Journal of Big Data

RESEARCH Open Access

Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network

Salim Salmi* and Lahcen Oughdir

*Correspondence:
Salim.salmi@usmba.ac.ma

Engineering, Systems and Applications Laboratory, ENSA, Sidi Mohamed Ben Abdellah University, Fez, Morocco

Abstract

Wireless sensor networks (WSNs) are increasingly being used for data monitoring and collection purposes. Typically, they consist of a large number of sensor nodes that are used remotely to collect data about the activities and conditions of a particular area, for example, temperature, pressure, motion. Each sensor node is usually small, inexpensive, and relatively easy to deploy compared to other sensing methods. For this reason, WSNs are used in a wide range of applications and industries. However, WSNs are vulnerable to different kinds of security threats and attacks. This is primarily because they are very limited in resources like power, storage, bandwidth, and processing power that could have been used in developing their defense. To ensure their security, an effective Intrusion detection system (IDS) need to be in place to detect these attacks even under these constraints. Today, traditional IDS are less effective as these malicious attacks are becoming more intelligent, frequent, and complex. Denial of service (DOS) attack is one of the main types of attacks that threaten WSNs. For this reason, we review related works that focus on detecting DoS attacks in WSN. In addition, we developed and implemented several Deep learning (DL) based IDS. These systems were trained on a specialized dataset for WSNs called WSN-DS in detecting four types of DoS attacks that affects WSNs. They include the Blackhole, Grayhole, Flooding, and Scheduling attacks. Finally, we evaluated and compared the results and we discuss possible future works.

Keywords: Wireless sensor networks (WSNs), Intrusion detection system (IDS), Denial of service (DOS), Deep learning (DL)

Introduction

Wireless sensor networks (WSNs) have gained an increasing demand from industries and researchers for monitoring environmental and physical conditions in recent years. They are simple, effective, much easier to deploy, and relatively inexpensive compared to other sensory devices. This has made them suitable for a wide range of applications in different fields such as the medical, health care, telecommunications, military, and environmental fields.

WSN usually consist of a collection of sensor nodes that are distributed across the area of interest, to gather and monitor the environmental and physical conditions of that area. These sensor nodes occasionally communicate with other nodes within the network. The

collected data are then gathered using the programmed protocol and transmitted via a wireless connection to a more powerful node called the Sink Node or Base Station (BS) for storage and analysis. WSNs are often deployed in remote, hostile, and inaccessible environments to detect environmental disasters like floods, storms, forest fires, volcanoes, earthquakes, and less hostile conditions like monitoring a patient's vitals, developing smart homes, cities, transportation, traffic, and Internet of Things (IoT) [1, 2].

However, having a simple build comes with some disadvantages. WSNs are highly vulnerable to security threats and attacks [3]. Securing them is a major challenge because of their constrained resources such as battery power, memory, storage space, communication bandwidth, and processing capabilities. Furthermore, due to the unsupervised environments they are deployed in, sensor nodes are also exposed to physical attacks.

Denial of service (DoS) attack is one of the most frequent and common type of attack against WSNs. They come in different forms with the main objective of draining the node resources, especially power and its ability to carry out other tasks.

Hence, some defense mechanisms are needed to protect WSN from DoS attacks. Several research studies have proposed different Intrusion detection systems (IDSs) to help detect these security attacks. Machine learning (ML) and Deep Learning techniques have been use in several studies and have often show great accuracy.

In this paper, we employ DL techniques to detect and classify several DoS attacks. The goal of this research is to experiment with several DL-based algorithms to develop an efficient, lightweight, and accurate algorithm that can be used detecting DoS attacks in WSNs. The algorithms were trained and evaluated on a specialized WSN dataset called WSN-DS. The WSN-DS dataset contains four types of DoS attacks which are Blackhole, Grayhole, Scheduling, and Flooding attacks, and includes instance where there was not attack.

we also review previous works that has been done in developing ML-based IDS for WSNs. Afterwards, we present our proposed methods. We trained four DL algorithms on the WSN-DS dataset, which are Dense Neural Network (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and an algorithm that combines the CNN and RNN architectures. Of the 19 feature present in the dataset (including the attack type), 16 were used in training the algorithm. After training, the models are evaluated and compared with other algorithms using well-known comparison metrics such as accuracy.

The rest of paper is organized as follows. Section "Related works" provides a literature review of recent existing IDSs that employed machine learning techniques. Section "Methods" discusses the dataset and methods used. Section "Results" presents the experimental results obtained from the proposed methods. Finally, the conclusion and future works suggestions are presented in Section "Conclusion".

Related works

Denial of service (DoS) attack is one of the most common threats to WSN security as it is relatively easy to launch. Several studies have proposed different IDS for DoS attack detection over the years. In this section, we review some of this past works and focus, in particular, on recent Deep Learning and machine learning-based IDSs.

Kim et al. [4] proposed a model based on a Convolutional Neural Network (CNN) for detecting DoS attacks using the KDD-99 [5] dataset and the CICIDS2018 [6] dataset. In their experiment, they converted the input features into an "image", the proposed CNN model then receive a gray-scale or RGB image as input. Different number of layers was explored as they performed both binary (normal vs. attack) and multiclass classification. The proposed models were then evaluated against a Recurrent Neural Network (RNN) model with the proposed models performing better in both binary and multiclass classification. Their method achieved an accuracy of over 99% on both classification types.

Sabeel et al. [7] proposed DNN and LSTM models for binary prediction of unknown DoS and DDoS attacks. These models were trained on the CICIDS2017 dataset. The authors then generated a new test dataset, ANTS2019, in a simulated environment to measure performance of their proposed models. Their proposed DNN method was able to achieve an accuracy of 99.68% when it was trained on CICIDS2017 and part of ANTS2019 dataset.

Lee et al. [8] evaluated three algorithms: DNN, Self-Taught Learning (STL) Approach, and RNN, on their accuracy, precision, recall, and F1 score for detecting four security attack types. They used the KDD and NSL-KDD datasets separately for the training and evaluating the algorithms. They concluded that the STL approach performed best with an accuracy of 98.9%, whereas the LSTM model yielded 79.20% accuracy.

Wu et al. [9] proposed a hierarchical CNN+RNN neural network which they called LuNet. It consists of multiple levels of CNN and RNN where both network learns jointly from their input data. Their proposed model was tested on the NSL-KDD and UNSW-NB15 datasets [10]. They carried out binary and multiclass classification and achieved a maximum accuracy of 99.36% and 99.05% respectively. Both results were on the NSL-KDD dataset.

Almomani et al. [11] used eight different machine learning models in detecting DoS attacks which are: Naive Bayes (NB), Decision Trees (DT), Random Forests (RF), Support Vector Machine (SVM), J48, Artificial Neural Networks (ANN), K-Nearest Neighbor (KNN) and Bayesian Networks (BN). They used the WSN-DS dataset for their experiment and performed feature selection based on expert survey. The authors reported that the Random Forest algorithm achieved the best results with a True positive of 99.7% accuracy, out-performing the ANN model with a True positive of 98.3%.

Vinayakumar et al. [12] proposed a scalable and hybrid DNN framework called Scale-Hybrid- IDS-AlertNet, which can effectively monitor network traffic and host-level events in real-time to proactively alert for possible cyber attacks. The authors tuned the model on the KDD-99 dataset and applied it to other datasets such as NSL-KDD, UNSW-NB15, Kyoto, WSN-DS and CICIDS2017 as benchmark. For the WSN-DS dataset, they achieved accuracy of 99.2% and 98.0% for binary and multiclass classification respectively.

Park et al. [13] proposed a Random Forest (RF) classifier to detect the type of DoS attacks in the WSN-DS dataset. The proposed model achieved a best F1-score of 99%, 96%, 98%, 100%, and 96% for Blackhole, Flooding, Grayhole, Normal, and Scheduling (TDMA) attacks respectively. They achieved an overall accuracy of 97.8%.

Abdullah et al. [14] proposed used several ML classifiers for detecting intrusions in WSNs. These classifiers are SVM, Naive Bayes, Decision Tree, and Random Forest. They

used the WSN-DS dataset for training and the WEKA data mining tool for implementing their classifiers. The SVM classifier achieved the highest accuracy of 96.7% compared to the other classifiers.

Premkumar and Sundararajan [15] presents A Deep Learning-based Defense Mechanism (DLDM) to identify and isolate DoS assaults in the Data For-warding Phase (DFP). DoS assaults such as fatigue, jamming, homing, and flooding may now be detected more reliably thanks to a novel methodology described in research. It is more resistant to denial-of-service (DoS) assaults because we do extensive simulation studies to separate the enemies adequately. Their system's detection, throughput, packet delivery ratio, and accuracy in the simulation are all high. It also cuts down on wasted energy and the number of false alarms.

The complexity and frequency of Distributed Denial of Service (DDoS) attacks on Web-based services have grown tremendously with the emergence of modern wireless technology and current computing paradigms. As a result, it is critical to identify these attacks in the sea of data packets.

Asad et al. [16] provide a unique deep neural network detection technique for reliably detecting numerous application layer DDoS assaults in research using feed-forward back-propagation. On a state-of-the-art dataset encompassing several types of DDoS assaults, the neural network architecture suggested here can detect and utilize the essential high-level aspects of packet flows with a precision of 98 The primary threat to the WSN is posed by the fact that the nodes in the network broadcast their signals. As a result, the security of WSNs is an essential task that must be completed. As a result, to overcome these challenges or hazards, we are attempting to identify them utilizing artificial intelligence technologies. In order to categorize different sorts of assaults, using Machine Learning and Deep Learning, which are emerging domains, we may use a wide range of algorithms. Once we have identified the assault correctly, we may take the necessary steps to avoid it. We are making use of WSN-DS. It has four types of assaults: Grayhole, Blackhole, TDMA (Scheduling), and Flooding, all of which fall under Denial of Service attacks.

Loukas et al. [17] use LSTMs achieved 86.9% accuracy Covers all attack types, including DDoS, command injection, and network malware. this accuracy Better than what other standard machine learning methods have achieved. They also tested LSTM Outperforms Other Attacks Against Untrained Malware Attacks Again machine learning methods.

Shaaban et al. [18] recommend a CNN models to detect DDoS attacks. The authors compared their proposed model with classification algorithms KNN, DT, SVM, NN in More than two Dataset: (simulated network traffic) and (NSL-KDD) datasets. has been observed The proposed model compares well with this model The other four classification algorithms, such as KNN, DT, SVM, and NN with 99% accuracy two records. In this method, a single column is populated Used to convert data into matrix form. Therefore, it affects the learning of the model.

Wazirali and Ahmad [19] evaluated the effectiveness of machine learning classification algorithms in detecting (1) flooding, (2) gray hole, and (3) black hole distributed denial of service attacks in wireless sensor networks. We conducted our review using a WSN-based dataset, referred to as WSN-DS, and took the accuracy and speediness measures

into account. The results show that the J48 approach is the most accurate and fastest way for identifying grey hole and black hole attacks. At the same time, the Random Tree method is the most accurate and fastest method for detecting flooding assaults. The J48 approach is the most efficient for speed, requiring an average of 0.54 s of processing time per sample.

Salmi and Oughdir [20] presented a CNN-LSTM approach to detect and classify DoS intrusion attacks as Flooding,Blackhole, Normal, TDMA, or Grayhole.This research study uses a computer-generated wireless sensor network-detection system WSN-DS dataset;The developed model gives a promising outcome in the attack detection process and successfully classifies the given attacks with a accuracy of 97%.

Deshpande et al. [21] examines and compares the accuracy of five primary machine learning classification methods described in detail below. In addition, the paper examined the one deep learning algorithm that was used. The dataset has been used to train an ANN (Artificial Neural Network) and five machine learning algorithms. To further improve the accuracy of our predictions, we employed K-fold cross-validation to get even more accurate results. The results of our analysis of these algorithms led us to conclude, among other things, that Machine Learning algorithms such as the Random Forest, Support Vector Machines, and Deep Learning algorithms, such as the Artificial Neural Network, can assist us in detecting intrusions into a system or network.

Using this information, researchers will build their machine learning models on top of our proposed model. Machine learning categorization algorithms have been offered a viable option as an additional technique to deter service attacks. However, due to a lack of an appropriate and thorough evaluation of such approaches, it is difficult to determine their genuine contribution to improving the detection of denial-of-service (DoS) assaults in wireless sensor networks (WSNs).

Wazirali and Ahmad [19] adds to the assessment of the usage of machine learning algorithms in WSN node traffic and their influence on the lifetime of WSN networks by demonstrating their effectiveness. On a WSN dataset of varying sizes, authors investigated the performance metrics of various machine learning classification categories, including K-Nearest Neighbor (KNN), Logistic Regression (LR), Support Vector Machine (SVM), Gboost, Decision Tree (DT), Naive Bayes, Long Short Term Memory (LSTM), and Multi-Layer Perceptron (MLP). The tests demonstrated that the statistical and logical classification categories outperformed the numeric statistical datasets. The G boost algorithm outperformed the others on the average of all performance measures compared to other algorithms as shown in Fig. 1.

The performance indicators evaluated in these validation studies were accuracy, F1-score, false-positive ratio (FPR), false-negative ratio (FNR), and the total time it took to complete the training session. Furthermore, the results of the tests revealed that the Gboost algorithm achieved 99.6 percent accuracy, 98.8 percent F1-score, FPR, and FNR, and 0.4 percent 0.13 percent in F1-score, respectively. When it came to training execution time, the average of all training time execution datasets yielded 1.41 s on average.

Furthermore, this paper demonstrated that for the numeric statistical data type, the best results are obtained when the dataset size is between 3000 and 6000 records, and the percentage between categories is not less than 50% for each category when compared to the other categories, as demonstrated in the previous paper. The second part of

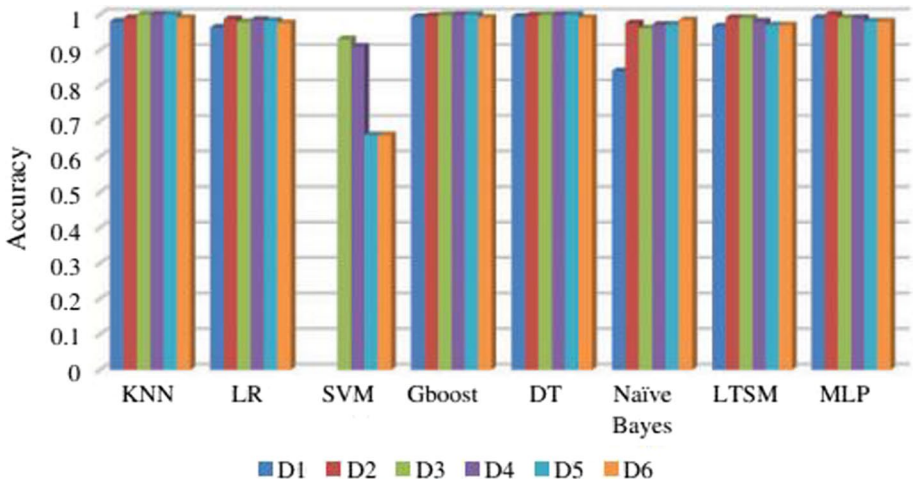

Fig. 1 Accuracy comparison between different datasets sizes and machine learning algorithms [19]

this article looked at the influence of Gboost on the lifespan of WSNs and found that it reduced the lifetime by 32 percent compared to scenarios in which Gboost was not used.

Throughout Gunduz et al. [22] looked at Denial of Service (DoS) attacks at each TCP/IP protocol stack layer. Author attention was drawn to the network layer assaults, which are more diversified than the other layers of attacks. The author examined several pieces of research that propose machine learning strategies for defending against network layer denial of service attacks in wireless sensor networks (WSNs). In addition, the author presents some comparison results to assist scholars who are researching this sector.

Table 1 shows the summarizes of reviewed detection methods and the datasets used with the higher accuracy achieved.

Previous works on most popular datasets for DoS and DDoS attacks detection based deep learning

CICDDoS2019 dataset

more than 80 traffic features have been extracted from raw data using CICFlowMeter V3 tool https://www.unb.ca/cic/datasets/ddos2019.html. CICDDoS 2019 includes benign and currently common DDoS attack. This record is generated with real traffic, Includes a number of different DDoS attacks generated by protocols using TCP/UDP. taxonomy Attacks include exploit-based and reflection-based attack. Reflection-based attacks include Microsoft SQL Server (MSSQL), Network Time Protocol (NTP), Simple Service Discovery Protocol (SSDP), CharGen, Trivial File Transfer Protocol (TFTP), Lightweight Directory Access Protocol (LDAP), Domain Name Server (DNS), Simple Network Management Protocol (SNMP), network Basic Input/Output System (NETBIOS) and PortMap. That Exploit-based attacks include UDP Flood, UDPLag, and SYN flood. In both cases, the dataset was collected within 2 days PCAP file and stream formats for training and testing Evaluate.Twelve DDoS attacks of the training day Contains DNS, LDAP, NTP, MSSQL, UDP, UDP-Lag, NetBIOS, SNMP, SSDP, WebDDoS, TFTP and SYN, On January 12, 2019, it was recorded that Test day includes NetBIOS, PortScan, LDAP, UDP, UDPLag, MSSQL and SYN collected on March 11, 2019 [23]. (Table 2) shows the previous deep learning approaches for DoS and DDoS attacks detection on CICDDoS2019 Dataset with their accuracies.

Table 1 the summarizes of reviewed detection methods,datasets used and accuracies

References	Datasets	Techniques	Accuracy %
Kim et al. [4]	KDD-99,CICIDS2018	CNN,RNN	99
Sabeel et al. [7]	CICIDS2017,ANTS2019	DNN,LSTM	99.68
Lee et al. [8]	NSL-KDD	DNN,STL,RNN	98.9
Wu et al. [9]	NSL-KDD,UNSW-NB15	CNN+LSTM,LuNet,RNN	99.36
Almomani et al. [11]	WSN-DS	NB,DT,RF,SVM,J48,ANN,KNN,BN	99.7
Vinayakumar et al. [12]	KDD-99,NSL-KDD ,WSN-DS,CICIDS2017,WSN-DS,CICIDS2017,Kyoto	DNN	99.2
Park et al. [13]	WSN-DS	RF	97.8
Abdullah et al. [14]	WSN-DS	SVM,NB,DT,RF	96.7
Premkumar and Sundararajan [15]	WSN CH	RBF	99
Asad et al. [16]	CICIDS2017	ANN	98
Loukas et al. [17]	malware (Net)	LSTM,LMP	86.9
Shaaban et al. [18]	simulated network traffic and NSL-KDD	CNN	99
Salmi and Oughdir [20]	WSN-DS	CNN+LSTM	97
Wazirali and Ahmad [19]	WSN dataset	KNN,LR,SVM,Gboost,DT,LSTM ,MLP	99.6
Deshpande et al. [21]	WSN-DS	ANN,SVM,RF,KNN,LR,NB	99

Table 2 Accuracy of deep learning approaches for DoS attacks detection on CICDDoS2019 Dataset

References	Source	DL Techniques	Accuracy %
A	Sbai and El boukhari (2020) [24]	DNN	100
B	de Assis et al. (2020) [25]	CNN	95.5
C	Hussain et al. (2020) [26]	CNN	100
D	Shurman et al. (2020) [27]	LSTM	99.2
E	Elsayed et al. (2020) [28]	RNN+AE	99
F	Amaizu et al. (2020) [29]	DNN	99.6
G	Cil et al. (2021) [30]	DNN	100
H	Assis et al. (2021) [31]	GRU	99.98

NSL-KDD dataset

This dataset is an extension of the KDDCUP99 dataset to exclude some problems of KDDCUP99 dataset https://www.unb.ca/cic/datasets/nsl.html. KDDCUP99 dataset contains numerous spare and indistinguishable records, and to fix these problems,the NSL-KDD dataset was proposed. The number of records in the train and test sets is reasonable in the NSL-KDD dataset. It contains roughly data points, and this dataset also contains emulated records [32]. The dataset is labelled and imbalanced and contains training records of and testing records of. It also includes four types of attacks DOS, Probe, R2L, U2R [33], (Table 3) shows the previous deep learning approaches for DoS and DDoS attacks detection on NSL-KDD Dataset with their accuracies.

Figure 2 displays the accuracy of the DDoS attack detection deep learning-based solutions on the CICDDoS2019 and NSL-KDD datasets respectively, only CNN approach (Shaaban et al. 2019) [18] showed an accuracy above 99% on NSL-KDD dataset.for

Table 3 Accuracy of deep learning approaches for DDoS attacks detection on NSL-KDD Dataset

References	Source	DL techniques	Accuracy %
A	Amma et al. (2019) [34]	CVNN+FCNN	94.2
B	Shaaban et al. (2020) [18]	CNN	99.2
C	Kasim (2020) [35]	AE-SVM	96.4
D	Bhardwaj et al. (2020) [36]	AE+DNN	98.4

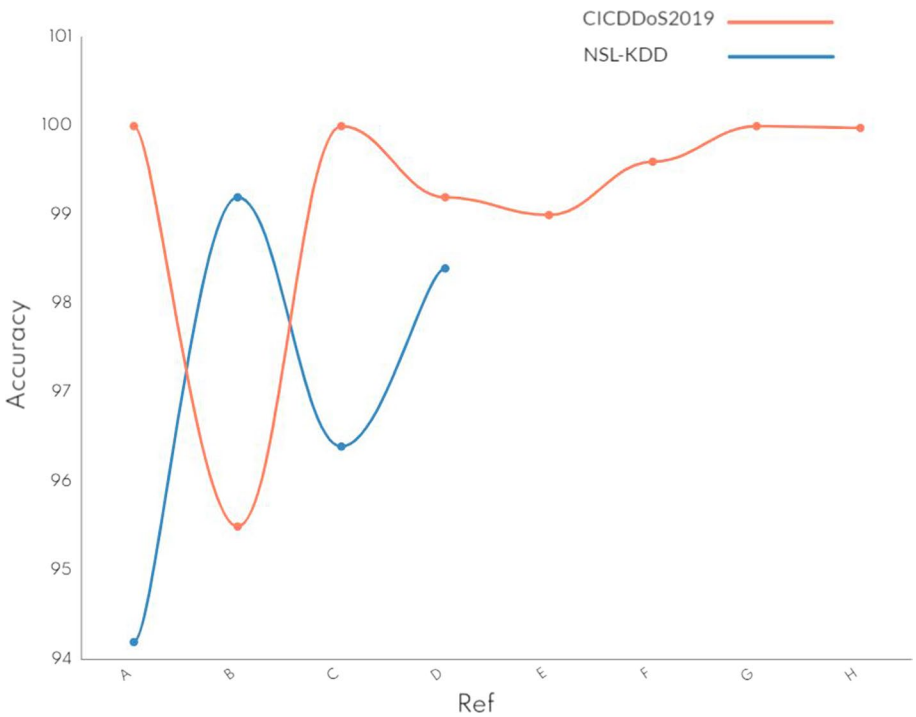

Fig. 2 Accuracy of deep learning approaches for DoS and DDoS attacks detection on CICDDoS2019 Dataset and NSL-KDD Dataset

CICDDoS2019 dataset It has been observed that the approaches CNN-based ResNet (Hussain et al. 2020) [26], LSTM (Shurman et al. 2020) [27], DNN (Sbai and El Boukhari 2020) [24], DNN (Amaizu et al. 2021) [29], DNN (Cil et al. 2021) [30], and GRU (Assis et al. 2021) [31] showed accuracy greater than 99%.

Intrusion detection system (IDS)

IDS (intrusion detection system), sometimes known as infiltration prevention system (IPS), is an active defensive mechanism deployed by the Internet of Things (IoT) that can recognize intrusion activity and trigger alerts. However, with the rising number of dangers in the Internet of Things, there are questions about present methods' long-term viability and practicality. These considerations are particularly relevant in light of the growing levels of adaptive performance and the inadequate levels of detecting precision.

Intrusion detection capabilities include: User and system monitoring and analysis Activity; Analyze system configuration and Vulnerability; System and file integrity assessment; Ability to identify attack patterns; Analysis of abnormal activity patterns; Track users for policy violations

The purpose of IDS is to help computer systems do this Responding to Attacks and IDS Gathering Information from several different sources within the computer system and the Internet, and compare this information with existing patterns of discrimination to see if there are any attack or weakness [37].

Comparison of IDS and Firewall: Both IDSs and firewalls are related to network security, but IDSs are different from firewalls in that IDS monitors the network, provides a real- time detection of attacks from the interior and exterior, and automatically informs firewall and dynamically alters the rules of firewall once an attack is found; on the other hand, firewall loads dynamic rules to hold up the intrusion, controls the data traffic of IDS and provides the security protection of IDS [38]. Figure 3 shows a very basic structure of Intrusion Detection System.

Wireless sensor networks (WSN)

WSNs are made up of several SENSOR NODES strategically placed across an area of concern. Essential characteristics such as temperature, pressure, humidity in the environment, soil quality, brightness, and a wide range of other data may be sensed by these sensor nodes, amongst other things. These sensor nodes can not only sense the parameters, but they can also communicate with one another, resulting in the formation of a network. They are carefully created in such a usual manner that they include a microprocessor that controls the monitoring, a radio transceiver for producing radio waves, various types of wireless communication devices, and an energy source such as a battery [39]. Figure 4 shows a typical wireless sensor network logical hierarchy diagram.

A sensor network comprises a collection of tiny, powered sensors connected to a networked infrastructure, either wireless or wired. It is possible to record circumstances in various contexts, such as industrial facilities, farms, and hospitals. The sensor network communicates with the Internet or computer networks to relay data for analysis and utilization. Sensor network nodes perceive and monitor their surroundings in a coordinated manner. They make it possible for persons or machines to communicate with and with the surrounding environment [41].

Characteristics of WSN

When picking a piece of WSN, there are many factors to consider.

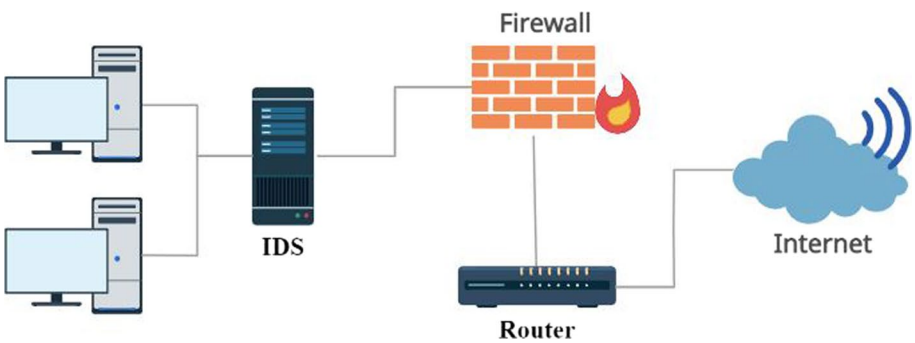

Fig. 3 Structure of intrusion detection system

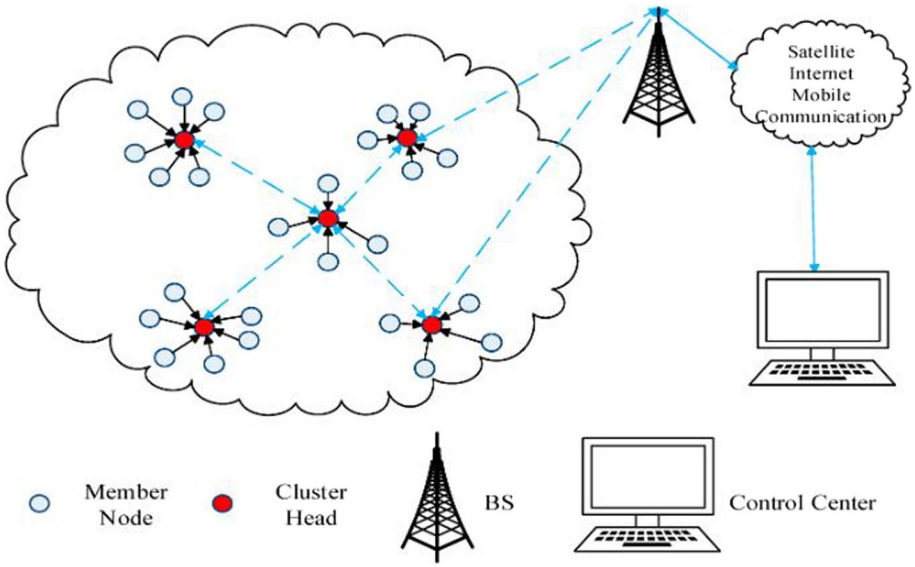

Fig. 4 A typical wireless sensor network (WSN) logical hierarchy diagram [40]

- *Type of Measurement*: It is critical to comprehend what is being measured. When it comes to wireless transmitters (which combine wireless process measurement and control), each one serves a specific purpose. Sensors are specially developed for various applications such as temperature, pressure, flow, etc., and must be chosen following these requirements.
- *Accuracy and Response Time*: Most wireless sensors are as exact as their wired equivalents in terms of accuracy; however, the data are often communicated every few seconds to save battery life. This must be considered while choosing the wireless transmitter if immediate measurement is required since specific devices may not require the reaction time.
- *Range*: The range of wireless sensors is quite variable. Some sensors are intended for short-range interior applications with a range of a few hundred feet or less, while others can transmit data to hundreds of miles distant receivers. Regardless of the capabilities of the sensors, the range of a wireless signal is permanently restricted by the presence of barriers. Signal strength and range capabilities are reduced when signals are sent via machinery, walls, and other structures. Consequently, the range of an inside transmitter is often substantially smaller than the range of an outside transmitter broadcasting in a vast open area.
- *Frequency*: The radio transmission frequency is also an essential factor to consider. The laws governing which sections of the wireless spectrum are open for usage without a license differ from nation to country and area to region. The primary frequencies that industries may utilize to broadcast signals in the United States are 915MHz and 2.4GHz (WiFi). Because these frequencies are part of the industrial, scientific, and medical spectrum, users do not need a radio license to use them. For the most part, wireless goods in Europe run on the 868MHz or 2.4GHz frequency bands. In some instances, items may only be sold in specific locations due to governmental limitations.

DoS attacks and WSN

Depending on their nature, WSN attacks may be divided into Invasive and non-invasive.

(i) Non-invasive assaults are often directed against the targeted channel's timings, power, and frequency.
(ii) Invasive attacks disrupt service availability, information transmission, routing, and other functions.

DoS attacks are attempts by hackers to render a service or system unreachable. More prevalent assaults, on the other hand, are experienced during the transmission of information. Routing attacks are often carried out from inside a network.

There are many different types of Denial of Service circumstances. These situations can potentially degrade the performance of WSN nodes and network operation. These may interfere with the network's usual operations, manifesting themselves in the form of resource depletion, any software problem, or any issue encountered when dealing with the application or infrastructure.

The term "denial of service" (DoS) refers to any such impediments in network functioning that impair the availability or complete operation of service; however, when the opponent causes these obstacles on purpose, they are referred to as "denial of service assaults." Dos attack refers to an opponent's deliberate attempt to demolish or destruct a network's infrastructure.

A distributed denial-of-service attack (DDoS) may have a more significant impact on network operation than anticipated. DoS attacks against WSN may occur at any layer of the OSI model [42]. DoS attacks are susceptible because they breach the efficiency of targeted networks by interfering with the protocols linked with them. DoS attacks can devour resources, destructs modifies the infrastructure configuration, and physically damage network components, among other things.Wood and Stankovic were the first to propose a layer-by-layer taxonomy of denial-of-service attacks [43].

Methods

Dataset overview

This paper uses the specially developed dataset for DoS attack detection in WSNs called WSN-DS [44]. The authors of the dataset used the LEACH protocol [45] during data collecting. For each data instance, 23 attributes were collected although only 19 were present in the dataset. Table 4 shows the attributes and their description, Fig. 5 presents the graphical representation of all the five kinds of attacks present in the dataset and their distribution.

Four types of Denial of Service attacks were simulated in the dataset which are: *Blackhole*, *Grayhole*, *Flooding*, and *Scheduling (TDMA)* attack. Each data instance is labeled as either Normal or one of the four attack types. The description of the attack types is as follows:

- *Blackhole attack*: In this DoS attack, the attacker node advertises itself as a cluster head (CH) which are responsible for transmitting data from their cluster member

Table 4 WSN-DS dataset attributes

No.	Attribute	Description
1	Node ID	Node ID number
2	Time	Node runtime
3	Is CH	Used to mark whether the node is a cluster head
4	Who CH	Cluster head ID
5	Distance to CH	Distance between node and cluster head
6	ADV CH sent	The number of the advertise CH's broadcast messages sent to the nodes
7	ADV CH received	The number of advertise CH messages received from CHs
8	Join REQ sent	The number of join request messages sent by the nodes to the CH
9	Join REQ received	The number of join request messages received by the CH from the nodes
10	ADV SCH sent	The number of join advertise TDMA schedule broadcast message sent
11	ADV SCH received	The number of scheduled messages received by the CH
12	Rank	Order of node TDMA scheduling
13	Data sent	The number of packets sent from the normal node to its CH
14	Data received	The number of packets received by the node from the CH
15	Data sent to BS	The number of packets sent to the BS
16	Distance CH to BS	Distance between CH and BS
17	Send Code	The cluster sending code
18	Consumed energy	The current energy for the node in the current round
19	Attack Type	Type of the node

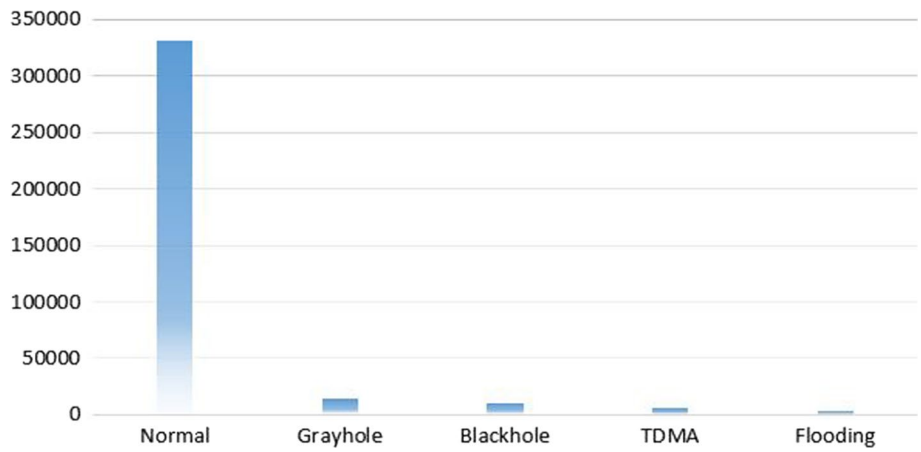

Fig. 5 WSN-DS attack type records

(CM) sensor nodes to the base station. The blackhole attack collects data packets from their CM but does not forward them to the base station.

- *Grayhole attack*: In this DoS attack, the attacker node advertises itself as a CH. When data packets are received from the CMs, it drops some of the packets randomly before forwarding the rest to the base station.
- *Flooding attack*: In this DoS attack, the attacker node advertises itself as a CH. This attack attempts to waste other sensor energy by sending large number of advertising CH messages (ADV_CH) with high transmission power. Nodes that are far away and would other have chosen a different CH would now have to use more energy in transferring their data packets.

- *Scheduling attack*: In this DoS attack, the attacker node acts as the CH and assigns the same data transmission time to several nodes. This causes packets collision and leads to loss of data.

Proposed methodology

Data pre-processing (data cleaning, data Transformation, Normalization) is one of the proposed methodology process. The proposed methodology's first step is preprocessed the dataset to produce refined data. t((We will perform two important steps during pre-processing: data cleaning and Transformation of cleaned data into numeric values (0,1,2,3,4). The final step is to train and test the (CNN,DNN,RNN,CNN+RNN) models on preprocessed data to evaluate its performance in detecting DoS attack patterns as shown in Fig. 6. All of these steps are detailed in the subsections that follow.

Data preprocessing

Data cleaning

Data cleaning refers to the process of getting ready data for analysis by removing the unnecessary or incorrect information. This is typically data that can have a deleterious impact on the model or algorithm into which it is fed by reinforcing an incorrect notion.

We identify and drop duplicates, redundant and NULL data,Handle missing data, We Detect and eliminate data anomalies by validating against known reasons.We Implement

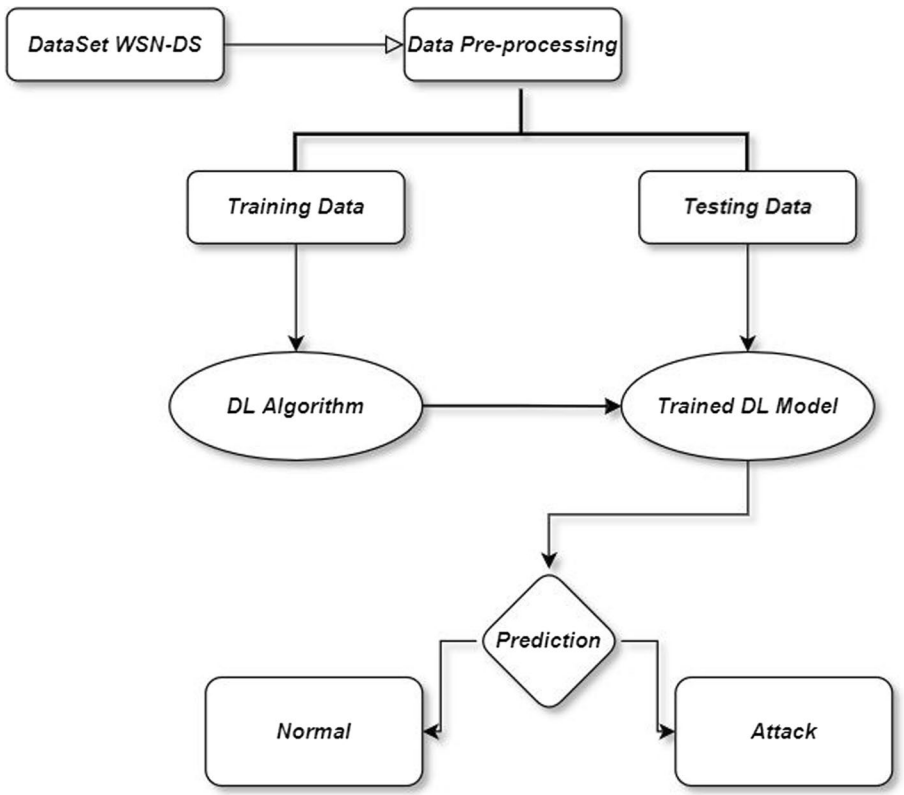

Fig. 6 Proposed methodology for detecting DoS attacks using DL techniques

proper data quality controls when importing new data by respecting five positive attributes that can be used to evaluate its quality:

Validity; Accuracy; Completeness; Consistency; Uniformity.

Data normalization and transformation

The process of converting raw data into a format or structure that is more suitable for the model, as well as data analysis in general, is known as data transformation.

Each attack type was converted into numeric values with Normal, Blackhole, Grayhole, Flooding, and TDMA, converted to 0, 1, 2, 3, and 4 respectively as shown in Table 5. Among the 19 attributes Node ID and Who CH are not used for DoS attack detection.

The main objective of normalization is to transform features so that they are all on the same scale. This improves the model's performance and training stability. One of the most common methods for normalizing data is min-max normalization described mathematically using Eq. 1. For each feature, the minimum value is converted to a 0, the maximum value is converted to a 1, and all other values are converted to a decimal between 0 and 1.

$$Z' = \frac{Z - minF}{maxF - minF}(new_{maxF} - new_{minF}) + new_{minF} \tag{1}$$

where Z' is the new value of each entry in data, Z is the old value of each entry in data, new_{maxF} and new_{minF} is the max and min value of the range.

Data splitting

To avoid overfitting, the data should be divided into two sets: training and testing. We train our model on the training set first, and then use the data from the testing set to evaluate the accuracy of the generated model.Empirical studies show that using 20–30% of the data for testing and the remaining 70–80% for training produces the best results [46]. We split Our dataset randomly into 80% training set and 20% testing set.The accuracy rate (when using an 80:20 splitting ratio) increases overall,The exact figure is given in Table 5.

Classification

We trained four DL algorithms on the WSN-DS dataset, which are Dense Neural Network (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and an algorithm that combines the CNN and RNN architectures. Of the 19

Table 5 Distribution of WSN-DS dataset

Attack type	Attack index	Training Set (80%)	Testing Set (20%)	Proportion
Normal	0	272,087	67,979	90.77%
Blackhole	1	8019	2030	2.68%
Grayhole	2	11,653	2943	3.9%
Flooding	3	2694	618	0.88%
TDMA	4	5275	1363	1.77%
Total		299728	74933	100%

feature present in the dataset (including the attack type), 16 were used in training the algorithm. After training, the models are evaluated and compared with other algorithms using well known comparison metrics such as accuracy, precision, recall, and F1-score. See the following Algorithm 1 for details:

Algorithm 1: DoS Attack Detection Algorithm

Input Training dataset WSN-DS, Testing dataset WSN-DS
Output Classification results: accuracy, precision, recall, and F1-score
Begin: Data preprocessing
z''=Data_Cleaning(Z');
z'=normalization(z);
x',y',z'=Transformation(x,y,z);
end
Begin: Feature extraction
Train the DNN,CNN,RNN,CNN+RNN using the WSN-DS Train+ dataset;
minimize the reconstruction error;
end
Begin: Classification
Train the DNN,RNN,CNN,CNN+RNN classifier;
Testing dataset WSN-DS Test are input into the trained DNN,RNN,CNN,CNN+RNN classifier to detect attacks;
end
Return the classification result ;

Algorithms

Four Deep Learning architectures were considered in this paper which are: DNN, CNN, RNN, and CNN+RNN.

- DNN models are made by combining feed forward neural networks with no feedback connections. The input, output, and hidden layers, which can be multiple, are the main components of the DNN. Each layer contains weighted units. These units carry out the activation processes of the units from the previous layer [47]. For the Dense Neural Network (DNN) model, we used a single hidden layer with an activation function with 32 neurons see Fig. 7. mathematical equation is given by 2:

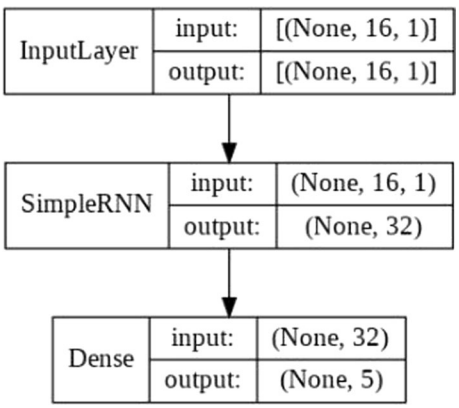

Fig. 7 Proposed DNN architecture

$$g(x) = f(x^T w + b) \tag{2}$$

where g is notations for hidden layers,f is activation function,x is input vector,b is output vector,w is weight vector of unit.

- The Convolutional Neural Network (CNN) is a typical artificial feed-forward neural network that extracts the characteristics of input data through the use of multiple filters. CNN continuously analyzes the extracted features to obtain the final features as the number of network layers increases. CNN has two features: local connectivity and weight sharing. The convolutional layer and the previous layer are linked by a local connection and weight sharing, which greatly reduces the number of parameters, network complexity, network robustness, and can effectively prevent overfitting. The convolutional neural network's basic structure is as follows: input layer, convolutional layer, pooling layer, fully connected layer, and output layer. In general, the convolutional and pooling layers appear alternately. Finally, the pooling layer's features are connected to form a feature vector, and the feature vector obtains a classification vector via the fully connected layer. *Convolutional layer* is made up of multiple feature maps, each of which is made up of multiple neurons. The convolutional kernel connects each neuron to the upper feature map. Convolutional layer uses convolution to extract features from different levels of input layer. The following 3 is the structure of a convolutional layer:

$$x_j^l = f\left\{\sum_{i \in M_j} x_j^{l-1} k_{ij}^l + b_j^l\right\} \tag{3}$$

where l indicates the current layer, b the bias of the current layer, k the convolutional kernel, and M_j the convolution window of the $j - th$ convolutional kernel. *pooling layer* Behind the convolutional layer, the pooling layer is also made up of multiple feature maps. Each feature map in the pooling layer corresponds to only one feature map in the previous layer, and the total number of feature maps remains constant. The convolutional layer is the pooling layer's input layer. The pooling layer takes the following shape 4:

$$x_j^l = f\left(\beta_j^l down\left(x_j^{l-1}\right) + b_j^l\right) \tag{4}$$

where $down(x_j)$ represents the j th - neuron's down sampling. Weight β and bias b are assigned to each output feature map. *Fully connected layer* One or more fully connected layers are connected after multiple convolutional layers and pooling layers. Each neuron in the fully connected layer is completely connected to every neuron in the preceding layer. Each neuron in the fully connected layer usually chooses the ReLu function as its activation function, and the output value of the last fully connected layer is delivered to an output layer that can be classified by Softmax. *ReLu* is a non-linear activation function used in deep neural networks and multilayer neural networks. This function can be written as:

$$f(x) = max(0, x) \tag{5}$$

where x is an input value; The maximum value between 0 and the input value is the output of ReLu, according to Eq. 5. When the input value is negative, the output is equal to 0, and when the input value is positive, the output is equal to the input value. As a result, we can rewrite Eq. 5 as follows:

$$f(x) = \begin{Bmatrix} 0, if\,(x) < 0) \\ x, if\,(x) \geq 0) \end{Bmatrix} \tag{6}$$

where x is an input value; Our CNN proposed model had a single hidden layer consisting of a 1D convolutional network with a kernel size of 5 and 32 filters, 1D max-pooling of kernel size 2 and stride 2, and the ReLU activation function. see Fig. 8

- Recurrent Neural Networks, or (RNNs), are a type of neural network that is used to process sequential data. Sequential data is a collection of data points.An RNN, unlike a typical neural network, does not limit its input or output to a set of fixed-sized vectors. It also does not limit the number of computational steps needed to train a model. Instead, it enables us to train the model with a series of vectors (sequential data). *Bidirectional RNNs* A BRNN is a combination of two RNNs, one of which moves forward from the start of the data sequence and the other which moves backward from the end of the data sequence. To accommodate the backward training process, a BRNN has an additional hidden layer. The forward (*f*) and backward (*b*) hidden states are described as follow 7 and 8 at any given time *t*.

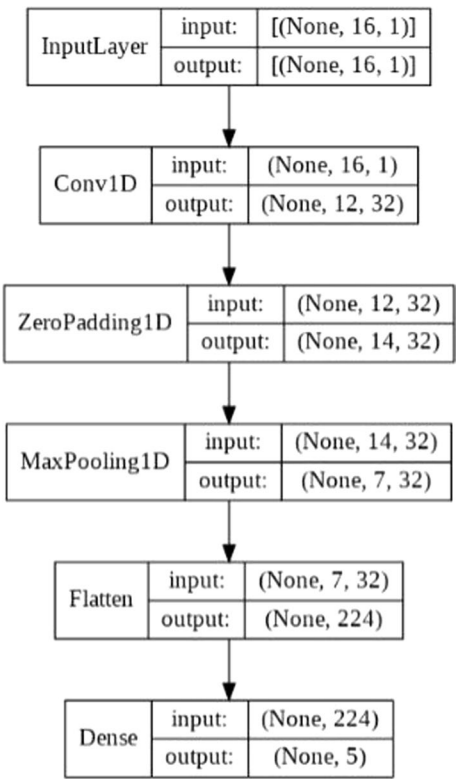

Fig. 8 Proposed CNN architecture

$$A_t(f) = \phi\left(X_t * W_{XA}^f + A_{t-1}(f) * W_{AA}^f + b_A^f\right) \tag{7}$$

$$A_t(b) = \phi\left(X_t * W_{XA}^b + A_{t+1}(b) * W_{AA}^b + b_A^b\right) \tag{8}$$

where ϕ is the activation function, W the weight matrix, and b is the bias. For Our Recurrent Neural Network (RNN) Proposed, a single bidirectional hidden layer was used with a hidden size of 32 see Fig. 9.

- Combining convolutional neural networks (CNN) and recurrent neural networks (RNN) obtains many interesting properties RNN can learn temporal and contextual features, particularly long-term dependency among different entities, whereas CNN can capture more potential features. The CNN+RNN proposed model is the same as having the CNN model without the output layer and directly stacking the RNN model on it see Fig. 10.

The output layer is the same for all the models. It includes a single dense neural network followed by a softmax activation function. A dropout of 0.2 is applied to the values before being passing to a dense network. Table 6 shows the summarizes of Algorithms used.

The same training setup was used for each model. They were trained for 25 epochs with a learning rate of 0.001, Adam optimizer, batch size of 16, and cross-entropy loss,as shown in Table 7.The evaluation results are discussed in the next section.

Experimental results and analysis

Experimental environment

We conduct experiments to evaluate the performance of the proposed DNN,CNN, RNN, RNN and CNN. The proposed Models was implemented withing Python 3.7.7 using TensorFlow with Keras environment with 16 GB RAM, GTX-2080Ti Nvidia GPU and 64-bit Windows 10 operating system.

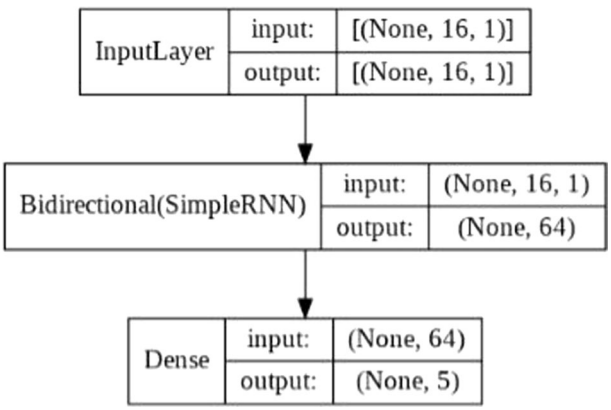

Fig. 9 Proposed RNN architecture

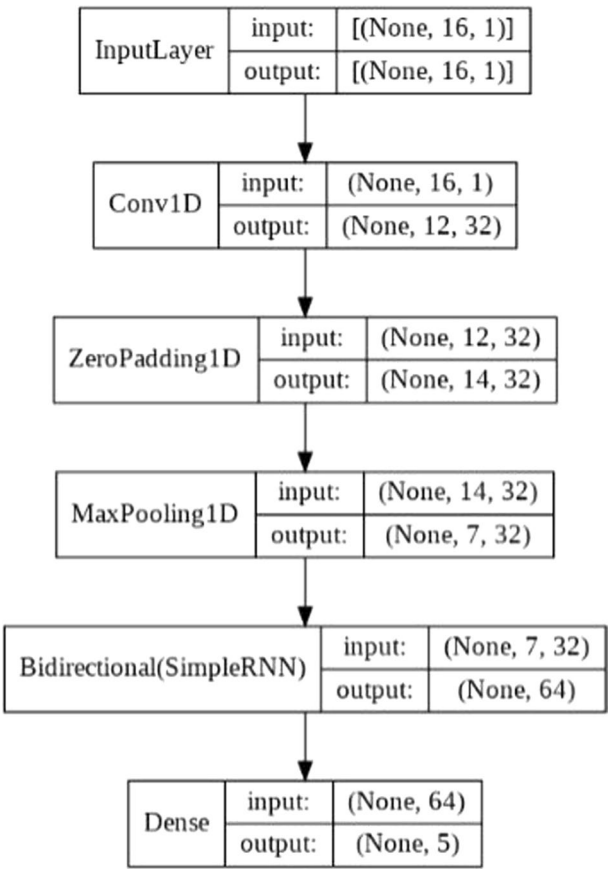

Fig. 10 Proposed CNN+RNN architecture

Table 6 Summary of Algorithms used

Algorithm	Hidden layers	Hidden size / Filter size
DNN	1	32
CNN	1	32
RNN	1	32
CNN+RNN	2	[32, 32]

Table 7 Models hyperparameter

Hyperparameter	Value
Epoch	25
Activation Function	ReLU, Softmax
Batch size	16
Loss function	Categorical cross entropy (CCE)
Optimization algorithm	Adam
Learning rate	0.001
Verbose	1

Performance metrics

The suggested methodology's performance is evaluated using four generally used performance metrics: *precision, recall, accuracy, and F1-measure*. For multi-class classification, the confusion matrix is used to calculate each of these parameters separately for each class. These parameters are described as:

Precision: It specifies the ratio of truly detected attacks to all packets classified as attacks. It is expressed mathematically as:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

Recall: It is the system's ability to correctly detect an attack upon the occurrence of a security breach. The true positive rate is another name for it. It is mathematically described as:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

Accuracy: It is defined as the system's ability to correctly identify an attack packet as a "attack packet" and a normal packet as a "normal packet." It describes the proportion of correct predictions in relation to all samples. It is expressed mathematically as:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \times 100 \tag{11}$$

F1-Score: The frequency mean of precision and recall is defined. It is represented mathematically as:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

Where **TP** is The number of attack cases correctly classified as attacks,**TN** is the number of normal cases correctly classified as normal (no-attack),**FP** is the number of normal cases incorrectly classified as attacks. **FN** is the number of attack cases that were incorrectly classified as normal (no-attack).

Results and discussion

In this experiment, we compare (CNN,RNN,DNN,CNN+RNN) models performance of detection rate After training the proposed models on 25 epochs with a learning rate of 0.001. Figure 11 show the accuracy of each Algorithm.

After training each algorithm, we evaluated their performance as shown in Table 8. we see that the CNN model achieved the highest performances on all metric with an accuracy of 98.75%. The DNN model had the next highest performance with an accuracy of 97.07%, followed by the CNN+RNN model with an accuracy of 96.84%. The RNN model had the worst performance with an accuracy of 96.50% which is a difference of 0.34% compared to the CNN+RNN model and 2.25% compared to the best performing model.

The CNN model is also seen to have the best F1 Score on every attack type.

We also evaluated the models on the test data with the results in Table 9.

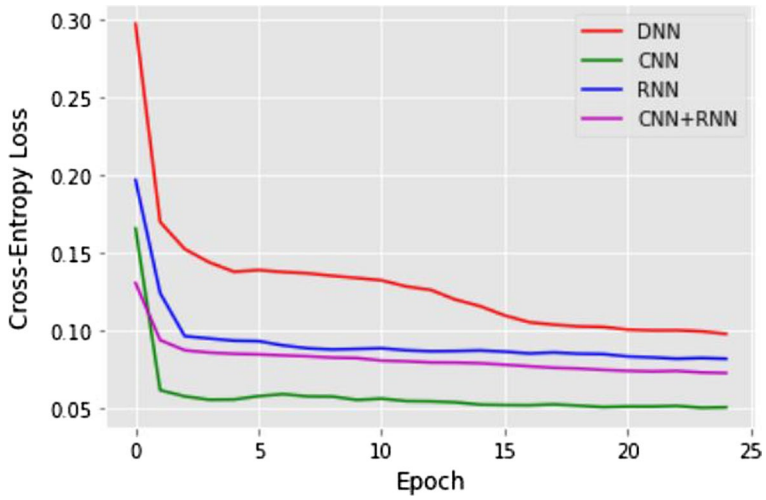

Fig. 11 Plot of loss vs. epoch

Table 8 Performance of each algorithm on training data

Algorithm	Attack type	Precision, %	Recall, %	F1 Score, %	Accuracy, %
DNN	Normal	99.64	99.19	99.41	–
	Blackhole	64.06	83.78	72.61	–
	Grayhole	69.56	67.72	68.63	–
	TDMA	97.56	86.93	91.94	–
	Flooding	88.57	77.50	82.67	–
	Overall	83.88	83.02	83.05	97.07
CNN	Normal	99.23	99.78	99.50	–
	Blackhole	94.67	95.12	94.89	–
	Grayhole	91.77	83.65	87.52	–
	TDMA	99.06	87.23	92.77	–
	Flooding	90.40	96.47	93.33	–
	Overall	95.03	92.45	93.60	98.75
RNN	Normal	98.25	99.91	99.07	–
	Blackhole	81.06	47.56	59.94	–
	Grayhole	66.02	72.82	69.25	–
	TDMA	97.30	76.46	85.63	–
	Flooding	89.77	51.97	65.83	–
	Overall	86.48	69.74	75.94	96.50
CNN+RNN	Normal	99.57	99.27	99.42	–
	Blackhole	53.13	99.77	69.34	–
	Grayhole	85.53	43.63	57.79	–
	TDMA	100.00	83.66	91.10	–
	Flooding	87.62	96.17	91.69	–
	Overall	85.17	84.50	81.87	96.84

Here also, we observe the CNN model performing best in terms of accuracy (98.79%), followed by the DNN model (97.04%), CNN+RNN model (96.86%), and the RNN model (96.48%), just like we observed on the training data.

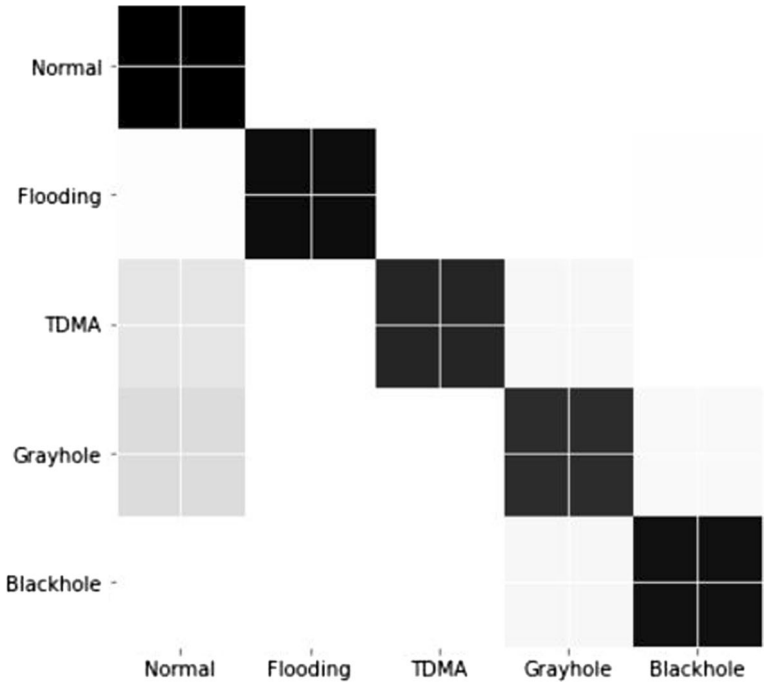

Fig. 12 CNN model confusion matrix on test data

Table 9 Performance of each algorithm on testing data

Algorithm	Attack type	Precision, %	Recall, %	F1 Score, %	Accuracy, %
DNN	Normal	99.64	99.21	99.42	–
	Blackhole	64.56	82.07	72.27	–
	Grayhole	67.89	66.51	67.19	–
	TDMA	98.00	88.31	92.90	–
	Flooding	83.91	73.95	78.62	–
	Overall	82.80	82.01	82.08	97.04
CNN	Normal	99.22	99.76	99.49	–
	Blackhole	95.12	96.44	95.78	–
	Grayhole	93.33	83.88	88.35	–
	TDMA	99.59	87.05	92.90	–
	Flooding	87.03	97.70	92.06	–
	Overall	94.86	92.97	93.72	98.79
RNN	Normal	98.19	99.88	99.03	–
	Blackhole	81.31	47.01	59.58	–
	Grayhole	66.13	71.03	68.49	–
	TDMA	98.60	75.90	85.77	–
	Flooding	83.85	51.72	63.98	–
	Overall	85.62	69.11	75.37	96.48
CNN+RNN	Normal	99.58	99.28	99.43	–
	Blackhole	53.54	100.00	69.74	–
	Grayhole	86.97	44.70	59.05	–
	TDMA	100.00	87.05	93.08	–
	Flooding	85.86	95.40	90.38	–
	Overall	85.19	85.29	82.34	96.86

The CNN model also dominated on the F1 score metric in all attack type (Normal 99.49%, Blackhole 95.78%, Grayhole 88.35%, and Flooding 92.06%) except the TDMA attack which was by the CNN+RNN model with an F1 score of 93.08% compared with CNN's 92.90%, The CNN confusion matrix is shown in Fig. 12.

we show the confusion matrix plot for CNN model when tested with the test data set. The rows represent the predicted class (Output Class), while the columns represent the actual class (Target Class). The diagonal cells in the confusion matrix represent correctly classified observations (TP and TN). Off-diagonal cells correspond to observations that were incorrectly classified (FP and FN). Each cell displays both the number of observations and their percentage of the total number of observations.

In terms of speed of training, the DNN model was the fastest followed very closely by the CNN model. The RNN and CNN+RNN models used about 3 times the training time of the DNN model although the CNN+RNN model was faster than the RNN model.

It is also interesting to note the similarity in performance of the models on the training data and testing data. This could be attributed to the dropout layer applied in each model which are used in preventing over-fitting and improve generalization. Because of this closeness in values, we could expect that further training on the training data would improve the models on both dataset.

Conclusion

In this work, we proposed efficient and lightweight (single layer) IDSs for detecting DoS attacks in WSN. We experimented on four deep learning models: DNN, CNN, RNN, and CNN+RNN. From the results of our experiments, we conclude that the CNN model (98.79%) achieved the best performance. A longer training time could improve performance as we trained our models for 25 epochs (reaching a max accuracy of 98.79%) compared to the 500 epoch used by the authors of the dataset (with a max accuracy of 97.54%) [44]. For future work, we plan on performing feature selection to reduce the size of the training data while keeping the performance.

Acknowledgements
Not applicable.

Author contributions
SS developed and deployed deep learning models presented in this article, prepared and analyzed the data, interpreted the results, and authored the manuscript. The data collection and acquisition was conducted by LO. All mentioned authors contribute to the elaboration of the article. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The data that support the findings of this study are available on request from the corresponding author [sel@psu.edu.sa]. The data are not publicly available.This dataset was created by the support of the Security Engineering Lab (SEL), College of Computer and Information Sciences, Prince Sultan University, Saudi Arabia."Almomani, Iman, Bassam Al-Kasasbeh, and Mousa Al-Akhras.[WSN-DS: A dataset for intrusion detection systems in wireless sensor networks."Journal of Sensors 2016 (2016), http://dx.doi.org/10.1155/2016/4731953.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Received: 8 March 2022 Accepted: 21 January 2023
Published online: 07 February 2023

References

1. França RP, Iano Y, Monteiro ACB, Arthur R. Intelligent applications of wsn in the world: a technological and literary background. In: Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's, Springer, 2020; pp. 13–34.
2. Haseeb K, Ud Din I, Almogren A, Islam N. An energy efficient and secure iot-based wsn framework: an application to smart agriculture. Sensors. 2020;20(7):2081.
3. Bhushan B, Sahoo G. Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks. Wireless Pers Commun. 2018;98(2):2037–77.
4. 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.
5. Stolfo SJ, Fan W, Lee W, Prodromidis A, Chan PK. Cost-based modeling for fraud and intrusion detection: Results from the jam project. In: Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00, 2000;2: 130–144 IEEE.
6. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp. 2018;1:108–16.
7. Sabeel U, Heydari SS, Mohanka H, Bendhaou Y, Elgazzar K, El-Khatib K. Evaluation of deep learning in detecting unknown network attacks. In: 2019 International Conference on Smart Applications, Communications and Networking (SmartNets), 2019; pp. 1–6. IEEE.
8. Lee B, Amaresh S, Green C, Engels D. Comparative study of deep learning models for network intrusion detection. SMU Data Science Review. 2018;1(1):8.
9. Wu P, Guo H, Buckland R. A transfer learning approach for network intrusion detection. In: 2019 IEEE 4th International Conference on Big Data Analytics (ICBDA), 2019; pp. 281–285. IEEE.
10. Moustafa N, Slay J. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), 2015; pp. 1–6 . IEEE.
11. Almomani IM, Alenezi M. Efficient denial of service attacks detection in wireless sensor networks. J Inf Sci Eng. 2018;34(4):977–1000.
12. 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.
13. Park T, Cho D, Kim H, et al: An effective classification for dos attacks in wireless sensor networks. In: 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), 2018; pp. 689–692. IEEE.
14. Abdullah MA, Alsolami BM, Alyahya HM, Alotibi MH. Retracted: intrusion detection of dos attacks in wsns using classification techniques. J Fundam Appl Sci. 2018;10(4S):298–303.
15. Premkumar M, Sundararajan T. Dldm: deep learning-based defense mechanism for denial of service attacks in wireless sensor networks. Microprocess Microsyst. 2020;79: 103278.
16. Asad M, Asim M, Javed T, Beg MO, Mujtaba H, Abbas S. Deepdetect: detection of distributed denial of service attacks using deep learning. Comput J. 2020;63(7):983–94.
17. Loukas G, Vuong T, Heartfield R, Sakellari G, Yoon Y, Gan D. Cloud-based cyber-physical intrusion detection for vehicles using deep learning. Ieee Access. 2017;6:3491–508.
18. Shaaban AR, Abd-Elwanis E, Hussein M. Ddos attack detection and classification via convolutional neural network (cnn). In: 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), 2019; pp. 233–238. IEEE.
19. Wazirali R, Ahmad R. Machine learning approaches to detect dos and their effect on wsns lifetime. CMC-Comput Mat Contin. 2021;70(3):4921–46.
20. Salmi S, Oughdir L. Cnn-lstm based approach for dos attacks detection in wireless sensor networks. Int J Adv Comput Sci Appl. 2022;13(4).
21. Deshpande S, Gujarathi J, Chandre P, Nerkar P. A comparative analysis of machine deep learning algorithms for intrusion detection in wsn. In: Security Issues and Privacy Threats in Smart Ubiquitous Computing, 2021; pp. 173–193. Springer.
22. Gunduz S, Arslan B, Demirci M. A review of machine learning solutions to denial-of-services attacks in wireless sensor networks. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015; pp. 150–155. IEEE.
23. Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST), 2019; pp. 1–8 . IEEE.
24. Sbai O, El boukhari M. Data flooding intrusion detection system for manets using deep learning approach. In: Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications, 2020; pp. 1–5.
25. de Assis MV, Carvalho LF, Rodrigues JJ, Lloret J, Proença ML Jr. Near real-time security system applied to sdn environments in iot networks using convolutional neural network. Comput Electric Eng. 2020;86: 106738.

26. Hussain F, Abbas SG, Husnain M, Fayyaz UU, Shahzad F, Shah GA. Iot dos and ddos attack detection using resnet. In: 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020; pp. 1–6. IEEE.
27. Shurman MM, Khrais RM, Yateem AA, et al. Dos and ddos attack detection using deep learning and ids. Int Arab J Inf Technol. 2020;17(4A):655–61.
28. Elsayed MS, Le-Khac N-A, Dev S, Jurcut AD. Ddosnet: A deep-learning model for detecting network attacks. In: 2020 IEEE 21st International Symposium On" A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 2020; pp. 391–396. IEEE.
29. Amaizu GC, Nwakanma CI, Bhardwaj S, Lee J, Kim D-S. Composite and efficient ddos attack detection framework for b5g networks. Comput Netw. 2021;188: 107871.
30. Cil AE, Yildiz K, Buldu A. Detection of ddos attacks with feed forward based deep neural network model. Expert Syst Appl. 2021;169: 114520.
31. Assis MV, Carvalho LF, Lloret J, Proença ML Jr. A gru deep learning system against attacks in software defined networks. J Netw Comput Appl. 2021;177: 102942.
32. Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A survey of network-based intrusion detection data sets. Comput Security. 2019;86:147–67.
33. Protić DD. Review of kdd cup '99, nsl-kdd and kyoto 2006+ datasets. Vojnotehnički glasnik/Military Technical Courier. 2018;66(3):580–96.
34. Amma NGB, Subramanian S. Vcdeepfl: Vector convolutional deep feature learning approach for identification of known and unknown denial of service attacks. In: TENCON 2018-2018 IEEE Region 10 Conference, 2018; pp. 0640–0645. IEEE.
35. Kasim Ö. An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks. Comput Netw. 2020;180: 107390.
36. Bhardwaj A, Mangat V, Vig R. Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of ddos attacks in cloud. IEEE Access. 2020;8:181916–29.
37. Ashoor AS, Gore S. Importance of intrusion detection system (ids). Int J Sci Eng Res. 2011;2(1):1–4.
38. Krishna VR, Subhashini R. Mimicking attack detection at hybrid level. EAI Endorsed Trans Energy Web. 2020;7(30):9–9.
39. Wang Q, Balasingham I. Wireless sensor networks-an introduction. Wireless sensor networks: application-centric design, 2010; 1–14.
40. Zhao Z, Xu K, Hui G, Hu L. An energy-efficient clustering routing protocol for wireless sensor networks based on agnes with balanced energy consumption optimization. Sensors. 2018;18(11):3938.
41. Zheng J, Jamalipour A. Introduction to wireless sensor networks. Wirel Sensor Netw Networking Perspect. 2009;1:1–18.
42. Kaplantzis S, Shilton A, Mani N, Sekercioglu YA. Detecting selective forwarding attacks in wireless sensor networks using support vector machines. In: 2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, 2007; pp. 335–340. IEEE.
43. Wood AD, Stankovic JA. A taxonomy for denial-of-service attacks in wireless sensor networks. Handbook of sensor networks: compact wireless and wired sensing systems. 2004;739:763.
44. Almomani I, Al-Kasasbeh B, Al-Akhras M. Wsn-ds: a dataset for intrusion detection systems in wireless sensor networks. J Sens 2016;2016.
45. Al-Shalabi M, Anbar M, Wan T-C, Khasawneh A. Variants of the low-energy adaptive clustering hierarchy protocol: survey, issues and challenges. Electronics. 2018;7(8):136.
46. Gholamy A, Kreinovich V, Kosheleva O. Why 70/30 or 80/20 relation between training and testing sets: a pedagogical explanation. 2018.
47. Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: a survey. Appl Sci. 2019;9(20):4396.

Publisher's Note