl is the length of the flight. R_j represents an arbitrary number between 0 and 1.

fl significantly impacts search capability. fl with a low-value result in a local search, whereas fl with aa high-value result in a worldwide search.

Case 2: Crow z is aware that Crow j is following it. Crow Z will shift its position in the search space to preserve its cache. The two preceding cases are described as follows:

Table 3
Fuzzy inference modelling.

Cost	Time (hours)	Behaviour	Chance of intrusion
low	Ordinary	Good	Normal user
low	Ordinary	Normal	Normal user
low	Ordinary	Bad	Normal user
low	Night time	Good	Normal user
low	Night time	Normal	Normal user
low	Night time	Bad	Less probable intruder
low	Peak	Good	Normal user
low	Peak	Normal	Less probable intruder
low	Peak	Bad	Medium probable intruder
medium	Ordinary	Good	Normal user
medium	Ordinary	Normal	Normal user
medium	Ordinary	Bad	Less probable intruder
medium	Night time	Good	Normal user
medium	Night time	Normal	Less probable intruder
medium	Night time	Bad	Medium probable intruder
medium	Peak	Good	Less probable intruder
medium	Peak	Normal	Medium probable intruder
medium	Peak	Bad	High probable intruder
high	Ordinary	Good	Normal user
high	Ordinary	Normal	Less probable intruder
high	Ordinary	Bad	Medium probable intruder
high	Night time	Good	Less probable intruder
high	Night time	Normal	Medium probable intruder
high	Night time	Bad	Definite intruder
high	Peak	Good	Medium probable intruder
high	Peak	Normal	High probable intruder
high	Peak	Bad	Definite intruder

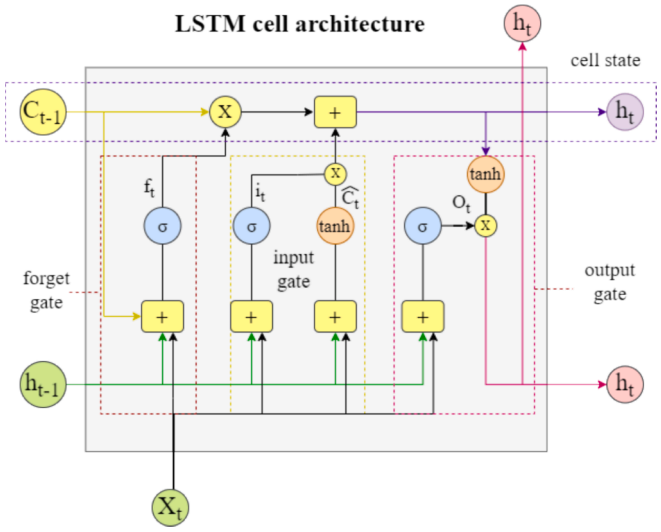

Fig. 3. LSTM with internal structure.

$$y^{j,t+1} = \begin{cases} y^{j,t} + R_j \times fl^{j,t} \times \left(N^{z,t} - y^{j,t}\right), R_z \geq qAP^{j,t} \\ Choosearandposition, otherwise \end{cases} \quad (2)$$

where R_z represents an arbitrary number between 0 and 1, and $AP^{j,t}$ is the knowledge probability of the crow z when iterating t. AP is in charge of balancing exploration and exploitation. Small AP values search in small regions (exploitation), whereas large values search globally (exploration) in the search space.

The limitations D, tMax, M, AP, and fl are initially established by CSA. In the search space, the y-position of each crow is chosen randomly. Crows are inexperienced at hiding food at first. As a result, they conceal the food in the starting position N. Every crow is examined using a specified fitness function throughout algorithm execution. The crow then adjusts its positioning using Equation (2) based on the fitness value. Examine each new position's viability. Crows refresh their memories by,

$$N^{j,t+1} = \begin{cases} y^{j,t+1}, F_n\left(y^{j,t+1}\right) isbetterthanFn\left(N^{j,t}\right) \\ Choosearandposition, otherwise \end{cases} \quad (3)$$

The objective function is defined as F n(). The optimum position is presented as the ideal solution as long as the dismissal requirements are met [31]. Algorithm 1 defines the CSA pseudo code.

Algorithm 1 Crow search Algorithm

1: Configure the parameters of M, AP, fl, and t_{Max}
2: Set the crow location y at random
3: Determine each crow's fitness function Fn(y)
4: Set the search crow N's memory N
5: Initialize the counter with t = 1
6: Repeat
7: for (j = 1: j ≤ M) do
8: At random, select one of the crows to follow z
9: if $R_Z \geq AP^{z,t}$ then
10: $y^{j,t+1} = y^{j,t} + R_j \times fl^{j,t} \times \left(N^{z,t} - y^{j,t}\right)$
11: else
12: $y^{j,t+1} = Arandomlocationofthesearchspace$
13: end if
14: end for
15: Determine the feasibility of $y^{j,t+1}$
16: Assess the crow's new position $Fn(y^{j,t+1})$
17: Improve the crow's memory $N^{j,t+1}$
18: Increase the iteration counter by setting t = t + 1
19: Until $t < t_{Max}$ { Termination criteria satisfied }
20: Build the optimum solution N

3.4. Detection module

The prevention and detection subsystem feeds the test dataset, which it then delivers to the feature selection module to identify the optimal feature collection. A classifier transfers test data from the optimal set of attributes in the training dataset to feature vectors created by the classifier during training. We then use a classifier to separate the data into DoS and benign requests. The classification toolbox includes fuzzing techniques with deep LSTMs for classifying standard and DoS samples.

3.4.1. Fuzzy rules

Table 3 illustrates the process of creating fuzzy rules. The first rule is written if For instance, we can format the first rule like this: If the cost is high, the time is during peak hours, and the behavior is poor, the likelihood of intrusion is apparent. In the fuzzy decision tree method, the fuzzy inference system terminates the fuzzy rules and judges the type of incursion or expected behavior. As a result, the algorithm employs the triangle membership function depicted in equation (4) to make efficient decisions even in the face of ambiguity. [32].

$$\mu_A(X) = \begin{cases} 0, & x \leq a \\ \dfrac{x-a}{m-1}, & a < x \leq m \\ \dfrac{b-x}{b-m}, & m < x < b \\ 0, & x \geq b \end{cases} \quad (4)$$

3.4.2. Deep long short-term memory

Hochreiter and Schmidhuber proposed long short-term memory (an RNN evolution) to address the drawbacks of RNNs by adding extra interactions to each module (or unit). LSTMs are a sort of RNN that can learn long-term dependencies and remember information for an extended period by default. The next cell receives both the hidden state and the cell state. The master string for data flow is the cell state, which allows data to flow with slight alteration. Some linear modifications may occur, however. Sigmoid gates allow the addition or withdrawal of information from the cell state. Gates are analogous to a layer or sequence of matrix operations in that they contain several distinct weights [33].

The basic procedure is the same as for simple RNNs, although LSTMs use different cell states and gates. The sequence's processing uses data

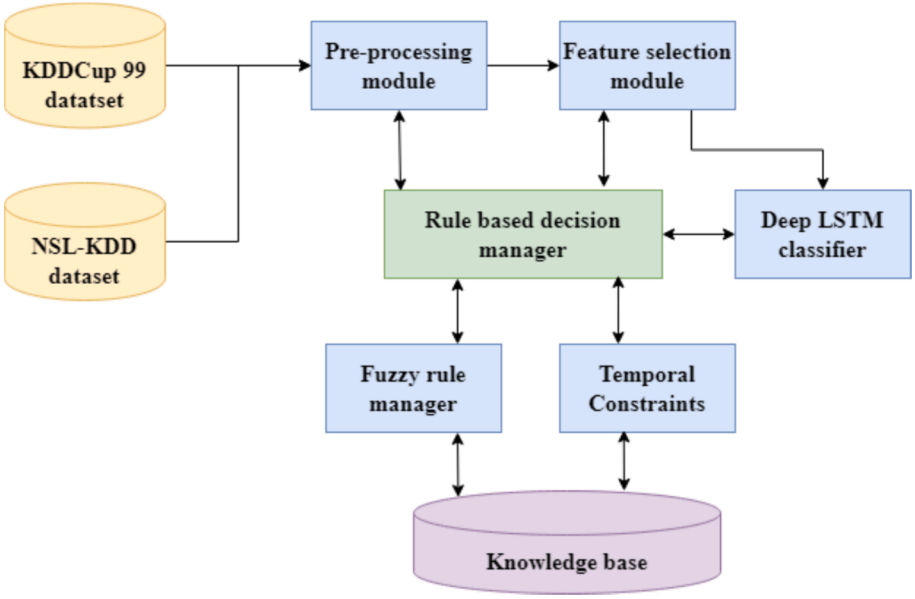

Fig. 4. Proposed system architecture.

from the cell state. Consequently, we can transfer data from earlier time steps to subsequent time steps, thereby reducing LSTM effects. We use gates to add or remove information from the cell state. Such gates are NNs that decipher underlying relevant patterns and decide which data to learn versus what to forget. The following gates determine the next concealed state of the network: input gate, forget gate, and output gate. If the input is multiplied by 0, the outcome of the process is "forgotten," however, if the input is multiplied by 1, then the outcome can maintain the same information, which is "retained." Assume the present time step is t, the input value for this time step is x(t), as illustrated in Fig. 3, the outcome is the hidden state expressed as h(t), the present state is represented by c(t), and the prior hidden state is represented by h(t-1).

The present state c(t) is formed by multiplying the forgotten state f(t) by the prior memory state c(t-1). If it is 0, it is deleted; otherwise, it is retained. The output gate decides the subsequent hidden state since it has the estimated value. Various activation functions can be applied to a unit. The sigmoid function is utilized on x(t) and h(t-1), while tanh is used in the new cell state to identify the hidden state, a combination of the sigmoid function and tanh outcomes. The ultimate final hidden state for the present time step is transmitted to the next time step.

LSTM is a recurrent neural network (RNN) framework that tackles the leaky gradient problem and captures long-term dependencies on sequential input. In contrast, "deep LSTM" refers to using numerous layers of LSTM cells stacked on each other to construct deeper structures. Deep LSTM refers to the method of stacking numerous layers of LSTM cells together to form deeper neural network architectures. A deep LSTM network's layers incorporate LSTM cells that process input data progressively, resulting in one layer serving as input to the next. Deep LSTMs use hierarchical representations learned across multiple layers to capture more complicated and abstract aspects of data. We refer to a single layer of LSTM drives as LSTM, but we stack different layers of LSTM drives to construct deeper structures. Deep LST M's purpose is to capture more complex features and long-term dependencies in sequence data by leveraging the expressive potential of numerous layers.

3.5. Proposed intrusion detection model of WSN

The suggested research encompasses the relevant elements, namely selecting and classifying certain traits. Fig. 4 depicts the system architecture. The system architecture comprises a preprocessing module, a feature selection module, a classification, a deep LSTM classifier, and a rule-based decision manager that utilizes a knowledge base. The suggested system gathers data using the network trace and KDD cup datasets. The preprocessing module receives the acquired data and preprocesses it. The feature selection module receives this dataset and extracts the relevant features. Fuzzy deep temporal LSTM classifiers categorize large datasets based on derived characteristics. A smart deep LSTM classifier classifies data based on fuzzy temporal restrictions. Lastly, the intrusion prevention phase receives the categorized data to prevent infiltration. The suggested system's most essential subsystem is the rule-based decision manager. To coordinate the intrusion detection process, the decision manager, based on the rules, manages and communicates with all system components.

The temporal constraint manager and the fuzzy rule manager are two sub-modules of the decision manager. The fuzzy manager creates rules for inference via the fuzzification process and implements them, utilizing the decision manager's forward-chaining rules. Following the routing module, the defuzzification module transforms fuzzy judgments into real-time decisions, thereby generating compelling routing outcomes. The time information management module maintains the time constraint manager for routing decisions. Both domain rules and empirical rules for practical routing inference based on network data make up the information database. After the authentication procedure, the BS gathers all the information the intrusion detection module provides. Lastly, intrusion prevention systems use fuzzy, ad hoc rules to identify and block attackers. The decision manager triggers it whenever it transmits information about a potential incursion or performs rule processing.

By enhancing the LSTM technique with temporal and fuzzy criteria, this work creates a deep LSTM method for rapid system training and testing. The deep LSTM method is most commonly employed for classification in deep learning applications. Time and space limits are not required for existing methods. However, the suggested deep LSTM method needs fuzzy rules and temporal limitations to produce excellent decisions. The algorithm is designed to execute binary classification and create each tree dimension. Furthermore, the error rate and cost function approaches used in this research's deep LSTM algorithm employ fuzzy temporal rules to determine the decision value.

For intrusion detection, WSNs produce continuous data streams, necessitating the capture of temporal patterns and trends in network behavior. Deep LSTM networks excel in this role because they effectively model and learn dependencies within sequential data. These LSTM-

based models look at the temporal relationships between sensor readings and network events to find strange or out-of-the-ordinary behaviour that could mean there has been a security breach. The incorporation of temporal constraints enables the IDS to differentiate between normal variations in network traffic and persistent suspicious activities, thereby enhancing the accuracy and dependability of intrusion detection.

The fuzzy temporal deep LSTM classifier categorizes a large dataset based on extracted features. It employs fuzzy temporal constraints to categorize the data. The intrusion prevention system then receives the classified data to prevent intrusion occurrences. The decision manager comprises two sub-modules: the fuzzy rule manager and the temporal constraint manager. The fuzzy rule manager formulates inference rules through a fuzzification process and implements them using forward chaining rules within the inference system. The defuzzification module then translates fuzzy decisions into real-world decisions, which inform efficient routing decisions subsequently executed by the routing module. The temporal information management module oversees the temporal constraints used to inform routing decisions. It maintains a knowledge base containing domain-specific and general rules essential for effective inference in directing collected data through the network. Once authenticated, the base station consolidates data from the intrusion detection module. As a result, the intrusion prevention module employs fuzzy temporal rules to identify and block attackers, classifying them as intruders. This system activates upon communication from the decision manager regarding potential intrusions, and it remains active throughout rule processing.

Fuzzy logic enhances deep LSTM networks by offering a systematic framework to interpret and integrate uncertain or imprecise information derived from sensor data. Fuzzy rules define linguistic variables and membership functions that show the levels of truth for different network conditions, like changes in traffic volume or packet sizes. This approach allows the IDS to reason and make decisions based on fuzzy sets rather than precise values, which is essential for accommodating the inherent uncertainties and variability present in WSN data. By combining fuzzy rules with deep LSTM-based models, the IDS can adeptly manage intricate patterns and contextual nuances in network traffic, thereby improving detection accuracy and minimizing false alarms.

The classification algorithm receives fundamental features identified by the proposed crow search technique. A deep LSTM is built every time the classification algorithm receives a feature. This work chooses the ideal deep LSTM algorithm based on the application by employing application-specific fuzzy temporal criteria. Furthermore, we examined the deep LSTM findings using both the data set and the optimal number of functions. In this study, we conducted tests using NSL-KDD and KDDCup 99. The research findings are explained in the following section.

Algorithm 2: DoS attack detection algorithm

Input: KDDCup 99 and NSL-KDD dataset
Output: Classification result: accuracy, recall, precision and F1 score
Begin: Data preprocessing;
$Z'' = Data\_Cleaning(Z')$;
$Z' = Normalization(Z)$;
$X', Y', Z' =$ Transformation (X, Y, Z);
end
Begin: Feature selection
Dynamic feature selection using Crow search algorithm
end
Begin: Classification
Train the fuzzy with deep LSTM classifier;
Testing dataset KDDCup 99 and NSL-KDD are input into the trained deep LSTM classifier to detect attacks;
End
Return the classification result

4. Results and discussion

The research utilizes a standard network dataset known as KDDCup99 and the NSL-KDD dataset to evaluate the suggested approach through various trials. Normal, Probe, DoS, R2L, and U2R are the training and testing examples. We implemented the suggested algorithm using MATLAB. We evaluate the suggested method against conventional techniques like intelligent decision trees [32], Deep Neural Network (DNN) [34], Recurrent Neural Network (RNN) [35], Convolutional Neural networks (CNN) [36], and Long short term memory (LSTM) [37]. Table 4 displays the simulation parameters.

Table 4
Simulation parameters.

parameter	Value
Area	200*200 m
Number of SNs	50–500
Energy of nodes	2 J
Initial energy	0.5 J
E_{elec}	50nJ/bit
Packet size	1024 bits

Table 5
Parameter values of CSO used for feature selection.

Parameter	Value
Population size	50
Maximum iteration	500
Awareness probability	0.1
Flight length	2

Table 6
Hyper-parameters of the deep LSTM.

Parameter	Value
Fully connected layer	5
Number of hidden units	100
State activation function	tanh
Output mode	last
Optimization algorithm	CSO
Gate activation function	hard-sigmoid
Minimum batch size	32
Maximum number of epochs	200
Gradient threshold	1
Initial learning rate	0.01

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

$$\text{Recall} = \text{TPR} = \frac{TP}{TP + FN} \tag{6}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{7}$$

$$\text{F} - \text{score} = 2X\frac{precisionXrecall}{Precision + recall} \tag{8}$$

$$\text{Attack predictive value (APV)} = \frac{TP}{TP + FP} \tag{9}$$

$$\text{Normal predictive value (APV)} = \frac{TN}{FP + TN} \tag{10}$$

The fitness function evaluates both the number of features included in the model and its performance, with the goal of lowering complexity and computational costs. Table 5 shows the parameter values for the CSO algorithm used in this study. It is important to note that the feature selection process for the models created using the proposed approach adheres to the parameter values listed in Table 5. Furthermore, Table 6 contains the specific parameter values for the proposed classifier algorithm.

Table 7
Performance evaluation of different methods (normal).

Algorithm	Precision (%)	Recall (%)	F-score (%)	FPR (%)
Intelligent DT	96.98	90.37	91.41	4.34
DNN	97.31	92.72	93.39	4.17
RNN	97.52	94.63	94.79	3.11
CNN	97.85	95.85	95.22	2.17
LSTM	98.42	96.92	98.93	2.05
Fuzzy with deep LSTM	99.61	98.58	99.15	1.8

Table 7 analyzes the effectiveness of the different methods in identifying normal data. Deep LSTM and other existing classifiers, such as intelligent decision trees, DNN, RNN, CNN, and LSTM, are used in normal attack analysis on the proposed fuzzy. We conducted five tests to examine the DoS attack, as shown in Fig. 5. The proposed approach outperforms existing classifiers in all five experiments because it uses intelligent fuzzy rules to detect DoS attacks.

Table 8 analyzes the effectiveness of the different methods in identifying DoS attacks. We conducted a DoS attack analysis on the proposed fuzzy system using deep LSTM and existing classifiers, such as intelligent decision trees, DNN, RNN, CNN, and LSTM. Five tests were conducted to examine the DoS attack depicted in Fig. 6. The proposed approach outperforms existing classifiers in all five experiments because it uses intelligent fuzzy rules to detect DoS attacks.

Table 9 shows the probe attack analysis we conducted on the suggested LSTM and conventional classifiers, including intelligent decision trees, DNN, RNN, CNN, and LSTM. We sent 1000, 2000, 3000, 4000, and 5000 data packets for attack analysis in this experiment, and Fig. 7 displays the results. It is clear from five distinct experiments that the suggested classifier outperforms conventional classifiers in detecting probing attacks. The use of intelligent fuzzy criteria enables the detection of probing attacks.

Table 10 shows the results of R2L attack analysis for the suggested deep LSTM and current classifiers, including intelligent decision trees, DNN, RNN, CNN, and LSTM. We sent 1000, 2000, 3000, 4000, and 5000

Table 8
Performance evaluation of different methods (DoS).

Algorithm	Precision (%)	Recall (%)	F-score (%)	FPR (%)
Intelligent DT	97.62	96.24	97.15	3.81
DNN	98.79	96.82	97.48	3.73
RNN	98.91	97.41	97.72	2.67
CNN	98.95	97.89	97.79	2.61
LSTM	99.1	98.31	98.31	2.14
Fuzzy with deep LSTM	99.78	99.42	99.49	2.05

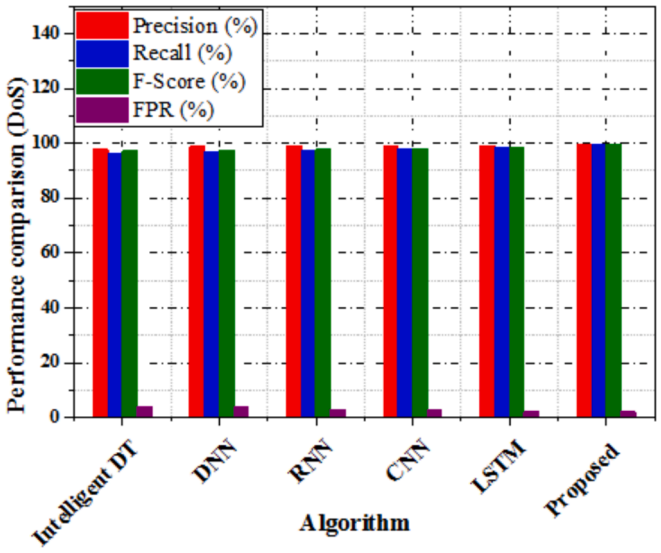

Fig. 6. DoS attack analysis.

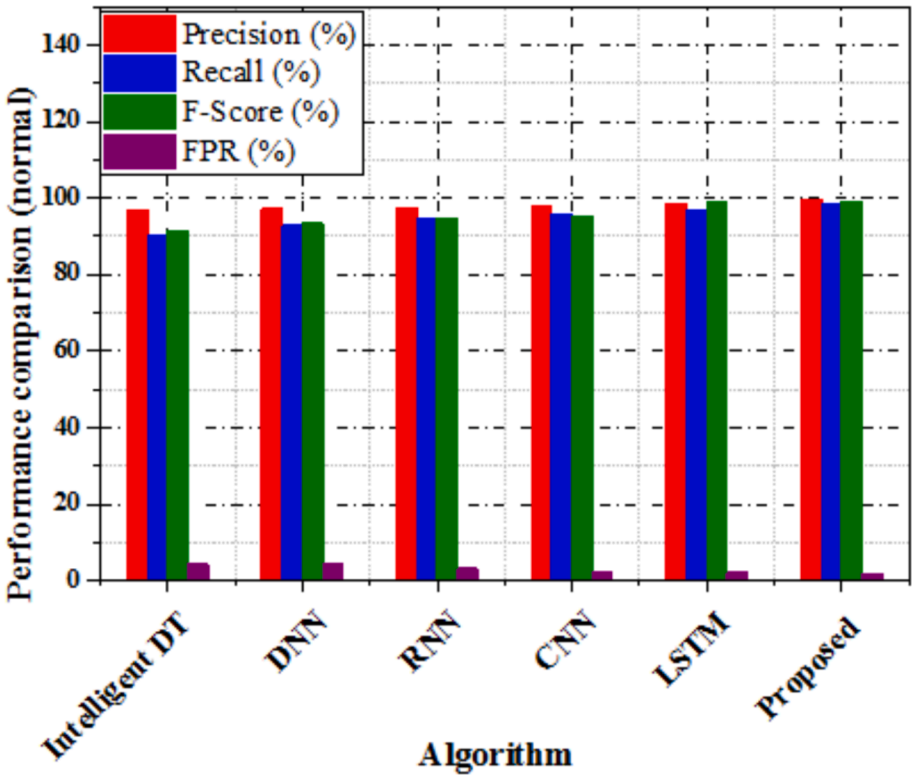

Fig. 5. Performance comparison of various models (Normal).

Table 9
Performance evaluation of different methods (PRB).

Algorithm	Precision (%)	Recall (%)	F-score (%)	FPR (%)
Intelligent DT	62.22	76.15	72.90	3
DNN	73.96	79.9	76.89	2.6
RNN	84.6	80.35	82.2	2.0
CNN	88.9	84.62	84.62	1.8
LSTM	90.4	87.4	88.9	1.6
Fuzzy with deep LSTM	93.9	90.8	89.15	1.5

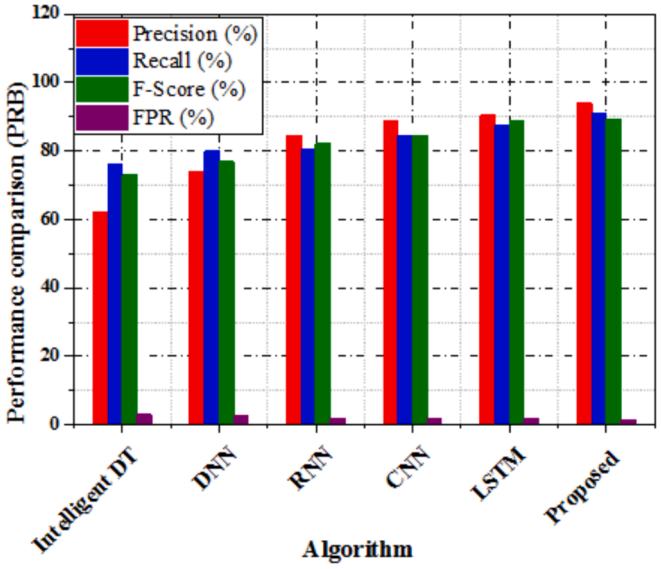

Fig. 7. PRB attack analysis.

Table 10
Performance evaluation of different methods (R2L).

Algorithm	Precision (%)	Recall (%)	F-score (%)	FPR (%)
Intelligent DT	66.56	20.65	20.94	4.5
DNN	69.37	26.79	26.50	3.0
RNN	71.68	30.73	28.59	2.6
CNN	73.96	34.61	30.96	2.0
LSTM	80.2	35.4	32.72	0.82
Fuzzy with deep LSTM	82.3	40.8	40.43	0.75

data packets for attack analysis in this experiment, and Fig. 8 displays the results. Five distinct experiments demonstrate that the suggested classifier outperforms conventional classifiers in detecting probing attacks. The use of intelligent fuzzy criteria enables the detection of R2L attacks.

Fig. 9 displays the U2R attack analysis results for the suggested deep LSTM and conventional classifiers such as intelligent decision tree DNN, RNN, CNN, and LSTM. Table 11 demonstrates that in five distinct experiments, the proposed classifier outperforms the traditional classifiers in detecting probing assaults. The use of intelligent fuzzy criteria aids in the detection of probing attacks.

Table 12 shows how the suggested method stacks up against other methods (like intelligent DT, DNN, RNN, CNN, and LSTM) on Dataset 1 in terms of recall, accuracy, and precision. The suggested deep LSTM fuzzy classifier outperforms conventional classifiers, as seen in Fig. 10. The other methods, on the other hand, all have signs that are higher than 90 %, which means that these methods can find things on the KDDCup 99 dataset. This means that the model used in this study has a bigger effect. The suggested approach has the highest accuracy rate of 99.58 %, the highest precision rate of 98.42 %, the highest recall rate of 98.45 %, the highest f-score of 98.36 %, the highest APV of 98.46 %, and the highest NPV of 99.8 %. Figs. 10 and 11 compare the proposed method's efficiency to other methods on the KDDCup 99 dataset.

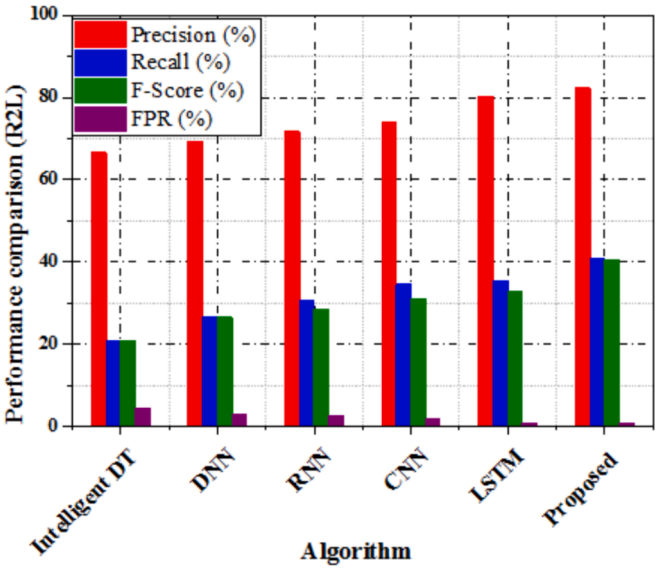

Fig. 8. R2L attack analysis.

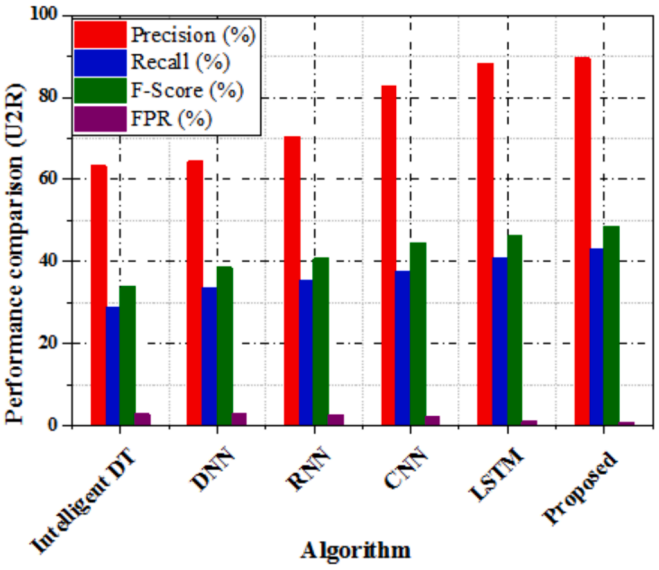

Fig. 9. U2R attack analysis.

Table 11
Performance evaluation of different methods (U2R).

Algorithm	Precision (%)	Recall (%)	F-score (%)	FPR (%)
Intelligent DT	63.33	28.81	33.76	2.79
DNN	64.44	33.58	38.53	2.80
RNN	70.5	35.28	40.78	2.63
CNN	82.8	37.7	44.61	2.11
LSTM	88.3	40.8	46.38	1.10
Fuzzy with deep LSTM	89.6	42.9	48.63	0.8

Table 13 shows how the suggested method measures up against other methods (such as intelligent DT, DNN, RNN, CNN, and LSTM) on Dataset 2 in terms of accuracy, precision, and recall. Figs. 12 and 13 show that the proposed deep LSTM fuzzy classifier outperforms existing classifiers. However, the indications of the existing approaches are greater than 90 %, showing that these methods effectively detect the NSL-KDD dataset.

Table 12
Various methods on the KDDCup 99 (dataset-1).

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	APV (%)	NPV (%)
Intelligent DT	91.77	92.87	91.26	92.32	94.39	97.4
DNN	91.24	92.93	91.77	93.26	94.90	97.3
RNN	93.84	94.93	93.94	94.93	95.93	98.5
CNN	94.40	94.99	94.40	94.99	96.98	98.7
LSTM	95.57	96.37	95.28	95.37	97.38	99.9
Proposed	99.58	98.42	98.45	98.36	98.46	98.8

The findings demonstrate that the approach described in this study produces highly effective detection outcomes. Compared to traditional techniques, neural networks exhibit quicker training and convergence. The proposed approach achieves notable metrics with 99.76 % accuracy, 99.90 % precision, 99.91 % recall, 99.63 % f-score, 99.57 % average precision value (APV), and 99.9 % negative predictive value (NPV). Figs. 12 and 13 depict performance comparisons with alternative methods on the KDDCup 99 dataset.

Fig. 14 and Table 14 depict the system's PDR analysis with and without the suggested IDS. This work ran five trials with different record sets, including 1000, 2000, 3000, 4000, and 5000. The intelligent fuzzy temporal rules in the suggested IDS lead to an improvement in network performance compared to a network without IDS.

Table 15 depicts the energy consumption between the suggested deep LSTM fuzzy classifier and existing classifiers (intelligent DT, DNN, RNN, CNN, and LSTM). The proposed technique consumes less energy than other classification algorithms, as shown in Fig. 15. Fuzzy temporal rules enhance the classifier's decision-making ability. Table 16 and Fig. 16 depict the network's latency analysis, with and without the suggested IDS. This work ran five distinct trials using various record sets. The proposed IDS enhances the network's performance compared to a network without the use of intelligent fuzzy time rules contributes to this improvement used.

Table 17 compares the proposed deep LSTM fuzzy classifier with existing classifiers. The suggested classifier outperforms existing classifiers such as intelligent DT, DNN, RNN, CNN, and LSTM, as shown in Fig. 17. The introduction of intelligent fuzzy temporal rules for decision-making enhances this work. The proposed approach has 95 % detection accuracy for PRB attack detection, 97 % for Dos, 45 % for R2L, and 48 % for U2R respectively.

This section compares the proposed approach's computational and time complexities to existing methods such as Intelligent DT, DNN, RNN, CNN, and LSTM.

(i) Computational complexity

where n is the number of nodes in the decision tree, L is the number of layers, N is the number of neurons per layer, T is the sequence length, K is the kernel size, M is the number of feature maps, and D represents the input data's dimensionality. Table 18 depicts the computational complexity analysis of the proposed and existing algorithms. We optimize LSTM networks to effectively capture long-term dependencies in sequential data. Unlike DNNs and CNNs, which typically necessitate multiple layers and intricate parameter adjustments, LSTM networks generally feature fewer parameters per unit, thereby reducing computational burdens. In WSNs, where data frequently displays temporal dependencies, LSTM's ability to preserve and transmit information across extended sequences—without encountering the vanishing gradient issue due to its gated structure—enables more efficient processing of sequential data.

This contrasts with traditional RNNs or CNNs, which might require deeper architectures to achieve similar performance levels in handling temporal data dependencies. The fuzzy logic in the deep LSTM model helps it make better decisions by using linguistic variables and rules. This could mean that it doesn't needs as many complex layers or as much data preprocessing as other models might. It takes less work to run the proposed deep LSTM model for finding DoS attacks in WSNs because it handles temporal dependencies better, handles parameters more efficiently, uses fuzzy logic to make decisions, and doesn't need as much

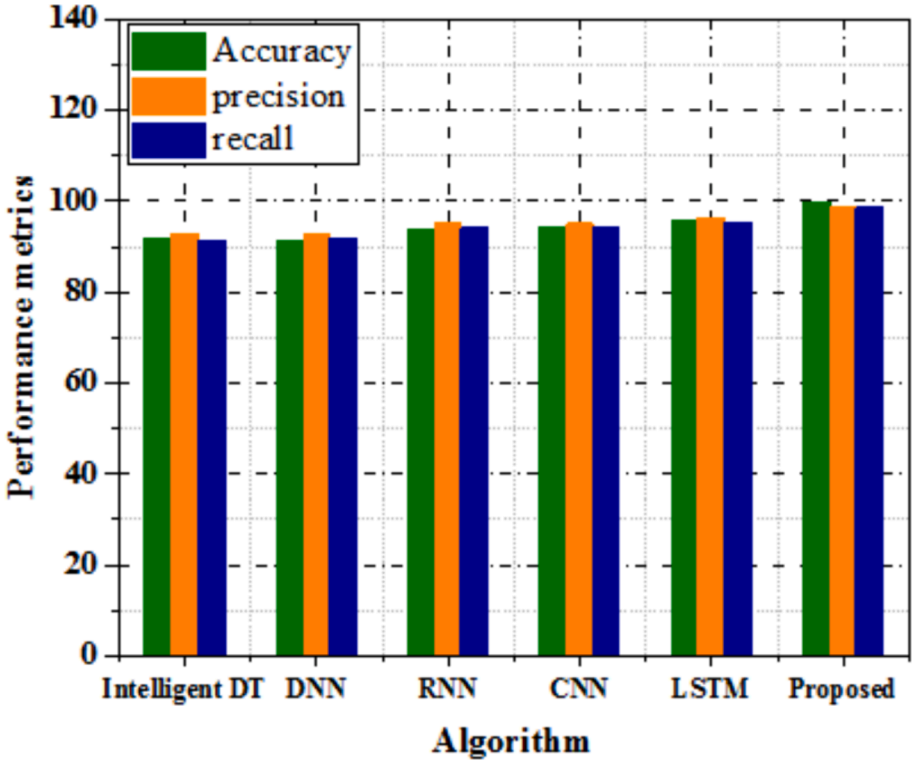

Fig. 10. Graphical representations for accuracy, precision, and recall for dataset-1.

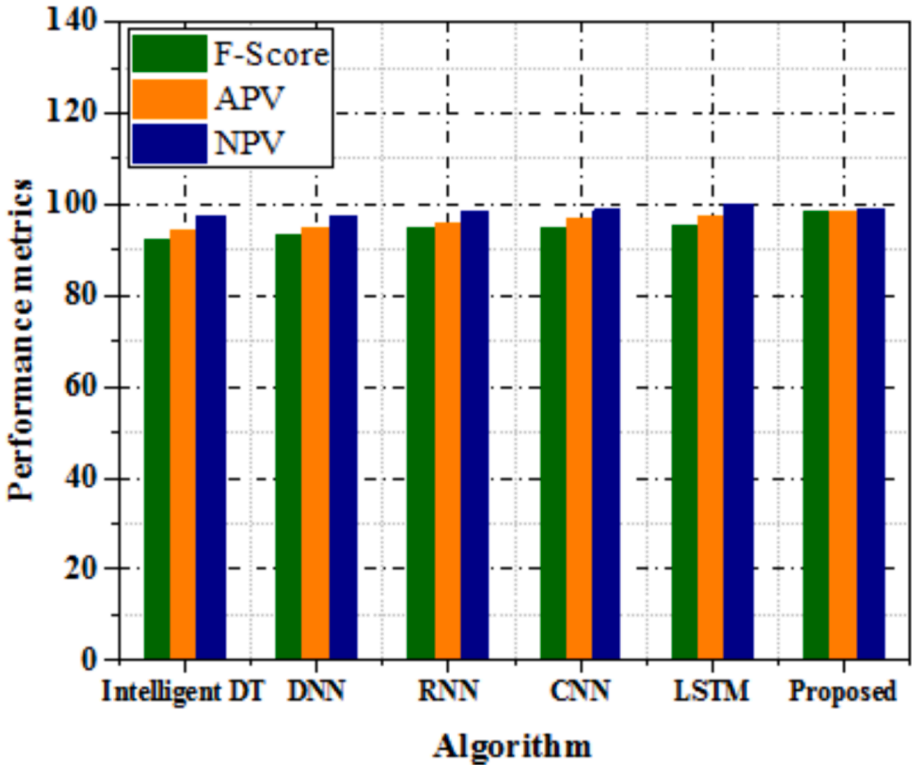

Fig. 11. Graphical representations for f-score, APV, and NPV for dataset-1.

Table 13
Various methods on the NSL-KDD (dataset-2).

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	APV (%)	NPV (%)
Intelligent DT	93.8	95.33	88.50	90.75	95.40	98.5
DNN	95.87	94.24	93.95	94.84	97.91	98.6
RNN	97.30	96.65	96.75	96.99	98.94	99.5
CNN	97.98	97.39	97.98	97.89	99.98	99.7
LSTM	98.53	98.67	98.23	97.88	99.49	99.8
Proposed	99.76	99.90	99.91	99.63	99.57	99.9

feature engineering. These factors collectively position it as a promising approach suitable for resource-limited environments such as WSNs.

(ii) Time complexity

Table 19 displays the temporal complexity analysis of both the proposed deep LSTM fuzzy classifier and the current classifiers. The suggested method is faster than current intrusion detection methods. Fig. 17 shows that the suggested approach minimizes execution time (18 s) compared to existing methods.

The comparison of computational and structural complexities takes into account simulation time, which includes the total amount of time spent training and testing the model. Table 19 compares the time complexity of the proposed fuzzy classifier with deep LSTM to that of existing classifiers. Fig. 18 shows that the suggested approach has shorter processing times than the Intelligent DT, DNN, RNN, CNN, and LSTM methods. Fuzzy logic integration allows for more streamlined decision-making processes by incorporating human-like reasoning and linguistic variables. In comparison to DNNs or CNNs, this integration may reduce the complexity of feature engineering and data preprocessing. Furthermore, LSTM networks capture long-term dependencies in sequential data with fewer parameters than traditional RNNs, resulting in more efficient temporal modeling and faster inference times. The hybrid approach also uses fuzzy rules to guide LSTM-based predictions, making the best use of computing power by focusing on relevant data sets and reducing the number of computations that aren't needed. This makes the proposed model very good at quickly finding and stopping DoS attacks in WSNs, which is very important for keeping the networks safe and working well in environments that are always changing and have limited resources.

5. Discussion

It is crucial to compare the proposed model with other existing models in the literature to evaluate the results, especially considering the increasing popularity of Federated Learning (FL) for safeguarding data privacy across diverse applications. Among these advanced models, the proposed fuzzy model with deep LSTM architecture stands out. Unlike current approaches, the suggested method utilizes deep learning (DL) to achieve enhanced detection accuracy. Consequently, the proposed technology excels at identifying network intrusion scenarios while maintaining privacy through local data storage. Table 12 presents a comparison between the proposed technique and previous studies using the KDDCup 99 dataset. Despite advancements in existing algorithms for intrusion detection, the suggested method demonstrates a slight advantage in accuracy across the datasets analyzed. This improved performance highlights the effectiveness of the proposed fuzzy-deep LSTM model, particularly when paired with the robust privacy protections provided by the proposed architecture. Thus, the proposed study suggests a promising way to create intrusion detection systems that put both data privacy and detection performance first by combining FL and DL.

Table 13 provides more comprehensive comparative results and analysis of the NSL-KDD dataset. The proposed approach achieves notable metrics with 99.76 % accuracy, 99.90 % precision, 99.91 % recall, 99.63 % f-score, 99.57 % average precision value (APV), and 99.9 % negative predictive value (NPV). It can be seen in Table 17 that the hybrid fuzzy logic with deep LSTM algorithm is consistently better at

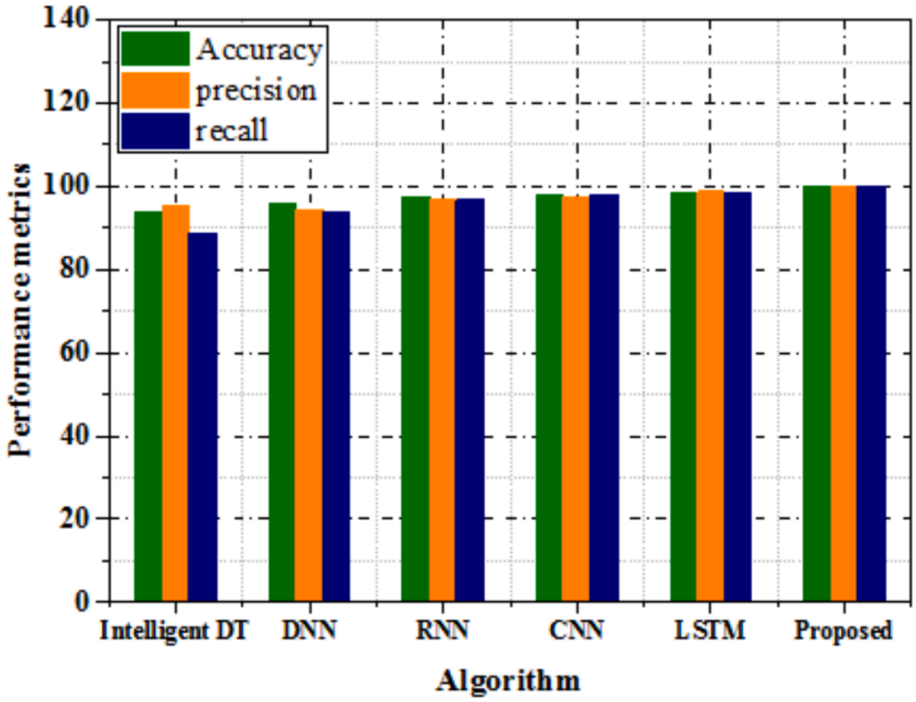

Fig. 12. Graphical representations for accuracy, precision, and recall for dataset-2.

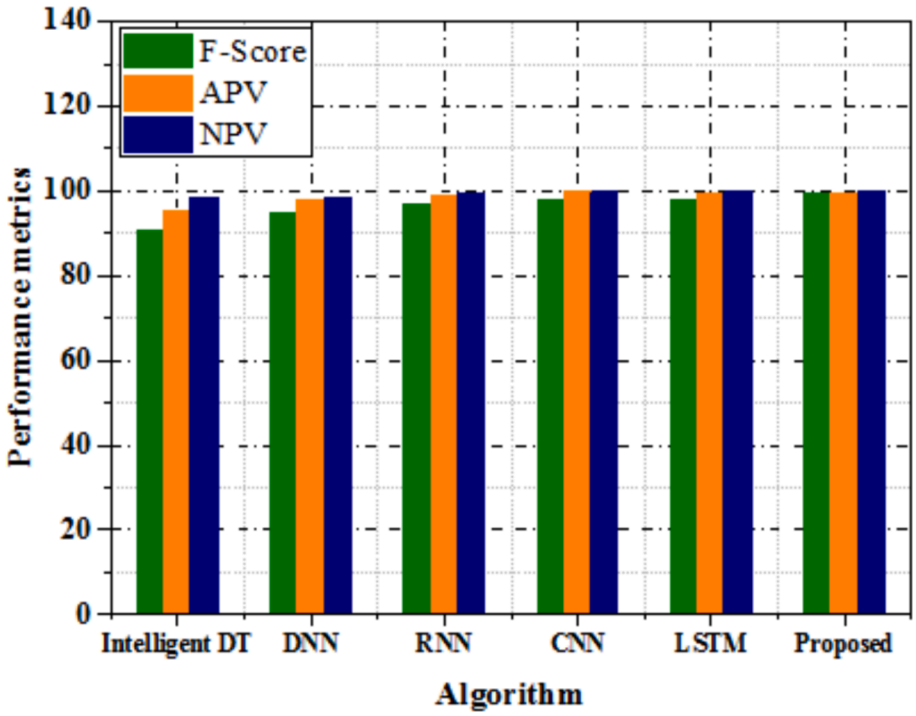

Fig. 13. Graphical representations for f-score, APV, and NPV for dataset-2.

detection than well-known methods such as intelligent DT, DNN, RNN, CNN, and LSTM-based approaches. The proposed approach has 95 % detection accuracy for PRB attack detection, 97 % for Dos, 45 % for R2L, and 48 % for U2R, respectively. The suggested approach minimizes execution time (18 s) compared to existing methods. Adding fuzzy logic to the model helps it deal with the errors and unknowns that come with WSN data, which lets us look at network behaviour in a more complex and context-aware way. Models relying solely on statistical or deterministic methods might overlook subtle anomalies or deviations that could signify security threats due to this capability. Additionally, LSTM networks are capable of capturing extended dependencies and temporal patterns within sequential data, such as network traffic logs and sensor readings. The hybrid model improves WSN's ability to find complex and changing intrusion patterns by combining fuzzy logic's ability to understand linguistic variables and fuzzy rules with LSTM's strong temporal modelling.

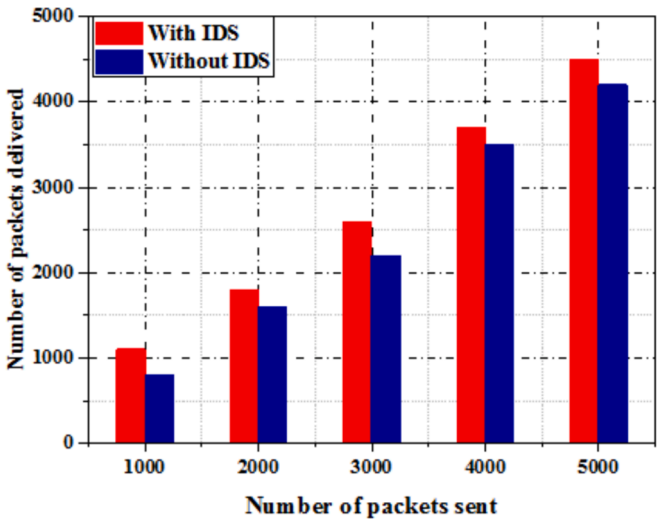

Fig. 14. Graphical representations of PDR.

Table 14
Packet delivery ratio analysis.

Number of packets sent	Packet delivery ratio	
	With IDS	Without IDS
1000	1100	800
2000	1800	1600
3000	2600	2200
4000	3700	3500
5000	4500	4200

Table 15
Energy consumption analysis.

Number of nodes	Intelligent DT	DNN	RNN	CNN	LSTM	Proposed
50	15	12	10	9	8	5
100	18	15	10	9.5	8	7
150	20	18	15	10	9	8
200	22	20	19	15	12	10
250	28	25	20	19	18	15

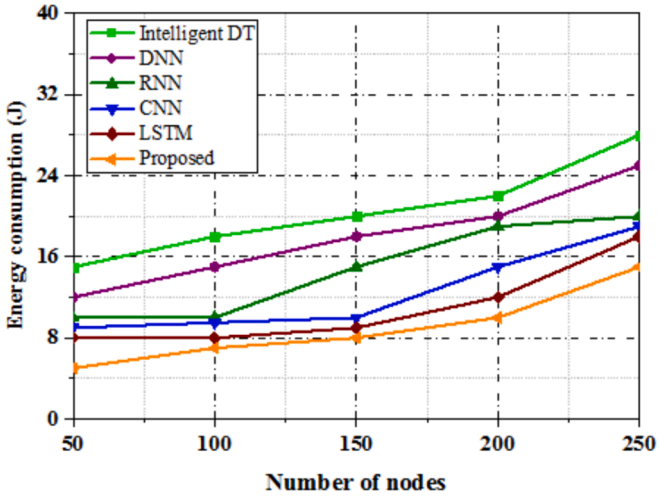

Fig. 15. Graphical representations of energy consumption.

Table 16
Delay analysis.

Experiments	With IDS	Without IDS
E1	12	16
E2	9	12
E3	8.5	13
E4	8	14.5
E5	10	13

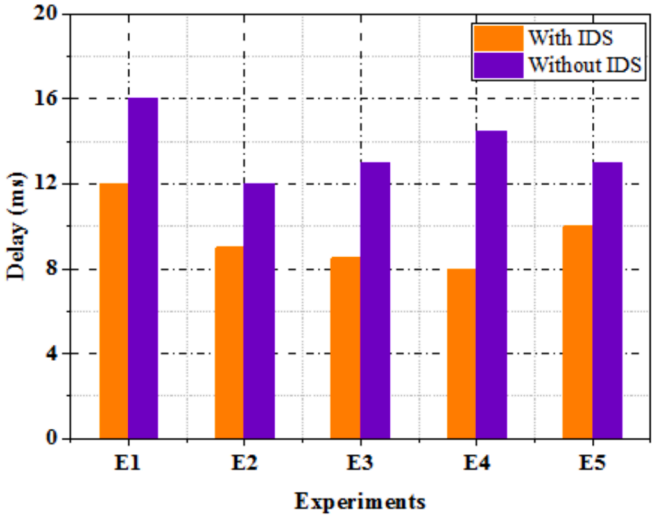

Fig. 16. Graphical representations of Delay.

Table 17
Detection accuracy analysis (%).

Algorithm	PRB	DoS	R2L	U2R
Intelligent DT	80	90	30	32
DNN	85	92	35	38
RNN	88	94	32	40
CNN	90	96	40	45
LSTM	93	97	45	48
proposed	95	99	50	55

6. Conclusion

Deep learning-based categorization approaches help increase DoS attack detection of network traffic, which is essential to network security. This work provides a deep learning-based DoS attack detection technique divided into the following modules: preprocessing, feature selection, and detection system. The CSA is proposed in this work as a novel feature selection technique that finds the optimal number of features for analysis and classification. Furthermore, an effective modification of the deep LSTM technique with fuzzy temporal constraints is proposed to classify network traffic and users more accurately. The suggested approach was evaluated utilizing the datasets KDDCup99 and NSL-KDD. The suggested FL-LSTM approach achieved the highest accuracy (99.58 %), the highest precision (98.42 %), the highest recall (98.45 %) and the highest f-score (98.36 %). Experiments demonstrate that the model improves intrusion detection accuracy, increases packet delivery rate, and improves network performance while decreasing false positive rate and network delay. Future work in this area will involve using intelligent agents to communicate in a distributed environment and testing the network in a real-time test bed to boost performance further.

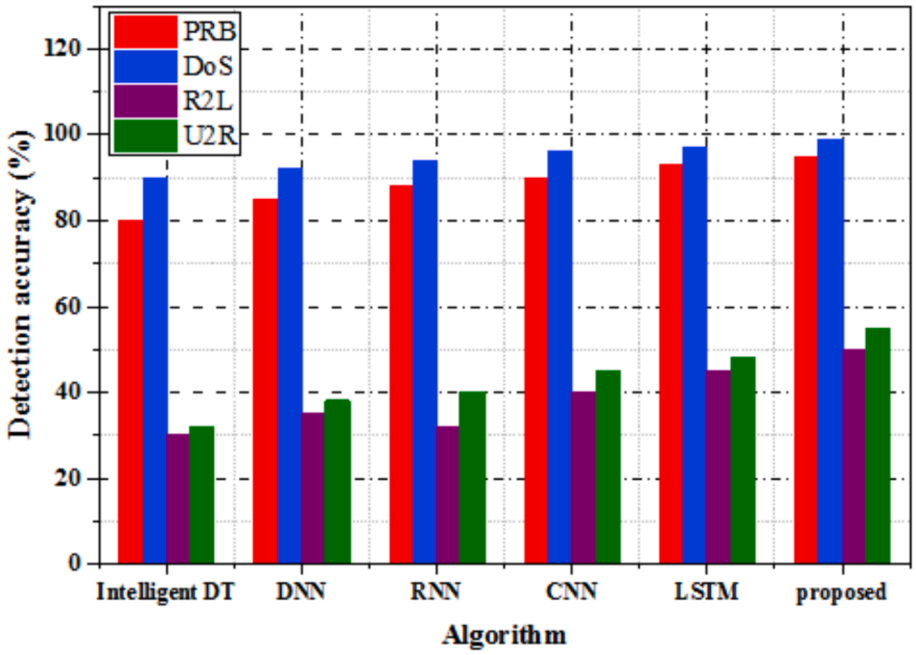

Fig. 17. Graphical representations of detection accuracy.

Table 18
Computational complexity.

Algorithm	Computational complexity
Intelligent DT	O(log n)
DNN	O(L·N·M)
RNN	O(T·L·N^2)
CNN	O(L·K·N^2·M)
LSTM	O(T·N^2)
Proposed	O(L·T·N^2·D)

Table 19
Time complexity analysis.

Algorithm	Simulation time (s)
Intelligent DT	250
DNN	200
RNN	100
CNN	70
LSTM	55
Proposed	18

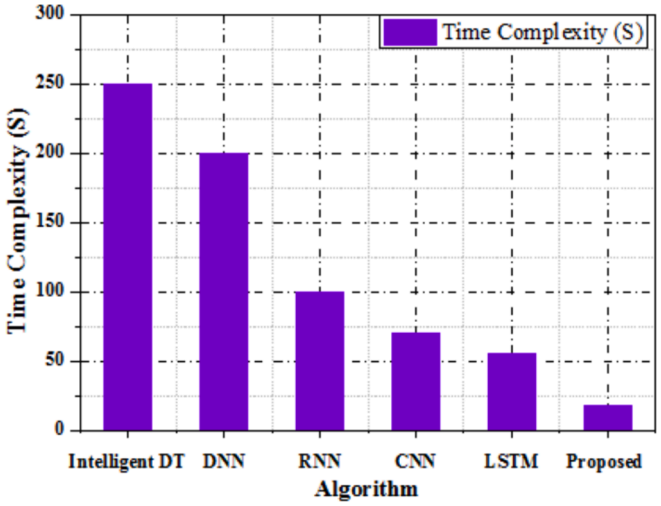

Fig. 18. Graphical representations of simulation time.

CRediT authorship contribution statement

P. Sathishkumar: Conceptualization, Writing – original draft. **A. Gnanabaskaran:** Supervision. **M. Saradha:** Investigation. **R. Gopinath:** Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Bhushan K, Gupta BB. Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment. J Ambient Intell Humaniz Comput 2019;10:1985–97. https://doi.org/10.1007/s12652-0 18-0800-9.
[2] Bhattacharyya Hoque DK, Kalita JK. Botnet in DDoS attacks: trends and challenges. IEEE Commun Surv Tutorials 2015;17(4):2242–70.
[3] Jayarajan P, Kanagachidambaresan GR, Sundararajan TVP, Sakthipandi K, Maheswar R, Karthikeyan A. An energy-aware buffer management (EABM) routing protocol for WSN. J Supercomput 2020;76(6). In this issue.
[4] Ahmed SA, Popov VL, Topalov AV, et al. Environmental monitoring using a robotized wireless sensor network. AI Soc 2018;33:207–14. https://doi.org/10.1007/s00146-018-0815-y.
[5] Singh K, Singh P, Kumar K. User behavior analytics-based classification of application layer HTTP-GET flood attacks. J Netw Comput Appl 2018;112:97–114.
[6] Salmi S, Oughdir L. Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network. J Big Data 2023;10:17. https://doi.org/10.1186/s40537-023-00692-w.
[7] Alsheikh MA, Lin S, Niyato D, Tan HP. Machine learning in wireless sensor networks: algorithms, strategies and applications. Commun Surveys Tutorials 2014;16:1996–2018. https://doi.org/10.1109/COMST.2014.2320099.

[8] Dwivedi, R.K., S. Pandey and R. Kumar, 2018. A study on machine learning approaches for outlier detection in wireless sensor network. Proceedings of the 8th International Conference on Cloud Computing, Data Science and Engineering, Jan. 11-12, IEEE Xplore Press, Noida, India, pp: 189-192.
[9] Ramesh S, Yaashuwanth C, Prathibanandhi K, et al. An optimized deep neural network based DoS attack detection in wireless video sensor network. J Ambient Intell Human Comput 2021. https://doi.org/10.1007/s12652-020-02763-9.
[10] Premkumar M, Sundararajan TVP. DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks. Microprocess Microsyst 2020;79:103278. https://doi.org/10.1016/j.micpro.2020.103278.
[11] Chandan RD, Balobaid A, Lakshmi N, Cherukupalli S, Flammini F, Natarajan R. Secure Modern Wireless Communication Network Based on Blockchain Technology. Electronics 2023;12:1095. https://doi.org/10.3390/electronics12051095.
[12] Al-Naeem M, Rahman M, Abubakar A, Rahman MM. AI-Based Techniques for DDoS Attack Detection in WSN: A Systematic Literature Review. J Comput Sci 2020;16:848–55. https://doi.org/10.3844/jcssp.2020.848.855.
[13] Rameshkumar S, Ganesan R, Merline A. Progressive transfer learning-based deep q network for ddos defence in wsn. Comput Syst Sci Eng 2023;44(3):2379–94.
[14] Li J, Liu M, Xue Z, Fan X, He X. RTVD: A real-time volumetric detection scheme for DDoS in the Internet of Things. IEEE Access 2020;8:36191–201.
[15] Liu, Z.; Thapa, N.; Shaver, A.; Roy, K.; Yuan, X.; Khorsandroo, S. Anomaly detection on iot network intrusion using machine learning. In Proceedings of the 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 6–7 August 2020; pp. 1–5.
[16] Bhunia, S.S.; Gurusamy, M. Dynamic attack detection and mitigation in IoT using SDN. In Proceedings of the 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, Australia, 22–24 November 2017; pp. 1–6.
[17] McDermott, C.D.; Majdani, F.; Petrovski, A.V. Botnet detection in the internet of things using deep learning approaches. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
[18] Sharma PK, Singh S, Park JH. OpCloudSec: Open cloud software defined wireless network security for the Internet of Things. Comput Commun 2018;122:1–8.
[19] Diro AA, Chilamkurti N. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener Comput Syst 2018;82:761–8.
[20] Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Shabtai A, Breitenbacher D, et al. N-baiot—Network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Comput 2018;17:12–22.
[21] Karasu S, Altan A. Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization. Energy 2022;242:122964.
[22] Özçelik YB, Altan A. Overcoming nonlinear dynamics in diabetic retinopathy classification: a robust AI-based model with chaotic swarm intelligence optimization and recurrent long short-term memory. Fractal Fract 2023;7:598. https://doi.org/10.3390/fractalfract7080598.
[23] Shaaban AR, Abd-Elwanis E, Hussein M. Ddos attack detection and classifcation via convolutional neural network (cnn). In: In: 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS). IEEE; 2019. p. 233–8.
[24] Sontowski S, Gupta M, Chukkapalli S, Abdelsalam M, Mittal S, et al. "Cyber attacks on smart farming infrastructure", in 2020 IEEE 6th Int. Conf on Collaboration andInternet Computing (CIC) 2020;1:135–43.
[25] Mohamed Amine F, Lei Shu D, Hamouda D, Kim-Kwang C. Deep learning-based intrusion detection for distributed denial of service attack. Journal of Agriculture 2021;11:221–35.
[26] Sahi A, Lai D, Li Y, Mohammed D. An Efficient DDoS TCP flood attack detection and prevention system in a cloud environment. IEEE Access 2017;3:56–74.
[27] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the kdd cup 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE. pp. 1–6.
[28] Choudhary S, Kesswani N. Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT. Procedia Comput Sci 2020;167:1561–73. https://doi.org/10.1016/j.procs.2020.03.367.
[29] Ullah S, Mahmood Z, Ali N, Ahmad T, Buriro A. Machine Learning-Based Dynamic Attribute Selection Technique for DDoS Attack Classification in IoT Networks. Computers 2023;12:115. https://doi.org/10.3390/computers12060115.
[30] Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 2016;169:1–12.
[31] Ismail Sayed, Gehad & Hassanien, Aboul Ella & Azar, Ahmad. (2019). Feature selection via a novel chaotic crow search algorithm. Neural Comput Appl. 31. 10.1007/s00521-017-2988-6.
[32] Nancy, Periasamy & Sannasy, Muthurajkumar & Ganapathy, Satish & Kumar Svn, Santhosh & Munuswamy, Selvi & Arputharaj, Kannan. (2020). Intelligent Intrusion Detection System Using Fuzzy and Deep Learning Approach for Wireless Sensor Networks. IET Commun. 14. 10.1049/iet-com.2019.0172.
[33] Adefemi Alimi KO, Ouahada K, Abu-Mahfouz AM, Rimer S, Alimi OA. Refined LSTM based intrusion detection for denial-of-service attack in internet of things. J Sens Actuator Netw 2022;11:32. https://doi.org/10.3390/jsan11030032.
[34] Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. IEEE Access 2019;7:41525–50.
[35] Wu P, Guo H, Buckland R. A transfer learning approach for network intrusion detection. In: In: 2019 IEEE 4th International Conference on Big Data Analytics (ICBDA). IEEE; 2019. p. 281–5.
[36] Kim J, Kim J, Kim H, Shim M, Choi E. Cnn-based network intrusion detection against denial-of-service attacks. Electronics 2020;9(6):916.
[37] H. -C. Chu and C. -Y. Yan, "DDoS Attack Detection with Packet Continuity Based on LSTM Model," 2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2021, pp. 44-47, doi: 10.1109/ECICE52819.2021.9645650.

P. Sathishkumar is a part-time Ph.D. student and working as Associate Professor in the Department of computer Science and Engineering in K.S Rangasamy College of Technology, Tiruchengode, Namakkal, Tamilnadu.He received B.E Computer Science and Engineering from Bharthidasan University, Tiruchirappalli and received M.E Computer Science and Engineering from Anna University, Chennai. His research interests include Security and Wireless Sensor Networking.

Dr.A. Gnanabaskaran received the M.C.A.(Master of Computer Applications) degree in the year 1996, M.E.(Computer Science and Engineering) degree in the year 2005 and Ph.D. degree in the year 2013. He is working as an Professor in the Department of Computer Science and Engineering, K.S.Rangasamy College of Technology (Autonomous), Tiruchengode. He has 24 years of teaching experience. He has published papers in 09 International Journals, 16 International Conference and 02 Books. His areas of interest are Data Mining, Big Data Analytics, Machine Learning and Cyber security.

M. Saradha is working as Assistant Professor in the Department of computer Science and Engineering in K.S Rangasamy College of Technology, Tiruchengode, Namakkal, Tamilnadu. She completed B.E Computer Science and Engineering(2013) and M.E Computer Science and Engineering(2015) at K.S.Rangasamy College of Technology, Tiruchengode, Namakkal. Her area of interests are Machine Learning and Data Analytics.

Dr.R. Gopinath received the B.E.(Computer Science and Engineering) degree in the year 2001, M. E.(Computer Science and Engineering) degree in the year 2005 and Ph.D. degree in the year 2017. He is working as an Associate Professor in the Department of Computer Science and Engineering, K.S.Rangasamy College of Technology (Autonomous), Tiruchengode. He has 19 years of teaching experience. He has published papers in 07 International Journals,10 International Conferences and 02 Books. His areas of interest are Machine Learning, Cryptography, Cloud Computing and Cyber security.