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

Ingénierie des Systèmes d'Information
Vol. 25, No. 5, October, 2020, pp. 569-577
Journal homepage: http://iieta.org/journals/isi

A Semi-supervised Deep Auto-encoder Based Intrusion Detection for IoT

Samir Fenanir[1*], Fouzi Semchedine[2], Saad Harous[3], Abderrahmane Baadache[4]

[1] Department of Computer Science, Faculty of Exact Sciences, University of Bejaia, Bejaia 06000, Algeria
[2] Mechatronics Laboratory - E1764200, Optics and Precision Mechanics Institute, University of Sétif 1, Sétif 19000, Algeria
[3] College of Information Technology, United Arab Emirates University, Abu Dhabi 15551, UAE
[4] Mathematics and Computer Science Department, University of Algiers 1, Algiers 16000, Algeria

Corresponding Author Email: samir.fenanir@univ-setif.dz

https://doi.org/10.18280/isi.250503

Received: 29 June 2020
Accepted: 17 September 2020

Keywords:
access control, anomaly detection, autoencoder, intrusion detection system, machine learning

ABSTRACT

The main problem facing the Internet of Things (IoT) today is the identification of attacks due to the constrained nature of IoT devices. To address this problem, we present a lightweight intrusion detection system (IDS) which acts as a second line of defense allowing the reinforcement of the access control mechanism. The proposed method is based on a Deep Auto-Encoder (DAE), which learns the pattern of a normal process using only the features of the user's normal behavior. Whatever deviation from the expected behavior is considered an anomaly. We validate our approach using two well-known network datasets, namely, the NSL-KDD and CIDDS-001. The experimental results demonstrate that our approach provides promising results in terms of accuracy, detection rate and false alarm rate.

1. INTRODUCTION

In recent years, the Internet of Things, which connects various things to Internet through gateways, including Radio Frequency Identification (RFID), sensors, mobile phones, etc., has become one of the prominent research topics. The IoT is seen as an extension of the Internet in the physical world, which covers a wide range of applications and touches many areas we face in our daily lives.

While the IoT offers an impressive set of benefits, it also presents a set of challenges. A major challenge for the IoT is to be able to manage the heterogeneity of objects coupled with multitudes of applications and users in terms of security service. Indeed, how to ensure the individual authentication of several million heterogeneous objects equipped with heterogeneous communication technologies. In addition, the components of the IoT are characterized by low resources in terms of computation and energy capacity. Therefore, proposed solutions should pay particular attention to resource efficiency in addition to obvious scalability issues. On the other hand, IoT security remains a great challenge and a major concern. Actually, the open nature of IoT makes it more vulnerable to both insider and outsider attacks [1]. These attacks aim to gain unauthorized access and over-privileged behaviors to compromise the user's security and privacy [2]. Denial of service (DoS) attack is one of the major important threats that prevent legitimate user from accessing and using the desired resources [3]. Moreover, DoS attacks on sensitive application like healthcare can affect their services, resulting in serious situations problems [4].

Securing IoT has become a fundamental issue due to the relevant information exchanged through the IoT networks. To address this issue, the intrusion detection system is widely deployed as a second block of defense when the access control fails [5]. It aims at detecting intrusions. Generally, the IDSs are categorized into two general approaches: misuse and anomaly detection [6]. Misuse-based IDSs detect the known attacks, by comparing new data with predetermined signatures of known attacks [7]. Anomaly-based IDSs compare the current profiles against observed behavior to identify any deviation [8]. The user profiles are learned using machine learning techniques.

In this paper, we present an anomaly-based IDS in the IoT environment using deep learning algorithms. Deep learning has gained a great interest in recent years in various domains, such as fraud detection, speech recognition, computer vision, etc. Moreover, deep learning has been applied to intrusion detection in a supervised, unsupervised and semi-supervised way [9]. Deep autoencoder is a kind of deep learning model, which has been used in various applications including automatically extracting features and some classification problems.

The objective of this study is to develop an intrusion detection system for the IoT environment. The IDS acts as a second block of defense, in addition to the access controller, which forms a protective layer necessary to effectively identify intruders. This kind of system must be efficient and suitable to the limited capacity and the heterogeneity of IoT devices. For this purpose, we presented a centralized architecture of an IDS that is composed of two defense blocks. The first one allows controlling the access to the network and blocking the external and internal attacks while the second allows reinforcing the access control mechanism.

In our experiments, we considered the intrusion detection task as a Deep AutoEncoder (DEA) binary classification; it is essentially a semi-supervised trained technique that uses user profiles to train the model. We evaluated our approach using TensorFlow by analyzing two benchmark datasets: NSL-KDD and CIDDS-00, and obtained promising results in classifying the attacks accordingly. We are aware of the limitations of

these datasets, but among similar works, they remain widely used benchmarks, which allow us to make direct comparisons. Our DEA based on the threshold method gave the best results on the CIDDS-00 dataset: 97% of accuracy, 90% of precision, 100% of recall, and 100% of UAC with a threshold value of 0.07. In addition, our approach is lightweight in terms of computation time and complexity. It provides better classification results than other common classifiers, as we will show in section 5.5.

The organization of this paper is as follows: Section 2 discusses related work. Section 3 presents briefly the technologies involved in the IoT. Section 4 details the proposed approach. The experimental results of our system are discussed in Section 5. Section 6 concludes the paper.

2. RELATED WORK

Various deep learning techniques have recently been developed in the fields of IoT and IDS in order to provide the best protection system. In this section, we discuss some works.

Works based on the supervised learning technique:

Ma et al. [10] proposed an intrusion classifier, which blends two techniques (spectral clustering and deep neural network (DNN)). The proposed method gives good performance on the KDDCUP99 and NSL-KDD datasets. However, the limit of this approach is that its cluster parameters are not set automatically. Pongle and Chavan [11] proposed an intrusion detection system for the IoT, which uses the location information of the nodes and their neighbors to identify the wormhole attack. The proposed method is suitable for resource-constrained environments with a detection rate of 94 %. Verma and Ranga [12] studied Anomaly-based IDS to detect DoS attacks in an IoT environment. To evaluate the performance of this approach, seven machine learning classification algorithms and a Raspberry Pi were tested on the NSL-KDD and CIDDS-001 dataset. Roy et al. [13] presented an IDS based on a deep learning technique to detect attacks for the IoT environment, using Bi-directional LSTM recurrent neural network. A binary classification (normal and attack) was adopted. UNSWNB15 dataset was used to evaluate the model, which achieved more than 95 % accuracy in detecting attacks. Roopak et al. [14] proposed a deep learning based IDS for IoT. The classification mode was identified through comparison between various deep learning algorithms. The hybrid CNN+LSTM model was selected owing to its outstanding performance, which could reach 95% accuracy.

Works based on the semi-supervised learning technique:

Sharmila and Satish [15] Proposed a semi-supervised learning algorithm for intrusion detection using a boosting framework. The proposed method aims to reduce the false alarm rate and improve the detection rate of IDS. The main advantage of the proposed method lies in its ability to solve the problem of labeled data unavailability. The experiment is carried out with the KDD CUP 99 dataset. The results show that the problem of the unavailability of labeled data can be solved using semi-supervised learning. Wenjuan el al. [16] presented a semi-supervised disagreement-based learning algorithm for the intrusion detection systems. The evaluation is performed using data sets and in real IoT network environments, with the aim of improving detection rate and reduction of false alarms. The experimental results show good performance in detecting intrusions compared to traditional supervised classifiers by automatically exploiting unlabeled data. However, its effectiveness is unknown in the case of large samples. Moreover, in real world applications, it needs to update the database regularly, in order to maintain the effectiveness of the training phase. Rana, Aamir et al. [17] proposed a new fuzzy-based semi-supervised learning approach using unlabeled samples assisted by a supervised learning algorithm to improve classifier performance for IDS. The main objective of this work is to find the relation between the fuzziness and the misclassification of the classifier on unlabeled samples. The experimental results on the NSL-KDD dataset show that unlabeled samples belonging to the low and high fuzziness groups contribute significantly to improving the classifier performance compared to the traditional classifiers. Rathore and Park [18] presented a distributed attack detection based on semi-supervised learning for IoT, using a fuzzy c-means algorithm to solve the problem of labeled data unavailability. The proposed approach reached 86.53% accuracy rate on the NSL-KDD dataset.

Works based on the unsupervised learning technique:

Choi et al. [19] developed a network intrusion detection system using an unsupervised learning algorithm autoencoder. This model achieved an accuracy of 91.70%. Deng et al. [20] presented a lightweight IDS combined with a Principal Component Analysis (PCA) algorithm to reduce the input data dimensionality and an FCM clustering algorithm proposed by Bezdek as a classifier. This model was evaluated on the KDD-CUP99 dataset and achieved a detection rate of 96.8% and an error rate of 1.6%. Ieracitano et al. [21] introduced IDS based on autoencoder and statistical analysis. This model was evaluated on the NSL-KDD database. It achieved around 84% accuracy in binary classification and an accuracy of 87 % in multi classification.

3. BACKGROUND

This section presents the basic information necessary to understand the concepts behind the model proposed in this paper.

3.1 IoT architectural model

The Internet of Things consists of many objects connected to the Internet. These objects are enabled to act on data collected from their environments with the goal of making smart environments and applications like healthcare, transportation, agriculture, energy, cities, and many other areas more intelligent. The IoT general architecture can be composed of three layers (Figure 1) [22, 23].

Perception layer: contains various devices for sensing and gathering environmental information, and then transmitting it to the network layer.

Network layer: consists basically of a gateway that serves as a link between the perception layer and the cloud. This interaction and cooperation requires the integration of several wired and wireless communication technologies for exchanging information between objects, such as WiFi, 5G, Zigbee, etc.

Application layer: uses the data received from the network layer to provide the services or operations required by users.

While the three layers have different objectives, they are vulnerable to various forms of attacks. The perception layer can have attacks like injecting malicious code, capturing nodes, eavesdropping, interference and sleep deprivation. The

network layer is vulnerable to the following attacks: Sinkhole, denial of service, spoofing, Hello Flood, man-in-the-middle, routing information, Sybil, etc. In the application layer, the main objective is to provide user-requested services; therefore it is vulnerable to the following attacks: Phishing attacks, worms and viruses [24].

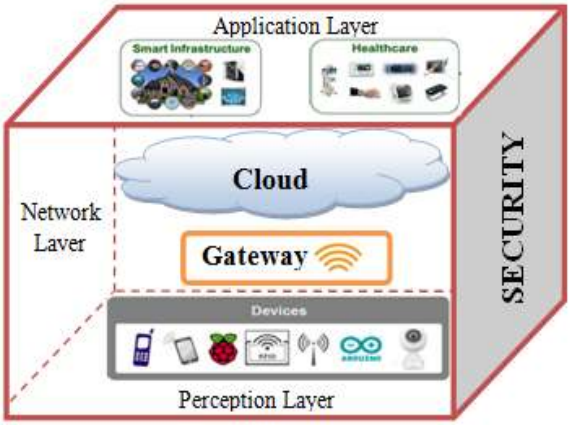


Figure 1. General architecture of IoT

3.2 Intrusion detection system

The intrusion detection system is a mechanism used to identify the intrusions or attacks by monitoring and analyzing network and system activity. There may be internal or external intrusions [23]. Internal intrusions are performed by legitimate users who wish to improve their access rights by abusing unauthorized rights. Whereas, external intrusions are performed by users outside the network seeking unauthorized access to the network [25]. The IDS monitors event sequences of a host or network, and then generates an alert when an intrusion is detected [26]. There are two types of IDS available [24]:

- Host-based IDS (HIDS) is connected to a host/device and monitors malicious activity from various sources such as file system logs and process activities;
- Network based IDS (NIDS) track network traffic data using sniffing tools such as TCPDUMP.

The IDSs may also be classified into three categories [27, 28]:

Signature Based IDS: detects intrusions using well-known attack patterns. It is simple to use. It is easy to use, but as the number of attacks increases, it requires more computation and storage space. The major drawback of such system is, it detects only the intrusions recognized by their signatures [29]. Therefore, it needs regular updates of the database with new attack signatures [30].

Anomaly Based IDS: establishes the normal user behavior at the start and compares the system activities to a normal activity profile and alerts the system administrator when a deviation reaches a threshold [3, 28]. This approach is useful in detecting unknown attacks; however, it generates several false positives, because a deviation from the normal behavior does not necessarily lead to an attack.

Specification Based IDS: combines the two previous techniques, due to their complementary nature [28]. It takes advantage of both techniques to detect unknown threats on the one hand, and reduce false positives on the other. However, this method is very expensive in terms of energy and resource consumption [26].

In this study we focus on anomaly-based IDS using machine learning techniques which are an important approach for creating a detection mechanism to identify new types of attacks [31].

3.3 Machine learning

Machine Learning (ML) refers to algorithms that allow computers to learn automatically without human guidance. More precisely, ML algorithms build models from sample data inputs through learning to make future predictions or decisions based on the newly input data. ML algorithms based detect intrusions system can be grouped into the following three groups:

Supervised learning: algorithms learn from labeled training data, in order to predict outcomes for unforeseen data.

Unsupervised learning: algorithms do not need labeled training datasets. They attempt to find structure in the input data by extracting and analyzing useful features.

Semi supervised learning: is intermediate between supervised learning and unsupervised learning. It builds a model from a normal behavior training dataset, then tests this model on unknown instances [32].

3.4 Deep learning

Deep learning (DL) is an advanced sub-domain of machine learning that is based on the Artificial Neural Network (ANN) model. Traditional ANNs typically contain very few hidden layers, while a Deep Neural Network (DNN) may have more hidden layers, allowing better generalization compared to ANN [33].

3.5 Autoencoder

An autoencoder is a special type of neural network composed mainly of three layers [34], the input, a hidden layer for encoding, and the output decoding layer, as shown in Figure 2. The encoder transforms the input data into low-dimensional codes and the decoder reconstructs the origin inputs from the corresponding codes, and finally a similar result of input and output is expected, i.e. X=$\grave{X}$.

Formally, the encoder takes the input data $x \in \mathbb{R}^n$ and maps it to $h \in \mathbb{R}^m$ using Eq. (1):

$$h = \sigma(Wx + b) \tag{1}$$

Here, σ is an activation function such as a sigmoid function or a rectified linear unit, W is a weight matrix and b is a bias vector. After that, the decoder maps h to the reconstruction $\grave{X}$ using Eq. (2):

$$\grave{X} = \sigma(Wh + b) \tag{2}$$

where, σ, W and b for the two previous equations can be unrelated. The autoencoder is trained to minimize reconstruction errors (such as quadratic errors), often called "losses", defined by Eq. (3):

$$J = \frac{1}{2n}\sum_{i=1}^{n} |x_i - \grave{x}_i|^2 \tag{3}$$

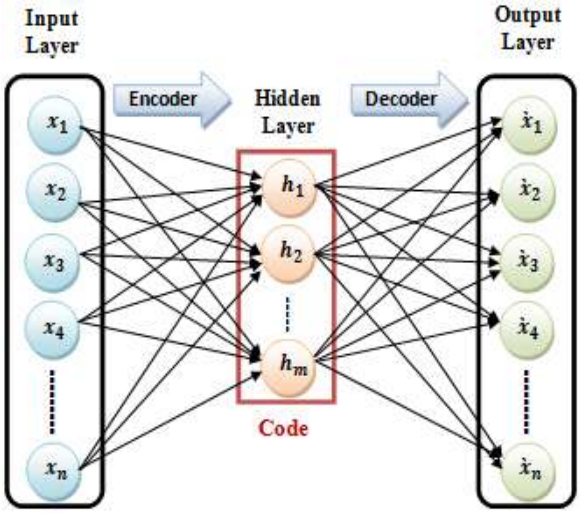


Figure 2. Architecture of an autoencoder

4. PROPOSED MODEL

One of our main challenges has been to develop an IDS which is lightweight and can adapt to the processing capacities of the IoT devices. Thus, according to Roman et al. [35], there should not be a specific intrusion detection module for each IoT node, because of the reduced computing capacity and energy consumption. For this reason, we have implemented a centralized intrusion detection system that allows solving the limited capacity problem as well as the problem of heterogeneity [36]. Figure 3 presents the architecture of our model, which consists of four modules:

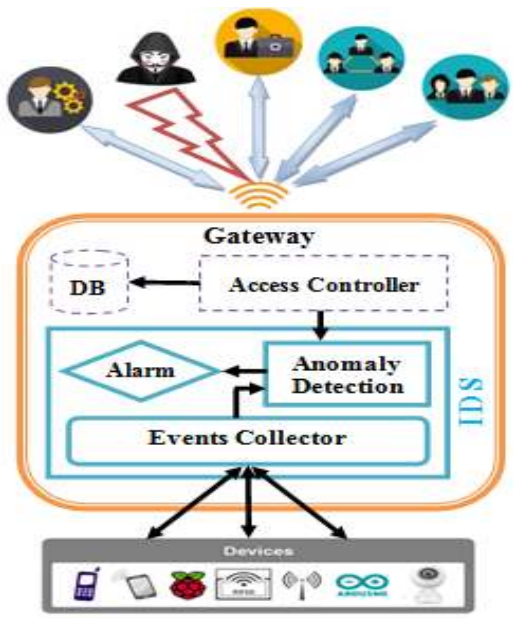


Figure 3. The IDS architectural model

A- Access control: This phase is considered as a first line of defense. It allows users to access the IoT services through authentication. This authentication limits access to services. For each new user, the user profile is created on the database. It represents a normal behavior of a user. This profile is composed of a set of privileges represented by the feature vector $V_i(t)$. Let $V_i(0) = (p_1, p_2, ..., p_n)$, such that: $V_i(0)$ is the feature vector at time 0 of size n, which represents the normal profile of user *i*.

B- Events collection: This component monitors and records user activity to create current user behavior as follows: $V_i(t) = (c_1, c_2, ..., c_n)$.

C- Anomaly detection: This phase identifies abnormal vectors of user. It measures the similarity between its current behavior $V_i(t)$ and its normal profile $V_i(0)$. If the difference exceeds a predefined threshold, it marks $V_i(t)$ as abnormal. The threshold is predefined by the administrator for each newly created user according to its type and characteristic through a learning process.

$$\begin{cases} d\big(V(t) - V(0)\big) \le k, & Normal \\ d\big(V(t) - V(0)\big) > k, & Abnormal \end{cases} \quad (4)$$

where, the similarity is measured by Eq. 5:

$$d\big(V(t) - V(0)\big) = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(c_i - p_i)^2} \quad (5)$$

Anomaly detection is performed using a semi-supervised deep Autoencoder that classifies user behavior as normal/abnormal. This classifier is presented in the following section.

D- Alarm: In the event of an intrusion, this component blocks the suspect user and then informs the system administrator in order to take the necessary measures.

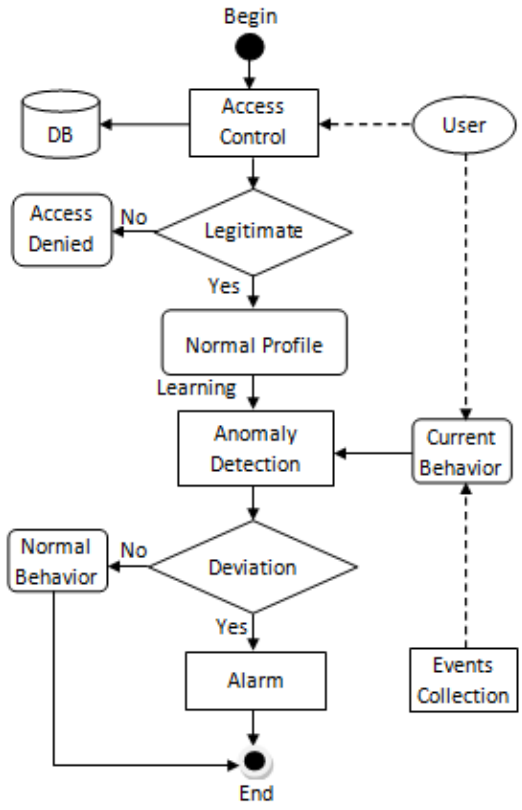


Figure 4. Activity diagram of IDS

Figure 4 presents the operation diagram of our intrusion detection approach. When a user attempts to log into the system, the system processes the user authentication by

checking their credentials against a database. If the authentication information is correct and valid, the user can log into the system, and his/her authentication information is saved in the normal profile database. This database is used to train the Deep autoencoder in a semi-supervised fashion. After entering the system, the user can perform various tasks which will be collected by the event collector and creating the current user behavior. Then, the anomaly detection component uses the Deep autoencoder model that is already trained to classify the current user behavior. If there is a deviation from the normal profile, then an intrusion is detected and an alert will be triggered to inform the administrator of the situation otherwise it is normal behavior.

The advantages of our approach are:

(1) the ability to detect internal and external attacks provided by dual protection; (2) there is not a need to update the learning dataset regularly, in order to maintain the effectiveness of training phase; (3) there is not a need a human supervisor to label the training dataset.

5. EVALUATIONS AND RESULTS

This section presents a detailed discussion of the experimental results using a semi-supervised deep autoencoder performed on two datasets: NSL-KDD and CIDDS-001. The autoencoder learns the pattern of a normal process using only the features of the normal behavior. Any behavior that does not follow this pattern is considered as an anomaly. The proposed approach is detailed in the following Algorithm 1.

Algorithm 1: Semi-Supervised Deep AutoEncoder for Intrusion Detection

Training Step

Input: Training dataset $X = \{x_1, x_2, \ldots, x_n\}$,
Output: SDEA Model; Threshold k

1. Data preprocessing: transformation and normalization of data in the training dataset
2. Removing the data labeled as attack from the training dataset
3. Splitting of training data into two disjoint subsets: training and validation dataset
4. Initialization of weights using pre-training method
5. Training the model using only the data labeled as normal
6. Hyper-parameters tuning using grid search technique
7. Using validation dataset to identify the threshold by applying the equation 5

Testing Step

Input: Testing dataset $\dot{X} = \{\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n\}$, Threshold k
Output: Result of the classification model

8. Data preprocessing: transformation and normalization of data in the testing dataset
9. Calculate the Reconstruction:
   Error (RE) = input data - reconstructed data
10. **If** RE > threshold k **then** Attack **else** Normal
11. Evaluate model performance by calculate various accuracy metrics

The process of our approach is composed of two stages: training and testing. During the training phase, the training dataset is used to train the semi-supervised autoencoder model on only data labeled as normal. After a preprocessing stage, the training dataset is splitted into two parts training and validation dataset, then the autoencoder is pre-trained to determine the initial weights that will be used by the model in order to avoid the vanishing gradient problem and enhance classification performance. During the learning process, a hyper-parameters tuning is carried to choose a set of optimal hyper-parameters for a learning algorithm using the grid search technique, where we use all combinations of the hyper-parameters. Then, we used equation 5 to calculate and set the threshold value for the validation data set. During the testing phase, the input of the test dataset is reconstructed on the output of the model. Using Equation 5, we calculated the reconstruction error which is equal to the deviation between the original data and the reconstructed data. When the deviation exceeds the threshold value, these data are then classified as abnormal; otherwise, they are classified as normal. At the end, we calculated the different performance metrics to evaluate our model.

5.1 Dataset

To validate our proposed model, we selected two reference data sets, which are well known in the field of the IDS, such as NSL-KDD [37] and CIDDS-001 [38].

NSL-KDD dataset: This dataset is constructed from the KDD Cup 99 [39] dataset by eliminating the redundant records. The NSL-KDD dataset is composed of two separate files, "KDDTrain+" and "KDDTest+" which represent the training dataset and the testing dataset, respectively. Each record in the NSL-KDD dataset consists of 41 features labeled with different categories of attack: Probane, denial of service (DoS), user to root (U2R), and remote to local (R2L) in addition to the Normal type. Table 1 illustrates the number of records.

Table 1. Label Categories in the NSL-KDD dataset

Category	Training-Set	Testing-Set
Normal	67,343	9,711
Probane	45,927	7,460
DoS	11,656	2,421
U2R	995	2,885
R2L	52	67
Total	**12,5973**	**22,544**

CIDDS-001 dataset: Coburg Intrusion Detection Data Set is a labeled dataset used to evaluate the anomaly-based IDS generated by Ring et al. [38]. This dataset is composed of ten features and five categories: normal, suspicious, unknown, attacker, and victim. In our experiments, we used only normal and attacker category as the final dataset. Therefore, it contains 153,026 instances and 12 features. The composition of this dataset is presented in Table 2.

Table 2. Label Categories in the CIDDS-001 dataset

Category	Training-Set	Testing-Set
Normal	4,032	1,218
Suspicious	62,539	19,567
Unknown	21,772	6,734
Attacker	5,888	1,890
Victim	3,705	1,197
Total	**97,936**	**30,606**

5.2 Data preprocessing

We performed the following preprocessing steps on the two datasets:

A. Data binarization: In our experiences, we have adopted a binary classification by considering only two label categories: Attack and Normal.

B. Data transformation: All the categorical values are mapped into numeric values by using the one hot encoding technique, e.g. in the NSL-KDD dataset, the feature “type protocols” has three categories "tcp", "udp" and "icmp" which will be mapped to (1,0,0), (0,1,0) and (0,0,1), respectively. Finally, 41 features are transformed into 117 features.

C. Data normalization: The key aim of data normalization is to enhance the performance by reducing the impact of features with a huge variance before they are fed to any learning algorithm. Therefore, all the values of the numerical entities must be arranged between 0 and 1, by applying Eq. (6):

$$Z = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{6}$$

After these transformations, the new structure of the datasets is presented in Table 3.

Table 3. New structure of the datasets

	Features	Normal	Attack	Total
NSL-KDD dataset				
Training	117	67,343	58630	125,973
Testing	117	97,11	12833	22,544
CIDDS-001 dataset				
Training	43	4,032	93,904	97,936
Testing	43	1,218	29,388	30,606

5.3 Accuracy metrics

Many accuracy metrics are used to evaluate the proposed model, which are based on the elements of the confusion matrix: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

The most widely used accuracy measures are formulated as follows:

$$TP\ Rate\ (Recall) = \frac{TP}{TP + FN} \tag{7}$$

$$TN\ Rate = \frac{TN}{TN + FP} \tag{8}$$

$$FP\ Rate = \frac{FP}{FP + TN} \tag{9}$$

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

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

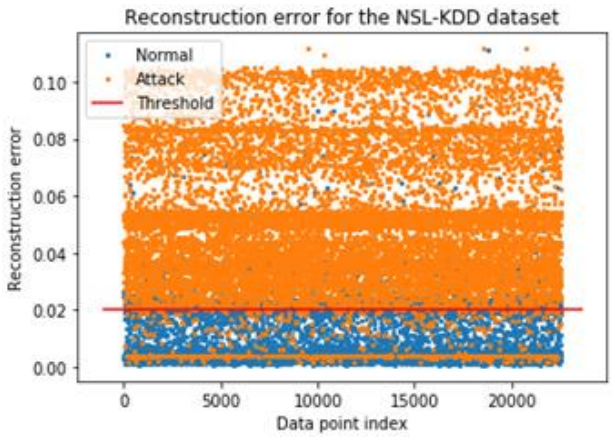
$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$F1\ Score = 2 * \frac{Precision\ * Recall}{Precision + Recall} \tag{13}$$

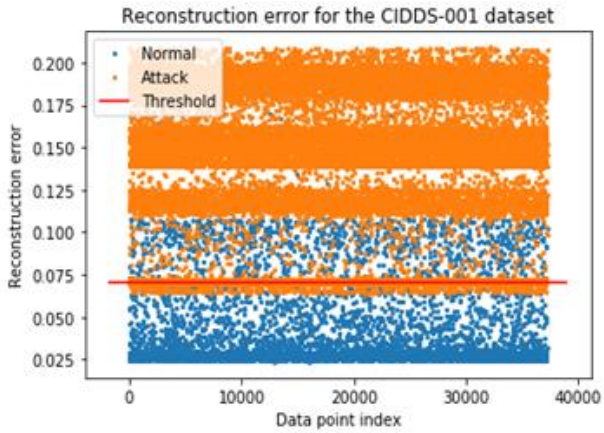
5.4 Performance evaluation

This section provides analysis of the performance results of the proposed model. First, to tune the hyper-parameters for the model, we performed a grid search which generates a certain number of candidate parameters values. After selecting the best hyper-parameter values, the final model is composed of five hidden layers with 64, 32, 16, 32 and 64 units, with the ReLU activation function for all hidden layers, the activation function of the output layer is Sigmoid. We used the Adam optimizer, L2 regularization with 0.0001 learning rate, train epochs of 10 and the batch size of 100.

After training the models on normal data, we calculated the mean square error (MSE) between the input data and the reconstructed data on the samples of the validation dataset. Then, we established an optimal empirical threshold which separates the normal and abnormal samples. Figure 5 shows the reconstruction error distribution with the thresholds.



(a) Reconstruction error for the NSK-KDD dataset



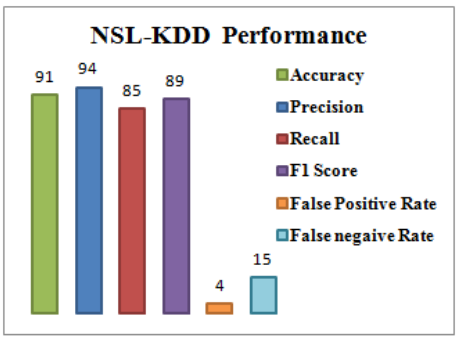
(b) Reconstruction error for the CIDDS-001 dataset

Figure 5. Reconstruction error with threshold

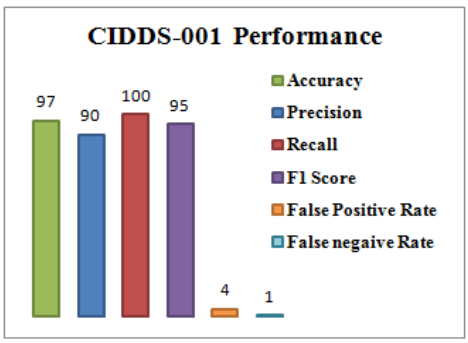
To set the construction error threshold, we have drawn the precision/recall curve for different thresholds, where we want to establish an optimal balance between recall and precision. Therefore, we have selected a reconstruction error threshold at 0.02 for the NSL-KDD dataset and 0.07 for the CIDDS-001 dataset.

Afterwards, once the reconstruction error has been fixed, we evaluated the performance of the proposed model on the test dataset using several evaluation metrics. An efficient IDS must aim to maximize accuracy, along with minimizing False Positives and False Negative. The accuracy refers to the proportion of correctly classified data by the model, On the other hand, a False Positive indicates that the IDS has wrongly detected a particular type of attack, while a False Negative indicates a failure to identify malicious activity. Since the

accuracy alone is not enough to evaluate the performance of the classifier, we used other evaluation criteria, such as: Precision, Recall and F1 Score. The Precision measures the proportion of samples classified as attacks that are actually attacks; the objective of this measure is to limit the number of false positives (FP). While the Recall (or sensitivity) measures the proportion of attacks that are identified correctly; the goal is to limit the number of false negatives (FN). Finally, The F1-score is the harmonic mean of Precision and Recall. Figure 6 presents the accuracy metrics results.



(a) NSL-KDD Performance



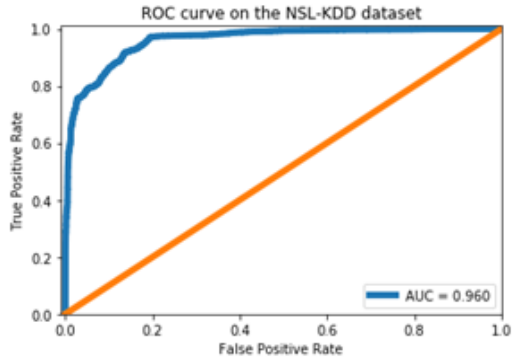
(b) CIDDS-001 Performance

Figure 6. Model's test performance

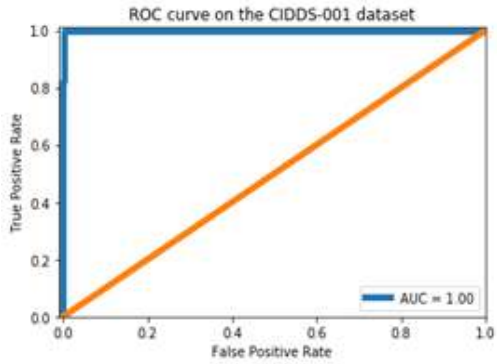
As is evident from Figure 6, the proposed model achieved the best detection performance on the CIDDS-001 dataset. It achieves about 97% of accuracy, 99% of detection rate, and 100% of recall. Furthermore, it also provides a low false negative and false positive rate. For the NSL-KDD dataset, the model is less efficient compared for NSL-KDD dataset, where the accuracy reached a rate of 91%, 94% precision, 85% recall and 89% F1 Score. Furthermore, the false positive rate was the same for the two datasets at 4%, while the false negative rate is higher, it reached 15%.

Receiver Operating Characteristic (ROC) is one of the most popular methods used to evaluate a binary classification problem. The ROC curve is useful for examining the general performance of the model by plotting the true positive rate (sensitivity) versus the false positive rate (1-specificity). The Area Under ROC Curve (AUC) is another important metric that is used to evaluate the performance of the classifiers. Figure 7 shows the ROC curve plot for the two datasets.

From Figure 7, the proposed model shows a good shape in the ROC curve. It could reach an AUC value of 1.00 on the CIDDS-001 dataset, which may produce near perfect performance. On the NSL-KDD dataset, our model could also provide excellent performance, reaching up to 0.96 of AUC. It is lower than that performed on the CIDDS-001 dataset. Indeed, some types of attacks are difficult to distinguish from normal behavior due of their great similarity, such as U2R and DOS attacks. Finally, we can conclude that our proposed model performs well. Therefore, it can effectively detect malicious nodes and targeted intrusive traffic on the IoT network.



(a) Roc curve on the NSL-KDD dataset



(b) Roc curve on the CIDDS-001 dataset

Figure 7. Model's ROC curve for the two datasets

5.5 Comparison and discussion

Many research studies have been conducted in the field of IDS in order to provide security for IoT. A comparison between our proposed method and some other previous propositions is illustrated in Table 4, focusing mainly on Intrusion Detection Systems for IoT.

Table 4. Comparison of IDS for IoT

Ref.	IDS Architecture	Learning type	Method	Dataset	Acc.
[20]	Centralized	Unsupervised	PCA and FCM	KDD-CUP99	96.8%
[18]	Distributed	Semi-supervised	fuzzy c-means	NSL-KDD	86.53%
[13]	Distributed	Supervised	LSTM Recurrent Neural Network	UNSWNB15	95%
[14]	Distributed	Supervised	CNN+ LSTM	CICIDS2017	97%
[21]	Distributed	unsupervised	Autoencoder	NSL-KDD	87%
PM	Centralized	Semi-Supervised	Autoencoder	NSL-KDD CIDDS-001	91% 97%

As shown in Table 4, most of approaches use a distributed architecture, which is more suitable for IoT environment than centralized architecture given the distributed nature of IoT

devices. However, for a distributed IDS, there should be a specific intrusion detection module for each IoT node, which is not suitable for IoT, because of the reduced computing capacity and energy consumption, in addition to the heterogeneity of IoT devices. Moreover, centralized IDS detect better the distributed attacks (such as distributed denial-of-service (DDoS) attack) than the distributed IDS. In the supervised learning, the output of the training set is already known; this makes it much easier to build, while the main drawback is the need for a human supervisor to label the training set. In unsupervised learning, training data is not labeled; hence, the learning is performed independently without any human intervention, which overcomes the problem of labeled data unavailability and the detection in real time. The main disadvantage of unsupervised learning is that it is less accurate than supervised learning. Whereas, semi supervised learning is intermediate between supervised learning and unsupervised learning.

Based these comparisons, the results obtained using the proposed model (PM) are very promising.

6. CONCLUSION

In this paper, a centralized IDS architecture suitable for securing IoT is proposed. This IDS is based on semi-supervised learning autoencoders, which trains the model using only the normal behavior. Then, using the reconstruction error threshold, it classifies the unknown data as normal or abnormal. The parameters of the model are tuned using the grid search technique. Evaluating of the classifier is performed on NSL-KDD and CIDDS-001 datasets, by measuring various performance metrics including accuracy, precision, recall, F1-score, false positive rate, true positive rate and ROC curve. The performance results allow us to conclude that the proposal framework gives promising results, which proves the efficient of this model for the construction of IDS based on anomaly detection suitable for IoT environment.

In our future research, we plan to develop our evaluation on real-world data to demonstrate the benefits of the proposed model. We also plan to adopt a hybrid architecture (centralized and distributed) as well as a hybrid IDS system that combines anomaly detection and scenario detection methods that meet IoT requirements.

REFERENCES

[1] Mosenia, A., Jha, N.K. (2017). A comprehensive study of security of internet-of-things. IEEE Transactions on Emerging Topics in Computing, 5(4): 586-602. https://doi.org/10.1109/TETC.2016.2606384

[2] Meng, Y., Zhang, W., Zhu, H., Shen, X. (2018). Securing consumer IoT in the smart home: Architecture, challenges, and countermeasures. IEEE Wireless Communications, 25(6): 53-59. https://doi.org/10.1109/MWC.2017.1800100

[3] Liu, J., Xiao, Y., Chen, C.L. (2012). Authentication and access control in the internet of things. 2012 32nd International Conference on Distributed Computing Systems Workshops, Macau, China. https://doi.org/10.1109/ICDCSW.2012.23

[4] Moosavi, S.R., Gia, T.N., Rahmani, A.M., Nigussie, E., Virtanen, S., Isoaho, J., Tenhunen, H. (2015). SEA: A secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways. In Procedia Computer Science, 52: 452-459. https://doi.org/10.1016/j.procs.2015.05.013

[5] Hussein, S.M., Ali, F.H.M., Kasiran, Z. (2012). Evaluation effectiveness of hybrid IDs using snort with naive Bayes to detect attacks. Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), Bangkok, Thailand, pp. 256-260. https://doi.org/10.1109/DICTAP.2012.6215386

[6] Xian, J.Q., Lang, F.H., Tang, X.L. (2005). A novel intrusion detection method based on clonal selection clustering algorithm. 2005 International conference on Machine Learning and Cybernetics, Guangzhou, China. https://doi.org/10.1109/ICMLC.2005.1527620

[7] Murali, A., Rao, M. (2005). A survey on intrusion detection approaches. International Conference on Information and Communication Technologies, Karachi, Pakistan. https://doi.org/10.1109/ICICT.2005.1598592

[8] Patcha, A., Park, J.M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks, 51(12): 3448-3470. https://doi.org/10.1016/j.comnet.2007.02.001

[9] Pan, Y., Sun, F., Teng, Z., White, J., Schmidt, D.C., Staples, J., Krause, L. (2019). Detecting web attacks with end-to-end deep learning. Journal of Internet Services and Applications, 10(1). https://doi.org/10.1186/s13174-019-0115-x

[10] Ma, T., Wang, F., Cheng, J., Yu, Y., Chen, X. (2016). A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. Sensors (Basel, Switzerland), 16(10): 1701. https://doi.org/10.3390/s16101701

[11] Pongle, P., Chavan, G.T. (2015). Real time intrusion and wormhole attack detection in internet of things. International Journal of Computer Applications, 121(9): 1-9. https://doi.org/10.5120/21565-4589

[12] Verma, A., Ranga, V. (2019). Machine learning based intrusion detection systems for IoT applications. Wireless Personal Communications, 111: 1-24. https://doi.org/10.1007/s11277-019-06986-8

[13] Roy, B., Cheung, H. (2018). A deep learning approach for intrusion detection in internet of things using Bi-directional long short-term memory recurrent neural network. 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, pp. 1-6. https://doi.org/10.1109/ATNAC.2018.8615294

[14] Roopak, M., Tian, G., Chambers, J. (2019). Deep learning models for cyber security in IoT networks. 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA. https://doi.org/10.1109/CCWC.2019.8666588

[15] Wagh, S.K., Kolhe, S.R. (2014). Effective intrusion detection system using semi-supervised learning. In 2014 International Conference on Data Mining and Intelligent Computing (ICDMIC), New Delhi, India, pp. 1-5. https://doi.org/10.1109/ICDMIC.2014.6954236

[16] Li, W.J., Meng, W.Z., Au, M.H. (2020). Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments. Journal of Network and Computer Applications, 161: 102631. https://doi.org/10.1016/j.jnca.2020.102631

[17] Ashfaq, R.A.R., Wang, X.Z., Huang, J.Z., Abbas, H., He, Y.L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. Information Sciences, 378: 484-497. https://doi.org/10.1016/j.ins.2016.04.019

[18] Rathore, S., Park, J.H. (2018). Semi-supervised learning based distributed attack detection framework for IoT. Applied Soft Computing, 72: 79-89. https://doi.org/10.1016/j.asoc.2018.05.049

[19] Choi, H., Kim, M., Lee, G., Kim, W. (2019). Unsupervised learning approach for network intrusion detection system using autoencoders. The Journal of Supercomputing, 75: 1-25. https://doi.org/10.1007/s11227-019-02805-w

[20] Deng, LB., Li, D.M., Yao, X., Cox, D., Wang, H.X. (2019). Mobile network intrusion detection for IoT system based on transfer learning algorithm. Cluster Computing, 22: 9889-9904. https://doi.org/10.1007/s10586-018-1847-2

[21] Ieracitano, C., Adeel, A., Morabito, F.C., Hussain, A. (2019). A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing, 387: 51-62. https://doi.org/10.1016/j.neucom.2019.11.016

[22] Atzori, L., Iera, A., Morabito, G. (2010). The internet of things: A survey. Computer Network, 54(15): 2787-2805. https://doi.org/10.1016/j.comnet.2010.05.010

[23] Zhao, K., Ge, L.N. (2013). A survey on the Internet of Things security. 2013 Ninth International Conference on Computational Intelligence and Security, Leshan, China, pp. 663-667. https://doi.org/10.1109/CIS.2013.145

[24] Santos, L., Rabadao, C., Goncalves, R. (2018). Intrusion detection systems in internet of things: A literature review. 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), Caceres, Spain. https://doi.org/10.23919/cisti.2018.8399291

[25] Huang, S.H. (2003). Dimensionality reduction in automatic knowledge acquisition: A simple greedy search approach. IEEE Transactions on Knowledge and Data Engineering, 15(6): 1364-1373. https://doi.org/10.1109/TKDE.2003.1245278

[26] Leo, M., Battisti, F., Carli, M., Neri, A. (2014). A federated architecture approach for internet of things security. In Euro Med Telco Conference (EMTC), Naples, Italy, pp. 1-5. https://doi.org/10.1109/EMTC.2014.6996632

[27] Sherasiya, T., Upadhyay, H., Patel, H.B. (2016). A survey: Intrusion detection system for internet of things. International Journal of Computer Science and Engineering (IJCSE), 5(2): 91-98.

[28] Zarpelão, B.B., Miani, R.S., de Alvarenga, S.C. (2017). A survey of intrusion detection in internet of things. Journal of Network and Computer Applications, 84(C): 25-37. https://doi.org/10.1016/j.jnca.2017.02.009

[29] Liao, H.J., Richard Lin, C.H., Lin, Y.C., Tung, K.Y. (2013). Review intrusion detection system: A comprehensive review. Journal of Network and Computer Application, 36(1): 16-24. https://doi.org/10.1016/j.jnca.2012.09.004

[30] Maharaj, N., Khanna, P. (2014). A comparative analysis of different classification techniques for intrusion detection system. International Journal of Computer Applications, 95(17): 22-26. https://doi.org/10.5120/16687-6806

[31] Buczak, A.L., Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2): 1153-1176. https://doi.org/10.1109/comst.2015.2494502

[32] Chandola, V., Banerjee, A., Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3). https://doi.org/10.1145/1541880.1541882

[33] Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. Pattern Recognition Letters, 119: 3-11. https://doi.org/10.1016/j.patrec.2018.02.010

[34] Bengio, Y., Courville, A., Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8): 1798-1828.

[35] Roman, R., Zhou, J.Y., Lopez, J. (2006). Applying intrusion detection systems to wireless sensor networks. In IEEE Consumer Communications & Networking Conference, Las Vegas, NV, USA. https://doi.org/10.1109/CCNC.2006.1593102

[36] Fenanir, S., Semchedine, F., Baadache, A. (2019). A Machine learning-based lightweight intrusion detection system for the internet of things. Revue d'Intelligence Artificielle, 33(3): 203-211. https://doi.org/10.18280/ria.330306

[37] NSL-KDD Dataset, https://www.unb.ca/cic/datasets/nsl.html, accessed on Mar. 16, 2020.

[38] Ring, M., Wunderlich, S., Grüdl, D., Landes, D., Hotho, A. (2017). Flow-based benchmark data sets for intrusion detection.

[39] KDD Cup 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, accessed on Oct. 22, 2007.