Journal of Computer Science

Original Research Paper

IoT Intrusion Detection using Auto-Encoder and Machine Learning Techniques

Ahmed Ridha Khudhu and Khairulmizam Samsudin

Department of Computer and Communication System Engineering, Universiti Putra Malaysia, Malaysia

Article history
Received: 08-05-2022
Revised: 29-07-2022
Accepted: 03-09-2022

Corresponding Author:
Ahmed Ridha Khudhur
Department of Computer and Communication System Engineering, Universiti Putra Malaysia, Malaysia
Email: gs58251@student.upm.edu.my

Abstract: IoT network refers to the capability of connecting smart and various devices to a single network for the sake of performing a particular task. Similar to conventional networks, IoT networks are vulnerable to several attacks. Therefore, IoT Intrusion Detection has caught much research attention. Several studies have examined the task of intrusion detection for IoT networks. Within such studies, the focus was set to accommodate a feature selection process for identifying the most relevant features per the intrusions. Yet, the feature selection techniques used in the literature were based on feature selection rather than a reduction in which individual solutions are being selected. This could lead to a fall in local minima problems where the optimal solution is not determined but instead, another near-optimal solution is identified. This study proposes a dimensionality reduction approach rather than feature selection using Auto-Encoder architecture for IoT intrusion detection. A benchmark dataset of UNSW-NB15 has been used within the experiment. In addition, a data preparation process of feature transformation has been applied to convert the categorical features into numeric ones. Then, the proposed autoencoder has been carried out upon the transformed data for the sake of dimensionality reduction. The reduced dimension produced by the proposed autoencoder has been utilized by four classifiers including DT, LR, NN, and RF for conducting the intrusion detection. Results showed that the proposed autoencoder with RF classifier has obtained the highest F1-score of 99% and the lowest FAR value of 0.78%. Such results are competitive in terms of the state of the art.

Keywords: Internet of Things, Intrusion Detection, Auto Encoder, Decision Tree, Logistic Regression, Random Forest, Neural Network

Introduction

The development of network and communication technologies in recent years has contributed to the appearance of a new type of network known as the Internet-of-Things (IoT) (Nauman *et al.*, 2020). This special type of network refers to the capability of connecting smart and various devices to a single network for the sake of performing a particular task. Considering a facility with cameras, sensors, computers, printers, and personnel's smartphones. A network that can keep all the aforementioned devices connected to the internet would open a wide door for various types of processing. From securing the facility, and detecting abnormal activities, to tracking the workflows, all these tasks can be accomplished through such a network (Khan *et al.*, 2020). Hence, the availability of providing remote access, reporting, and controlling would facilitate top managers or executives to monitor and assess the activities within such a facility. These promising features and capabilities have contributed toward the dramatic extension of IoT networks where the emergence of smart houses, smart hospitals, smart manufacturing, and others are widely witnessed recently (Qadri *et al.*, 2020).

Similar to any sort of network, IoT tends to be vulnerable to a wide range of attacks and threats. From traditional attacks such as Denial of Service (DoS) to viruses and worms, IoT is subject to different threats. In particular, attackers nowadays are developing specific types of attacks that specifically target IoT networks (Ullah and Mahmoud, 2019). Due to the sensitivity of information located over the IoT networks which are usually related to specific businesses or industries, the security of IoT networks represents the

most common challenge. Identifying any possible threat that might target IoT networks would be an essential task (Stoyanova *et al.*, 2020).

Therefore, the task of Intrusion Detection (ID), which has been widely explored in the last two decades, would be the right choice for securing IoT networks. ID is the process of detecting any potential threat that seems to be harmful or at least intended to impact the performance of a particular network (Masdari and Khezri, 2020). Numerous ID systems have been presented in the past for various types of networks. Recently, ID has been examined in terms of IoT networks where the literature was highly dependent on Machine Learning Techniques (MLT) (Khaliq, 2020).

Detecting intrusions and any possible threat that might target the IoT network is considered an essential task. The literature showed a great interest in using machine learning techniques to identify intrusions. This can be depicted by training MLT on network traffic features such as duration of connections, internet protocol used by the connection, size of sending and receiving packets, and so on. Hence, there will be a large dimension of features that might be utilized. However, MLT is highly impacted by the features that will be used within the training. This means that if there are irrelevant features used within the training, the accuracy of detecting intrusions will considerably be affected. Therefore, most state-of-the-art in IoT intrusion detection is considering a feature selection task before the detection itself.

To do the feature selection, the literature was highly relying on bio-inspired techniques that are intended to find optimal or near-optimal solutions such as Artificial Bee Colony (ABC) and Artificial Fish Swarm (AFS) (Hajisalem and Babaie, 2018) or Genetic Algorithm (Papamartzivanos *et al.*, 2018; Roopak *et al.*, 2020). Yet, these techniques suffer from falling into local minima problems where the optimal solution is not determined but instead another near-optimal solution is identified. This is because these techniques are investigating individual features or solutions therefore, they will generate possible permutations of these solutions. In other words, such techniques are based on feature selection rather than reduction.

As a new opportunity, Deep Learning (DL) techniques refer to modern sophisticated architectures of Neural Networks (NN) that substitute the use of feature engineering. Auto-Encoder is a DL architecture that aims to accommodate a dimensionality reduction by learning the original feature space and attempting to predict the same feature space. Unlike the way of bio-inspired techniques in terms of finding optimal solutions, the Auto-Encoder technique offers a distinctive approach to learning the network feature space in which the aim is to reduce the dimensionality of such a space into a very representative learned to code.

This study aims to propose an Auto-encoder architecture as a dimensionality reduction for IoT intrusion detection to improve feature learning. Consequentially, this study has utilized the learned features produced by Auto-encoder for accommodating classification using four classifiers including Decision Tree (DT), Logistic Regression (LR) Neural Network (NN), and Random Forest (RF). Lastly, an evaluation based on the performance of the proposed Auto-encoder in terms of intrusion detection accuracy has been depicted.

Related Work

Many scholars have recently looked at feature selection in IoT detection. For example, Gharaee and Hosseinvand (2016) looked at the issue of dimensionality of feature space in IoT intrusion detection. The authors have concentrated on the difficult task of lowering the false positive rate in intrusion detection systems. The authors suggest a feature selection/reduction strategy combining the Genetic Algorithm (GA) and a Support Vector Machine (SVM) classifier for this aim. The studies were conducted on the UNSW-NB15 dataset, which had a detection accuracy of 93.25% and a FAR of 8.6.

Khammassi and Krichen (2017) developed a wrapper technique-based feature selection strategy. The authors aimed to determine the most important characteristics that could influence intrusion detection accuracy. As a result, a wrapper methodology was applied, in which a Genetic Algorithm was employed as a feature selection strategy, and a Decision Tree (DT) was employed as a classification method. The studies were conducted on the UNSW-NB15 dataset, with the best subset of features achieving an accuracy of 81.42% and a FAR of 6.39.

Moustafa and Slay (2017) developed an Association Rule Mining strategy for feature selection/reduction in IoT intrusion detection, in addition to the existing meta-heuristic feature selection methodologies. The suggested method focuses on the major points of important features that influence intrusion detection. The dataset utilized in the trials was UNSW-NB15, and the proposed technique achieved an average accuracy of 83% with a FAR of 14.2.

In a similar vein, Mogal *et al*. (2017) used the Apriori method to identify the most important features in IoT intrusion detection. The proposed algorithm has been used to rank the features based on their importance, with the unnecessary features being removed. The data examples were then classified using two classifiers, Nave Bayes, and Logistic Regression, based on the selected features. The UNSW-NB15 dataset was employed, and the average accuracy obtained by the suggested technique was 90% with a FAR of 10.5.

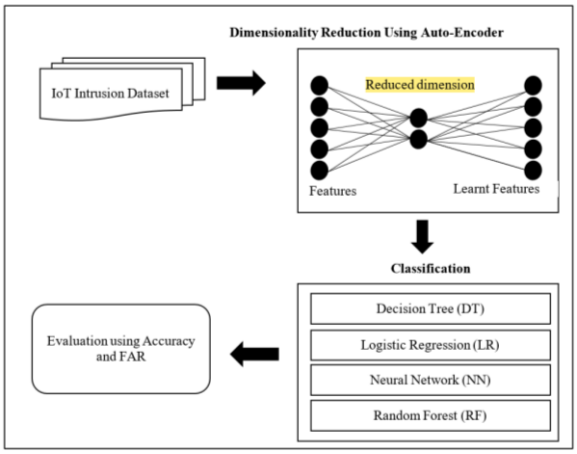

Fig. 1: Framework of the proposed method

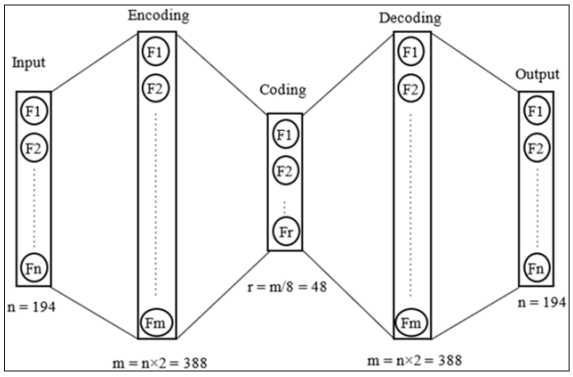

Fig. 2: Architecture of the proposed AE

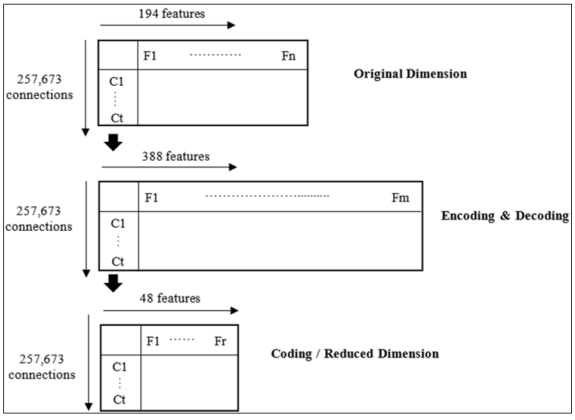

Fig. 3: Size of matrices produced by AE

Papamartzivanos *et al.* (2018) conducted another work on feature selection in IoT intrusion detection, proposing a combination of Genetic Algorithm and Decision Tree for this purpose. To make rule induction for the rules created by the DT, GA was used. The UNSW-NB15 dataset was employed in the tests, as it has been in all previous studies on IoT intrusion detection. The best subset of characteristics has an accuracy of 84.33%, with a FAR of 8.9.

In a similar vein, Hajisalem and Babaie (2018) presented a hybrid of Artificial Bee Colony (ABC) and Artificial Fish Swarm (AFS) algorithms to handle a holistic feature selection assignment on IoT intrusion detection. To obtain the optimal solution of features, the authors used the two methods together. Finally, the intrusion was classified using a CART Association Rule classifier based on the specified features. The studies employed the UNSW-NB15 dataset, which had an average accuracy of 85% and a FAR of 14.9.

Some writers, on the other hand, have employed feature selection approaches to improve IoT intrusion detection classifiers. To find the optimal classifier parameters, Tama and Rhee (2019) suggested a grid search technique. Each classifier has its own set of parameters, and examining each one individually can be challenging at times. As a result, the proposed grid search was utilized to find the optimal parameters for four different classifiers: Neural networks, support vector machines, and fuzzy classifiers. The proposed grid search improved all classifiers, with the combination of grid search and neural networks achieving the maximum accuracy on the UNSW-NB15 dataset, with an average accuracy of 82.6% with a FAR of 16.2.

For the issue of IoT intrusion detection, Ullah and Mahmoud (2019) introduced the Recursive Feature Elimination (RFE) approach, which is a linear method for feature selection. The suggested method will recursively analyze each feature by iteratively dividing the feature space into progressively smaller subsets. The experiment employed the UNSW-NB15 dataset, with a 97% average accuracy as well as a FAR of 7.8.

An artificial neural network for intrusion detection has been proposed by Lopez-Martin *et al*. (2019). The proposed technique achieved a 77.8% accuracy using the UNSW-NB15 dataset.

Hanif *et al*. (2019), utilizing the UNSW-NB15 dataset, developed an ANN for intrusion detection. The proposed technique attained an accuracy of 84% with a FAR of 8.0.

Roopak *et al*. (2020) proposed an enhanced version of GA called Non-dominated Sorting Genetic Algorithm (NSGA) for feature selection in IoT intrusion detection. The authors have utilized an ML technique known as Extreme Learning Machine (ELM).

Recently, studies by (Ullah and Mahmoud, 2021; Ullah and Mahmoud, 2022) have examined a deep learning architecture of Convolutional Neural networks for IoT intrusion detection. Using the UNSW-NB15 dataset, these studies showed an F1 score ranging from 99.60 to 99.99%.

The state of the art in IDS feature selection, as indicated in the previous section, relied on classic bio-inspired approaches. The accuracy of classification obtained by these techniques was excellent. However, the False Alarm Rate (FAR) must be lowered. FAR refers to connections that have been mistakenly categorized as intrusive when they are not.

Table 1: Dataset details

Attributes	Details
Connections number	257,673
Training portion	175341
Testing portion	82332
Features number	43
Attacks number	9
Attack types	Fuzzers: Feeding networking with randomly generated data Analysis: Contains different attacks such as scanning and probing Backdoors: Seeking weak points in a network DoS: Allocate the resources of a network Exploits: Seeking weak points in an operating system Generic: Targets the block ciphers and their keys Reconnaissance: Strikes that can simulate attacks that gather information Shellcode: Exploits specific software on the network Worms: Replicates itself to spread to other computers

Table 2: Sample of the dataset

Connection ID	Protocol	Service	Duration	….	Class	Class (Binary)
1	TCP	FTP	0.121478		Normal	0
2	TCP	HTTP	0.649902		Normal	0
3	UDP	HTTP	1.623129		Exploits	1
4	TCP	HTTP	1.681642		Normal	0
5	UDP	FTP	0.449454		DoS	1

Table 3: Sample of categorical features from the dataset

Connection	Protocol	Service	….	Class
Con 1	UDP	HTTP		Intrusion
Con 2	TCP	HTTP		Normal
Con 3	UDP	HTTP		Intrusion
Con 4	TCP	HTTP		Normal
Con 5	UDP	FTP		Normal

Table 4: Feature transformation

Connection	Protocol TCP	Protocol UDP	Service FTP	Service HTTP	….	Class
Con 1	0	1	0	1		Intrusion
Con 2	1	0	0	1		Normal
Con 3	0	1	0	1		Intrusion
Con 4	1	0	0	1		Normal
Con 5	0	1	1	0		Normal

Table 5: Hyperparameter of autoencoder

Hyperparameter	Description
Normalization	2 Batch normalization
Activation function	2 Leaky ReLU
Epochs	20
Batch size	16
Optimizer	Adam
Number of layers	5 including input, encoding, coding, decoding, and output
Number of neurons	Input: 194 Encoding: 388 Coding: 48 Decoding: 388 Output: 194

Table 6: Confusion matrix

Predicted actual	Intrusion	Legitimate
Intrusion	True positive	False negative
Legitimate	False positive	True negative

Table 7: Results of the four classifiers

Classifier	Weighted average F1-score	FAR
DT	97%	1.89%
LR	89%	9.21%
NN	92%	2.01%
RF	99%	0.78%

Table 8: Comparison against baseline studies

Baseline Study	Method	Dataset	F1-score	FAR
Gharaee and Hosseinvand (2016)	GA + SVM	UNSW-NB15	93.25%	8.60%
Khammassi and Krichen (2017)	GA + DT	UNSW-NB15	81.42%	6.39%
Moustafa and Slay (2017)	Association Rule	UNSW-NB15	83.00%	14.20%
Mogal *et al*. (2017)	Apriori + LR	UNSW-NB15	90.00%	10.50%
Papamartzivanos *et al*. (2018)	GA + DT	UNSW-NB15	84.33%	8.90%
Hajisalem and Babaie (2018)	ABC + AFS + CART	UNSW-NB15	85.00%	14.90%
Tama and Rhee (2019)	Grid search + SVM	UNSW-NB15	82.6%	16.20%
Ullah and Mahmoud (2019)	RFE	UNSW-NB15	97.00%	7.80%
Lopez-Martin *et al*. (2019)	ANN	UNSW-NB15	77.8%	-
Hanif *et al*. (2019)	ANN	UNSW-NB15	84.00%	8.00%
Roopak *et al*. (2021)	NSGA + ELM	UNSW-NB15	89.17%	-
Ullah and Mahmoud (2021)	CNN	UNSW-NB15	99.60%	-
Ullah and Mahmoud (2022)	CNN	UNSW-NB15	99.92%	-
Proposed Method	AE + RF	UNSW-NB15	99.00%	0.78%

The Proposed Method

In this study, an intrusion dataset will be used first. Then, a dimensionality reduction method using Auto-Encoder will be utilized. Unlike the bio-inspired dimensionality reduction techniques which aim at selecting individual features, the proposed Auto-Encoder will accommodate a dimensionality reduction rather than selecting permutations. This can be depicted through a learning process where the original feature space will be processed as input and the same feature space will be predicted within the output. Hence, the middle layer also referred to as the hidden layer will represent the reduced dimension. Such a reduced dimension will be then examined through four different machine learning techniques including Decision Tree (DT), Logistic Regression (LR), Neural Network (NN), and Random Forest (RF). Figure 1 shows the framework of the proposed method. Lastly, the evaluation will take a place to assess the classification results.

Dataset

Apart from traditional intrusion detection datasets such as KDD-CUP99 and NSL-KDD, which simulated older networks, the dataset used in this study is focusing on a special type of network, this dataset called UNSW-NB15 introduced by (Moustafa and Slay, 2015). Such a dataset simulates both normal connections and intrusions that could target current networks such as Wireless Sensor Networks (WSN) and the Internet of Things (IoT). The primary difference between this dataset and previous ones is the introduction of new threats and attacks, such as Shellcode, which tries to exploit specific software in a specific network. The details of the dataset used are shown in Table 1, while Table 2 shows a sample of the dataset.

Data Preparation

The data contains around 40 features that capture specific characteristics of the network connection such as duration, packets sent and received along with other features. Most of these features are represented numerically yet, a few numbers of them have been represented in a discrete or categorical form. For example, the feature service determines the type of service used by such a connection such as FTP or HTTP. On the other hand, the protocol feature determines the protocol used by the connection such as TCP or UDP. Since the machine learning techniques are only handling numeric features, it is necessary to transform the discrete values into numeric ones. To do that, the binary representation can be used to articulate the categorical values. Let's assume a sample of the categorical feature from the dataset such as in Table 3.

To transform the categorical values of 'UDP', 'TCP', 'HTTP', and 'FTP', the categorical feature column will be divided into multiple columns that correspond to the number of these possible categorical values. Then, a binary representation of '1' and '0' will be utilized to indicate one of these values as shown in Table 4.

Dimensionality Reduction using Auto Encoder

Auto Encoder (AE) is one of the deep neural network architectures that aims to accommodate a dimensionality reduction (Balın *et al.*, 2019). This can be articulated where the original dimensionality of the data is being expanded into the larger dimension, this process is known as encoding. Then, another step is conducted where the encoding is decomposed into a smaller vector that is smaller than both the encoding and the original dimension, this process is known as coding. After that, a process known as decoding

will aim at expanding the coding vector into a larger dimension which is usually equivalent to the encoding. Lastly, the output will articulate the process of predicting the original dimension of the data.

This process of encoding, decoding, and coding aims at learning the features of the dataset by decomposing these features and then reconstructing them again. The coding vector represents the reduced dimension of the original dataset which can be used later for the classification (Zhang *et al.*, 2019). Figure 2 shows the architecture of the proposed AE.

As shown in Fig. 2, the architecture of the proposed AE is compromised of the same components of input, encoding, decoding, coding, and output.

In particular, the input is the original dimension of the dataset after categorical transformation which consists of 196 features. Consequentially, the encoding will examine a larger portion where the original dimension is multiplied by 2 which results in 388 feature dimensions. After that, a coding process will be examined where the encoding dimension will be decomposed into a very smaller portion that is even smaller than the original.

For this purpose, the encoding dimension of 388 will be divided by 8. Such coding will determine the actual significant feature dimension. Then, a reconstruction process will take a place where the coding will be decoded into a larger portion (multiplied by 8) which makes it correspond to the encoding dimension. Finally, the output articulates the prediction of the original dimension which is 194.

To put the dimensionality in context, let's assume the original dimensionality of the original dataset as a matrix of 194×257673 where 194 indicates the feature dimension and 257673 corresponds to the total number of connections (See Table 3.1). The encoding will expand this matrix horizontally where the feature dimension will be doubled which results in a matrix of 388×257673. After that, the coding will examine a smaller portion of the feature dimension where the encoding dimension will be divided by 8. This results in a matrix of 48×257673. This coding matrix will be then treated as the reduced dimension that would undergo the classification process. Figure 3 depicts these matrices in detail.

In particular, the proposed auto encoder contains several parameters and hyperparameters that need to be specified. First, it is worth discussing the architecture of the proposed autoencoder. In this regard, the proposed autoencoder is composed of five layers including input, encoding, coder, decoder, and output layers.

The first layer which is the input reflects the dimension of the feature space of the dataset after transformation where the size is 194. The second layer is the encoder where the input size is multiplied by 2 and the size becomes 388. Note that, the encoder is supplemented with two sub-layers of Batch-Normalization and Leaky-ReLU activation function. Both sub-layers aim at standardizing the values which contribute toward stabilizing the learning process.

The third layer is the coder where the expanded dimension will be reduced dramatically. This layer represents the core of the dimensionality reduction task within the autoencoder. For this purpose, the expanded dimension of 388 will be divided by 8 which results in a layer with a size of 48.

The fourth layer is the decoder in which the reduced dimension space is reconstructed to form the expanded dimension. In this regard, the reduced size of 48 will be multiplied by 8 which results in a layer with a size of 388. Once again, the decoder layer is supplemented with two sub-layers of Batch-Normalizer and Leaky-ReLU activation function.

Lastly, the fifth layer is the output layer in which the reconstructed and expanded dimension resulting in the decoder will be processed to predict the original dimension of the dataset. Therefore, the output layer would have a size of 194. Figure 3.6 represents the architecture of the proposed autoencoder.

After discussing the architecture of the proposed autoencoder, it is necessary to highlight the hyperparameters utilized by the autoencoder. Table 5 shows these hyperparameters.

Classification

As soon as the reduced matrix produced by AE is acquired, the classification process will take a place. This can be depicted where a classification algorithm will handle the reduced dimension to classify each connection into intrusion and normal classes. For this purpose, four classifiers will be used including DT, LR, NN, and RF. The reason behind using such classifiers is that they can cover different aspects of machine learning where four paradigms can be included tree-based classification, probabilistic-based classification, neural network-based, and linear-based classification.

To perform the classification, it is necessary to split the data into two sets including a training set and a testing set. Hence, the reduced dimension matrix will be divided into two sets vertically where the total number of connections (i.e., 257673) will be divided into a training set of 175341 connections and a testing set of 82332. This ratio of division for the training and testing has been followed by the original study of the dataset (Moustafa and Slay, 2015).

Evaluation

Once the classification task by the four classifiers is done, this phase will take a place to assess the classification results. To this end, it is imperative to consider the confusion matrix of the classes as shown in Table 6.

As shown in Table 6, TP indicates the correctness of prediction when the connection is an intrusion and it has been classified as an intrusion. Similarly, TN indicates the correctness of prediction when the connection is non-intrusion and it has been classified as non-intrusion. Whereas, FN indicates the incorrectness of prediction when the connection is intrusion it has been classified into non-intrusion. Similarly, FP indicates the incorrectness of prediction when the connection is non-intrusion but it has been classified as an intrusion.

According to Kolias *et al*. (2015), two main metrics can be used to assess the intrusion detection task which is Accuracy and False Alarm Rate (FAR). Both of them can be calculated using the aforementioned variables of TP, FP, FN, TN, Precision, and Recall as follows:

$$\text{Pr}ecision = TP / TP + FP \quad (1)$$

$$\text{Re}call = TP / TP + FN \quad (2)$$

$$F1\text{–}score = 2 \times Precision \times \text{Re}call / Precision + Recall \quad (3)$$

$$FAR = FP / total\ number\ of\ connection \quad (4)$$

A comparison will be conducted against the state-of-the-art studies including Ullah and Mahmoud (2019) and Roopak *et al*. (2020) who used traditional feature selection techniques. The following pseudo-code shows the steps of the proposed method.

Results

Model Fitness

Before the classification evaluation, it is necessary to assess the performance of the autoencoder. For this purpose, the model of the autoencoder will be assessed based on the loss function. Such assessment aims at detecting underfitting or overfitting. Underfitting refers to the cases where the model produces a high error rate during training and testing. Whereas overfitting refers to the cases where the model produces a low error rate during training, meanwhile, produces a high error rate during testing. Both underfitting and overfitting indicate that the model is not performing well. Therefore, it is worth examining the model of the autoencoder by comparing the error rate (i.e., loss) during training and testing. Figure 4 shows the results of loss for both training and testing.

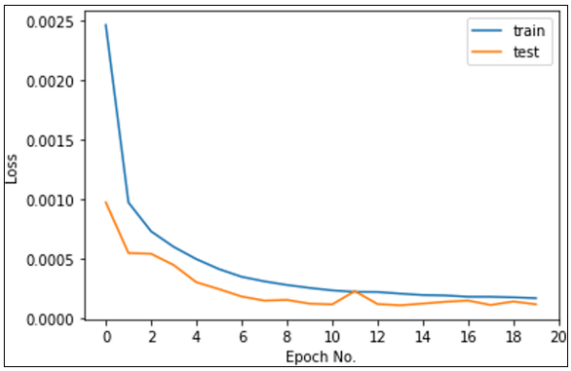

Fig. 4: Loss function through the iterations

As shown in Fig. 4, the error rate in training begins with a relatively high value. Then, the model showed an ongoing decline in error rate through the iterations (i.e., epochs) until reaching a very low value of error rate. Similarly, in testing the model begins with a relatively high value and then shows a decrease in error rate until reaching a very low value of error. Both training and testing loss values are relatively similar which indicates the absence of both underfitting and overfitting. This implies the efficacy of the proposed autoencoder.

Results of Classifiers

In this section, the results of the four classifiers will be highlighted. Such results will be depicted based on the weighted average F1 score and FAR. Table 7 shows the comparison.

As shown in Table 7, the highest weighted average F1 score was depicted by the RF classifier achieving 99%. As well as, the lowest FAR value was depicted by RF where it got 0.78%. This implies the efficacy of the RF classifier in terms of classifying intrusions. This was followed by the DT classifier which achieved a weighted average F1 score of 97% along with a FAR value of 1.89%. After that, the results of NN come with a weighted average F1 score of 92% and a FAR value of 2.01%. The lowest weighted average F1 score obtained by the LR where it has 89%, meanwhile, LR got the highest FAR value of 9.21%.

Discussion

In this section, the highest results of the F1-score and FAR within this study will be compared against the baseline studies. Table 8 shows such a comparison.

As depicted in Table 8, the proposed autoencoder with RF classifier showed a competitive F1 score compared to other studies. Although some studies such as those (Ullah and Mahmoud, 2021; Ullah and Mahmoud, 2022) have managed to achieve a high F1-score (i.e., ~ 99%) yet, they have not considered the FAR rate. The proposed method demonstrated a very low FAR value. This emphasizes the efficacy of autoencoder in terms of dimensionality reduction

compared to the traditional approaches such as Genetic Algorithm and others.

Conclusion

This study has implemented an autoencoder architecture as a dimensionality reduction approach for IoT intrusion detection. Such a reduction has led to the accumulation of the most significant features associated with intrusions. Consequentially, this study has utilized the reduced dimension with four classifiers including DT, LR, NN, and RF to conduct the intrusion detection task. A comparison between the classifiers has been conducted where the RF classifier had superior performance. Lastly, a comparison against the baseline studies has been conducted and showed an outperformance of the proposed autoencoder in terms of dimensionality reduction in IoT intrusion detection.

Examining deep learning classifiers such as Convolutional Neural Network or Long Short Term Memory (LSTM) within future research could contribute toward enhancing the classification accuracy.

Acknowledgment

This study has been supported by the Universiti Putra Malaysia.

Author's Contributions

Ahmed Ridha Khudhur: Participated in all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

Khairulmizam Samsudin: Planned the experiments and contributed to the interpretation of the results and review the manuscript.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Balın, M. F., Abid, A., & Zou, J. (2019, May). Concrete autoencoders: Differentiable feature selection and reconstruction. *In International conference on machine learning* (pp. 444-453). *PMLR*.

Gharaee, H., & Hosseinvand, H. (2016, September). A new feature selection IDS based on genetic algorithm and SVM. *In 2016 8th International Symposium on Telecommunications (IST)* (pp. 139-144). IEEE. https://doi.org/10.1109/ISTEL.2016.7881798

Hajisalem, V., & Babaie, S. (2018). A hybrid intrusion detection system based on the ABC-AFS algorithm for misuse and anomaly detection. *Computer Networks*, 136, 37-50. https://doi.org/10.1016/j.comnet.2018.02.028

Hanif, S., Ilyas, T., & Zeeshan, M. (2019, October). Intrusion detection in IoT using artificial neural networks on UNSW-15 dataset. In 2019 IEEE 16th international conference on smart cities: *Improving the quality of life using ICT & IoT and AI (HONET-ICT)* (pp. 152-156). *IEEE*. https://doi.org/10.1109/HONET.2019.8908122

Khaliq, S. (2020). Intrusion Detection Survey: *A Survey and Taxonomy*. https://doi.org/10.20944/preprints202006.0065.v1

Khammassi, C., & Krichen, S. (2017). A GA-LR wrapper approach for feature selection in network intrusion detection. *Computers & Security*, 70, 255-277. https://doi.org/10.1016/j.cose.2017.06.005

Khan, W. Z., Rehman, M. H., Zangoti, H. M., Afzal, M. K., Armi, N., & Salah, K. (2020). Industrial internet of things: Recent advances, enabling technologies and open challenges. *Computers & Electrical Engineering*, 81, 106522. https://doi.org/10.1016/j.compeleceng.2019.106522

Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2015). Intrusion detection in 802.11 networks: An empirical evaluation of threats and a public dataset. *IEEE Communications Surveys & Tutorials*, 18(1), 184-208.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2019). Shallow neural network with kernel approximation for prediction problems in highly demanding data networks. *Expert Systems with Applications*, 124, 196-208.
https://doi.org/10.1016/j.eswa.2019.01.063

Masdari, M., & Khezri, H. (2020). A survey and taxonomy of the fuzzy signature-based intrusion detection systems. *Applied Soft Computing*, 92, 106301. https://doi.org/10.1016/j.asoc.2020.106301

Mogal, D. G., Ghungrad, S. R., & Bhusare, B. B. (2017). NIDS using machine learning classifiers on UNSW-NB15 and KDDCUP99 datasets. *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, 6(4), 533-537.

Moustafa, N., & Slay, J. (2017). A hybrid feature selection for network intrusion detection systems: Central points. *arXiv preprint arXiv*:1707.05505. https://doi.org/10.48550/arXiv.1707.05505

Moustafa, N., & Slay, J. (2015, November). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *In the 2015 military communications and information systems conference* (MilCIS) (pp. 1-6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Nauman, A., Qadri, Y. A., Amjad, M., Zikria, Y. B., Afzal, M. K., & Kim, S. W. (2020). Multimedia Internet of Things: A comprehensive survey. *IEEE Access*, 8, 8202-8250. https://doi.org/10.1109/ACCESS.2020.2964280

Papamartzivanos, D., Mármol, F. G., & Kambourakis, G. (2018). Dendron: Genetic trees drove rule induction for network intrusion detection systems. *Future Generation Computer Systems*, 79, 558-574. https://doi.org/10.1016/j.future.2017.09.056

Qadri, Y. A., Nauman, A., Zikria, Y. B., Vasilakos, A. V., & Kim, S. W. (2020). The future of healthcare internet of things: A survey of emerging technologies. *IEEE Communications Surveys & Tutorials*, 22(2), 1121-1167. https://doi.org/10.1109/COMST.2020.2973314

Roopak, M., Tian, G. Y., & Chambers, J. (2020). Multi-objective-based feature selection for DDoS attack detection in IoT networks. *IET Networks*, 9(3), 120-127. https://doi.org/10.1049/iet-net.2018.5206

Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., & Markakis, E. K. (2020). A survey on the Internet of Things (IoT) forensics: Challenges, approaches, and open issues. *IEEE Communications Surveys & Tutorials*, 22(2), 1191-1221. https://doi.org/10.1109/COMST.2019.2962586

Tama, B. A., & Rhee, K. H. (2019). An in-depth experimental study of anomaly detection using gradient boosted machine. *Neural Computing and Applications*, 31(4), 955-965. https://doi.org/10.1007/s00521-017-3128-z

Ullah, I., & Mahmoud, Q. H. (2021). Design and development of a deep learning-based model for anomaly detection in IoT networks. *IEEE Access*, 9, 103906-103926. https://doi.org/10.1109/ACCESS.2021.3094024

Ullah, I., & Mahmoud, Q. H. (2019, January). A two-level hybrid model for anomalous activity detection in IoT networks. *In 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)* (pp. 1-6). IEEE. https://doi.org/10.1109/CCNC.2019.8651782

Ullah, I., & Mahmoud, Q. H. (2022, January). An Anomaly Detection Model for IoT Networks based on Flow and Flag Features using a Feed-Forward Neural Network. *In 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)* (pp. 363-368). *IEEE*. https://doi.org/10.1109/CCNC49033.2022.9700597

Zhang, C., Liu, Y., & Fu, H. (2019). Ae2-nets: Autoencoder in autoencoder networks. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2577-2585). https://doi.org/10.1109/COMST.2015.2402161