Proc. of the 7th International Conference on Engineering and Emerging Technologies (ICEET)
27-28 October 2021, Istanbul, Turkey

Autoencoder-based Intrusion Detection System

Firuz Kamalov
Department of Electrical Engineering
Canadian University Dubai
Dubai, UAE
firuz@cud.ac.ae

Rita Zgheib
Department of Computer Science
Canadian University Dubai
Dubai, UAE
rita.zgheib@cud.ac.ae

Ho Hon Leung
Department of Mathematics
UAE University
Al Ain, UAE
hohon.leung@uaeu.ac.ae

Ahmed Al-Gindy
Department of Electrical Engineering
Canadian University Dubai
Dubai, UAE
agindy@cud.ac.ae

Sherif Moussa
Department of Electrical Engineering
Canadian University Dubai
Dubai, UAE
smoussa@cud.ac.ae

2021 International Conference on Engineering and Emerging Technologies (ICEET) | 978-1-6654-2714-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICEET53442.2021.9659562

***Abstract*—Given the dependence of the modern society on networks, the importance of effective intrusion detection systems (IDS) cannot be underestimated. In this paper, we consider an autoencoder-based IDS for detecting distributed denial of service attacks (DDoS). The advantage of autoencoders over traditional machine learning methods is the ability to train on unlabeled data. As a result, autoencoders are well-suited for detecting unknown attacks. The key idea of the proposed approach is that anomalous traffic flows will have higher reconstruction loss which can be used to flag the intrusions. The results of numerical experiments show that the proposed method outperforms benchmark unsupervised algorithms in detecting DDoS attacks.**

***Index Terms*—intrusion detection systems, autoencoders, unsupervised learning, cybersecurity, anomaly detection**

I. Introduction

Networks underpin the functioning of the modern society. From local area networks to the world wide web, our daily lives depend on secure and smooth operation of networks. The increased importance of networks has also made them more targeted by hackers and malicious users. Therefore, effective intrusion detection systems (IDS) that guard networks from unwanted traffic have become as important as ever. In this paper, we propose an IDS based on autoencoders. The proposed approach has several advantages over the existing methods and outperforms similar benchmark methods in numerical experiments.

In general, an IDS is a computer-security application designed to detect a range of security violations - from break-ins by outsiders to system infiltration and exploits by insiders. Traditional IDS require a significant amount of effort from domain experts to design and maintain. Signature-based IDS have a high missed alarm rate, lack the ability to detect unknown attacks, and need a huge signature database. To address the issues with the traditional IDS, new approaches based on machine learning algorithms have been proposed in the literature. Machine learning methods do not require extensive domain knowledge and can operate in an end-to-end fashion with little supervision. Machine learning algorithms achieve high accuracy when trained on large amounts of data. Given the availability of network data machine learning-based IDS are well-suited to detect malicious traffic.

Machine learning algorithms can be grouped into two categories: supervised and unsupervised. Supervised algorithms such as decision trees, multilayer perceptron, support vector machines (SVM) and others require a labeled dataset for training, while unsupervised methods can be trained on unlabeled data. In many applications including intrusion detection there is a limited availability of labeled data. Labeling data manually is expensive and time consuming. In the context of intrusion detection, we also need to account for unknown future attacks whose signature and patterns are not currently available. So it would be impossible to train a supervised classifier to detect future unknown attacks. Therefore, it is not always possible to design an IDS based on a supervised machine learning algorithm. To address the issue of the lack of labeled data and unknown future attacks, we propose to employ unsupervised learning to construct IDS. In particular, we propose to use stacked autoencoders to build an unsupervised learning detection system.

Autoencoder-based IDS can be trained on regular traffic data to learn normal traffic patterns. Then, given a new traffic flow it can be classified as regular or anomalous based on the reconstruction error. Since regular traffic data is much more readily available than attack data, autoencoder offers a practical alternative to the traditional machine learning algorithms. In this paper, we apply autoencoders to detect DDoS attacks. To test the performance of the proposed approach we benchmark it against existing outlier detection methods. The results of the experiments reveal that the proposed approach achieves a robust accuracy and outperforms benchmark unsupervised algorithms.

The paper is structured as follows. In Section II, we give a brief review of the current literature regarding intrusion detection. In Section III, we provide the methodology and present the results of the numerical experiments. We conclude

978-1-6654-3897-1/21/$31.00 ©2021 IEEE

with a few closing remarks in Section IV.

II. Literature

There exists a wide range of machine learning-based intrusion detection methods in the literature [2]. The majority of the existing methods are based on supervised learning algorithms such as neural networks, SVM, and decision trees. Unsupervised approaches include generative adversarial networks (GAN), K-means, and autoencoders. In most, cases the performance of unsupervised methods is inferior to supervised methods. However, unsupervised methods have the advantage of being trained on unlabeled data.

Deep learning methods have been frequently used in constructing intelligent IDS. The author in [15] experimented with various neural network architectures and concluded that the network with fewer parameters achieves better accuracy on DDoS attacks. The authors in [6], propose a hybrid deep learning model consisting of long short-term memory network and convolutional network for real-time web intrusion detection. In addition to identifying malicious attacks, the proposed method helps write rules for signature-based IDS. Although deep learning is considered the state-of-the-art in machine learning, it does not always produce the optimal results. The authors in [7], compared 10 different machine learning algorithms and found that kNN, decision tree, and Naive Bayes algorithms outperform the deep learning models. The same study also found that supervised methods outperform unsupervised approaches.

Unsupervised algorithms are applied in a number of different ways to detect intrusions. One approach is based on clustering the data into normal and anomalous groups [12]. In a surprising result, the authors in [14] compared the performance of five machine learning algorithm in DDoD detection and found that the k-means clustering achieves the best accuracy. Unsupervised learning is also used for feature extraction. The authors in [16] first extracted features using a sparse autoenconder and then applied the XGBoost classifier to distinguish between benign and malicious signals. In [1], the authors employ two autoencoders - one based on normal traffic and another based on malicious signals - to learn class-specific features. Then in the supervised stage, a multi-channel parametric convolution is adopted. GANs offer a novel approach to building IDS. Adversarial learning can be used to attack IDS while simultaneously training the IDS to detect the adversarial traffic flow [8].

One of the main issues affecting traffic flow data is imbalanced class distribution [13]. In most cases, regular (benign) samples vastly outnumber anomalous (malicious) samples. Imbalanced class distribution leads to bias in classification algorithms [5]. In [16], the authors applied SMOTE to balanced the data. Other data preprocessing steps such as feature selection were also shown to improve the performance of IDS [3], [4].

III. Methodology

A. Autoencoders

An autoencoder is a special type of neural network designed to reduce the dimensionality of the input data. The key idea is to reconstruct the original input after passing it through a bottleneck layer. An autoencoder consists of two parts: encoder and decoder. The encoder compresses the data into the bottleneck layer, while the decoder tries to reconstruct the bottleneck vector into the original input. The loss function used in autoencoder is the mean squared error between the input and output arrays. In its simplest form - when both encoder and decoder are linear functions - the autoencoder is equivalent to principal component decomposition. Nonlinear autoencoders have a variety of applications including data compression, image denoising, feature extraction, anomaly detection, and image generation.

In our paper, we employ a fully connected stacked autoencoder with three layers in the encoder and three layers in decoder networks. The proposed architecture is illustrated in Fig. 1. The encoder initially reduces the dimension of the input vector from 63 to 4, while the decoder reconstructs the bottleneck vector to the original input size. The network hyper parameters where tuned based on the mean absolute error on the training set. In the final model, we used ReLU activation in all hidden layers and Adam optimizer with learning rate of 0.001.

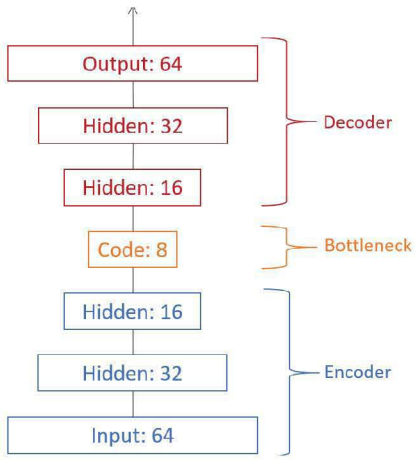

Fig. 1: The autoencoder architecture used in the proposed IDS.

B. IDS Algorithm

The proposed approach to detecting malicious traffic flows involves initially training the autoencoder on normal data. Then classifying new traffic flows based on the reconstruction loss. If the reconstruction loss is large, then the IDS flags the traffic flow. Otherwise, it is deemed benign. Since normal traffic data is widely available, autoencoders offer a more practical approach to constructing an IDS than traditional machine learning methods. The details of the proposed method are outlined below.

Algorithm

1. Train the autoencoder on regular (non malicious) traffic data.
2. Calculate the reconstruction loss on the training data and determine the mean μ and standard deviation σ of the loss distribution.
3. Select the threshold for flagging traffic flows: $\mu + z \cdot \sigma$, where z is a defined by the domain expert.
4. Given new traffic flow, calculate the reconstruction loss using the trained autoencoder and classify the traffic as benign/malicious based on the threshold.

C. Dataset and benchmarks

To test the performance of the proposed approach we use the dataset CSE-CIC-IDS2018 developed by the Canadian Institute for Cybersecurity [11]. The dataset consists of traffic flow information based a simulated network. Flow data contains grouped packets over a time interval and is the most common source of data for IDS especially for DOS. The dataset is generated using the notion of profiles. Benign profiles capture the behaviors of typical users on the network. The encapsulated features are distributions of packet sizes of a protocol, number of packets per flow, certain patterns in the payload, size of payload, and request time distribution of a protocol. Malicious profiles are designed to mimic an attack scenario. In particular, the DDoS attack is simulated using Low Orbit Ion Canon. The network is implemented using a common LAN topology on the AWS computing platform. In our paper, we use a dataset of 80 extracted features from the network traffic flow. The dataset contains 10,000 traffic flows divided equally between benign and DDoS cases.

We employ two standard anomaly detection algorithms as our benchmarks: one-class SVM and Robust Covariance. One-class SVM algorithm is trained to obtain the boundary of the distribution of the initial observations [10]. Then new observations are categorized according to their distance from the boundary. Robust Covariance is a model-based anomaly detection algorithm. It is based on the assumption that the regular data is generated via some statistical distribution. The distribution parameters such as the mean and standard deviation are calculated based on the sample data and an ellipse is fitted to the central data points, ignoring points outside the central mode [9].

IV. Numerical Experiments

In this section, we present the results of the numerical experiments aimed at comparing the proposed IDS against benchmark outlier detection methods. The experiments are based on the dataset described in Section III-C. The data is divided into training and testing subsets based on 80/20 ratio. Furthermore, the malicious samples are discarded from training set and only the normal samples are retained. Note that the numbers of benign and malicious instances are approximately the same in the original dataset. Thus, given the 80/20 split, the number of normal training samples is approximately 4,000 out of the total of 10,000 samples. The number of samples in each subset is provided in Table I. As described in the algorithm in Section III-B, we train the autoencoder on the normal training samples. We use the test set as the validation set during training.

TABLE I: The number of samples in training and testing sets.

	Benign	Malicious
Training	3989	0
Testing	1011	990

As shown in Fig. 2, the training error is significantly lower than the validation error. It is a desirable outcome because we would like the reconstruction error for the normal traffic to be lower than the reconstruction error for the mixed traffic which contains malicious signals. It allows us to threshold the malicious signals and flag them based on higher reconstruction loss. Otherwise, if the autoencoder had very close training and validation error, then it would not be able to distinguish between the normal and malicious samples.

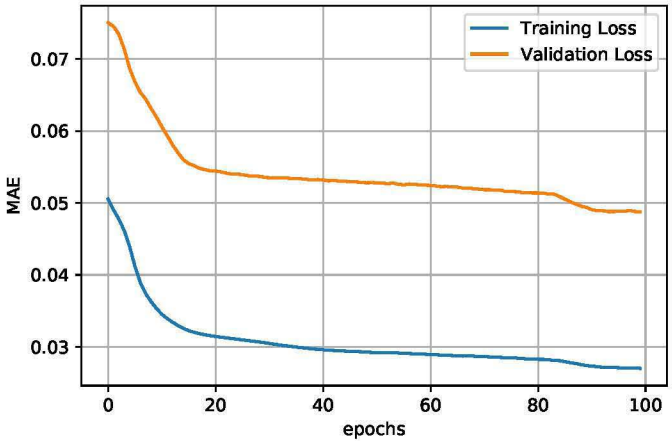

Fig. 2: Training and validation loss of the autoencoder.

In Fig. 3, we present the reconstruction error on the training set. Recall that the training set consists of only the regular traffic flows. It can be seen that the majority of samples have very low reconstruction loss and fall below the threshold. So most of the normal traffic flow is classified correctly as benign.

In Fig. 4, we present the reconstruction loss for the malicious samples in the testing set. Recall that the testing set consists of both the regular and malicious traffic flows. It can be seen that while the majority of the malicious data lies above the threshold, there is a portion that lies below the threshold. It indicates that the autoencoder fails to capture all the malicious signals. Note that by varying the threshold we can control the trade off between the recall and precision of the classifier. If we decrease the threshold, then we would obtain higher precision as more malicious samples will lie above the threshold. However, by lowering the threshold will well concurrently decrease the recall.

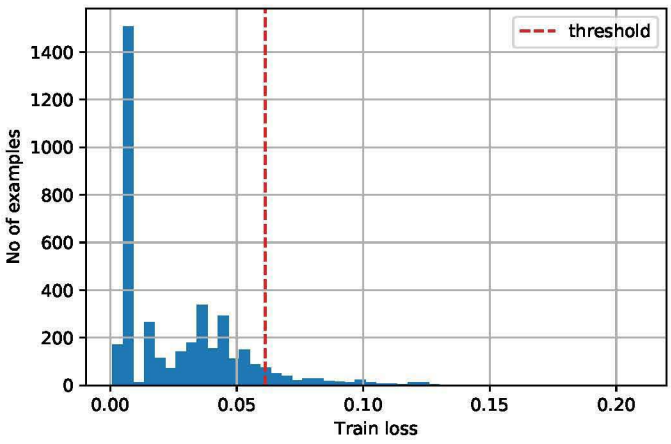

Fig. 3: Reconstruction loss on the training samples. The majority of samples have very low reconstruction loss and are classified correctly as benign.

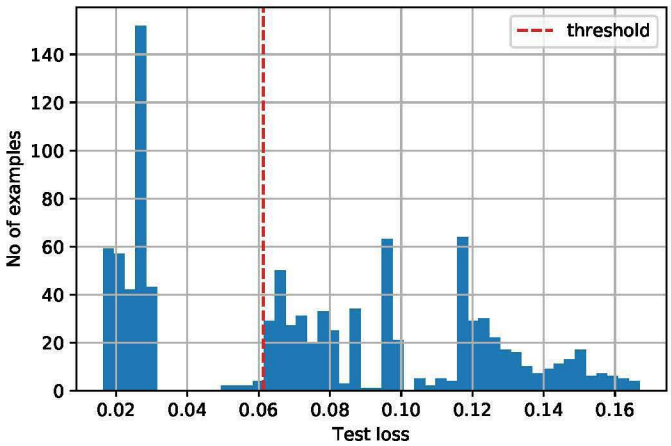

Fig. 4: Reconstruction loss on the malicious testing samples. A portion of the samples losses lie below the threshold which indicates that the autoencoder misses some of the malicious signals.

To gauge the performance of the proposed method, we calculate the accuracy, precision, and recall of the autoencoder-based IDS on the test set. We also calculate the same metrics for the benchmark methods. The results are presented in Table II. It can be seen that the autoencoder-based IDS significantly outperforms the benchmark methods in accuracy. Indeed, the accuracy of the autoencoder-based IDS is 20% higher than the benchmarks. It also has the best recall but yields to Robust Covariance in terms of precision. The high recall rate shows that the proposed method correctly identifies the regular samples as benign. The precision rate shows that 38.56% of the samples that were identified as benign were in fact malicious. Note that the benign/malicious sample distribution in the test set is roughly the same (Table I). Therefore, if we use a random-guess approach to classify the traffic flows then we would get approximately 50% accuracy, precision, and recall rates. It shows that the results achieved by the autoencoder are non-trivial.

TABLE II: Performance of the autoencoder-based IDS against benchmark methods.

	Autoencoder	one-class SVM	Robust Covariance
Accuracy	**0.7671**	0.5647	0.5657
Precision	0.7144	0.4866	**0.8882**
Recall	**0.8981**	0.5829	0.5429

Although the proposed approach achieves solid results, it is definitely less accurate than the existing supervised methods. In Sharafaldin, the authors used the ID3 algorithm to achieve accuracy of up to 98% on a similar dataset. Similarly, in [4], the authors achieved over 99% accuracy on the same dataset. It is not surprising that the supervised methods achieve higher accuracy than unsupervised methods. During the training process, supervised methods have the information about which traffic flows are malicious and which are normal. It allows the supervised methods to better learn the distinguishing characteristics between benign and malicious signals and consequently achieve high classification accuracy. Despite their effectiveness, supervised methods cannot be always employed in IDS due to the lack of labeled data. Another important drawback supervised methods is that by definition they cannot be trained to recognize the future unknown attacks.

V. Conclusion

In this paper, we considered an autoencoder-based IDS. Unlike supervised methods, the proposed approach can be applied without labeled data. Numerical experiments showed that it significantly outperforms the benchmark outlier detection algorithms. Since the proposed approach is only trained on regular traffic flow data it is more likely to be effective against unknown attacks than supervised methods. On the other hand, supervised method do perform well when labeled data is available.

As a direction for future research, the proposed method can be extended to identify other types of attacks such as web attacks and infiltration attacks. In addition, the performance may be improved if in place of the extracted features the raw data is used for training.

References

[1] Andresini, G., Appice, A., Di Mauro, N., Loglisci, C., & Malerba, D. (2020). Multi-channel deep feature learning for intrusion detection. IEEE Access, 8, 53346-53359.

[2] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20), 4396.

[3] Kamalov, F. (2021). Orthogonal variance decomposition based feature selection. Expert Systems with Applications, 182, 115191.

[4] Kamalov, F., Moussa, S., Zgheib, R., & Mashaal, O. (2020, December). Feature selection for intrusion detection systems. In 2020 13th International Symposium on Computational Intelligence and Design (ISCID) (pp. 265-269). IEEE.

[5] Kamalov, F., & Denisov, D. (2020). Gamma distribution-based sampling for imbalanced data. Knowledge-Based Systems, 207, 106368.

[6] Kim, A., Park, M., & Lee, D. H. (2020). AI-IDS: Application of deep learning to real-time Web intrusion detection. IEEE Access, 8, 70245-70261.

[7] Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., & Foozy, C. F. M. (2021). Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE Access, 9, 22351-22370.
[8] Rigaki, M., & Garcia, S. (2018, May). Bringing a gan to a knife-fight: Adapting malware communication to avoid detection. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 70-75). IEEE.
[9] Rousseeuw, P. J., & Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. Technometrics, 41(3), 212-223.
[10] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural computation, 13(7), 1443-1471.
[11] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018, January). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In ICISSp (pp. 108-116).
[12] Peng, K., Leung, V. C., & Huang, Q. (2018). Clustering approach based on mini batch kmeans for intrusion detection system over big data. IEEE Access, 6, 11897-11906.
[13] Sahu, A., Mao, Z., Davis, K., & Goulart, A. E. (2020, May). Data processing and model selection for machine learning-based network intrusion detection. In 2020 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR) (pp. 1-6). IEEE.
[14] Tuan, T. A., Long, H. V., Kumar, R., Priyadarshini, I., & Son, N. T. K. (2019). Performance evaluation of Botnet DDoS attack detection using machine learning. Evolutionary Intelligence, 1-12.
[15] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525-41550.
[16] Zhang, B., Yu, Y., & Li, J. (2018, May). Network intrusion detection based on stacked sparse autoencoder and binary tree ensemble method. In 2018 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.