A Feature Selection Method Based on Auto-Encoder for IoT Intrusion Detection

Ahmed Fahad Alshudukhi, Hashimi Sallehudin

Faculty of Information Science and Technology, University Kebangsaan Malaysia
43600 Bangi, Selangor Darul Ehsan, Malaysia

Email: p103545@siswa.ukm.edu.my, hasimi@ukm.edu.my

ABSTRACT

The dramatic evolution in gadgets where wide range of devices have become connected to the internet such as sensors, cameras, smartphones and others, has led to the emergence of Internet-of-Things (IoT). As any network, IoT is facing a challenging issue represented by the security. Several research studies have addressed the intrusion detection task in IoT. Most of them have concentrated on determining robust set of features which may contribute toward improving the accuracy of intrusion classification based on statistical and bio-inspired feature selection techniques. Deep learning is a family of techniques that demonstrated remarkable performance in the field of classification. The emergence of deep learning techniques has led to configure new neural network architectures that is designed for the feature selection task. This study proposes a deep learning architecture known as Auto-Encoder (AE) for the task of feature selection in IoT intrusion detection. A benchmark dataset for IoT intrusions has been considered in the experiments. In addition, multiple normalization tasks have been applied to transform the data into an appropriate format for processing. After that, the proposed AE has been carried out for the feature selection task along with a simple Neural Network (NN) architecture for the classification task. Experimental results showed that the proposed AE showed an accuracy of 99.97% with a False Alarm Rate (FAR) of 1.0. Comparing these results against the ones obtained by the related work proves that the AE has superior performance over the statistical and bio-inspired feature selection techniques.

Key words: Intrusion Detection, Feature Selection, Neural Network, Auto Encoder, Deep Learning

INTRODUCTION

The last decade has witnessed a dramatic evolutions in gadgets where wide range of devices have become connected to the internet such as sensors, cameras, smartphones and others (Da Xu *et al.*, 2014). Such evolution has led to the emergence of Internet-of-Things (IoT) as a new research area that explores the utilization of the massive number of connected devices in order to perform specific tasks (Al-Fuqaha *et al.*, 2015). One of the IoT applications is the smart house that can take the advantage of cameras, sensors and smartphone to build an intelligent system for alerting the owners regarding suspicious and emerging events that could happen during his absence. In addition, a framework for a smart hospital is also proposed by utilizing medical devices to determine the priority and emergency list of patients (Eskofier *et al.*, 2017). This massive evolution of technology has brought numerous challenges, one of the concerning challenges is the security. The protection of IoT network from traditional threats or intrusions such as viruses, worms, Trojan horses and others is the main challenge. The security is representing an essential demand especially if the IoT network is related to medical or private agencies which makes the violation of private information is intolerable (Ullah and Mahmoud, 2019). In fact, Intrusion Detection (ID) is a research field that is examining the identification of any abnormal activity conducted in certain networks (Mishra *et al.*, 2019). ID has been investigated extensively in the last two decades where wide range of techniques have been proposed for the detection task. However, the intrusions on specified networks such as the IoT would have different characteristics which requires new techniques that can address these differences.

One of the significant techniques that can address the characteristics of IoT intrusion detection is the feature selection where the aim is to analyse the features of IoT intrusions in order to identify the

Copyright@FTSM

most important subset of features. With the release of UNSW-NB15 dataset (Moustafa and Slay, 2015) which has been dedicated for monitoring the intrusions in IoT networks, numerous researches have been proposed for the feature selection purposes (Hajisalem and Babaie, 2018; Khammassi and Krichen, 2017; Papamartzivanos *et al.*, 2018; Tama and Rhee, 2019). Yet, the techniques used in such studies are still require improvement in terms of the accuracy or time consuming of determining the best solution. This is because these techniques suffer from specific limitations regarding the identification of robust subset of features. Several researchers have proposed feature selection techniques for this purpose. Some of these researches have utilized the traditional methods such as the Apriori and Association Rules (Mogal et al. 2017; Moustafa & Slay 2017). Other researchers have used the meta-heuristic methods such as Genetic Algorithm and Swarm-based (Hajisalem & Babaie 2018; Papamartzivanos et al. 2018; Tama & Rhee 2019).As noticed from the state of the art in feature selection for IoT intrusion detection, most of the studies have relying on traditional methods such as the rule-based and meta-heuristic. The drawback behind these methods lies on its inability to find optimal solution where the best subset of the features can be identified. Apart from the feature selection techniques, the classification methods used in literature for detecting intrusions are still facing some limitations regarding the performance of detection. This is because most of the classifiers used in the literature were standard such as Support Vector Machine (SVM), Naïve Bayes (NB) or Decision Tree (DT). These classifiers do not have extensive training paradigm like in Neural Network (NN) where an error-tuning procedure is considered. This has caused a limitation in achieving high detection accuracy with low False Alarm Rate (FAR).

RELATED WORK

Recently, many researchers have examined the feature selection in IoT detection for example, Gharaee & Hosseinvand (2016) have examined the problem of dimensionality of feature space within the intrusion detection in IoT. The authors have focused on the challenging task of reducing the false positive rate within the intrusion detection. For this purpose, the authors have proposed a combination of Genetic Algorithm (GA) as a feature selection/reduction technique along with Support Vector Machine (SVM) classifier. The dataset used in the experiments was UNSW-NB15 in which the average accuracy of detection was 93.25% with a FAR of 8.6. Similarly, Khammassi & Krichen (2017) have proposed a feature selection approach based on a wrapper technique. The authors have attempted to identify the most significant features that might impact the accuracy of intrusion detection. Therefore, a wrapper technique has been used where a Genetic Algorithm is being used as a feature selection approach with Decision Tree (DT) as a classification method. The dataset used in the experiments was UNSW-NB15 where the best subset of features has acquired an accuracy of 81.42% with a FAR of 6.39.

Apart from the traditional meta-heuristic feature selection approaches, Moustafa & Slay (2017), have proposed an Association Rule Mining technique for the feature selection/reduction in IoT intrusion detection. The proposed method has concentrated on central points of significant attributes that impact the detection of intrusion. The dataset used in the experiments was UNSW-NB15 where the average accuracy obtained by the proposed method was 83% with a FAR of 14.2. Similarly, Mogal et al. (2017) have utilized the Apriori algorithm in order to determine the most significant features within IoT intrusion detection. The proposed algorithm has conducted to rank the features based on its significance where the irrelevant ones will be dismissed. After that, two classifiers of Naïve Bayes and Logistic Regression have been used to classify the data instances based on the selected features. The dataset of UNSW-NB15 has been used where the average accuracy obtained by the proposed method was 90% with a FAR of 10.5.

Another study that addressed the feature selection in IoT intrusion detection conducted by Papamartzivanos et al. (2018) where a combination of Genetic Algorithm and Decision Tree has been proposed for this purpose. GA has been applied in order to make rule induction for the rules produced by the DT. As all the studies on IoT intrusion detection, the UNSW-NB15 dataset has been used in the experiments. Results of accuracy for the best subset of features showed 84.33% with a FAR of 8.9. In the same regard, Hajisalem & Babaie (2018) have proposed a combination of Artificial Bee Colony (ABC) and Artificial Fish Swarm (AFS) algorithms in order to accommodate a holistic feature selection task on IoT intrusion detection. The authors have took the advantage of the two algorithms in order to find the best solution of features. Finally, an Association Rule classifier of CART has been used to classify the intrusion based on the selected features. UNSW-NB15 dataset has been used in the experiments with an average accuracy of 85% with a FAR of 14.9. On the other hand, some authors have used the feature selection approaches in order to improve the classifiers themselves in IoT intrusion detection. For example, Tama & Rhee (2019) have proposed a grid search algorithm in order to search for the best parameters of classifiers. In fact, every classifier its own parameters, and

Copyright©FTSM

sometimes, it is difficult to examine every parameter individually. Therefore, the proposed grid search has been used to identify the best parameters for three classifiers including Neural Network, Support Vector Machine and Fuzzy classifier. Results showed that the proposed grid search has improved all the classifiers in which the combination of grid search and neural network has got the highest accuracy on UNSW-NB15 dataset where the average accuracy was 82.6% with a FAR of 16.2.

Ullah & Mahmoud (2019) have proposed a linear method for the feature selection which is called Recursive Feature Elimination (RFE) for the task of IoT intrusion detection. The proposed method will iteratively divide the feature space into much smaller subsets and recursively evaluate each feature. UNSW-NB15 dataset has been used in the experiment and the average accuracy obtained was 97% with a FAR of 7.8.

MATERIALS AND METHODS

The framework of the proposed method consists of five components. The first component would be the dataset where the details of such dataset and other statistics can be tackled. The second component of the framework would be normalization tasks where the attributes would be filtered and transformed. Since this study is examining the task of feature selection within intrusion detection therefore, the third component in the framework would be a technique that performs feature selection. For this purpose, the proposed Auto-Encoder will be represented in this phase where the normalized data passed from the previous component will be used to train the auto-encoder on determining significant features. On the other hand, any feature selection technique (especially wrapping ones) requires the use of a classifier in order to test the subset of features selected by the feature selection technique. Hence, the fourth component would be the classification task. For this purpose, Neural Network (NN) classifier will be used to classify the connection data into intrusion and non-intrusion based on the selected features by the proposed auto-encoder form the previous component. For any machine learning classification tasks, it is important to conduct an evaluation process in order to assess the learning capabilities of the proposed classifier. In this regard, the fifth component would be the evaluation where the results of NN classifier can be validated based on the number of correctly and incorrectly classified connections. Figure 1 shows the components of the framework.

Copyright@FTSM

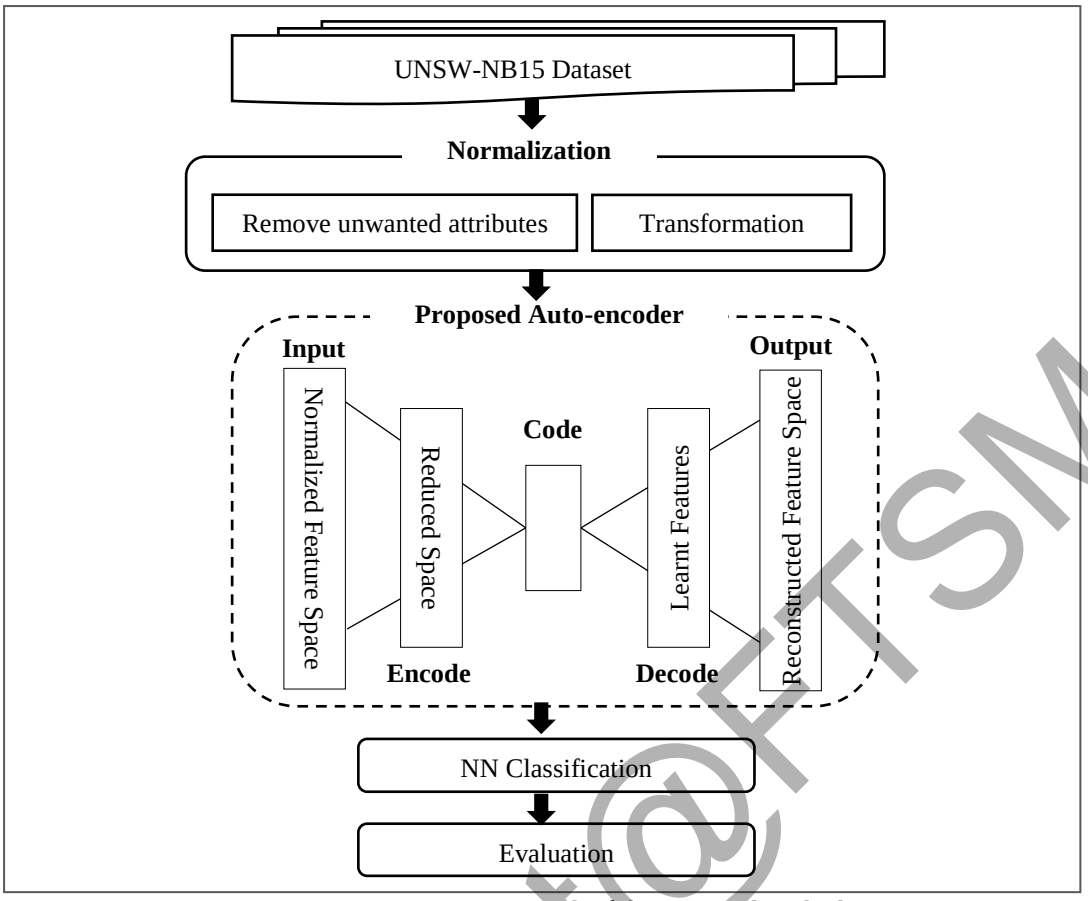

Figure 1. Framework of the proposed method

The rest of sections in this chapter will depict each component from the framework separately.

Dataset

This stage aims to utilize the benchmark dataset of UNSW-NB15. Unlike previous datasets of intrusion detection such as KDD-CUP99 and NSL-KDD where the simulation was conducted using traditional networks, UNSW-NB15 dataset is a simulation for both normal connections and intrusions that might target modern networks such as Wireless Sensor Networks (WSN) and Internet-of-Things (IoT) (Hanif *et al.*, 2019). The key distinguish between this dataset from the previous ones lies on the new threats and attacks that have been introduced such as Shellcode which aims to exploit specific software in a particular network. The dataset is composed of 43 features which can be described as in Table 1.

Table 1 Features of UNSW-NB15 dataset

No.	Type	Feature	Description
1.	Flow Features	srcip	Source IP address
2.		sport	Source port number
3.		dstip	Destination IP address
4.		dsport	Destination port number
5.		proto	Transaction protocol
6.	Basic Features	state	Indicates to the state and its dependent protocol
7.		dur	Record total duration
8.		sbytes	Source to destination transaction bytes
9.		dbytes	Destination to source transaction bytes
10.		sttl	Source to destination time to live value
11.		dttl	Destination to source time to live value
12.		sloss	Source packets retransmitted or dropped
13.		dloss	Destination packets retransmitted or dropped
14.		service	Service used such as http, ftp, smtp, or others
15.		Sload	Source bits per second
16.		Dload	Destination bits per second
17.		Spkts	Source to destination packet count

Copyright@FTSM

18.		Dpkts	Destination to source packet count
19.	Content Features	swin	Source TCP window advertisement value
20.		dwin	Destination TCP window advertisement value
21.		stcpb	Source TCP base sequence number
22.		dtcpb	Destination TCP base sequence number
23.		smeansz	Mean of packet size transmitted by the source
24.		dmeansz	Mean of packet size transmitted by the destination
25.		trans_depth	Represents the pipelined depth into the connection of http request/response
26.		res_bdy_len	Actual uncompressed content size of the data transferred from the server's
27.	Time Features	Sjit	Source jitter (mSec)
28.		Djit	Destination jitter (mSec)
29.		Stime	record start time
30.		Ltime	record last time
31.		Sintpkt	Source interpacket arrival time (mSec)
32.		Dintpkt	Destination interpacket arrival time (mSec)
33.		tcprtt	TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.
34.		synack	TCP connection setup time, the time between the SYN and the SYN_ACK
35.		ackdat	TCP connection setup time, the time between the SYN_ACK and the ACK
36.	Connection Features	is_sm_ips_ports	If source (1) and destination (3) IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0
37.		ct_state_ttl	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38.		ct_flw_http_mthd	No. of flows that has methods such as Get and Post in http service.
39.		is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.
40.		ct_ftp_cmd	No of flows that has a command in ftp session.
41.		ct_srv_src	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42.		ct_srv_dst	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43.		ct_dst_ltm	No. of connections of the same destination address (3) in 100 connections according to the last time (26).

Normalization

Unlike old datasets such as KDD-CUP99 or NSL-KDD where numerous noisy data is being located along with redundant records, the UNSW-NB15 dataset has been carefully designed. However, there are still some issues need to be tackled in such dataset. Therefore, this section aims to examine these issues. Table 2 shows a sample of connections from the dataset.

Table 2 Sample of connections from UNSW-NB15 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

As shown in Table 2, multiple connections brought from the dataset. First observation would reveal that there are various features for each connection for example, the ID of such connection, its protocol, its service and the duration of the connections. Lastly, there is a column for the class label where the connection is being categorized into 'normal' or any intrusion classes such as 'exploits' or 'DoS'. Another attribute related to the class is located also which is the 'Binary Class'. Such attribute contains only two values either '0' for normal connection, or '1' for the intrusion classes. Now, some attributes are not needed within the machine learning processing such as the ID in which the ID cannot indicate the status of any connection. In addition, the datatypes within the features are vary, this can hinder the machine learning from gaining a good training of such features. Hence, some normalization tasks are needed, following subsections are tackling these tasks.

Remove Unnecessary Attributes

As mentioned earlier, there are some attributes that do not have any importance in terms of identifying the status of a connection. The first attribute is the ID in which the identification number of the connection would not have any significance in terms of determining the connection status. In

Copyright©FTSM

addition, the 'Binary class' attribute is also unnecessary because it has only binary values (0 for normal connection and 1 for intrusion). In order to train the machine learning adequately, all the classes should be fed. Therefore, the aforementioned attributes must be removed. Table 3 and Table 4 depict the data before and after removing the unnecessary attributes.

Table 3 Removing unnecessary attributes

~~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 4 Data after removing unnecessary attributes

Protocol	Service	Duration	….	Class
TCP	FTP	0.121478		Normal
TCP	HTTP	0.649902		Normal
UDP	HTTP	1.623129		Exploits
TCP	HTTP	1.681642		Normal
UDP	FTP	0.449454		DoS

Note that, the number of features after removing the unnecessary attributes is 43 along with one attribute for the class label.

Attribute Transformation

As mentioned earlier, the features contain variant datatypes where some attributes consist of numeric values (e.g. 0.12), while other attributes consist of nominal values (e.g. 'FTP' and 'TCP'). For adequate feature learning in MLT, it is important to transform the attributes. In this regard, the one-hot encoding approach is being used to turn the nominal values into numeric (Seger, 2018). Table 3.5 shows an example of applying one-hot encoding on the data in Table 5.

Table 5 Example of applying one-hot encoding

Protocol_TCP	Protocol_UDP	Service_FTP	Service_HTTP	Duration	….	Class
1	0	1	0	0.121478		Normal
1	0	0	1	0.649902		Normal
0	1	0	1	1.623129		Exploits
1	0	0	1	1.681642		Normal
0	1	1	0	0.449454		DoS

As shown in Table 5, the one-hot encoding was intended to examine all the possible values in nominal attribute and then, turn these values into independent/additional attributes. For example, the 'Protocol' attribute was containing two values including 'TCP' and 'UDP' thus, the attribute has been divided into two attributes including 'Protocol_TCP' and 'Protocol_UDP'. Once the nominal attributes are being splitted based on its values, the matching value will be filled with '1' while the mis-match will be represented as '0'. In this way, the datatype of all attributes will be unified into numeric values. Note that, after splitting the nominal attributes, the number of features has been increased into 196 attributes.

Feature Selection Using Auto-Encoder

Auto-Encoder (AE) is one of Neural Network architectures which has unique and customized layers. Before discussing AE, it is necessary to describe the original neural network and how it works.

Neural Network

Let consider the sample data in Table 5, in order for the neural network to process the data, it is important to turn all the attributes into numerical values. Since the features have been turned into numeric, the class label should be converted as in Table 6.

Table 6 Encoding class label for neural network processing

Protocol_TCP	Protocol_UDP	Service_FTP	Service_HTTP	Duration	Class
1	0	1	0	0.121478	Normal (0)

Copyright@FTSM

1	0	0	1	0.649902	Normal (0)
0	1	0	1	1.623129	Exploits (1)
1	0	0	1	1.681642	Normal (0)
0	1	1	0	0.449454	DoS (2)

As shown in Table 6, the three classes Normal, Exploits and DoS have been converted into '0', '1' and '2'.

Apparently, any neural network would have three main layers including input, hidden and output. The input is the layer that takes the features of a connection, while the output layer would represent the class label of the connection. However, the hidden layer is the part of neural network where the features are being analysed to find the deep relationship among them. Now let us input the first connection in Table 6 to a simple neural network as in Figure 2.

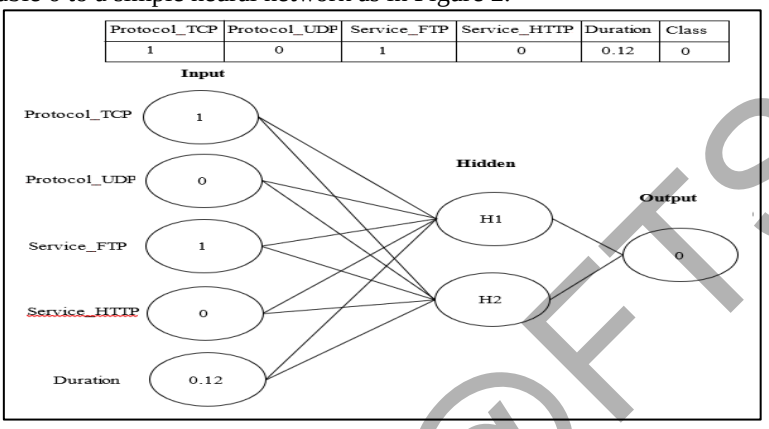

Figure 2 Simple neural network

Auto-Encoder (AE)

After describing the original neural network, now we can discuss the AE algorithm. In fact, the main aim behind AE is to learn a compressed and distributed representation of a given data (Mighan and Kahani, 2018). In other words, AE aims to process a data as input and output the same data itself as shown in Figure 5.

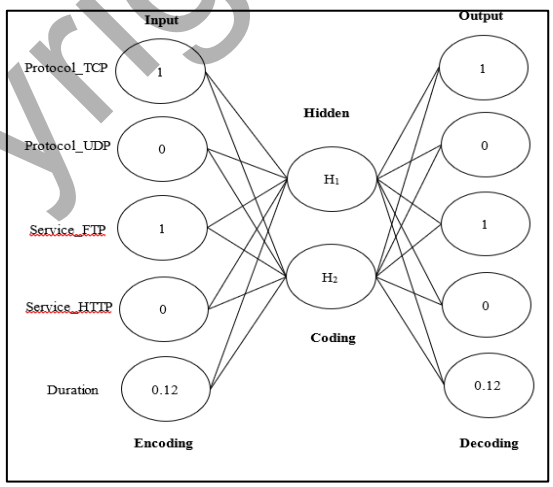

Figure 3 AE architecture

As shown in Figure 3, the input of AE are the features of a connection, while the output are the same values of the features. The first layer is also known as encoding within the AE where the data is being encoded until getting the coding and then, decoding the data. In order to understand such mechanism, let recall the standard neural network (described in the previous subsection). The input weights will be initiated with random values, then the hidden will be computed using ConcreteAutoEncoder package. After that, the hidden weights will be initiated with random values to compute the output. After considering an activation function, the predicted output will be compared against the actual output to calculate the error. If there is error, the Backpropagation will be used to

Copyright@FTSM

reduce the error rate until predicted output corresponds the actual output. Once the error is being minimized to zero where precited output is identical to the actual output, the hidden neuron values will be considered as the selected and reduced feature space. In fact, setting the hidden layer size is a challenging issue. Therefore, in this study different size values will be experimenting to find the most accurate one.

Neural Network Classification

After acquired the selected features by the proposed AE, a simple neural network will be used to classify the connections into intrusion and normal. The input of this neural network is the set of selected features produced by the proposed AE.

RESULT

The results of classification have been evaluated using accuracy and False Alarm Rate (FAR). Table 8 shows the results.

Table 7. Experimental results

Epoch	Features =30		Features = 20		Features = 10		Feature = 5		Feature = 4	
No.	Accuracy	FAR	Accuracy	FAR	Accuracy	FAR	Accuracy	FAR	Accuracy	FAR
100	0.0780	26.52	0.1318	57.45	0.1269	57.12	0.1433	57.45	0.0103	57.45
200	0.3421	6.84	0.1715	32.63	0.1857	32.45	0.1844	32.63	0.0122	32.63
300	0.5836	1.76	0.2388	18.53	0.2413	18.43	0.2286	18.53	0.0292	18.53
400	0.5664	1.0	0.6516	10.53	0.6820	10.53	0.7205	10.53	0.1347	10.53
500	0.5471	1.0	**0.8037**	5.98	0.7498	5.98	0.7977	5.98	0.9834	5.98
600	**0.6782**	1.0	0.7913	3.39	0.7493	5.94	0.8014	3.39	0.9972	3.39
700	0.6697	1.0	0.7605	1.93	**0.8145**	3.39	0.8717	1.93	0.9990	1.93
800	0.6682	1.0	0.7520	1.09	0.7027	1.91	0.8944	1.09	0.9993	1.09
900	0.6631	1.0	0.7514	1.0	0.7032	1.09	0.8954	1.0	0.9995	1.0
1000	0.6504	1.0	0.7520	1.0	0.7035	1.0	0.8977	1.0	0.9996	1.0
1100	0.6410	1.0	0.7525	1.0	0.7038	1.0	0.9002	1.0	0.9996	1.0
1200	0.6384	1.0	0.7529	1.0	0.7037	1.0	**0.9033**	1.0	**0.9997**	1.0

As shown in Table 8 the accuracies for the three feature numbers have been increased as the number of epochs were increased. However, the highest accuracy depicted by 10 number of features where the accuracy was 81.45%.

In terms of FAR as shown in Table 8, all the number of features showed similar rates of FAR in which the values have been decreased as the number of epochs increased where the minimal value of FAR was 1.0.

This can demonstrate that the best number of features selected was 10 where it has the highest accuracy. Therefore, the best choice is to examine lower dimension of features. Hence, next section will depict such examination.

As shown in Table 8, when number of features was 4, the accuracy has reached to 99.97% compared to the maximum accuracy obtained when the number of features was 5 which is 90.33%. This can prove that 4 number of features is the most accurate reduction of the features produced by AE. Finally, for FAR, both number of features showed similar performance where the minimal value of FAR was 1.0.

CONCLUSION

This study proposed the Auto-encoder as a feature selection approach in IoT intrusion detection in order to improve the accuracy of classification by enhancing the feature learning. A benchmark dataset of IoT intrusions has been considered in the experiments. In addition, different normalization tasks have been conducted including irrelevant attribute removal and converting categorical attributes into numeric ones. After that, the proposed AE has been carried out where the connection features have been processed as an input and the same features have been processed as an output. Within the hidden layer, the reduced feature space or the selected features is being acquired. Based on such selected features, a simple neural network will be fed to classify the connection into its class label.

Copyright@FTSM

Using various hidden size for the proposed AE architecture, results of accuracy and FAR showed superiority for the proposed AE over the traditional feature selection techniques where the best results have been obtained when the hidden size was 4 by achieving an accuracy of 99.97% with a 1.0 of FAR.

REFERENCES

Al-Fuqaha, et al. 2015. Internet of things: A survey on enabling technologies, protocols, and applications, IEEE Communications Surveys & Tutorials, Vol. 17 No. 4, pp. 2347-2376 (Access 2015

Da Xu, et al. 2014. Internet of things in industries: A survey, IEEE transactions on industrial informatics, Vol. 10 No. 4, pp. 2233-2243 (Access 2014

Eskofier, et al. 2017. An Overview of Smart Shoes in the Internet of Health Things: Gait and Mobility Assessment in Health Promotion and Disease Monitoring, Applied Sciences, Vol. 7 No. 10, pp. 986 (Access 2017

Gharaee and Hosseinvand. 2016. A new feature selection IDS based on genetic algorithm and SVM, *2016 8th International Symposium on Telecommunications (IST)*, pp. 139-144.

Hajisalem and Babaie. 2018. A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection, Computer Networks, Vol. 136, pp. 37-50 http://www.sciencedirect.com/science/article/pii/S1389128618301014 (Access 2018

Hanif, et al. 2019. Intrusion Detection In IoT Using Artificial Neural Networks On UNSW-15 Dataset, *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT)*, pp. 152-156.

Khammassi and Krichen. 2017. A GA-LR wrapper approach for feature selection in network intrusion detection, Computers & Security, Vol. 70, pp. 255-277 http://www.sciencedirect.com/science/article/pii/S0167404817301244 (Access 2017

Mighan and Kahani. 2018. Deep Learning Based Latent Feature Extraction for Intrusion Detection, *Iranian Conference on Electrical Engineering (ICEE)*, pp. 1511-1516.

Mishra, et al. 2019. A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection, IEEE Communications Surveys & Tutorials, Vol. 21 No. 1, pp. 686-728 (Access 2019

Mogal, et al. 2017. NIDS using machine learning classifiers on UNSW-NB15 and KDDCUP99 datasets, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 6 No. 4, pp. 533-537 (Access 2017

Moustafa and Slay. 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), *2015 military communications and information systems conference (MilCIS)*, IEEE, pp. 1-6.

Moustafa and Slay. 2017. A hybrid feature selection for network intrusion detection systems: Central points, arXiv preprint arXiv:1707.05505, (Access 2017

Papamartzivanos, et al. 2018. Dendron : Genetic trees driven rule induction for network intrusion detection systems, Future Generation Computer Systems, Vol. 79, pp. 558-574 http://www.sciencedirect.com/science/article/pii/S0167739X16305465 (Access 2018

Seger. 2018. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.

Tama and Rhee. 2019. An in-depth experimental study of anomaly detection using gradient boosted machine, Neural Computing and Applications, Vol. 31 No. 4, pp. 955-965 https://doi.org/10.1007/s00521-017-3128-z (Access 2019

Ullah and Mahmoud. 2019. A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks, *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1-6.

Copyright of FTSM