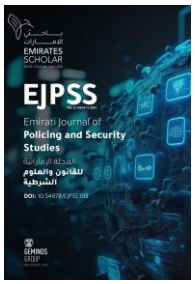


Emirati Journal of Policing and Security Studies
Vol 2 Issue 1 (2023) 4 - 12
DOI: 10.54878/EJPSS.298

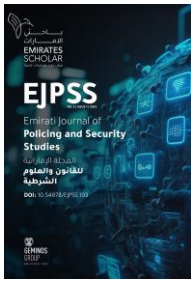
Available at www.emiratesscholar.com

Case Study on the Application of Deep Learning to Network Intruder Detection

Chithik R. Mohamed
University of Technology and Applied Sciences Salalah, Oman
chithik43@gmail.com (ESID 3383 5360 2023)

Abstract
Deep learning has seen considerable success in several application sectors. Unfortunately, little research has been done on its efficacy in the context of network intrusion detection. This article includes case studies that use deep learning to identify network anomalies both supervised and unsupervised. It has been demonstrated that deep neural networks (DNNs) outperform current machine learning-based intrusion detection systems in the presence of shifting IP addresses. We also demonstrate how auto encoders can support network anomaly detection.

Keywords:
Deep Learning, Intruder Detection System, Deep Neural Network, threshold, Auto encoder, Anomaly Detection, Host-based intruder.



Emirati Journal of Policing and Security Studies
Vol 2 Issue 1 (2023) 4 - 12
DOI: 10.54878/EJPSS.298

Available at www.emiratesscholar.com

Introduction

Organizations need to constantly adapt to deal with the dynamic nature of developing cyber threat actors as the scope of cyberattack and the volume of network data both grow tremendously. The quantity of security events being generated keeps going up as more security tools and sensors are put in contemporary company networks, making it harder to identify malicious activity. In order to assist human analysts in monitoring, preventing, identifying, and responding to cyberattacks and potential threats, organizations must adopt new methodologies. Many predict that machine learning will revolutionize cyber protection. Despite considerable success in other different applications, the value of deep learning in the context of network intrusion detection systems (NIDSs) has not been fully explored (e.g., image recognition).

Literature Review

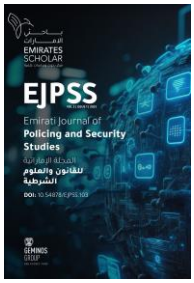
Our Contributions

This project contributes in two ways. Then, using a feedforward-connected deep neural network, we suggest training her NIDS through supervised learning (DNN). We also advise employing an auto encoder to recognize and categorize attack traffic using unsupervised learning in the absence of malicious information that has been identified as malicious. Then, using pre-established baselines of hostile and benign traffic, these models are assessed against two of her most recent network attack detection datasets. I We demonstrate that DNN outperforms existing network intrusion detection systems based on machine learning. (ii) When employing IP addresses as features for training a DNN, DNNs are resilient to dynamic IP addresses assigned by Dynamic Host Configuration Protocol (DHCP). (iii) Auto encoders are effective at detecting anomalies.

Related Work

The focus of this study is network-based intrusion detection, as opposed to host-based intrusion detection. There are many approaches to network-based intrusion detection. Anomaly detection has statistical origins and has been around since the 19th century [1]. The topic of cybersecurity intrusion detection was first studied in 1987 [2]. According to Fiore et al. A semi-supervised model based on discriminately constrained Boltzmann machines can be used to detect network intrusions. [3]. However, his KDD 99 dataset [4] on which their work is based is now outdated [5]. In addition to the KDD 99 dataset, CAIDA dataset [6], DARPA/Lincoln Lab packet traces [7], [8], Lawrence Berkeley National Laboratory (LBNL), and ICSI Enterprise Tracing Project [9. A comparative analysis reveals that some of them are outdated and unreliable due to a lack of traffic and volume, lack of attack types, lack of anonymization, and in some cases lack key payloads. . Lack of other features and metadata. The CAIDA dataset [6], DARPA/Lincoln Lab Packet Trace [7],[8], Lawrence Berkeley National Laboratory (LBNL) ,and ICSI Enterprise Tracing Project [9] are other data sets that exist alongside the KDD 99 dataset. It's a set. However, comparative analysis of these datasets reveals that some are outdated and unreliable due to lack of traffic and volume, attacks, anonymization, and lack of valuable payload information, feature sets, and metadata. became.

These new datasets not only contain the latest attacks, but also the accepted criteria (realism, evaluability, complete coverage, completeness, and in terms of malicious activity) [11]. These materials have been used in the evaluation work of many other studies. However, many ISCX IDS 2012 dataset interpreters only use part of the data and various ground truth construction techniques [12]-[16]. Studies on the CIC IDS 2017 dataset [11] [17]-[23] used deep learning as well as other machine learning techniques. Recent research has started using the CIC IDS 2017 dataset in deep learning. However, only a subset of the data is used to detect certain types of attacks (port scans, DDoS, etc.) [24]-[26], or to create your own stream instead of using the truth stream There are also things [27]. This study differs in that it evaluates both supervised and unsupervised deep



Emirati Journal of Policing and Security Studies
Vol 2 Issue 1 (2023) 4 - 12
DOI: 10.54878/EJPSS.298

Available at www.emiratesscholar.com

learning approaches against various attacks using the full dataset and given ground truth.

One of the pillars of the Dynamic Cybersecurity Framework, which aims to model and quantify cybersecurity from a holistic perspective, is the analysis of cybersecurity data [28]-[36], which includes It includes intrusion detection as an important area of investigation. His two other pillars are known as cybersecurity metrics [46] and first principles modeling and analysis [39]-[45]. With a holistic approach, the Cybersecurity Dynamics Framework [37, [38]] aims at modeling and quantifying cybersecurity. The second and third pillars are first principles modeling and analysis [39]-[45] and cybersecurity metrics [46]-[49]. The rest of the essay is structured as follows. Considered Part II. In Part III presents preliminaries and case study. Section IV discusses the limitations of this study. Part V concludes the work.

Preliminaries and Methodology

DNNs are effective supervised learning tools. By adding more layers and more units per layer to the neural network, they can represent progressively more complex functions [50]. DNNs can be used in NIDS to find both good and bad traffic patterns that are hidden in massive amounts of structured data. A common deep learning representation is shown in Figure 1. Superscript (i) denotes the i-th training example, superscript [l] denotes the l-th layer, and nodes denote inputs, edges denote weights. DNNs are the main focus of our case study because of their capacity.

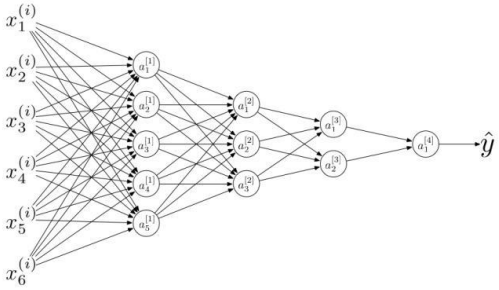


Figure 1: Deep neural network representation

Another type of neural network is an auto encoder. It is trained to find a low-dimensional latent spatial representation of the input data to copy the input to the output [50]. Unlike other popular dimensionality reduction techniques such as principal component analysis (PCA), it achieves its goal in a nonlinear manner. A diagram of a typical auto encoder with an equal number of input and output neurons is shown in Figure 2. We chose the auto encoder for our anomaly detection case study because it is useful when there is a lot of good data and it is difficult to detect anomalous data.

Case Study

Large amounts of training data are a defining characteristic of deep learning [50]. This indicates that the case study should make use of a fresh dataset.2.54 million samples (including 2.47 million benign and 68,910 malignant samples) are included in the ISCX IDS 2012 dataset [11]. In the CIC IDS 2017 dataset [10], there are over 2.83 million cases (including 2.27 million benign and 557,646 malignant ones). Older datasets, in contrast, are frequently small (the KDD 99 dataset [4] has only 148,517 flows, of which 77,054 are benign and 71,463 are malignant).

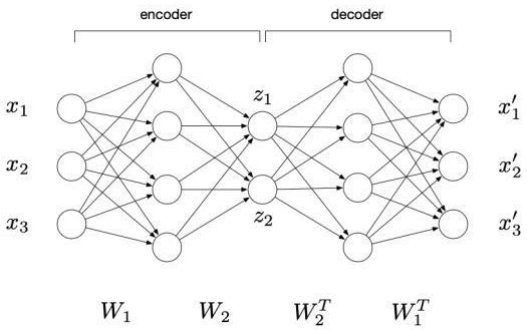
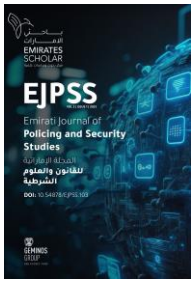


Figure 2: Diagram of neural network architecture: automatic encoding.

We chose DNNs because they can handle tabular data with high-cardinality categorical variables, as demonstrated by two recent datasets we work with. Dealing with high-cardinality categorical features is a



Emirati Journal of Policing and Security Studies
Vol 2 Issue 1 (2023) 4 - 12
DOI: 10.54878/EJPSS.298

Available at www.emiratesscholar.com

central topic [51]. The goal is to use entity embedding to transform high-cardinality categorical features into low-dimensional real vectors while preserving close relationships between similar values [52, 53]. There are many examples of normal data, but it's hard to define what is abnormal data, so we choose auto encoders. useful in such situations [11]. An auto encoder learns a compressed representation of the input data, so the output is a specific reconstruction of the input data. By preserving the errors in the reconstructed input (that is, in order to reconstruct the input, the auto encoder learns to change the weights. In contrast to typical flow reconstruction, the auto encoder learns to Encoders experience high reconstruction errors when faced with malicious flows.

Data Description

The ISCX IDS 2012 dataset [11] was created by using a testbed to simulate a specific network environment and launch into the testbed network. This data set includes realistic network configurations from real test environments, realistic traffic from real attacks/exploits, clear ground truth for benign and malicious traffic, full communication captures, and various egress comparisons. attack scenario. - Latest datasets [4], [7], and [8]. We used an IBM QRadar appliance to create a custom network flow XML file for this dataset. It also contains ground truth labels. Both formats are provided. Note that a network flow is created from a collection of IP packets and consists of a protocol, source and destination port numbers, and source and destination IP addresses. Additionally, this study focuses on flows rather than IP packets, as flows are often used as the unit of attack detection. A summary of this data set is shown in Table I.

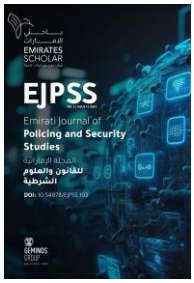
TABLE I: Overview of the ISCX IDS 2012 dataset, where "# of attacks" is the subset of flows that contain an attack.

Date	# of Flows	# of Attacks	Description
11/6/20212 to 17/6/2012	474,278	0	Benign network activities
	133,193	2,086	Brute-force against SSH
	275,528	20,358	Infiltrations internally
	171,380	3,776	HTTP DoS attacks
	571,698	37,460	DDoS using IRC bots
	522,263	11	Brute-force against SSH
	397,595	5,219	Brute-force against SSH
Total	2,545,935	68,910	2.71% malicious

Table II summarizes the 14 features that can be extracted from the labeled XML file of network flows.

TABLE II: Description of the 14 features of the ISCX IDS 2012 dataset, where "uniques" means the number of possible values of a categorical feature.

No.	Feature_ Name	Description	Type	Unique
1	SrcIP	Source IP address	Categorical	2,478
2	DstIP	Dest. IP address	Categorical	34,552
3	SrcPort	Source port	Categorical	64,482
4	DstPort	Dest. port	Categorical	24,238
5	AppNam e	Application name	Categorical	107
6	Directio n	Direction of flow	Categorical	4
7	Protocol	IP protocol	Categorical	6
8	Duration	Flow duration	Continuous	N/A



Emirati Journal of Policing and Security Studies
Vol 2 Issue 1 (2023) 4 - 12
DOI: 10.54878/EJPSS.298

Available at www.emiratesscholar.com

9	TotalSrcBytes	Total source bytes	Continuous	N/A
10	TotalDstBytes	Total dest. bytes	Continuous	N/A
11	TotalBytes	Total bytes	Continuous	N/A
12	TotalSrcPkts	Total source packets	Continuous	N/A
13	TotalDstPkts	Total dest. packets	Continuous	N/A
14	TotalPkts	Total packets	Continuous	N/A

In addition to harmless traffic, the CIC IDS 2017 dataset [10] included attack traffic from seven different attack types (Heartbleed bug, botnets, distributed denial of service (DDoS), cross-site scripting, etc.). , extending the ISCX IDS 2012 dataset. (XSS) or SQL injection attacks on websites, intrusions, etc.). This dataset contains both raw PCAP data and network flow data preprocessed from raw PCAP data using the CICFlowMeter tool [54]. A CSV file is provided that you can use to feed this preprocessed network flow data into your machine learning pipeline. The preprocessed network flow data is split into 83 columns (e.g. G., duration, number of packets, number of bytes, packet length), plus a label column and a flow ID column that can be used as features. This dataset contains 7 types of attacks, allowing multi-class classification investigations.
A summary of this data set is shown in Table III.

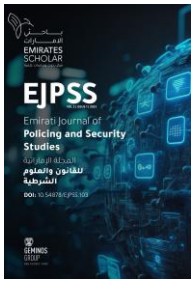
TABLE III :A summary of data set.

Day	# of Flows	# of Attacks	Description
Monday-Friday	529,918	0	Normal activities
	445,909	7,938	FTP-Patator
		5,897	SSH-Patator
		5,796	DoS slowloris
		5,499	DoS Slowhttptest
	692,703	231,073	DoS Hulk
		10,293	Dos GoldenEye
		11	Heartbleed
	170,366	1507	Web - Brute Force
		652	Web - XSS
		21	Web - SQL Injection
	288,602	36	Infiltration
	191,033	1966	Bot
	286,467	158,930	PortScan
	225,745	128,027	DDoS
Total	2,830,743	557,646	19.70% malicious

The columns in TABLE IV have the same meanings as in TABLE II and provide descriptions of some of the seventy-four features of the CIC IDS 2017 dataset.

TABLE IV: The 'useful' feature from the dataset

No.	Feature_name	Description	Type	Uniques
1	SrcIP	Source IP address	Categorical	17,002
2	DstIP	Dest. IP address	Categorical	19,112
3	SrcPort	Source port	Categorical	64,638
4	DstPort	Dest. port	Categorical	53,791
5	Protocol	IP protocol	Categorical	3
6	Duration	Flow duration	Continuous	N/A
7	tot fpacal _kets	Total num. forward packets	Continuous	N/A
8		Total num.	Continuous	N/A



Emirati Journal of Policing and Security Studies
Vol 2 Issue 1 (2023) 4 - 12
DOI: 10.54878/EJPSS.298

Available at www.emiratesscholar.com

	bpa tot cket al_ s	backward packets		
9	total fpktl	Total size of forward packets	Continuous	N/A
10	bpa tot cket al_ s	Total size of backward packets	Continuous	N/A
. . .	. . .	. . .	. . .	. . .
70	st acti d_ ve	Std. dev time flow active before idle	Continuous	N/A
71	min idle	Min time flow idle before active	Continuous	N/A
72	mean idle	Mean time flow idle before active	Continuous	N/A
73	max idle	Max time flow idle before active	Continuous	N/A
74	std idle	Std. dev time flow idle before active	Continuous	N/A

Table IV is some of the 74 features that were 'useful' from the CICIDS 2017 dataset, but of the remaining 85-74=11 features, 3 FlowIDs, timestamps, and Shows the labels (used for predicted classes). , each characterized by the currently available 74 traits (5 categories and 69 statistics).

C. Using DNNs for network intrusion detection

1) Pre-processing: To enable input to the DNN, I propose formatting the dataset (more specifically, the network flow). You should take note that the ISCX IDS 2012 dataset is offered not only in PCAP but also in a unique XML file that contains ground truth labels for network flows (indicating whether they are malicious or benign flows). Please give to me. A CSV file containing flows that are used as input in the machine learning pipeline is created after the XML file has been parsed. Remember that the PCAP and flow formats of the CIC IDS 2017 dataset each have 74 available characteristics (five categories and 69 statistics).

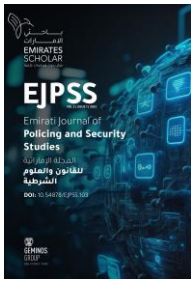
$$X = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where the minimum and maximum values of feature X are denoted by Xmin and Xmax. A DNN must be trained on categorical data, which must first be converted to numbers. We recommend the entity embedding technique [51] for this purpose, as it can handle categorical features that can take a wide range of possible values. This is also true for the dataset we analyzed, as the source IP address, destination IP address, source port number, and destination port number can all have different values. Embedding dimensions for entity embedding methods are chosen according to the following general principles [52]:

$$deviation = 4\sqrt{Posibile\ values} \quad (1)$$

Where the number of partial values for the categorical feature is indicated by possible values. In particular, categorical features are first mapped to integers between 0 and n-1 .s, where n is the maximum number of unique values the feature can take. Then, they are mapped as a dense vector using the equation given in equation (1). (1). The embedding results for each of the four categorical traits in the CIC IDS 2017 data set are listed in Table V.

Table V: creating a data set for the CIC IDS 2017 that includes four categorical traits.



Emirati Journal of Policing and Security Studies
Vol 2 Issue 1 (2023) 4 - 12
DOI: 10.54878/EJPSS.298

Available at www.emiratesscholar.com

Feature_name	Possible Values	Embedded Dimensions
Source IP	17,002	12
Destination IP	19,112	12
Source Port	64,638	16
Destination Port	53,791	15

Initialization of the parameters (weights) of the categorical feature vector representation supports [0:05; 0:05]. Not only is this representation computationally efficient, but the entity embedding layer also learns the intrinsic properties of each categorical feature, and deeper layers of neural networks combine these intrinsic properties [51]. These vectors are input to the first layer of the neural network, and a backpropagation step updates the weights of these vectors at every point.

Training:

The neural network has 3 layers, each with 64 units. These three hidden layers are fed from an initial input layer consisting of embedded categorical variables associated with statistical input features. The activation function for each hidden layer is the ReLU activation function R(z) = max(0;z), unlike the last output layer used. uses

A sigmoid activation function [z] = 1 / 1+e.In each hidden layer, dropout rates of 0:40 are used. It uses the RMSProp optimizer, which has a default learning rate of 0.001. Binary cross-entropy is the loss function employed.

$$H_p(q) = \frac{-1}{N}\sum_{i=1}^{N} y_i \log\left(P\left(y_i\right)\right) + (1-y_i) \log\left(1-P\left(y_i\right)\right)$$

(2)

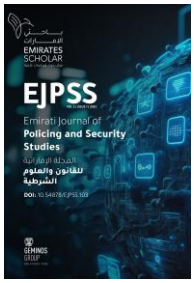
where the label is y_i, the predicted probability of a given flow is p(yi), the total number of flows is N, and yi is the label (1 for malicious, 0 for benign). The formula at a glance, according to Figure 2, the loss is log(p(yi), the log probability that the flow is malignant, and log(1 p(y_i)), the log probability that the flow is malignant. current (y_i = 1).

3) Experiments and findings: I would like to answer two questions in my experiment. (i) Is deep learning more effective than other machine learning techniques? (ii) Is deep learning robust in the presence of dynamic IP addresses? (ii) Training using IP addresses as a key feature DNNs can easily be rendered useless in the presence of dynamic IP addresses generated by networks that use Dynamic Host Configuration Protocol (DHCP). Using true positive rate (TPR) and false positive rate (FPR) as two common measures [46], we examine the effectiveness of deep learning and other machine learning techniques to answer the question (i) above. Compare

Table VI: A comparison of deep learning-based intrusion detection with another machine learning-based intrusion detection method using the CICIDS 2017 dataset [17].

Technique	TPR	FPR
Hypbrid IDS Decision Tree + Rule-based [17]	0:94475	:01145
WISARD [18]	0:48175	0:02865
Forest PA [19]	:92920	0:03550
J48 Consolidated [20]	0:92020	0:06645
LIBSVM [21]	0:54595	0:05130
FURIA [22]	0:90500	0:03165
Random Forest [17]	0:93050	0:01880
REP Tree [17]	0:91640	0:04835
MLP [17]	0:77830	0:07350
Naive Bayes [17]	0:82510	0:33455
Jrip [17]	0:93400	0:04470
J48 [17]	0:91990	0:05040
DNN with IPs	0.9993	0.0003
DNN without IPs	0.9677	0.0052

Table VI uses the CICIDS 2017 dataset to compare deep learning with other methods evaluated in [17]. Note that DNN achieves the highest true positive rate



Emirati Journal of Policing and Security Studies
Vol 2 Issue 1 (2023) 4 - 12
DOI: 10.54878/EJPSS.298

Available at www.emiratesscholar.com

(detection rate) and the lowest false positive rate when using IP addresses.
This leads to insight 1:

Compared to other machine learning techniques studied in the literature, DNN achieves the highest efficiency using IP addresses. To answer the aforementioned question (ii), we train a deep learning model using a subset of IP addresses. This makes sense because DHCP often uses the same network, which means the network identity (eg the original 24-bit IP addresses of a Class C network). The results for the two datasets with and without properties associated with IP addresses are shown in Figure 3. The results using only the first three octets of the source and destination IP addresses are shown in Figure 3e. As shown in Figure 3b, when the IP address feature is removed, the performance suffers significantly for the TPR and FNR of the ISCX IDS 2012 dataset. Figure 3d for the CIC IDS 2017 dataset shows that compared to ISCX IDS 2012, the performance in removing an IP address is only slightly worse. Compared to ISCX IDS 2012 (3:32 percent), CICIDS 2017 (19:68 percent) has significantly more malicious examples. Furthermore, we see that embedding the IP address with only the first three octets (Figure 3e) gives results comparable to those shown with the full IP address (Figure 3c). The result is as follows.

Insight 2: DNN is able to use IP addresses with the first three octets just as well as it does use a full IP address. The first three octets of an IP address can be used by DNN just as effectively as a full IP address. Generally speaking, the presence of DHCP greatly strengthens the detection of Deep Learning Based Attacks. Nevertheless, it's crucial to use full IP addresses if your data set is unbalanced that is, if the proportion of traffic that has been classified as malicious is low.

D. Using Auto encoders for network intrusion detection

1) Pre-processing: All 69 available features of the continuous flow statistics of the CIC IDS 2017 dataset are used in the auto-encoder experiment and normalized using the min-max method described above. There is also a categorical feature of the "protocol" used, which is converted to float using one-hot encoding. This function has only three unique values. Including the high cardinality properties of IP addresses and ports in auto encoder training is left to future research. ao encoders

2) Training: The auto encoder assembly consists of seven layers, the first and last layers using sigmoidal activation functions and the other hidden layers using ReLU. The hidden layers consist of 140, 35, 16, 16, and 35 units respectively, and the first and last layers consist of 72 units (representing all input features). Additionally, the first input layer is L1 regularized. The squared error serves as the auto-encoder's target function. This function is in equation (3) as a function of the following weights and inputs.

$$J = \left|X - sigmoid\ (\ sigmlid(x * w)\ *\ w\right|_F^2 \quad (3)$$

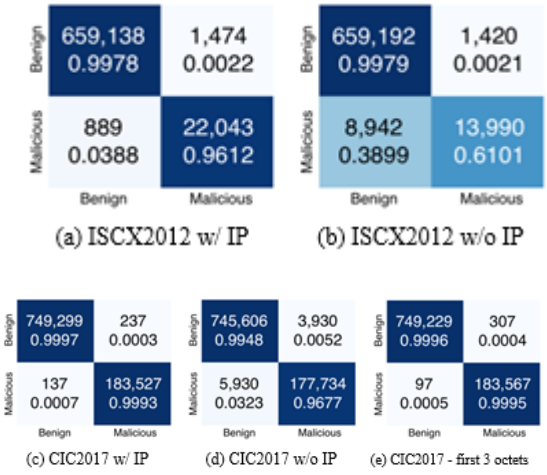
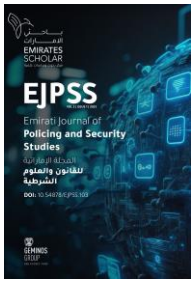


Figure 3: Confusion matrix results for both datasets, where the x-axis is the predicted class, and the y-axis is the true class.

3) Examinations and Findings: Experimental results are shown in Figure 4. We can see that the malicious traffic flow has a larger reconstruction



Emirati Journal of Policing and Security Studies
Vol 2 Issue 1 (2023) 4 - 12
DOI: 10.54878/EJPSS.298

Available at www.emiratesscholar.com

error than the benign traffic flow. The number of false positives can be changed according to the selected threshold. The current threshold of 0 generates only 89 false positives. With 03 reconstruction errors, the false positive rate is 0. 00013. High false negative rate 0: However, the 7670 does exist. Additionally, note that the majority of malicious flows are grouped together to provide support for future work that may include time domains as a feature. This leads to the following insight and conclusions.

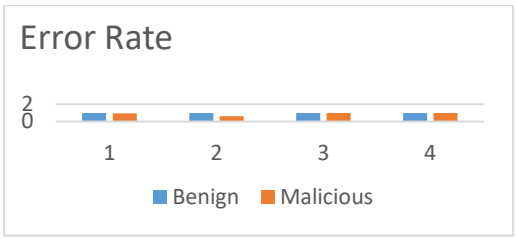


Figure 4: Experimental results using the CIC IDS 2017 dataset:

Auto encoder reconstruction error rate.

Insight 3:
If trained only on harmless data, auto encoders can be efficient anomaly detection systems for detecting network attacks (in terms of their low false positive rate).

Limitations

This study has some limitations. From a methodological point of view, we briefly examined two different types of neural networks. Other neural network types should be considered in future research. To determine if a DNN can achieve the same result as using the entire 32-bit IP address, more testing should be done on just the first two octets (first 16 bits) or the first octet. It should be investigated whether the IP address and port feature can be used to reduce the false negative rate of auto encoders. In terms of records, ISCX IDS 2012 records do not include HTTP traffic, only binary ground truth classifications (malicious and benign).

Conclusion

We have illustrated how well DNN performs in supervised network intrusion detection. Additionally, we showed that DNN is robust in the existence of DHCP by successfully handling the role of dynamic IP addresses when only the first three octets of the IP address are used. We also revealed how, after being trained on standard flows, auto encoders may be implemented for anomaly identification. We further demonstrated that auto encoders can be adopted for anomaly detection when they are trained on benign flows.

References

1. Mariam Aljouhi and Sara Al Hosani. Windows Forensics Analysis. *EJPSS.* 2022. Vol. 1(1):4-11. DOI: 10.54878/EJPSS.179

2. Riktesh Srivastava. Service Quality Control using Queuing Theory. *EJBESS.* 2022. Vol. 1(1):31-38. DOI: 10.54878/EJBESS.169

3. Tobias Haschke, Mathias Hüsing and Burkhard Corves. Bots2ReC - Analysis of Key Findings for the Application Development of Semi-Autonomous Asbestos Removal. *IJADT.* 2022. Vol. 1(1):4-12. DOI: 10.54878/IJADT.166

4. Tosin Ekundayo and Osama Isaac. Open Data: A National Data Governance Strategy for Open Science and Economic Development - A case study of the United Arab Emirates. *EJBESS.* 2023. Vol. 1(2):98-109. DOI: 10.54878/EJBESS.208