Measurement: Sensors 32 (2024) 101046

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Mitigation of attacks via improved network security in IOT network environment using RNN

Surendra Yadav [a,*], Hina Hashmi [b], Daxa Vekariya [c], Zafar Ali Khan N [d], Vijay Fidelis J [e]

[a] Department of Computer Science & Engineering, Vivekananda Global University, Jaipur, India
[b] Department of Computing Sciences & Information Technology, Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India
[c] Department of Computer Science and Engineering, Parul University, Vadodara, Gujarat, India
[d] School of Computer Science Engineering and IS, Presidency University, Bengaluru, India
[e] Department of MCA, Presidency College, Bengaluru, India

ARTICLE INFO

Keywords:
RNN
Attack mitigation
IoT
KDD datasets
Deep learning

ABSTRACT

Internet of Things (IoT) has become one of the emerging communication paradigms in recent years. Distributed denial of service (DDoS) attacks is becoming more common in IoT implying a growing need for security and authentication. Current advancements in IoT indicate fundamental changes in global communication infrastructures. Many facets of urban lives in smart cities are significantly impacted by increased interoperability of smart communication technologies. This paper proposes an IoT network-based threat mitigation strategy based on Recurrent Neural Network (RNN) algorithm. Using pre-processed and feature-extracted data, RNN seeks to categorise attributes associated with attacks. XBoost model selects features after the datasets have been pre-processed using min-max scaling technique. The simulations are executed over KDD datasets and assessed for the metrics of accuracy, precision, recall, and f-measure. The findings of this work demonstrated that in both training and testing datasets, the proposed RNN based schema achieves high degrees of classification accuracy.

1. Introduction

When it comes to wireless communications, the Internet of Things (IoT) is commonly considered as the next big thing [1] and as a paradigm shift that has the potential to radically transform the game. This is because IoT has the capacity to connect virtually every object in existence to the internet. Radio-frequency identification tags (RIFD), sensors, mobile phones, digital machines, and even humans all have the potential to be a part of the IoTs [2] since they are all provided with unique identifiers (UIDs) and can communicate data wirelessly over a network.

The IoTs has a wide variety of applications, and the number of such applications is only going to increase in the coming years [3]. The IoT is currently under attack by several malicious actors, who typically have one of two goals in mind: either to prevent consumers from accessing specific services or to seize control of the network itself. On the other hand, DDoS attacks are likely to be the most difficult obstacle to overcome for a system that is connected to the IoT [4]. A DDoS attack, more commonly referred to as a DDoS attack, is a sort of DOS attack in which the attack is conducted concurrently from several different locations throughout a network. This type of attack is frequently referred to as a DDoS attack. In an IoTs network, there will be a finite quantity of resources that can be accessed. When a DDoS attack is launched against a network, the network in question will begin to place a higher premium on answering the demands of the attack once the attack has been successfully launched. However, if the number of requests exceeds the maximum number that the network can process, the network will cease processing those requests once it reaches that point. Any request, regardless of whether it was made by authorised users, would be denied, which would result in an instant to the delivery of services related to the IoTs [5]. Most of the DoS and DDoS attacks rely on transmission control protocol (TCP) SYN floods or internet control message protocol (ICMP) Smurf attacks to bring down systems and networks. These protocols are used to communicate with computers over the internet.

Most denial of service (DoS) and DDoS attacks against IPv6 networks rely on the ICMPv6 protocol and make use of time series data in some manner. This is true even for the attacks that rely on other protocols. In addition, the Transmission Control Protocol, also known as TCP, and the

* Corresponding author.
E-mail addresses: surendra.yadav@vgu.ac.in (S. Yadav), hinahashmi170@gmail.com (H. Hashmi), daxa.vekariya18436@paruluniversity.ac.in (D. Vekariya), zafaralikhan@presidencyuniversity.in (Z.A.K. N), vijai-college@presidency.edu.in (V.F. J).

https://doi.org/10.1016/j.measen.2024.101046
Received 9 March 2023; Received in revised form 18 December 2023; Accepted 24 January 2024
Available online 13 February 2024

Internet Control Message Protocol, commonly known as ICMP, are also vital components of today and tomorrow networks, such as the IPv6 configuration. Both protocols are also known by their acronyms. Because of this, it is necessary for the safety of IoT networks to be able to identify attacks that are founded on TCP and ICMP [6]. IoT is a technology that has made it possible to run a network in a way that is both flexible and safe. Moreover, it has to respond quickly to changes that occur within the network. This has been made feasible by the system's capacity to respond quickly to network changes. With the use of IoT, suspicious traffic patterns can be discovered more rapidly due to the enhanced precision of detection given by IoT-enabled switches. This makes the identification of these patterns possible much more swiftly.

IoT can assist in the prevention of attacks on IoT devices and the dynamic flow control of those devices by stifling or blocking any traffic that may be suspicious. This can help prevent attacks on IoT devices. Identifying attacks on IoTs networks early can assist in limiting damages and isolating devices that have already been affected. If a form of attack that uses up a considerable amount of network bandwidth, such as a DoS or DDoS, is recognised while it is still in its early phases, the amount of bandwidth that is spent can frequently be decreased. In addition, the purpose of the IoT controller is not to carry out in-depth analyses of the traffic that is going across the network. The researchers will need to perform some sort of modification on the IoT controller for it to be more complicated and advanced [7].

Deep learning, also known as DL, has recently shown a great lot of promise in a variety of different fields, such as natural language translation, image processing, and facial recognition, to name just a few of these field applications. Instead, then relying on people to perform it, DL can automatically extract the underlying qualities from the information that has been collected. This contrasts with traditional data analysis methods. It can recognise correlations automatically even in data that has not been processed, which is one of the factors that contributes to its great performance. The percentage of successfully identified attacks has climbed to levels that have never been seen before thanks to the installation of DL-based models, which has resulted in the establishment of a new record. Deep learning algorithms, on the other hand, are often trained on the most fundamental kind of traffic, which is sequential in nature [8–15].

The Recurrent Neural Network (RNN) algorithm serves as the foundation for the design of a technique that we develop in this research to mitigate the effects of attacks on networks connected to the IoTs. The RNN is used to classify attack-related attributes into several different categories. The RNN performs an inputs analysis as well as an examination of the outcomes of the computations that were completed at the previous level at each level. These kinds of approaches to training models result in extremely little loss of data in comparison to other possible outcomes.

2. Related works

It is possible that the proliferation of IoTs devices in settings as disparate as supply chains, factories, municipal infrastructures, and private residences are largely attributable to seamless integrations of smart devices with the internet and communications between individual elements. This is largely attributable to the fact that IoT devices are now commonly found in environments such as these. The data that is provided by these devices as well as the optimised performance metrics are beneficial to a wide variety of different industries in a wide variety of different ways.

Some of them include the conservation of energy, the improvement of agricultural output, the monitoring of patient conditions, and the enhancement of the output of industrial devices. IoT devices are vulnerable to intrusion attacks, and they can also be used to execute high-volume DDoS attacks against other networks, including the cloud. These attacks can disrupt service and prevent users from accessing their data. Flooding a target network with requests for a service that the network is unable to supply is an example of a DDoS. The passwords that are utilised to provide security for various electronic devices almost always have a general application.

An attack known as Mirai utilised a method known as brute-force to infect thousands of devices by controlling these devices using existing password lists. This was accomplished by employing a technique known as dictionary attacks. However, before we get into the specific access points, we are going to talk about the vulnerabilities of IoT devices, which can be abused to construct a botnet.

Doshi et al. [14] proposed using machine learning approaches to monitor network traffic and guard systems against DDoS assaults in IoT environments and save them from complete destruction. Their suggested technique demonstrated the ability to identify local IoT node traffic linked to DDoS attacks with an accuracy of 0.99 using an approach that used very little or no CPU resources. IoT devices have low processing capacity, which means they are unable to successfully execute complex machine learning. The detection mechanism makes use of a detection strategy with a limited feature-set to reduce the amount of time that is wasted on processing that is not required. This occurs at the packet level. It was determined that the method of detection had an accuracy of 0.99 after being tested, and it was able to operate in real time on devices that were connected to the IoTs.

Kawamura et al. [15] recommended the introduction of NTP based DDoS attack detections for IoT environments. The clock on the device will eventually become out of sync with the time on the server because of the broad delays in processing that a DDoS attack creates across the system. This will happen because the DDoS attack produces widespread delays in processing. Once NTP client successfully establish connections with NTP servers, primary server clocks are utilised to compute deviations between clock timings of devices and primary servers. If the clock is unable to keep up with the server, it is quite likely that a distributed denial-of-service attack is being carried out. The existence of a time difference between the two can be used to draw this conclusion about the relationship between them.

McDermott et al. [16] proposed the usage of machine learning in detecting botnet formations on IoT-connected devices and networks using bidirectional long-short-term memory recurrent neural network (BLSTM-RNN). It is possible to recognise a DDoS attack within the system by performing an analysis of the patterns of traffic within the system. The proposed method lays a greater emphasis, in comparison to past methods of flow identification, on recognising text embedded inside characteristics. This contrasts with earlier methods of flow identification. Word embedding is a procedure that is used for the purpose of text recognition.

According to the findings of the investigation that was carried out by Nguyen et al. [17], there is a method that can potentially identify DDoS traffic at the edge gateway of an SDN. To carry out the task of collecting samples, several collectors have been placed in strategic locations across the switches. The data that is coming in is received by an IDS, which then performs additional inspections on it before sending it on. They proposed positioning the sample collector at the edge gateway to assist in preventing any malicious data from reaching its destination. As soon as the controller has done putting the updated network policy into effect, it will begin sending updated policy instructions to the collectors.

A Complex Event Processing Intrusion Detection System (CEPIDS) that could analyse traffic in real time was presented by Cardoso et al. [18]. The intrusion detection system (IDS) can be found right at the very edge of the network, which is also referred to as the perimeter of the network. To compile information regarding the actions taking place across the network, CEPIDS will make use of an event filter. The packet analyzer and the attack detection modules are both contained within the event processor. These modules oversee doing an analysis of the characteristics of the traffic packets to determine the type of attack that is being carried out against the network. It will be the responsibility of the CEP to communicate rules to the Action Engine, and the Action Engine will then notify the proper parties of the detrimental behaviour and shut

down the relevant services.

Yuan et al. [19] proposed employing a system named DeepDefense that was founded on Deep Learning to make DDoS detection more accurate. To identify malicious packets, a recurrent neural network, also known as an RNN, was utilised. To accomplish this, they came up with the concept of DDoS detection using a window and rethought the issue of DDoS detection based on packets as a challenge involving sequence classification. Because of its superior capacity for learning, RNN performed far better than other models available on the market when it came to representing the identification of fraudulent packets.

Mondal et al. [20] proposed using fuzzy logic as a strategy to defend against DDoS attacks in the cloud. The fuzzy logic system (FLS) must perform an evaluation on each incoming connection before allowing any of them to be utilised within the cloud. This system is in the network layer of the stack. The IF-THEN logic will be utilised by the fuzzy inference system to detect and report on abnormally high data flow to the cloud. The severity of the attack status will increase whenever a greater number of packets are obtained in a predetermined amount of time.

One method for identifying cyberattacks is the use of an intrusion detection system (IDS). Feature selection (FS) is necessary for reducing dimensionality of data and enhance the effectiveness of IDS. Most FS methods are based on limited objectives like data relevance or accuracy which are not sufficient and can be misleading in attack detections. DDoS assaults in IoT networks were detected in Ref. [21] using FS approach based on multi-objective optimisations. This work contributed to the development of an appropriate FS method. IDS also fail when incorrect features are selected. The study used non-dominated sorting algorithm with changed jumping gene operator for handling optimisation challenges. They subsequently used extreme learning machine as the classifier for FS based on six crucial goals in IoT networks. Their suggested approach reduced features by over 90% and achieved 99.9% success rate for FS, in their experimental findings. In terms of IDS's ability to detect DDoS assaults, their suggested technique performed better than other suggested FS methods.

The work in Ref. [22] detected invasions more precisely using an intelligent rule-based upgraded Multiclass Support Vector Machine classification method along with rule-and Multi-Objective PSO-based FS approach. It was evident from their experimental results which used the KDD'99 Cup and CIDD data sets for assessments, that their suggested IDS could identify intrusions with enhanced detection accuracy and reduced false positive rates.

Numerous jumping gene-based NSGA-II variants including NSGA–II–mJG, NSGA–II–saJG, NSGA–II–aJG, and NSGA–II–sJG have been studied [23]. The study in Ref. [24] concluded that NSGA–II–ajG outperforms other two methods in terms of computations and convergences. A semi-supervised learning-based DDoS attack detection technique for IoT networks utilising ELM classifier was presented in Ref. [25]. The method was evaluated using NSL-KDD and KDDCUP'99 datasets where it performed more accurately than centralised attack detection frameworks. Their schema reduced runtimes by 11 m s while maintaining maximized accuracy of 86.53%. Six key goals namely maximizing relevance, minimising redundancy, reducing features, maximizing classifier's values for accuracy, recall, and precision, were applied for jumping gene adaptations of NSGA-II in this work. The comparative results of existing methods are displayed as Table 1.

Table 1
Comparative analysis of the existing approaches.

Author	Approaches	Results	Drawbacks
Doshi et (2018)	IoT-specific network characteristics that guide feature selection, such as a restricted number of endpoints and consistent packet intervals	Achieving an accuracy of 0.99	High sensitivity to errors. While being autonomous, machine learning is quite error-prone. Assuming tiny data sets train an algorithm, it is not inclusive, resulting in a biassed training set and biassed predictions.
Kawamura et al. (2017)	Event detection modules for DDoS attacks on IoT	Achieves high recall and precision values,	The local clocks will continue to run but become less accurate over a period as they cannot synchronise with external time servers.
McDermott et al. (2018)	Detection model based on BLSTM-RNN	It emerges as a superior progressive model over a period of time.	It require more computational resources and memory than standard RNN
Yuan et al. (2017)	Deep learning based DDoS attack detection approach (DeepDefense)	The error rate reduced to 2.103% from 7.517% in comparisons with the usage of traditional machine learning approaches.	It also has some limitations, such as high computational cost, overfitting, lack of interpretability
Roopak et al. (2020)	Multi-objective-based feature selection for DDoS attack detection in IoT networks	The suggested approach achieved 99.9% success rate for FS and reduced features by around 90%.	Crowded comparisons can limit convergences. Non-dominated sorts on 2 N sizes.

3. Proposed method

To address the problem of multiclass imbalance, a machine learning-based IDS methodology has been appended, and an XGBoost-RNN method has been supplied for use with IoT datasets. Both methodologies can be utilised to solve the problem. Fig. 1 is an illustration of an example of an IoT high-level architecture. It starts with the IoT perception layer and then continues to the IoT network and processing layer as well as the IoT application layer.

Hence by analysing the existing approaches, to address the problem of multiclass imbalance, a machine and deep learning-based IDS methodology has been appended, and an XGBoost-RNN method has been supplied for use with IoT datasets.

Fig. 1 provides an illustration of the ML-based intrusion detection system in the context of an overall IoT network design. At the perception layer, the gathering of sensory data, the monitoring of the environment around them, and the delivery of raw materials are the fundamental functions of these devices.

Connectivity networks are the ones that oversee provision of perceptual information from IoTs devices positioned in network, processing, and perception layers as it is the application layer's responsibility to ensure that user specified criterion is satisfied. A solution for the application layer of an intrusion detection system network architecture that is based on machine learning is described in Fig. 1.

The XGBoost model is presented as one possible solution for the imbalanced multiclass classification that can be observed in the IoT-IDS datasets, as seen in Fig. 2. The preprocessing, the XGBoost-based IoT intrusion detection system, and the evaluation of the classification scheme are the three elements that are of the utmost significance.

Pre-processing: We began with the original TON IoT and X-IoTID datasets [26] and performed the following steps: normalised the features, encoded the labels, and divided the datasets into training and testing groups. The formula that follows demonstrates how the min-max scale approach was used as part of the procedure for normalising the characteristics. This formula is as follows:

$$X^N(i) = \frac{X^N(i) - \min X^N(i)}{\max X^N(i) - \min X^N(i)} \quad (1)$$

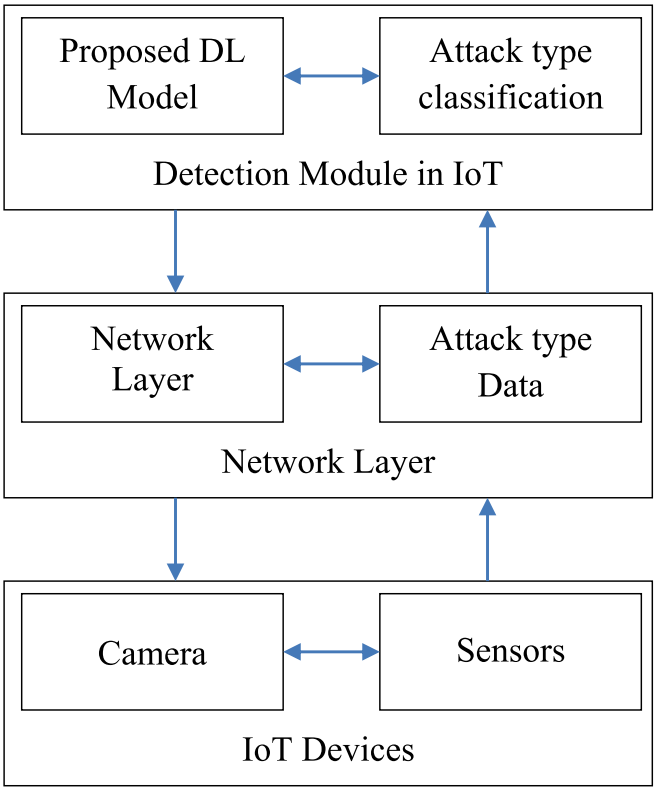

Fig. 1. ML-security model.

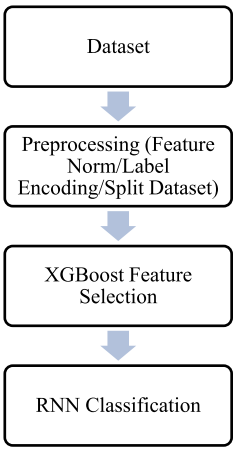

Fig. 2. XGBoost model.

The method of label encoding that we used to manage the multiclass output values made it possible for us to transform the data from a non-numerical kind into a numerical kind that the machine learning model could use for training purposes. This was accomplished by encoding the labels with the multiclass output values. An encoding was assigned to the target labels, which are denoted by Y. This encoding took the form of a number that varied from 0 to (n classes), where n classes stand for the entire range of potential values for Y.

We were able to get this result by utilising the LabelEncoder capability that is found in the Sklearn module of the Python programming language. The proposed approach utilised different quantities of sampling data for the training and testing operations; the total number of samples was established based on the ratio of training data to testing data that was utilised. The number of records in the training set compared to those in the test set was distributed in a proportion of 70:30.

There is currently available a solution for the identification of intrusions in the IoT that is based on XGBoost. To give just one example, XGBoost is a sequence model. The XGBoost model was selected because it possesses a wide variety of advantageous characteristics, such as the ability to improve itself by learning from its previous mistakes, the adaptability to fine-tune vast hyperparameters, the capacity to scale uneven data, and the ability to deal with null values. These are just a few of the many reasons why the XGBoost model was selected.

In order to rectify the errors that were produced by prior models, a sequential ensemble method known as boosting was developed. It has been established that the implementation of XGBoost can significantly improve the overall performance of weak learners in the context of classification and regression issues. An XGBoost tree can consider the tree historical prediction value for a particular set of input data to construct a new tree that maximises prediction gain when it is generated. This allows the tree to produce a new tree that is more accurate. In the first algorithm, we provide an explanation of the primary principle that lies at the IoT IDS datasets that are produced using XGBoost.

During the training phase, the installation of a new tree on an iterative basis corrects faults and residuals generated by earlier trees. These errors and residuals are caused by the trees that came before it. The algorithm compiles the data obtained from each of the preceding trees into a single prediction using the information obtained. The significance of the predictive equation value. The learning parameter is represented by A, the predicted value is designated byYp_T, and the weight prediction function is represented by the f_T.

$$Yp_T(X) = Yp_{T-1}(X)+\alpha^* f_T(X,w_T) \tag{2}$$

Where.

Yp_T-prediction output,

A - learning parameter, and

f_T - function for the weight w_T prediction.

During the entirety of the process of training, the model keeps a close check on the node loss and chooses a leaf node based on which has the biggest gain or loss. Every time, a new tree is added to the model by making use of a recently learnt function called ft to fit the residual of the forecast that came before it f_t (X,w_t). After training, T -trees are created, and within each leaf node of the tree is a score storage area. This score is proportionate to the data that the model was trained on.

The value that is predicted is determined by putting together all the scores that are associated with the various trees. If XGBoost is successful in locating a reasonable middle ground between the goals of complexity reduction and the object function, then problems caused by over-fitting, which can be avoided. XGBoost uses a regularisation term that is coupled to a Taylor expansion of the loss function up to the second order. This expansion goes all the way up to the second order. Thiscan be seenas a forecast that the XGBoost model provides is represented by Equation (3).

$$Y_{p_t} = \sum_{t=1}^{T} f_t(X_t) \tag{3}$$

where.

T - decision trees,

f_t (X_t) - input function of the tree, and.

Yp_t - predicted value.

The following equation provides an illustration of the two components that make up the training objective function in XGBoost, and those components are the training error and the regularisation.

$$X_{p_t} = \sum_{t=1}^{T} re(f_t) + \sum_{t=1}^{n} L(Y_i, Y_{p_t}) \tag{4}$$

where $\sum_{t=1}^{n} L(Y_i, Y_{p_i})$ is used to measure the difference between the predicted value and the real value of the loss function. $\sum_{t=1}^{T} re(f_t)$ is the

weak learner's regularisation term, and $re(f_t) = \gamma N + 0.5\lambda|s|^2$,

Where.

N - leaf nodes,

s - leaf node score,

γ - leaf penalty coefficient, and

λ - lesser leaf node score.

XGBoost takes use of a second-order Taylor equation to create an approximative representation of the step-tth object function. This is accomplished by using a Taylor expansion. The preceding t-1 prediction function, which is denoted by y head, can be interpreted as a variable representing the tth weakest learner, and $f_t(X)$ stands for the delta change.

$$X_{p_t} = \sum_{t=1}^{t} re(f_t) + \sum_{t=1}^{n} L\left(g_i f_t(X_i) + Y_i, Y_{p_{t-1}} + 0.5 h_i f_t^2(X_i)\right) \quad (5)$$

where.

g_i - first derivative gradient, and

h_i - second derivative hessian.

$$h_i = \partial^2 Y_{p_{t-1}} L\left(Y_i, Y_{p_{t-1}}\right) \quad (6)$$

$$g_i = \partial Y_{p_{t-1}} L\left(Y_i, Y_{p_{t-1}}\right) \quad (7)$$

Since the values of the t1-step prediction y head and all preceding t-1 regularisation are known at the current tth step, they are regarded as constants in the tth-step object function in the equation that was presented earlier. This is because the current tth step is the current step in the progression. The following is the result that we receive after deleting the constant terms (as there is no purpose for them in the process of optimising the object function), which is as follows:

$$X_{p_t} = re(f_t) + \sum_{i=1}^{n} \left(0.5 h_i f_t^2(X_i) + g_i f_t(X_i)\right) \quad (8)$$

The function known as the tree mapping function has the following definition:

$$I_j = \{i | q(X_i) = j\} \quad (9)$$

Where.

j - jth leaf;

I_j - data instances set that gets located in a jth leaf; and

q (x_i) – mapping function and obtains the index of the leaf:

$$f_t(X) = w_q(X) \quad (10)$$

Where.

W_i - ith score of the leaf.

f(X) - tree output.

We then apply a regularisation procedure to the object function as follows:

$$X(t) = \sum_{j=1}^{T} \left(\sum_{i \in I_j} g_i\right) w_j + 0.5 \left(\sum_{i \in I_j} h_i + \lambda\right) + \gamma T \quad (11)$$

where.

T - leaves in the

tth-weak learner and

λ and γ - regularisation hyperparameters.

The values of w_i, g_i, and h_i are already known to be accurate. This is because they are connected to the loss function as well as the values that were predicted at step t-1. To determine the value of w_i that will produce the lowest feasible value of the object function, we can make use of the following formula to accomplish this goal:

$$\partial_{wi} X_t = 0 \quad (12)$$

The selected loss function, the 1- and 2-order derivatives g_i and h_i are used, respectively, to determine the degree of similarity that exists between the output w_i that is generated by tree leaves and the tree nodes. This comparison is made so that the loss function can be optimised. A specific log loss function is the name given to the variety of loss function that is utilised in the process of classification.

$$L = Y_i \log(p_i) + (1 - Y_i) \log(1 - p_i) \quad (13)$$

Where.

p_i = sigmoid (Y_i) - class probability (output).

3.1. RNN classification

After the features have been fine-tuned, the RNN will be able to classify correctly either the attacking or the non-attacking labels. The addition of a memory state to the neurons in this network makes it functionally identical to a regular neural network, with the important distinction that it can store all data relevant to calculations for further use. This network name comes from the fact that it is functionally identical to a regular neural network.

RNN is a deep learning algorithm with a sequential approach. RNN maintains unique connections between hidden layers. It repeatedly duplicates hidden layers, applying same weights and biases to inputs at time steps. The networks execute looping procedures that update and modify concealed states while storing data in internal memory spaces. RNN construct models by training on appropriate data. They update and rebuild on every pass of time data. RNN works well with big datasets and is extremely easy comprehend data throughout the training process. It can handle both numerical and categorical data and requires minimum data preparations. RNN allows model validations using statistical tests [5,26]. RNN is a statistical technique that employs data cluster points in functional groups. Classification and clustering procedures get complicated and challenging in large datasets as there are more types of variables [26]. In addition, RNNs can store information in their neuron units by capturing changes, a distinct benefit of RNN in handling time series data. RNN can gather information from any lengths of sequences. When creating predictions, RNN considers word dependencies and sequential information found in inputs. RNN method for data series is depicted in Fig. 3. RNN's backpropagation capabilities include significant edges in data processing and in-depth analysis for concealed neural network layers.

Since the same operation is being performed on all the inputs or hidden layers (h) before the output is generated, the same parameters are used throughout the entirety of the process. This neural network, in contrast to many other neural networks, makes the parameters much simpler to comprehend. Because an RNN applies the same weights and biases to each layer, it can change independent activations into dependent ones. This is how it accomplishes this transformation. The need to adjust the settings and commit previous findings to memory is eliminated because of this, which makes the training process more efficient.

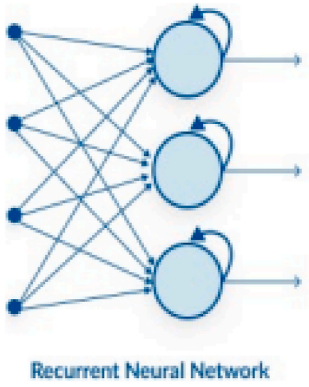

Fig. 3. RRN structure.

This implies that the weights and biases of all the hidden layers can be brought into consistency with one another by integrating all 3 levels into a single recurrent layer. It is possible to accomplish this by fusing all the layers together.

To categorise the attacks, an approach that involves training and testing is carried out. The RNN has already been trained to do classification, and the procedures that need to be carried out to train the RNN are outlined in more depth down below.

Find out what state it is in currently by combining a collection of recent input with the value that was recorded for its previous state. This will allow you to determine what state it is currently in. It is a known fact that it is possible to ascertain the current state by finding the solution to the equation.

$$h_t = f_t(h_{t\text{-}1}, i_t) \quad (14)$$

Where,

h_t – present state,

$h_{t\text{-}1}$ - input state.

$$h_t = \tan h(h_{t\text{-}1} W_{hh} + h_{it} W_i) \quad (15)$$

Where,

W_{hh} - recurrent weight, and.

W_{ih} - input weight.

Following the time step, the h_t that is now being utilised will be switched for $h_{t\text{-}1}$. The problem can be resolved by dividing it into as many phases as are necessary, and the data that is gathered during each of those stages can be linked to the data obtained during the other stages. After all the time steps have been completed, the output is computed by using the final state in the following manner to make use of it:

$$o_t = W_h o_{ht} \quad (16)$$

Where,

o_t - output, and.

W_{ho} - output weight.

The comparison of the actual output to the output that was planned is an essential part of the calculation of error.

$$E = o_A\text{-}o_t \quad (12)$$

The error will be back-propagated to the RNN once it has been ready for training, and then the weights will be adjusted accordingly. The results of the training session serve as a guide for conducting a test of the data values that are contained within each cluster. The result of the classification technique is the establishment of two categories, which are referred to as follows: standard data and attack data. These categories are the final outcome of the classification procedure.

4. Results and discussions

The test was conducted on a 64-bit version of Windows 10. NumPy and Panda are utilised throughout the processes of data cleansing and feature selection that we carried out, respectively.

4.1. Dataset description

The study uses a variety of systems through their paces by conducting analysis using a diverse collection of data sets. This study made use of data obtained from a variety of sources, one of which was a dataset referred to as the BoT-IoT. The environment was generated by the online traffic with traffic generated by botnets. Raw. pcap and comma-separated values are the two formats that can be used to access the data that was made public (CSV).

The term network packet capture is what the file extension. pcap refers to, and the Wireshark tool uses files with this extension quite frequently. This file is often consulted as part of the process of examining the peculiarities of networked data, and its contents are reviewed to do so. The files have been organised into numerous folders according to the various types of attacks and their associated subtypes, which will make it much simpler to classify them into the appropriate categories.

Fig. 4 Shows the performance comparison results for the proposed XGBoost-RNN and the existing FLS, CEPIDS and DeepDefense methods in terms of average accuracy for the CICIDS-2017 dataset. From the figure it is concluded that the proposed XGBoost-RNN produces the better average accuracy than other existing methods.

Performance comparison results for the proposed XGBoost-RNN and the existing FLS, CEPIDS and DeepDefense methods are shown in Fig. 5 interms of average precision for the CICIDS-2017 dataset. From the figure it is concluded that the proposed XGBoost-RNN produces the better average precision than other existing methods.

Average Recall metric performance comparison results forthe proposed XGBoost-RNN and the existing FLS, CEPIDS and DeepDefense methods are shown in Fig. 6 interms of average recall for the CICIDS-2017 dataset. From the figure it is concluded that the proposed XGBoost-RNN produces the better average recall than other existing methods.

Fig. 7. Shows the performance comparison results for the proposed XGBoost-RNN and the existing FLS, CEPIDS and DeepDefense methods interms of average f-measure for the CICIDS-2017 dataset. From the figure it is concluded that the proposed XGBoost-RNN produces the better average f-measure than other existing methods.

When evaluated with the CICIDS-2017 dataset, the proposed model was able to correctly categorise 99% of all DDoS attacks.

The findings of the experiments have demonstrated that the model that was developed is able to accurately detect distributed denial-of-service attacks with a high degree of precision. We compare the RLSTM model to the multilayer perceptron neural network, which is the most used method for machine learning, so that we can illustrate the RLSTM model effectiveness in feature extraction and intrusion categorization. MLP is the algorithm of choice for our business due to its adaptability in the process of identifying new connections, its facility in the generalisation of models, and its capacity for accurate prediction.

The classifications produced by the FLS algorithm were compared to the classifications produced by the CEPIDS model in the case of the NSL-KDD and CICIDS-2017 datasets. It was found that the FLS model that was built was outperformed by the CEPIDS model that was proposed in terms of accuracy as well as other metrics. The proposed model achieved superior results than its predecessors in the current body of research in terms of F-score, accuracy, precision, and recall. Previous models that were proposed in the literature may have yielded various outcomes

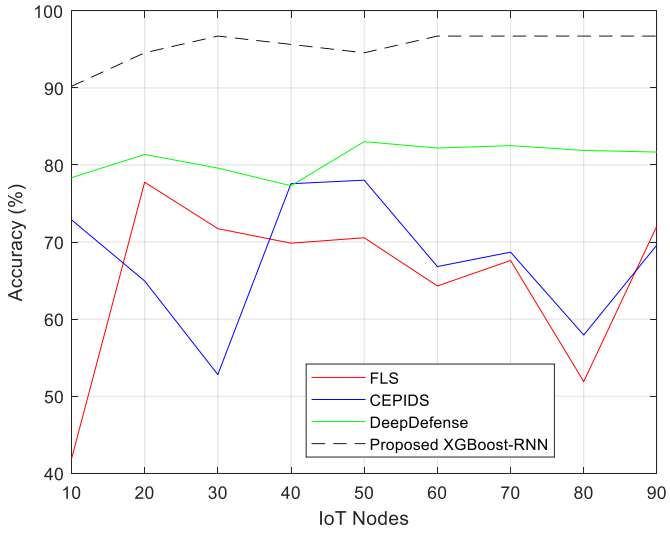

Fig. 4. Average accuracy.

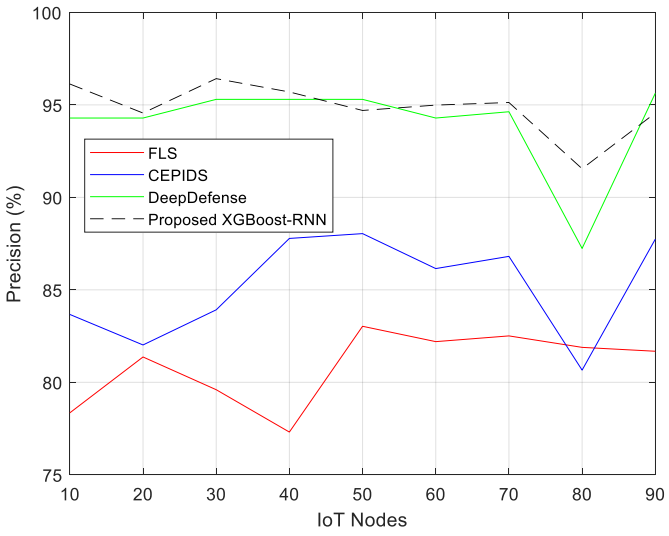

Fig. 5. Average precision.

Fig. 6. Average recall.

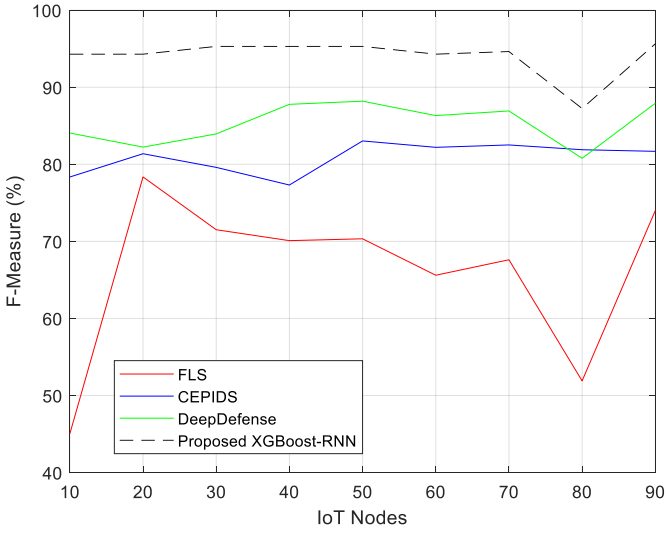

Fig. 7. Average F-Measure.

depending on the training parameters that were utilised and the classifiers that were put into operation. These factors were taken into consideration in this study. Despite this, the model that was proposed has a superior performance than the methods that were proposed in the published research, both in terms of the feature extraction it can accomplish and its scalability to a dataset.

5. Conclusions

In this paper, a secured authentication framework developed on IoT based RNN approach is proposed. The RNN categorise attack-related elements by drawing on data that has already been processed and features that have been retrieved from the data. This is done so that the RNN can fulfil its mission. The simulation is carried out with the use of a KDD dataset, and the outcomes of the simulation are analysed in terms of aspects like as accuracy, precision, recall, and f-measure. According to the findings, the RNN that was proposed is capable of attaining a high level of classification accuracy not only on the training dataset but also on the testing dataset.

In addition, the cyber-attacks in IoT systems can be identified using the deployment of deep learning algorithms. In the field of machine learning and deep learning detection approaches, researchers have investigated a wide range of malicious tactics. These include DoS and DDoS attacks, as well as probing, U2R and R2L attacks, botnets, spoofing, and MITM attacks. In addition, we offered detection datasets that had been developed through the application of machine learning and deep learning. To determine which strategies or procedures are the most effective at detecting these kinds of attacks, each of the learning methods was analysed and compared to one another in terms of the various types of attacks, feature selection methods, attack detection methods, and datasets. This was done to find out which approaches or procedures have the best success rate. Accordingly, it is recommended that semantic hierarchies be applied in future studies, usage of deep Long Short-Term Memory (LSTM) neural networks to improve attack detection processes and use reinforcement learning algorithms to improve DDoS attack detections in IoT.

Funding statement

Authors state no funding involved.

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

https://www.kaggle.com/general/155948

References

[1] O. Yousuf, R.N. Mir, DDoS attack detection in Internet of Things using recurrent neural network, Comput. Electr. Eng. 101 (2022) 108034.
[2] P. Bhale, D.R. Chowdhury, S. Biswas, S. Nandi, OPTIMIST: lightweight and transparent IDS with optimum placement strategy to mitigate mixed-rate DDoS attacks in IoT networks, IEEE Internet Things J. Vol. 10 (10) (2023) 8357–8370.
[3] M. Masood, Z. Anwar, S.A. Raza, M.A. Hur, EDoS Armor: a cost effective economic denial of sustainability attack mitigation framework for e-commerce applications in cloud environments, in: In Proceedings of the Multi Topic Conference (INMIC), 9–20 December 2013, pp. 37–42. Lahore, Pakistan.
[4] S. Roy, J. Li, B.J. Choi, Y. Bai, A lightweight supervised intrusion detection mechanism for IoT networks, Future Generat. Comput. Syst. 127 (2022) 276–285.
[5] S. Myneni, A. Chowdhary, D. Huang, A. Alshamrani, SmartDefense: a distributed deep defense against DDoS attacks with edge computing, Comput. Network. 209 (2022) 108874.
[6] M. Akshay Kumaar, D. Samiayya, P.M. Vincent, K. Srinivasan, C.Y. Chang, H. Ganesh, A hybrid framework for intrusion detection in healthcare systems using deep learning, Front. Public Health 9 (2022) 824898.

[7] M. Mayuranathan, S.K. Saravanan, B. Muthusenthil, A. Samydurai, An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique, Adv. Eng. Software 173 (2022) 103236.
[8] S.S. Sathiyadhas, M.C.V. Soosai Antony, A network intrusion detection system in cloud computing environment using dragonfly improved invasive weed optimization integrated Shepard convolutional neural network, Int. J. Adapt. Control Signal Process. 36 (5) (2022) 1060–1076.
[9] K. Thapa, N. Duraipandian, Malicious traffic classification using long short-term memory (LSTM) model, Wireless Pers. Commun. 119 (2021) 2707–2724, https://doi.org/10.1007/s11277-021-08359-6.
[10] T.H. Aldhyani, H. Alkahtani, Cyber security for detecting distributed denial of service attacks in agriculture 4.0: deep learning model, Mathematics 11 (1) (2023) 233.
[11] M.M. Jagtap, R.D. Saravanan, Intelligent software defined networking: long short term memory-graded rated unit enabled block-attack model to tackle distributed denial of service attacks, Transac. Emerg. Telecommun. Technol. 33 (11) (2022) e4594.
[12] R. Alghamdi, M. Bellaiche, A cascaded federated deep learning based framework for detecting wormhole attacks in IoT networks, Comput. Secur. 125 (2023) 103014.
[13] V. Ravi, R. Chaganti, M. Alazab, Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system, Comput. Electr. Eng. 102 (2022) 108156.
[14] R. Doshi, N. Apthorpe, N. Feamster, Machine learning ddos detection for consumer internet of things devices, in: 2018 IEEE Security and Privacy Workshops (SPW), IEEE, 2018, May, pp. 29–35.
[15] T. Kawamura, M. Fukushi, Y. Hirano, Y. Fujita, Y. Hamamoto, An NTP-based detection module for DDoS attacks on IoT, in: 2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), IEEE, 2017, June, pp. 15–16.
[16] C.D. McDermott, F. Majdani, A.V. Petrovski, Botnet detection in the internet of things using deep learning approaches, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, July, pp. 1–8.
[17] S.N. Nguyen, J. Choi, K. Kim, Suspicious traffic detection based on edge gateway sampling method, in: 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE, 2017, September, pp. 243–246.
[18] A.M. da Silva Cardoso, R.F. Lopes, A.S. Teles, F.B.V. Magalhães, Real-time DDoS detection based on complex event processing for IoT, in: 2018 IEEE/ACM Third International Conference on Internet-Of-Things Design and Implementation (IoTDI), IEEE, 2018, April, pp. 273–274.
[19] H.S. Mondal, M.T. Hasan, M.B. Hossain, M.E. Rahaman, R. Hasan, Enhancing secure cloud computing environment by Detecting DDoS attack using fuzzy logic, in: 2017 3rd International Conference on Electrical Information and Communication Technology (EICT), IEEE, 2017, December, pp. 1–4.
[20] X. Yuan, C. Li, X. Li, DeepDefense: identifying DDoS attack via deep learning, in: 2017 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2017, May, pp. 1–8.
[21] M. Roopak, G.Y. Tian, J. Chambers, Multi-objective-based feature selection for DDoS attack detection in IoT networks, IET Netw. 9 (3) (2020) 120–127.
[22] S. Subramani, M. Selvi, Multi-objective PSO based feature selection for intrusion detection in IoT based wireless sensor networks, Optik 273 (2023) 170419.
[23] Y.-L. Wan, J.-C. Chang, R.-J. Chen, et al., Feature-selection-based ransomware detection with machine learning of data analysis, in: 2018 3rd Int. Conf. On Computer and Communication Systems (ICCCS), 2018, pp. 85–88. Nagoya, Japan.
[24] M. Kumar, C. Guria, The elitist non-dominated sorting genetic algorithm with inheritance (i-NSGA-II) and its jumping gene adaptations for multi-objective optimization, Inf. Sci. 382 (2017) 15–37.
[25] S. Rathore, J.H. Park, Semi-supervised learning based distributed attack detection framework for IoT, Appl. Soft Comput. 72 (2018) 79–89.
[26] Moustafa, N. ToN_IoT and unsw15 Datasets. Available online: https://research.unsw.edu.au/projects/toniot-datasets (accessed on 3 April 2022)...