ISSN 2228-9860 eISSN 1906-9642 CODEN: ITJEA8

International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies

http://TuEngr.com

An Efficient Autoencoder-based Deep Learning Technique to Detect Network Intrusions

C. Haripriya[1*], M.P. Prabhudev Jagadeesh[1]

[1] *JSS Academy of Technical Education, VTU, INDIA.*
**Corresponding Author (Tel: 7760666000, Email: haripriya289 @gmail.com).*

Paper ID: 13A7P

Volume 13 Issue 7

Received 18 January 2022
Received in revised form 18 May 2022
Accepted 25 May 2022
Available online 01 June 2022

Keywords:
Intrusion Detection; Deep learning; Autoencoder, CSE_CIC-IDS 2018; Class Imbalance; SMOTE; Cybersecurity;

Abstract

With the tremendous advancements in internet technology, the amount of data generated over the network is very large. In a network connected with millions of computers, Terabytes/Zettabytes of data are generated every second. It is almost impossible to analyze this enormous data generated in the network manually. Companies have to incur huge losses if their network is compromised, hence timely detection of intrusions is very important to help the organizations prevent further attacks. Deep learning algorithms proved to be more effective when compared to Machine Learning algorithms. Earlier research works focused on old sets like KDDCup99 which do not reflect current-day attacks. All the Intrusion Detection Datasets are imbalanced and have severely skewed class distribution. Many researchers do not focus on class imbalance and their classification models tend to overfit. The major motivation of our research work is to focus on data pre-processing techniques and address the class imbalance problem using SMOTE (Synthetic Minority Oversampling Technique). We implement a deep autoencoder on the latest dataset which is the latest benchmark dataset that reflects current attacks. The average accuracy considering all the CSV (Comma Separated Values) files of the "CSE-CIC-IDS 2018" is 97.79 %. The proposed model achieved promising results and is more accurate since we considered all the records and attack types of the dataset.

Disciplinary: Cyber Security and Machine Learning (Deep Learning, Network Security)

Cite This Article:

Haripriya, C., Jagadeesh, M. P. P. (2022). An Efficient Autoencoder-based Deep Learning Technique To Detect Intrusions. *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies, 13*(7), 13A7P, 1-10. http://TUENGR.COM/V13/13A7P.pdf DOI: 10.14456/ITJEMAST.2022.142

1 Introduction

Researchers explored the application of Machine Learning (ML) in the domain of network security. The traditional approaches of ML are unsuitable to analyze large amounts of data in a short time. Researchers have explored the application of deep learning algorithms to detect intrusions. Although KDDcup 99 (Tavallaee, 2009) is one of the standard benchmark datasets used for intrusion detection, it no longer reflects current attacks. One of the latest benchmark datasets that reflects the current attacks is the "CSE-CIC-IDS 2018" dataset (Sharafaldi et al,2018). The dataset is a collaborative project between "The Communications Security Establishment" (CSE) and the "Canadian Institute of Cybersecurity". Network behaviour and patterns change over-time and intrusions evolve. It was important that researchers no longer use static datasets. Datasets used for intrusion detection should be dynamically generated to reflect current-day attacks. In addition to dynamicity, the dataset should also be modifiable, extensible and reproducible. The major challenge in detecting intrusions is detecting anomalies.

An IDS (Intrusion Detection System) is an efficient way used to identify and detect cyber-attacks and malicious activities. Implementing an anomaly-based IDS is a tedious process since it involves analyzing enormous traffic generated in the network. The main challenge is, that it requires more computational resources and memory. Another problem is the possibility of overfitting because of high-dimensional complex feature space. Research suggests that efficient techniques to detect anomalies are yet to be developed (Al-amri R et al., 2021).

2 Literature Review

Kanimozhi et al. (2019) implemented IDS on ANN (Artificial Neural Network) and MLP (Multi-Layer Perceptron) to detect only botnet attacks on the CSE-CIC2018 dataset. Their framework was on CPU with an accuracy of 99.97%. Ferraga et al. (2020) discuss in detail the different approaches and datasets used in IDS. They also perform a comparative study. Shone et al. (2018) implemented the IDS on a "Non-Symmetric Deep Auto Encoder" using the KDD99 dataset. They achieved an average accuracy of 97.85% for 5-class classification. Khan et al. (2019) implemented a "Stacked Encoder with Soft-max classifier" and achieved an accuracy of 99.97% and 89.13% on KDD99 and UNSW-NB15 datasets respectively.

Papamartzivanos et al. (2019) designed a self-adaptive autonomous IDS. They focused on misuse detection. They used the KDDcup 99 and NSL-KDD datasets and proved that the adaptive model was better than the static model. Yang et al. (2019) proposed ICVAE – DNN. They combined the Improved Conditional Variational Autoencoder with Deep Neural Networks and used NSL-DD and UNSW-NB15 datasets to evaluate the performance of their model. Abusitta et al. (2019) proposed a denoising auto encoder and used the KDDcup 99 dataset and achieved an accuracy of 89.09%. Wang et al. (2016) used stacked denoising autoencoders. Their model was used to detect malicious JavaScript code. Their dataset consisted of benign and malicious JavaScript samples and achieved an accuracy of 95%.

Moraboena et al. (2020) proposed Symmetric Deep Autoencoder (SDAE) by using CICIDS 2017 and achieved an accuracy of 91.76% and 91.88% in binary classification for batch sizes 512 and 1024 respectively. Fathima et al. (2021) used a two-stage deep-stacked autoencoder and achieved an accuracy of 87% on the CICIDS 2017 dataset. However, the authors have not addressed the class imbalance problem of the CICIDS 2017 dataset. Zhaojun et al. (2021) proposed ANDAE (Adam Nonsymmetric Deep Autoencoder) and implemented it on NSL-KDD and CICIDS 2017 and evaluated their model on metrics like precision, recall, and F1 score. Wenfeng et al. (2021) proposed interpretable intrusion detection using an autoencoder and additive tree on the UNSW-NB15 dataset with an accuracy of 99.95%.

Fahimeh et al. (2018) used Deep Auto-Encoder (DAE) on the KDDCup99 dataset with an accuracy of 94.71% (19). Kunang et al. (2018) used a simple autoencoder and SVM classifier with an accuracy of 99.35% and 88.64% on KDDCup99 and NSL-KDD datasets respectively. Narayana et al. (2021) proposed Sparse Auto Encoder (SAE) and achieved an accuracy of 99.03%, 99.71% and 99.98% for KDDCup99, NSL-KDD and UNSW-NB15 datasets respectively.

Networks today have become more vulnerable as attackers are launching new /sophisticated attacks. Many of the existing research work focused on the KDDcup99 or NSLKDD dataset. Both KDDcup99 and NSLKDD datasets were benchmark datasets for network intrusion detection which no longer reflect the current attacks. New attacks are launched every day. Attackers are also finding new ways to launch the existing attacks. Thus, in our research work, we focus on the latest IDS dataset which contains the current attacks.

3 Method

This section gives details about data preprocessing and how the Class imbalance problem is addressed. Figure 1 shows the architecture of our proposed model. In our research work, a deep autoencoder is used which is a type of unsupervised neural network (14). Autoencoders belong to generative deep learning models. The model learns by using a compressed representation of the input. The main advantage of using unsupervised neural networks in NIDS is, that they avoid the tedious task of class labelling and also help to detect zero-day attacks.

The CIC-CSE 2018 dataset is based on the concept of the creation of user profiles. CSE CIC 2018 dataset also uses the concept of profiles like the ISCX-IDS-2012 Intrusion Detection dataset (12). It consists of protocols such as HTTPS (Hyper Text Transfer Protocol Secure), IMAP (Internet Message Access Protocol) HTTP (Hyper Text Transfer Protocol), SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3), SSH (Secure Shell Protocol) and FTP (File Transfer Protocol). The CSV files of the CIC_IDS2018 dataset can be downloaded by accessing AWS services. The dataset was generated by deploying 50 malicious hosts, 420 client servers and 30 servers (Tavallaee et.al, 2009).

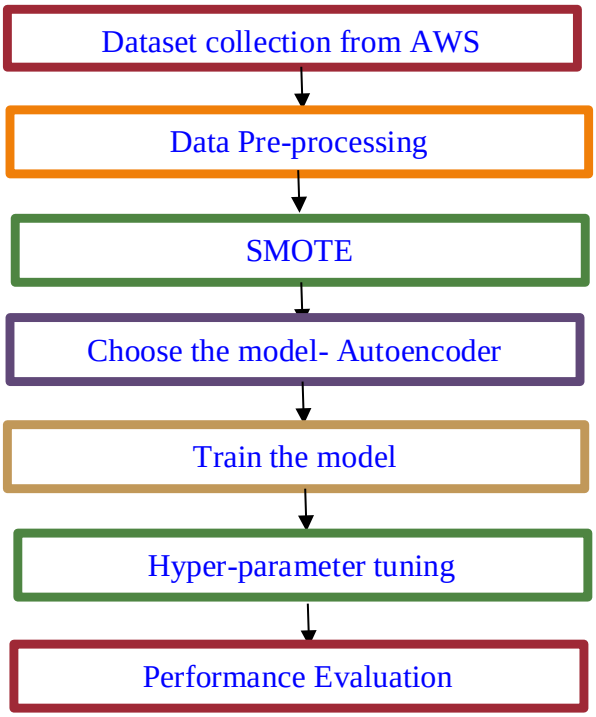

Figure 1: Flow diagram of this studied model.

3.1 Data Pre-Processing

Data pre-processing is a very important step to speed up training. The main aim is to prepare the raw data and make it suitable for the deep learning algorithm. Figure 2 illustrates the different steps followed during data pre-processing.

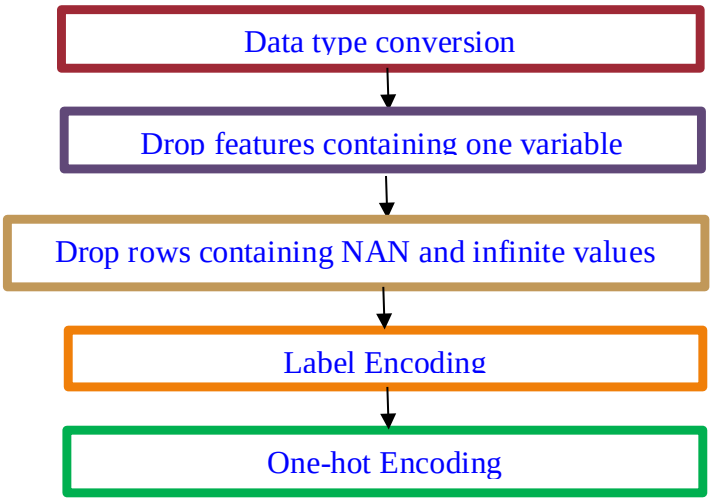

Figure 2: Steps in Data Pre-processing

Table 1: Total Number of Records in the CSE CIC 2018 Dataset before and after Pre-processing.

CSV	Date	Before removing nan and infinite rows	After removing nan and infinite rows	Total number of records removed
CSV 1	14-02-2018	1048575	1044751	3824
CSV 2	15-02-2018	1048575	1040548	8027
CSV 3	16-02-2018	1048575	1048575	0
CSV 4	20-02-2018	1500000	1494468	5532
CSV 5	21-02-2018	1048575	1048575	0
CSV 6	22-02-2018	1048575	1042965	5610
CSV 7	23-02-2018	1048575	1042867	5708
CSV 8	28-02-2018	607723	606902	821
CSV 9	01-03-2018	331125	328181	51056
CSV 10	02-03-2018	1048575	1044525	4050

3.2 Class Imbalance Problem

The CIC-CSE IDS 2018 dataset suffers from a class imbalance problem. Imbalanced datasets have severely skewed class distribution. The classification models tend to over-classify the larger class due to their increased prior probability. Therefore, the instances belonging to smaller classes are typically misclassified when compared to classes having larger instances. In the CIC-CSE IDS 2018 dataset, records containing normal traffic are more when compared to the attack traffic. For example, the Wednesday 14-02-2-18 CSV file consists of 63.69% of benign data, 18.44 % of FTP-Brute force attack data and 17. 88% of SSH – Brute force attack data. Thus, instances containing attack traffic namely FTP-Brute force attack and SSH Brute force are typically misclassified when compared to instances containing normal traffic. Figure 3 illustrates the different class labels of the 1st CSV file of the CSE CIC 2018 dataset. It also illustrates the class imbalance problem.

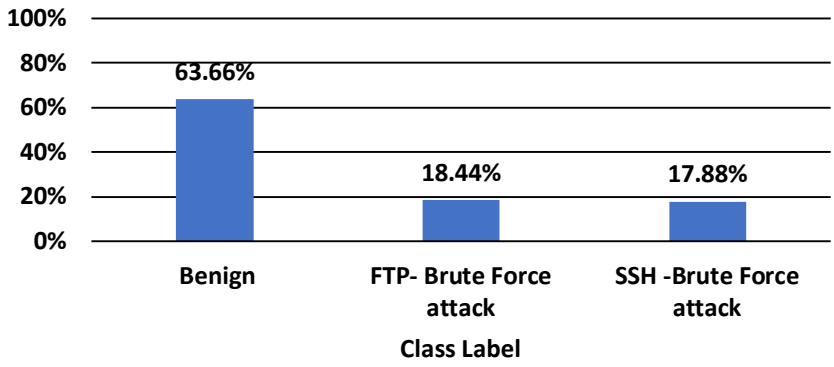

Figure 3: Different class labels of the CSE CIC 2018 dataset (CSV1) (Total of percentage records in the dataset before using SMOTE).

3.2.1 SMOTE

To address the class imbalance problem SMOTE is used (Chawla et.al, 2002). In this technique, synthetic samples are generated from the minority class. New samples can be synthesized from the existing samples. This technique also solves the problem of overfitting posed by random oversampling. In this work, the class imbalance problem is addressed by duplicating examples from minority classes. This should be done on the training set before fitting the model. After using SMOTE, the class imbalance problem can be effectively addressed. SMOTE does not provide any additional information to the model. Thus, SMOTE is a data augmentation technique for the minority class. In CSE CIC 2018 dataset, minority classes are instances containing attack traffic.

All the files were run on Google Colab which is a cloud-based notebook environment. Classifying all the attacks of the CSE-CIC2018 dataset is a compute-intensive task. Considering the large size of the dataset, the autoencoder model was run on GPU. Performance of the model on GPU proved to be more efficient than CPU, in detecting intrusions at a faster rate. Hence, the GPU

runtime environment was used to run the deep learning model as the main goal was to detect intrusions at a faster rate.

Stratify is used to preserve the dataset proportions. It is used for better reproducibility of results. For example, if we consider Friday 02-03-2018 CSV file, we have two main classes namely Benign and bot attack. Hence a binary classification should be performed on the examples. Considering 100 examples of the CSV file, if we split the train set into 80:20 ratio, assuming there would be 15 examples in class 0 i.e. Benign and 5 examples in class 1 i.e. Bot attack. We would have quite biased data. Thus, in order to provide a set of common relationships between the training and testing set, stratify is used. Without using dropout, the model is overfitted. To prevent overfitting, dropout is used in regularizing the neural network. By using drop out, we temporarily remove a node from the network by disconnecting all the incoming and outgoing edges. The idea here is to break up circumstances where the network layers co-adapt by correcting mistakes from the previous layers. The problem of overfitting arises when the co-adaptations do not generalize to new data. Thus, dropout tends to make the model more robust.

ReLU (Rectified Linear Unit) is used in the hidden layers whereas Sigmoid is used in the output layer. To determine the number of training examples to be utilized in one iteration batch size is used. If the CSV files consisted of only 2 classes i.e. normal or attack (Binary Classification), then binary cross-entropy is used as the loss function. If the CSV files consisted of more than 2 classes then categorical loss entropy is used as the loss function to perform multi-classification. Adam optimizer is used. Early stopping monitor with patience is used to stop training when the model does not further improve.

Table 2: Different Hyper parameters used in the proposed autoencoder model

Sl. No.	Hyper Parameters	Value
1	Activation Function: Hidden layer	ReLU
2	Activation Function: Output layer	Sigmoid
3	Batch Size	32,64,128
4	Number of Epochs	20
5	Loss function: Binary Classification	Binary Cross Entropy
6	Loss function: Multi Classification	Categorical Cross Entropy
7	Optimizer	Adam

4 Result and Discussion

True positive gives the number of correctly classified attack records. True Negative gives the number of correctly classified benign records. False Positive is the number of misclassified benign records. False Negative is the number of misclassified attack records. The following metrics were used to evaluate our classification model.

$$\text{Accuracy} = (TP + TN) / \text{Total number of instances} \quad (1),$$

$$\text{Precision} = TP/ (TP + FP) \quad (2),$$

$$\text{Recall} = TP / (TP + FN) \quad (3).$$

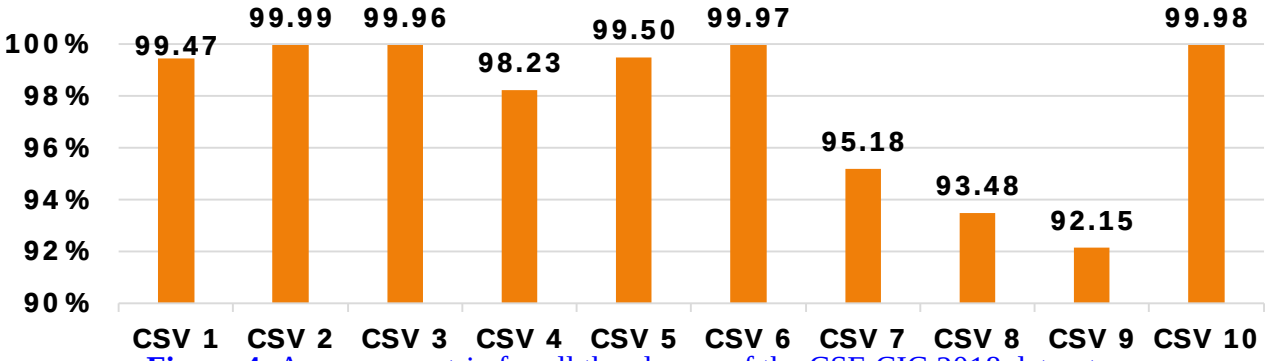

Figure 4: Accuracy metric for all the classes of the CSE CIC 2018 dataset.

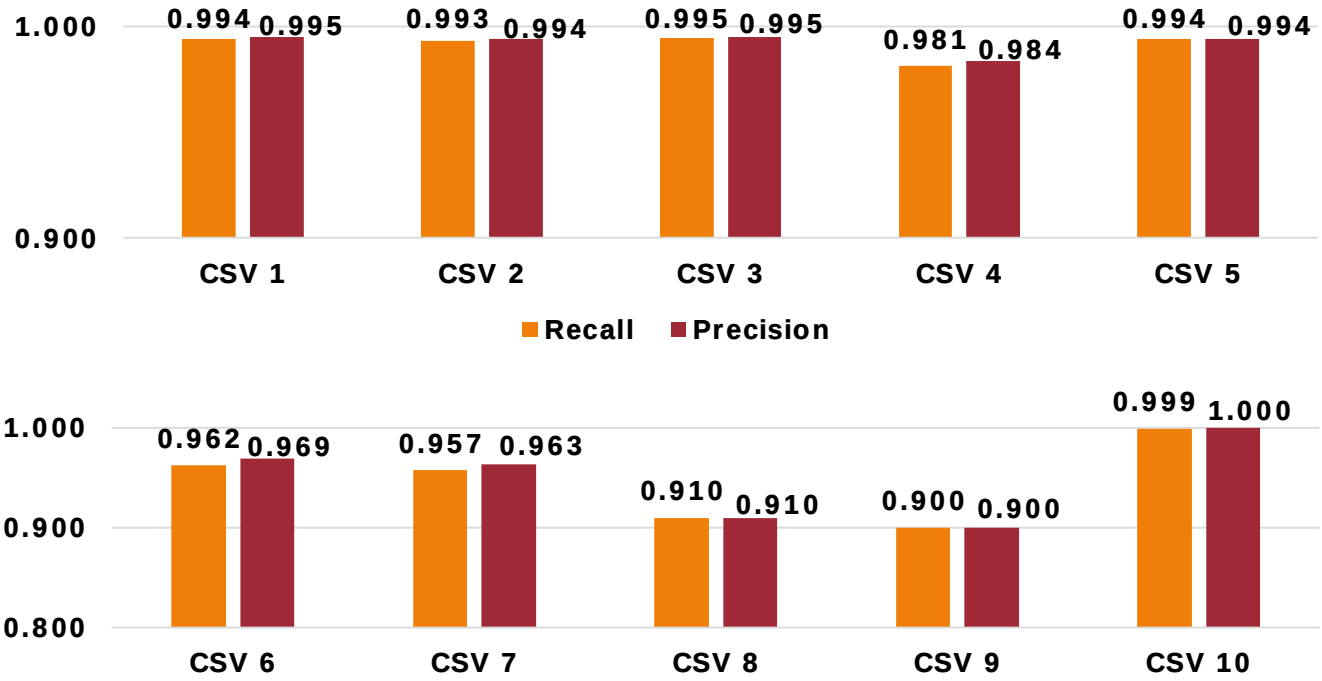

Figure 5: Recall and Precision metrics for all the classes of the CSE CIC 2018 dataset

Table 3: Comparative analysis of various techniques and datasets employed for network intrusion detection using various datasets

Sl.No	Authors	Model	Dataset Used	Accuracy	Year
1	V. Kanimozhi et al	ANN and MLP	CSE-CIC-ID S2018	99.97%.	2019
2	Shone et al	Non-symmetric deep Auto-encoder	KDD-Cup 99	97.85%	2017
3	Khan et. al	Stacked encoder with soft-max classifier	KDD-Cup 99 and UNSW-NB15	99.97% and 89.13%	2019
4	Abusitta et al	Denoising auto- encoder	KDD-Cup 99	89.09%.	2019
5	Moraboena et al.	Symmetric Deep Autoencoder	CICIDS 2017	91.76% and 91.88%	2020
6	Nasreen Fathima et al	Two-stage deep stacked autoencoder	CICIDS 2017	87%	2021
7	Wenfeng et.al	Interpretable Intrusion Detection using Autoencoder and Additive Tree	UNSW-NB15	99.95%.	2021
8	Fahimeh et al.	Deep Auto-Encoder (DAE)	KDD-Cup 99	94.71%.	2019
9	Y. N. Kunang et al	Simple autoencoder and SVM classifier	KDD-Cup 99and NSL-KDD	99.35% and 88.64%	2018
10	Narayana et al.	Sparse Auto Encoder (SAE)	KDD Cup99, NSL-KDD and UNSW-NB15	99.03%, 99.71% and 99.98%	2021

Table 3 gives a comparative analysis of various techniques and datasets employed for network intrusion detection using various datasets. From Table 3, we infer that none of the researchers have implemented their models on the latest datasets. However, Kanimozhi et al worked only on the botnet traffic of the CSE-CIC 2018 and achieved an accuracy of 99.97% (4). The authors used ANN, however they did not run their model on GPU. In addition to benign and botnet traffic we worked on all the classes of the dataset namely FTP-Brute force, SSH-Brute force, DOS attacks-Golden eye, DOS Attacks-Slow Loris, DDOS attacks-LOIC HTTP, Brute force-Web attack, Brute force-XSS attack, SQL Injection, Infilteration, DDOS attack-HOIC, DDOS attack LOIC-UDP.

Figures 4 and 5 show the different metrics used in our model evaluation. Accuracy, Recall and Precision values are tabulated for the different CSV files of the CSE-CIC 2018 dataset. From the results in Figures 4 and 5, it is observed that our model designed using a deep autoencoder provided very good results. The average accuracy obtained over all the files of the dataset is 97.79%. In addition to accuracy, recall and precision metrics were also used as shown in Figure 5.

5 Conclusion

This research mainly focuses on effectively addressing the class imbalance problem of the CSE-CIC 2018 dataset and implementing it on a deep autoencoder. Many researchers have not addressed the class imbalance problem. In this research paper, we focused mainly on the pre-processing techniques and used SMOTE to address the class imbalance problem. Considering all the 10 CSV Files of the dataset we have achieved an overall accuracy of 97.79%. We conclude that we have achieved very promising results by considering all the files of the CSE-CIC 2018 dataset. Many researchers only considered a few attack types of the dataset because it was a computed and data-intensive task to consider all the files. We conclude that for large IDS datasets deep learning algorithms like deep autoencoder are an effective approach to detect intrusions. In the future, we would like to implement more deep learning models and use ensemble techniques and compare their performance with shallow machine learning techniques. In future work, to detect intrusions faster and accelerate the performance, we are exploring parallel computing techniques.

6 Availability of Data and Material

Data can be made available by contacting the corresponding author.

7 References

Abusitta A, Bellaiche M, Dagenais M, Halabi T. A deep learning approach for proactive multi-cloud cooperative intrusion detection system. Future Generation Computer Systems. 2019 Sep 1;98:308-18.

Al-amri R, Murugesan RK, Man M, Abdulateef AF, Al-Sharafi MA, Alkahtani AA. A review of machine learning and deep learning techniques for anomaly detection in IoT data. Applied Sciences. 2021 Jan;11(12):5320.

Bank D, Koenigstein N, Giryes R. Autoencoders. arXiv preprint arXiv:2003.05991. 2020 Mar 12.

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research. 2002 Jun 1;16:321-57.

Farahnakian F, Heikkonen J. A deep auto-encoder based approach for intrusion detection system. In2018 20th International Conference on Advanced Communication Technology (ICACT) 2018 Feb 11 (pp. 178-183). IEEE.

Fathima N, Pramod A, Srivastava Y, Thomas AM. Two-stage Deep Stacked Autoencoder with Shallow Learning for Network Intrusion Detection System. arXiv preprint arXiv:2112.03704. 2021 Dec 3.

Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications. 2020 Feb 1;50:102419.

Gu Z, Wang L, Liu C, Wang Z. Network Intrusion Detection with Nonsymmetric Deep Autoencoding Feature Extraction. Security and Communication Networks. 2021 Dec 31;2021.

Kanimozhi V, Jacob TP. Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. In2019 international conference on communication and signal processing (ICCSP) 2019 Apr 4 (pp. 0033-0036). IEEE.

Khan FA, Gumaei A, Derhab A, Hussain A. A novel two-stage deep learning model for efficient network intrusion detection. IEEE Access. 2019 Feb 15;7:30373-85.

Kunang YN, Nurmaini S, Stiawan D, Zarkasi A. Automatic features extraction using autoencoder in intrusion detection system. In2018 International Conference on Electrical Engineering and Computer Science (ICECOS) 2018 Oct 2 (pp. 219-224). IEEE.

Moraboena, S., Ketepalli, G., Ragam, P. (2020). A deep learning approach to network intrusion detection using deep autoencoder. Revue d'Intelligence Artificielle, Vol. 34, No. 4, pp. 457-463. DOI: 10.18280/ria.340410

Papamartzivanos D, Mármol FG, Kambourakis G. Introducing deep learning self-adaptive misuse network intrusion detection systems. IEEE Access. 2019 Jan 22;7:13546-60.

Rao KN, Rao KV, PVGD PR. A hybrid intrusion detection system based on sparse autoencoder and deep neural network. Computer Communications. 2021 Dec 1;180:77-88.

Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp. 2018 Jan 22;1:108-16.

Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. computers & security. 2012 May 1;31(3):357-74.

Shone N, Ngoc TN, Phai VD, Shi Q. A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence. 2018 Jan 22;2(1):41-50.

Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In2009 IEEE symposium on computational intelligence for security and defense applications 2009 Jul 8 (pp. 1-6). IEEE.

Wang Y, Cai WD, Wei PC. A deep learning approach for detecting malicious JavaScript code. Security and Communication Networks. 2016 Jul 25;9(11):1520-34.

Xu W, Fan Y, Li C. I2DS: interpretable intrusion detection system using autoencoder and additive tree.

Security and Communication Networks. 2021 Mar 12;2021.

Yang Y, Zheng K, Wu C, Yang Y. Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Sensors. 2019;19(11):2528.

Haripriya C. is a Research Scholar at JSS Academy of Technical Education, Visvesvaraya Technological University (VTU), India. She got her Master's degree in Software Engineering from VTU, India. Her research interests are in the field of Deep Learning, Network Security.

Dr. Prabhudev Jagadeesh M.P. is a Professor at JSS Academy of Technical Education, Visvesvaraya Technological University (VTU), India. He got his Ph.D. degree in Computer Science and Engineering from the University of Mysore, India. His research interests are in the field of Image Processing, Deep Learning and Information Security.