Mansoura Engineering Journal

Volume 48 | Issue 1 Article 3

April 2023

Proposed Mitigation Framework for the Internet of Insecure Things

Mahmoud M. Elgindy

Sally M. Elghamrawy
Prof. & head of Communications & Computer Engineering Department, MET academy, Mansoura, Egypt., sally@mans.edu.eg

Ali I. El-Desouky

ORIGINAL STUDY

Proposed Mitigation Framework for the Internet of Insecure Things

Mahmoud A. Elgindy [a], Sally M. Elghamrawy [b,*], Ali I. El-Desouky [c]

[a] Department of Computer Engineering, Faculty of Engineering, Mansoura University, Egypt
[b] Department of Communications & Computer Engineering, MET Academy, Egypt
[c] Department of Computer Engineering, Mansoura University, Mansoura, Egypt

Abstract

The purpose of this paper is design, implementation and evaluation of a framework consisting of deep learning intrusion detection suitable for nodes that are part of sensor networks in the IoT and it takes advantage of the characteristics of deep learning to obtain high accuracy compared to other designs. A sensor node with a small amount of battery life and processing power can construct an appropriate IDS framework and almost operate it effectively. The framework is suitable for different detection methods which can be combined. This also leads to different demands of energy consumption because of the changing complexity of the solutions and the combination of them. The main goal of the IDS framework is to detect the following types of attacks against the nodes: 1. Generic DoS/flooding attacks against nodes. 2. Sinkhole attacks against 6LoWPAN. 3. Selective forwarding attack against routing. 4. Wormhole attack against routing. This paper proposes a framework to improve the detection performance as BiLSTM classifier, which gives a high with less time. This paper presents a deep learning Bidirectional long-short term memory BiLSTM intrusion detection framework with Principal Component Analysis PCA (PCA-LSTM-IDS). By using MATLAB 2021a on PC of 16-GB RAM Intel® Core™ i5-2.90 GHz CPU with an NVIDIA Quadro M2000M GPU, the results of proposed framework are 99% accuracy with 4.5s for testing time. Attacks identification has been dependent on anomaly IDS, which utilizes PCA for feature extraction and high accuracy BiLSTM deep learning to identify unknown attacks with acceptable timing.

Keywords: Deep learning, Internet of things, Intrusion detection system, Long-short term memory, Security

1. Introduction

Internet of Things (IoT) is a complex system that connects all electronic devices to the internet so that they can interact with each other via many protocols (Chen et al., 2014). In this approach, anything can be accessed by anyone, anywhere, at any time. Applications for the IoT can range from a basic smart home appliance to a complicated system in a smart grid (Li et al., 2015). However, a number of issues have appeared during the development of IoT applications, including security, compatibility, connectivity, intelligent analysis, and standards. Conventional networks use protocols like IPv6 over Low-Power Wireless Personal Area Network, IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), IEEE 802.15.4 WPAN, and Constrained Application Protocol. IoT networks use different protocols (CoAP). For IoT intrusion detection system (IDS), protocol heterogeneity brings new problems and challenges by introducing new vulnerabilities.

IoT network may be subject to a variety of attacks, including probe attacks, denial-of-service (DoS) attacks, attacks from the root to the local machine (R2L), and attacks from the user to the root (U2R). Therefore, an IoT IDS that can more accurately identify attacks on heterogeneous networks must be designed. Given its capabilities to detect minor changes in complicated systems, deep learning technique is a potential choice for the problem's solution. The primary purpose of an IDS is to

Received 10 September 2022; revised 5 December 2022; accepted 13 December 2022.
Available online 14 April 2023

* Corresponding author at: Department of Communications & Computer Engineering, Misr Higher Institute for Engineering and Technology, Mansoura, 35511, Egypt. Fax: 0502160893
E-mail address: sally@mans.edu.eg (S.M. Elghamrawy).

https://doi.org/10.58491/2735-4202.3025

identify malicious activity across all nodes and networks. As the second layer of security, it can operate behind a firewall to guard the network from malware and intruders (Anand and Patel, 2012). For all of the network's sensor nodes and objects, intrusion is a threat, malicious, or offensive action. IoT has experienced a number of threats that can be divided into four primary categories: DoS, conventional, routing, and man-in-the-middle assaults. In the IoT context, there are numerous attack types for each category, including sybil attacks, DoS attacks, hello flood attacks, sinkhole attacks, and DoS attacks.

However, with the improvement of network, hardware, and software technologies in recent years, network activities have evolved into a more complex form of human–computer interaction, linked to many protocols and user activities.

IDS might be a software component or hardware. Both can inspect and examine devices, sensor nods, and user activity in an attempt to identify patterns or signatures of well-known attacks, predict undiscovered ones, and recognize any malicious network activity (Maharaj and Khanna, 2014). IDS's major goal is to keep track of all sensor nodes and networks, identify various kinds of network intrusions, and notify administrators when an intrusion is discovered. By sounding an alarm when it notices any intruder or attacker attempting to attack the network, it avoids damage to the systems and networks. IDS can also identify network packets and determine if they are legit or malicious. It contains three modules: monitoring, detection, and alarm (Alrajeh et al., 2013).

Depending on the manner of detection, there are three major types of intrusion detection techniques:

(1) Anomaly-based systems compare a system's behavior at any moment to a profile of normal behavior that is saved, and they raise an alarm whenever the divergence from the normal reaches a specified threshold.
(2) The signature-based technique compares the present signature patterns with previously stored patterns or signatures.
(3) Finally, hybrid IDS, which combines two of the aforementioned ways to have the benefits of both, contains some rules and thresholds that define the typical behavior for network objects such as sensors nodes, protocols, and routing tables.

A number of research studies have motivated detection techniques using IoT environment. On the contrary, some research studies have focused on a variety of storage techniques as follows: Habibi et al. (Habibi et al., 2017) used the principle of constructing two layers of IDS, which depends on working correctly. Jan et al. (Jan et al., 2019) proposed a lightweight machine learning depended on an SVM IDS by using three features only for classification. Tao et al. (Tao et al., 2018) presented an IDS depends on feature selection, weight, and optimization of a GA-SVM. Liu et al. (Liu et al., 2012) proposed an anomaly IoT detection model that depends on an artificial system that uses agents.

In this context, this paper proposes a framework to improve the detection performance as QSVM classifier, which gives a high accuracy and if a malicious attack was detected, it will be stored in signature database, and if it happened again, then our proposed framework will identify faster with higher accuracy.

This paper is structured as follows: the related work is presented in section II. Section III introduces the IDS's deep learning-based framework, followed by experimental results in section IV. The discussion and conclusion are presented in the V and VI sections, respectively.

2. Related work

Early on in the development of IoT IDS research, many research studies focused on conventional detection techniques that underperformed as expected in the IoT environment. Others concentrated on a variety of storage techniques, such as storing source and destination IP addresses and traffic sensitivity. To check the source and distention of any packet for IoT devices in whitelist databases and online associated domains, Habibi et al. (Habibi et al., 2017) utilized the principle of constructing two layers of IDS, which depends on working correctly. Despite the fact that this is a terrific idea, they did not employ machine learning to dynamically update the whitelist and instead relied on VirusTotal to identify new or unlisted threats.

Jan et al. (Jan et al., 2019) proposed a lightweight machine learning that depended on an SVM IDS by using three features only for classification. The provided results showed that the proposed IDS has reduced the time and increase the accuracy but it depends only on three features, which are mean, maximum, and median, so its accuracy is not enough for most attacks that perform like normal data.

Tao et al. (Tao et al., 2018) introduced an IDS that depends on feature selection, weight, and optimization of a GA-SVM. This proposed approach

selects the features first, then concurrently optimizes the SVM parameters. Finally, this trained classifier is applied to detection and classification of network anomalies. Liu et al. (Liu et al., 2012) suggested an anomaly IoT detection model that depends on an artificial system that utilizes agents. These agents were placed at multiple gateways to gather and exchange statistical data with a central service. Lopez-Martin et al. (Lopez-Martin et al., 2017) introduced a conditional variational autoencoder-based IDS (CVAE). As an additional input, the labels of the samples are applied to the decoder block of a VAE. Khalvati et al. (Khalvati et al., 2018) combined SVM with naïve Bayes classifiers for increasing the efficiency of detection and classification of intrusions in a network. Teng et al. (Teng et al., 2018) proposed a self-adaptive and collaborative IDS based on SVM and decision tree algorithms. The experimental results show that the decision tree algorithm gave a better overall performance with the SVM in aspects of detection accuracy and recall rate. Han et al. (Han et al., 2018) proposed an IDS that depends on the game theory with an autoregressive model; the proposed system focuses on the system's energy efficiency, and the results show that it is performing satisfactorily. The first three main types of IDS can be explained as follows.

Signature-based IDS: this technique is easy to implement because it only needs the patterns or signatures of an attack or malicious activity. It matches the network's existing profile and uses preset attack signatures or patterns. Each intrusion or attack can be defined based on preset patterns or signatures that are maintained in an existing database. This method cannot identify any new threats unless new signatures or patterns are manually added to the database. As a result, the database needs to continually be updated with new signatures or attack patterns (Patel and Aggarwal, 2013).

Anomaly-based IDS: this method is often referred to as event-based detection. This method examines network events to identify malicious activity. It first identifies the network's typical activity. The activity is therefore labeled as an incursion if it differs in any way from this behavior Amaral et al. This method uses a comparison between the current protocol specification state and a predetermined protocol state to identify malicious objects or nodes. In comparison to signature-based IDS, which cannot detect unknown attacks, this approach is more effective at detecting all varieties of attacks. Although high false-positive rates might affect anomaly detection, combining this technology with other appropriate optimized software solutions or artificial intelligence (AI) methodologies is still promising for developing intelligent detection models, such as using machine learning or deep learning to simulate human brain decisions through neurons and optimization techniques to implement an IDS.

Specification-based IDS: this is a technique that is a bit close to the anomaly detection technique. In this technique, the normal behavior of the network is specified manually, thus giving fewer false-positive rates. This method aims to apply optimum between signature-based and anomaly-based techniques by attempting to illustrate deviations from normal behavioral signatures that are created by any method of machine learning methods. One of the disadvantages of this technique is manual performance of developing the attacks and protocol specifications, which takes longer (Zarpelao et al., 2017). Several data sets are available, such as NSL-KDD, KDD'99, DARPA, CAIDA DDoS, and others, that provide samples of DoS attacks and other types of attacks in various scenarios. NSL-KDD was used for the proposed system principal component analysis (PCA)-LSTM-IDS and compared with other several algorithms.

An IDS is used to monitor the suspicious activity in a specific node or network. It can operate as the next layer of defense behind a firewall, protecting the network from intrusions (Anand and Patel, 2012). An intrusion is a threat or malicious or offensive activity that is harmful to all of the network sensor nodes and objects. Several threats happened in IoT, which are classified into four main categories as follows: DoS, conventional, routing, and man-in-the-middle attacks. There are many attack types for each category in the IoT environment such as black-hole attack, sink-hole attack, DoS attack, sybil attack, wormhole attack, selective forwarding attack, and hello flood attack. With the evolution in the network's software and hardware technologies in the latest years, however, network behavior has evolved into a more complex way, involving several protocols and user activities.

3. The proposed intrusion detection framework

This study investigates the integration of the anomaly detection concept with cutting-edge AI methods like bidirectional long-short term memory (BiLSTM). The originality of this paper is in the implementation of a more accurate and less time-focused IDS for IoT networks, which overcomes the difficulties we face in the IoT environment such as

different protocols from the normal internet and power consumption. Using the PCA technique, the complex features are identified and selected. Additionally, the dimensions of the input vector affect how complicated a BiLSTM classifier is. BiLSTM becomes more challenging as the hidden layers increase for more accuracy, which increase the overall time.

The most effective results were obtained by combining the strengths of both types of IDSs – signature-based and anomaly-based – into a hybrid system. True-positive rates (sensitivity) and accuracy are higher for signature-based methods. High accuracy is achieved while identifying anomalies using machine learning.

The two main layers of PCA-LSTM-IDS are features extraction layer and BiLSTM deep learning classification layer. Fig. 1 illustrates the PCA-LSTM-IDS architectural IDS's layout.

3.1. Layer 1: features extraction

A method called feature extraction creates a new, smaller set of features that mostly contains the critical information. This step is carried out using the PCA technique. Then, using the collected features and a vector of labels, the BiLSTM classifier is trained to categorize the unidentified samples in the test data set. From the test samples in the test data set, comparable features to those used in the training phase are extracted. The classifier then uses those unlabeled test samples as an input to predict the results.

PCA offers two separate techniques for extracting features, explained variance and number of components. Explained variance refers to obtaining all characteristics that represent all data and removing overlapping features. The term 'number of components' refers to the setting of a particular number of features for the presentation of all data. Dimensionality, or the number of features, can be lowered by reducing the number of principal components by altering the cumulative explained variance, such as by keeping only as many principal components as are necessary to obtain a cumulative explained variance of 90%. By reducing redundant information with minimal loss, PCA's key benefit is decreasing the dimension.

3.2. Layer 2: deep learning classification

This layer predicts if the data are normal or malicious. There are many deep learning algorithms, but for our model, we use BiLSTM, which is unsupervised learning algorithm that we designed just have two hidden layers.

3.2.1. Structure of long short term memory

LSTM is an advanced type of RNN which includes different cell structures and contains cell state. The three gates that comprise LSTM neurons are the forget gate, input gate, and output gate. Information importance is evaluated by the forget gate, cell state is refreshed by the input gate, and hidden state value is computed by the output gate. The neural network's component X evaluates the input value x_t and produces the output h_t, as seen in Fig. 2.

Beyond a loop, data can be transmitted from one network phase to the next. In a recurrent neural network, many replicas of the same network transfer data from one successor to another. This chain-like feature demonstrates how closely sequences, lists, and recurrent neural networks are related. The neural network is the more suitable architecture for use with such data, and LSTM depends on them.

The LSTM unit design is shown in Fig. 3, in which c_t, h_t, and x_t represent the memory state cell, the LSTM output at time t, and the input at time t, respectively. The logistic sigmoid function is expressed as σ, and the tanh represents the hyperbolic tangent function. In an LSTM, C_t and $C_{t\text{-}1}$ represent the memory units for the current time and the previous time, respectively. h_t and $h_{t\text{-}1}$ represent the hidden units for the current time and the previous time. i_t is the input gate for the current time, f_t is the forget gate, o_t is the output gate, and X_{nmt} represents the value of the n characteristic vector in the m time period of the t day. Following is a calculation of the LSTM network status:

$$f_t = \sigma\left(W_f[h_{t-1}, X_{nmt}] + b_f \right) \quad (1)$$

$$i_t = \sigma(W_i[h_{t-1}, X_{nmt}] + b_i \quad (2)$$

$$C'_t = tanh\left(W_c\left[h_{t-1}, X_{nm(t-1)}\right] + b_c \right. \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * C'_t \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, X_{nmt}] + b_o \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

where b_c, b_f, b_i, and b_o are the associated bias coefficients and W_c, W_f, W_i, and W_o are the weights of the memory unit, forget gate, input gate, and output gate, respectively.

3.2.2. Structure of bidirectional long-short term memory

BiLSTM is an enhanced variant of LSTM that can perform high-level abstraction and nonlinear

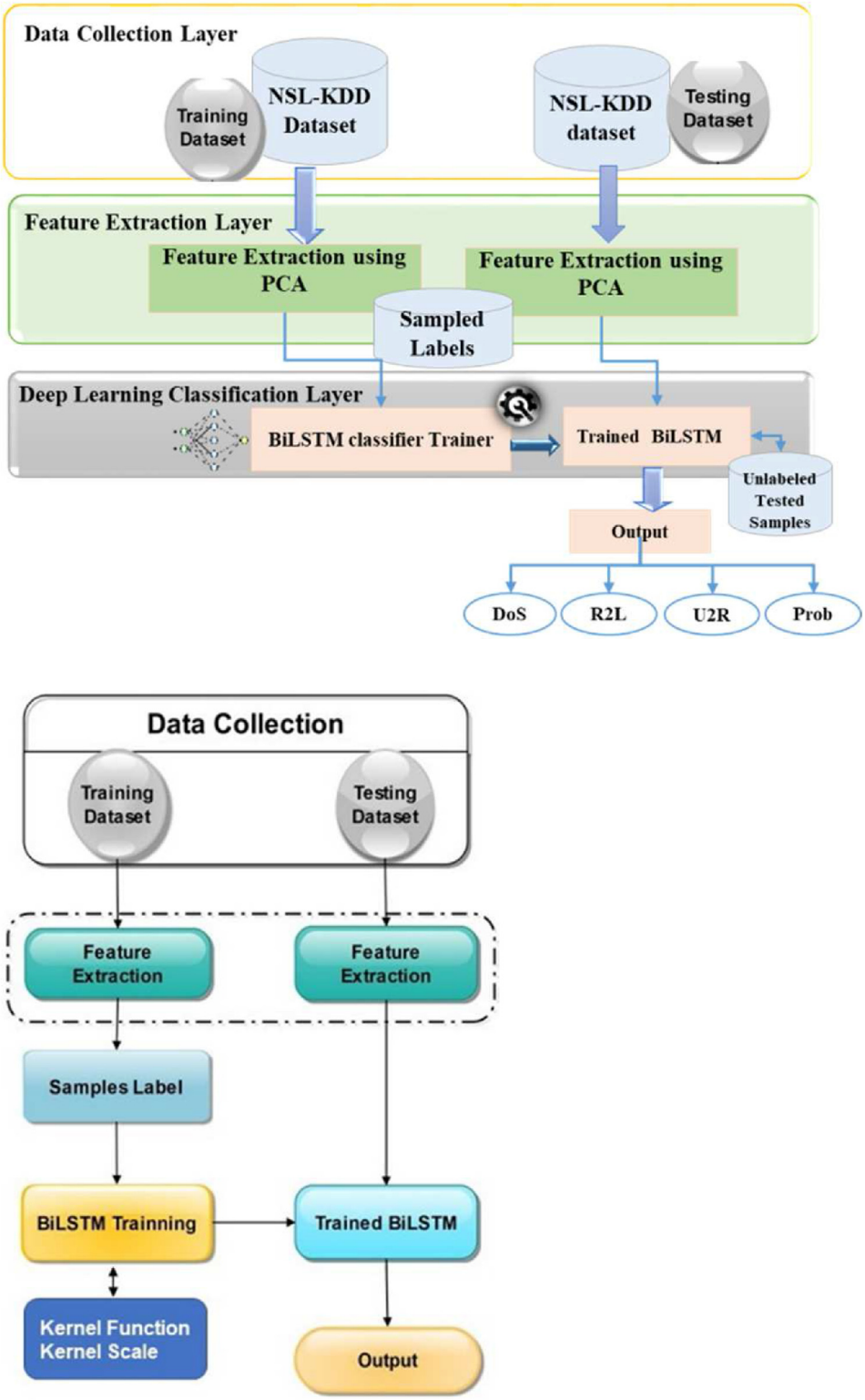

Fig 1. PCA-LSTM-IDS framework. IDS, intrusion detection system; LSTM, long-short term memory; PCA, principal component analysis.

transformation of intrusion data, evaluate two-way data information, and offer more fine-grained computation. It consists of many LSTM as backward and forward, which presents past and future, as shown in Fig. 4. The formula for calculating BiLSTM output is as follows:

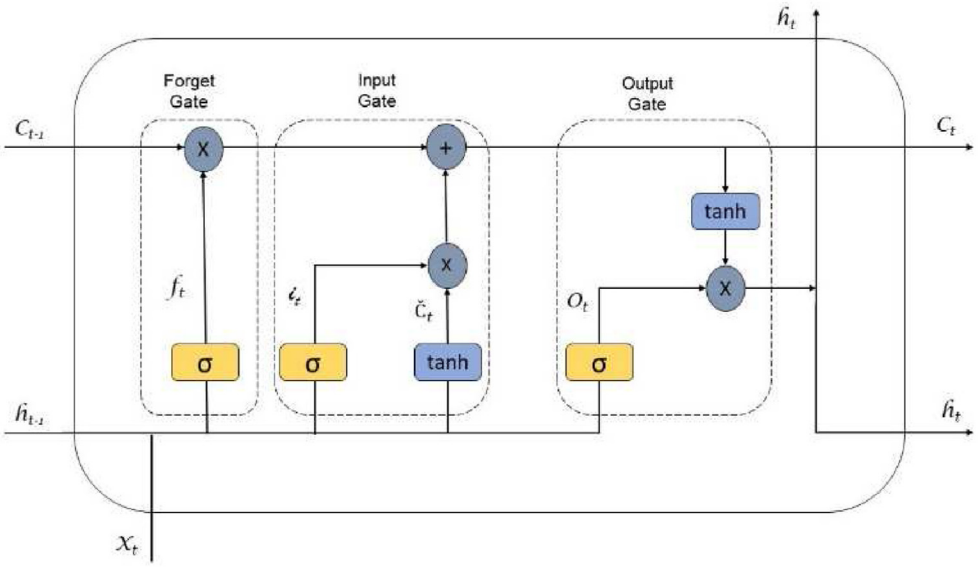

Fig. 2. Loop recurrent neural network.

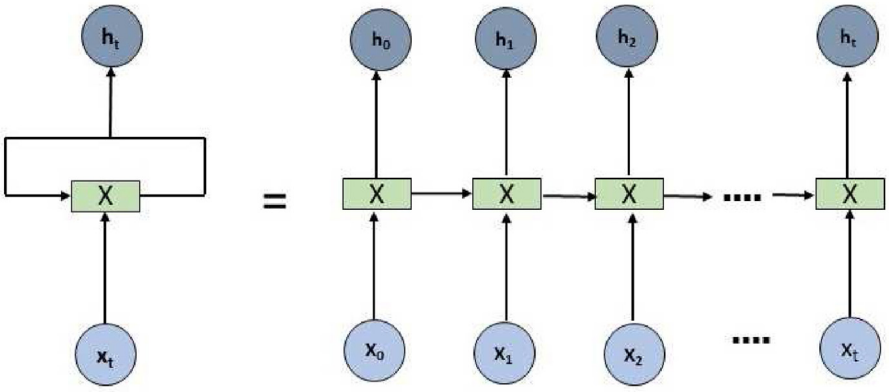

Fig. 3. LSTM architecture. LSTM, long-short term memory.

$$\overrightarrow{h}_t = f\left(\overrightarrow{W} \odot x_t + \overrightarrow{W} \odot \overrightarrow{h}_{t-1} + \overrightarrow{b}\right) \tag{7}$$

$$\overleftarrow{h}_t = f\left(\overleftarrow{W} \odot x_t + \overleftarrow{W} \odot \overleftarrow{h}_{t-1} + \overleftarrow{b}\right) \tag{8}$$

$$y_t = g\left(U\left[\overrightarrow{h}\,;\,\overleftarrow{h}\right] + c\right) \tag{9}$$

Where $\overrightarrow{W}$ and $\overleftarrow{W}$ refer to the parameters of the hidden layer, x_t refers to the input data, h_t and $\overrightarrow{h_t}$ refer to the output results of the LSTM layers at time t, $\overrightarrow{b}$ and b refer to the offset values, and finally y_t refers to the BiLSTM's output.

First, anomaly-based IDS extracts the features using the explained variance to establish a new set of the extracted features. This new set is then utilized for training and testing the BiLSTM classifier.

The training phase and the testing phase are the two stages of this layer process. During the training stage, a training data set with labeled samples is acquired. The input data were extracted using PCA to remove redundant features, and then the characteristics that were obtained were classified as normal or intruding data. After that, this new set of features will be prepared for BiLSTM training.

In the testing phase, a test data set with labeled samples is established, followed by extracting the features and sending it to trained BiLSTM classifier to predict the final result.

4. Experimental results

MATLAB, version 2021, a simulation tool, is used to implement all the experiments and data gathering procedures on a PC with a Core i5 and

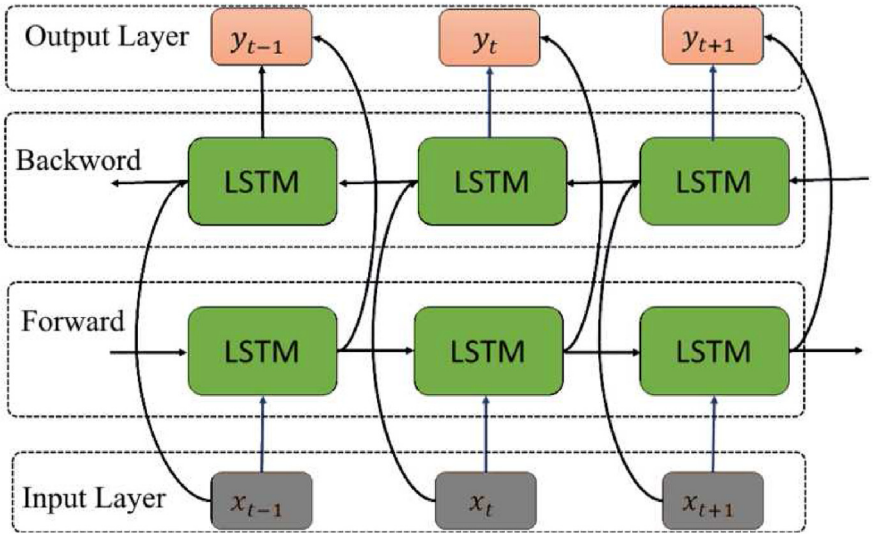

Fig. 4. BiLSTM architecture. BiLSTM, bidirectional long-short term memory.

12 G RAM. The experiments make use of the NSL-KDD data set. It is initially transformed into a prepared file of the type comma separated values. The experiment is then performed after it has been imported into MATLAB.

The experimental results demonstrated the proposed framework's performance, particularly in terms of accuracy. These are the paper's main contributions:

(1) PCA feature extraction is a useful method for reducing feature complexity.
(2) The performance of intrusion detection is enhanced as a result of a novel method for creating an intelligent anomaly IDS using BiLSTM with PCA.

We implemented our method using the NSL-KDD data set with 41 features once and 30 features once again using PCA. The results of the experiments showed that PCA-LSTM-IDS can achieve 99% accuracy, which is better than any other method.

4.1. Performance metrics

These are the definitions of the performance evaluation criteria used in this research. As stated in Table 1, which is used to determine the accuracy, the Confusion Matrix is used to measure the effectiveness of the suggested IDS and also the performance of the BiLSTM classification model.

The ensuing variables were computed:

True positive (TP): the volume of harmful codes that were accurately discovered.

True negative (TN): the volume of accurately detected benign codes.

False positive (FP): the number of benign codes that were falsely detected as harmful by a classifier.

False negatives (FN) are occurrences where harmful code was mistakenly identified as benign code by a classifier.

Total accuracy (ACC) is the percentage of the observation's numbers to the total sum of true-positive and true-negative values. The following equation can be used to determine the accuracy:

$$\mathrm{ACC} = (\mathrm{TP} + \mathrm{TN}) / \mathrm{N} \tag{10}$$

N: the total number.

4.2. Data set

Using the NSL-KDD data set, training and testing were implemented. The NSL-KDD data set consists of train, test, and validation data. Four different types of attacks, which are DoS, probe, U2R, and R2L, along with 23 different types of attack, are illustrated by the training data. The branch type of each attack type is listed in Table 2 along with the specifications of all four attack classes.

NSL-KDD's records each have 42 features. Typically, there are three types of features: continuous, discrete, and symbolic, each of which has a distinct range of values.

DoS attack prevents users from accessing a machine or network resource by making some memory

Table 1. Confusion matrix.

Types	Predicted normal	Predicated attack
Actual normal	TP	FN
Actual attack	FP	TN

FN, false negative; FP, false positive; TN, true negative; TP, true positive.

Table 2. NSL-KDD classification attacker types.

Class	Types
DoS	Smurf Land pod teardrop Back nepton
R2L	ftp_write guess_passwd imap multihop phf spy warezclient warezmaster
U2R	Rootkit perl loadmodule buffer_overflow portsweep
Prob	Ipsweep nmap satan

DoS, denial-of-service; R2L, root to the local machine; U2R, user to the root.

resources or network channels too full or busy to handle authorized requests like land, neptune, or pod.

R2L: in this technique, an attacker sends packets to a device across the network and takes use of the current vulnerabilities to obtain local access to that system without having an account or privileges on remote devices. Ftp write, the dictionary, guess passwd, and Imapt are some examples.

U2R: in this form of attack, the attacker first gains access to a user's local account on the system before using the vulnerabilities to take control of the system as the root user. Fdformat and Loadmodule are two examples.

In a probing attack, an attacker explores the network's computer systems to obtain data or identify known vulnerabilities. Attackers who have gained access to the network have compiled a list of the devices and services that can be used to exploit the system. Nmap, Ipsweep, and Satan are some examples.

4.3. Experiment: anomaly bidirectional long-short term memory intrusion detection system's validation

A BiLSTM classifier was used with PCA with 100% variance, which decrease the number of the features from 41 to 30, for extracting features to compare accuracy, prediction speed, and training time.

It is observed that using linear SVM gives high accuracy with 94.914% and acceptable prediction time. If a genetic algorithm is used, it gives a higher prediction time compared with other algorithms, which is 20.142 s, and the accuracy is lower than any other algorithm, as shown in Table 3 and Fig. 5.

As shown in Figs. 6 and 7, our approach has less time as it has only 397.304 and 4.548 s for training and testing data, respectively, with a higher accuracy among all other algorithms.

Table 3. Anomaly result for several classification algorithms.

Algorithm	Accuracy %	Training time (s)	Testing time (s)
SVM	94.914	620.312	16.542
Polynomial SVM	91.822	674.914	12.541
Genetic algorithm	85.731	718.048	20.142
PCA-LSTM-IDS	99.348	397.304	4.548

IDS, intrusion detection system; LSTM, long-short term memory; PCA, principal component analysis.

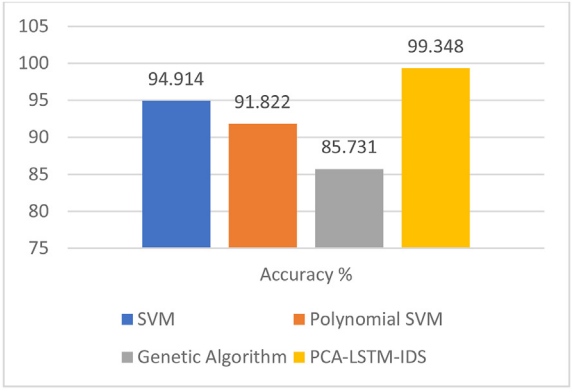

Fig. 5. Accuracy for PCA-LSTM-IDS with other algorithms. IDS, intrusion detection system; LSTM, long-short term memory; PCA, principal component analysis.

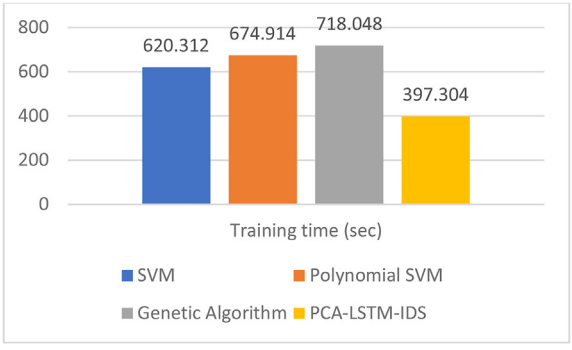

Fig. 6. Training time for PCA-LSTM-IDS with other algorithms. IDS, intrusion detection system; LSTM, long-short term memory; PCA, principal component analysis.

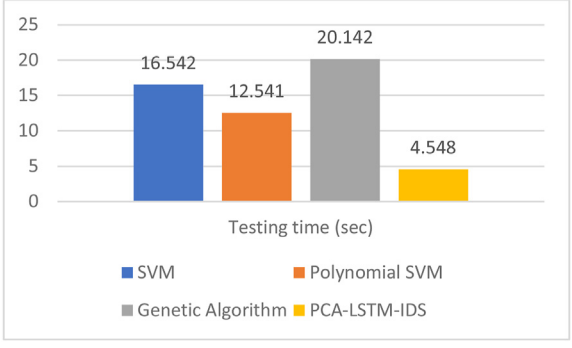

Fig. 7. Testing time for PCA-LSTM-IDS with other algorithms. IDS, intrusion detection system; LSTM, long-short term memory; PCA, principal component analysis.

Algorithm 1 training process
Input
Data set
Kernel_function Kernel_scale Cross_validation
Output
BiLSTM: trained classification model
STEPS
X ← extract features from data using PCA
Xlabeld ← label normal and intruder data
BiLSTM ← train BiLSTM model using Kernel_function, cross_validation

5. Discussion

The benefit of utilizing PCA for feature extraction before BiLSTM in anomaly IDS is to lower the number of features since doing so takes less time than using it for up to 42 features, which increases system processing time.

A recurrent neural network used mainly for natural language processing is called BiLSTM. Compared with traditional LSTM, BiLSTM provides greater accuracy as we can get the result from two directions, which are forward and backward for future and past.

A training data set with labeled samples is obtained during the training phase. PCA extracted the input data to remove unnecessary features and then selected features were labeled as normal or intruder data. Then selected features were processed to the training algorithm which is QSVM with default kernel function and scale with five folding cross-validations, as shown as follows:

5.1. Conclusions

The IoT is a cutting-edge technology that may be applied to a wide range of applications, from simple home automation systems to complex grid networks like smart grids. However, the reliability of these complex networks is compromised by their vulnerability to major attacks. Additionally, the nodes' limitations, which include memory, computing power, and battery life, have an impact on network security.

The issue with data stream deep learning applications is that their developers always think about how to submit applications with the highest accuracy possible while using the least amount of processing power. When using a neural network, this is challenging to accomplish because accuracy improves with the higher number of hidden layers which lead to takes longer time to implement.

Attacks identification has been dependent on anomaly IDS, which utilizes PCA for feature extraction and high accuracy BiLSTM deep learning to identify unknown attacks with acceptable timing.

Conflict of Interest

There are no conflicts of interest.

Authors contribution

(1) Conception or design of the work: Mahmoud M. Elgindy (30%), Sally M. Elghamrawy (60%), and Ali I. El-Desouky (10%). (2) Data collection and tools: Mahmoud M. Elgindy (32%), Sally M. Elghamrawy (58%), and Ali I. El-Desouky (10%). (3) Data analysis and interpretation: Mahmoud M. Elgindy (40%), Sally M. Elghamrawy (50%), and Ali I. El-Desouky (10%). (4) Funding acquisition: no funding. (5) Investigation: Mahmoud M. Elgindy (38%), Sally M. Elghamrawy (52%), and Ali I. El-Desouky (10%). (6) Methodology: Mahmoud M. Elgindy (25%), Sally M. Elghamrawy (65%), Ali I. El-Desouky (10%). (7) Project administration: Mahmoud M. Elgindy (28%), Sally M. Elghamrawy (55%), and Ali I. El-Desouky (17%). (8) Resources: Mahmoud M. Elgindy (40%), Sally M. Elghamrawy (65%), and Ali I. El-Desouky (5%). (9) Software: Mahmoud M. Elgindy (55%), Sally M. Elghamrawy (38%), and Ali I. El-Desouky (7%). (10) Supervision: Mahmoud M. Elgindy (20%), Sally M. Elghamrawy (60%), and Ali I. El-Desouky (20%). (11) Drafting the article: Mahmoud M. Elgindy (35%), Sally M. Elghamrawy (50%), and Ali I. El-Desouky (15%). (12) Critical revision of the article: Mahmoud M. Elgindy (25%), Sally M. Elghamrawy (50%), and Ali I. El-Desouky (25%). (13) Final approval of the version to be published: Mahmoud M. Elgindy (32%), Sally M. Elghamrawy (55%), and Ali I. El-Desouky (13%).

References

Alrajeh, N.A., Khan, S., Shams, B., 2013. Intrusion detection systems in wireless sensor networks: a review. Int. J. Distributed Sens. Netw. 2013, 167575.

Amaral JP, Oliveira LM, Rodrigues JJPC, Han G, Shu L. Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks, 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 2014, pp. 1796-1801

Anand, A., Patel, B., 2012. An overview on intrusion detection system and types of attacks it can detect considering different protocols. Int. J. Adv. Res. Comput. Sci. Software Eng. 2, 8.

Chen, Si, Xu, H., Liu, D., Hu, B., Wang, H., 2014. A vision of IoT: applications, challenges, and opportunities with China perspective. IEEE Internet Things J. 1, 4.

Habibi, J., Midi, D., Mudgerikar, A., Bertino, E., 2017. Heimdall: mitigating the internet of insecure things. IEEE Internet Things J. 4, 968–978.

Han, L., Zhou, M., Jia, W., Dalil, Z., Xu, X., 2018. Intrusion detection model of wireless sensor networks based on game theory and an autoregressive model. Inf. Sci. 476, 491–504.
Jan, S.U., Ahmed, S., Shakhov, V., Koo, I., 2019. Toward a lightweight intrusion detection system for the internet of things. IEEE Access 7, 42450–42471.
Khalvati, L., Keshtgary, M., Rikhtegar, N., 2018. Intrusion detection based on a novel hybrid learning approach. J. AI Data Mining 6, 157–162.
Li, S., Xu, L.D., Zhao, S., 2015. The internet of things: a survey. Springer Inform. Syst. Front. 17, 243–259.
Liu, C.M., Yu Chen, S., Zhang, Y., Chen, R., Guo, K.L., 2012. An IoT anomaly detection model based on artificial immunity. Adv. Mater. Res. 424, 625–628.
Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J., 2017. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT. Sensors 17, 1967.
Maharaj, N., Khanna, P., 2014. A comparative analysis of different classification techniques for intrusion detection system. Int. J. Comput. Appl. 95 (17).
Patel, M., Aggarwal, A., 2013. Security attacks in wireless sensor networks: a survey. In: International Conference on Intelligent Systems and Signal Processing (ISSP).
Tao, P., Sun, Z., Sun, Z., 2018. An improved intrusion detection algorithm based on GA and SVM. IEEE Access 6, 13624–13631.
Teng, S., Wu, N., Zhu, H., Teng, L., Zhang, W., 2018. SVM-DT-based adaptive and collaborative intrusion detection. IEEE/CAA J. Autom. Sinica 5, 108–118.
Zarpelao, B.B., Miani, R.S., Kawakani, C.T., de Alvarenga, S.C., 2017. A survey of intrusion detection in Internet of Things. J. Netw. Comput. Appl. 84, 25–37.