Comparison of Classifier-Based Machine Learning Algorithms for Intrusion Detection System

Author: Fauzan Isnaini (faisna@iu.edu)

Abstract

In the early days of the internet, network threats can be recognized by their specific bit patterns, called signatures. However, as network threats become more sophisticated, security researchers have taken an interest in applying machine learning for detecting unknown threats. In this paper, we compare the performances of the four of the most common classifier-based machine learning techniques for detecting intrusions: KNN, decision tree, SVM, and ANN. ANN performs best with f1-score of 98.9916%, while SVM performs worst with f1-score of 98.5678%. However, all our models surpass the baseline set by DARPA of 0.1% maximum false alarm rate.

Keywords: machine learning, classifier, intrusion detection system, security

Introduction

The rapid advances in the internet and communication fields have resulted in a huge increase in the network size and the corresponding data. With the increased traffic and more connected devices, securing the network is becoming more challenging. Moreover, network threats like malware, zero-day attacks, and Denial of Services (DoS) have become more sophisticated, forcing network administrators to adapt to new security approaches.

In the early days of the internet, network threats can be recognized by their specific bit patterns, called signatures, in affected files or packets. Network security tools, such as antivirus and the early version of Intrusion Detection System (IDS) would then use these signatures to detect attacks. The signature database has to be manually revised for each new type of intrusion that is discovered. A significant limitation of signature-based methods is that they cannot detect emerging cyber threats, since by their very nature these threats are launched using previously unknown attacks. With the rapid development of network threats, malware can also change its binary representations to evade signature-based detections. In addition, even if a new attack is discovered and its signature developed, often there is a substantial latency in its deployment across networks [6].

These limitations have led to an increasing interest in intrusion detection techniques based upon data mining. The first well-documented IDS came from Stanford Research Institute in 1983. It applied statistical methods to audit trails to identify anomalous activity [3]. However, for a long time, this method was not a favorite solution for network administrators due to its high false positive rate (FPR). Today, as computational power increases and more training data are available, many network security vendors are looking back at the potential of using data mining to prevent network threats [10].

Ahmad et al. [11] conducted a systematic study to select recent journal articles focusing on various machine learning-based IDS which were published during 2017-April 2020. Based on these observations, they reported that the most common machine learning algorithms used for IDS are: Decision Tree, K-Nearest Neighbor (KNN), Artificial Neural Network (ANN), Support Vector Machine (SVM), and K-Mean Clustering. Deep learning algorithms such as Auto Encoder (AE) also has gained some popularity.

Phadke et al. [1] conducted a similar study and reported SVM, K-Mean Clustering, and ANN as some of the machine learning techniques for modern IDS. They also reported the accuracy and false positive rate (FPR) of each method, but these numbers are from different studies and based on different datasets, thus it is not an apple-to-apple comparison.

In this paper, we want to provide a comparison between the recent trends in machine learning techniques for IDS. Unlike the previous studies, the key idea here is to do an apple-to-apple comparison by running them on the same dataset and comparing the results. We use the KDD-99 dataset, which is considered one of the benchmark datasets for intrusion detection and includes a wide variety of intrusions simulated in a military network environment [8].

Methods

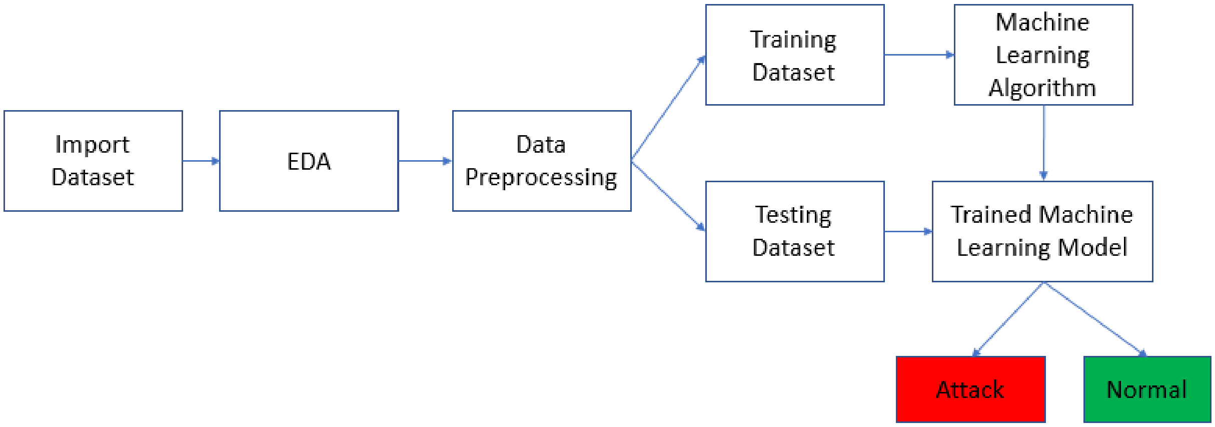

Figure 1 General machine learning-based intrusion detection methodology

Figure 1 shows our generalized methodology to run our classification models. First, we do an exploratory data analysis (EDA) to understand the dataset so we can transform the data accordingly. Our dataset contains 41,237 rows and 41 columns. Out of these 41 columns, 40 columns are the features, and the remaining one column contains the label information. Figure 2 shows the label distribution of the dataset. The problem of skewed class distribution is very apparent here since the anomalous traffics as the classes of interest are much smaller than the class representing normal network behavior. In such scenarios when the normal behavior may typically represent 98-99% of the entire population a trivial classifier that labels everything with the majority class can achieve 98-99% accuracy [6]. It is apparent that in this case classification accuracy is not sufficient as a standard performance measure, and other performance measures such as precision, recall, and f1-score will be required to evaluate our models.

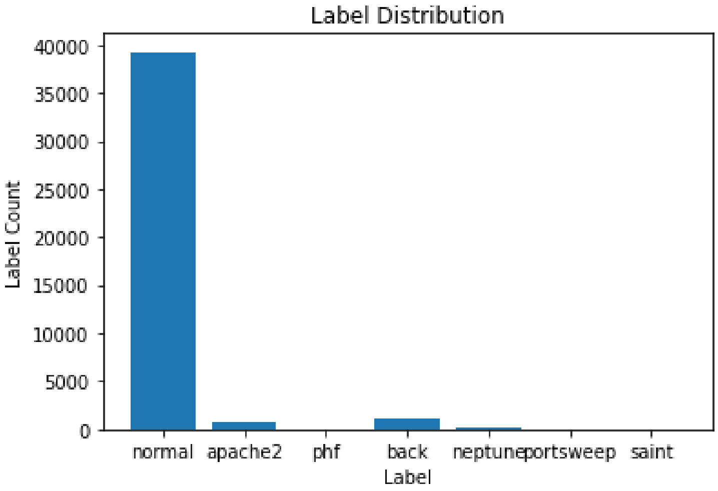

Figure 2 Label distribution of the dataset

For all the proposed solutions, the dataset is then preprocessed to transform it into the format suitable to be used by the algorithms. There are three important steps here: (1) all non-normal labels are converted into a single class, (2) all categorical features are converted into numerical formats, and (3) transforming all the features using their respective mean and variance.

After that, the data is split randomly into two portions: the training data and the testing data. We use two-thirds of the dataset as the training data, and the remaining one-third as the testing data, as suggested by [7].

We then apply four different machine learning techniques on the same data: (1) K-nearest neighbors, (2) decision tree, (3) SVM, and (4) ANN. These methods are selected as they were reported by [1] and [11] to be the most common classification techniques for IDS. We then used the trained models to classify the testing data as either normal or anomalous traffics. Accuracy, precision, recall, f1-score, and false positive rate (FPR) are then used to compare the performance of the four models. Table 2 shows the parameters of each model.

No.	ML Model	Parameters
1	KNN	Number of neighbors: 2, distance metric: Euclidean
2	Decision tree	Split criterion: Gini impurity, split strategy: best
3	SVM	Kernel type: linear (using LinearSVC), maximum number of iterations: 40,000
4	ANN	Input layer with 40 input dimensions, one hidden layer with 50 neurons and relu activation function, output layer with one neuron and sigmoid activation function, loss: binary crossentropy, optimizer: adam, epochs: 10

Table 1 Models' parameters

Results

Table 2 shows the comparison of the models' performances. KNN, decision tree, and ANN have the same levels of accuracy, but ANN has a higher f1-score. This is due to ANN having a higher recall compared to KNN and decision tree, while the number of anomalous traffic is smaller than the number of normal traffic. SVM gives the lowest performance among the four models, but it still gives an f1-score of 98.5678%.

No.	ML Model	Accuracy	Precision	Recall	F1-score	FPR
1	KNN	99.9030	99.6604	98.3250	98.9882	0.017
2	Decision tree	99.9030	99.6604	98.3250	98.9882	0.017
3	SVM	99.8626	99.1525	97.9899	98.5678	0.042
4	ANN	99.9030	99.3255	98.6600	98.9916	0.034

Table 2 Performance comparison in percent (%)

Discussion

There are several baselines that can be used to evaluate the performance of our intrusion detection models. For example, in [4], the authors cited that DARPA has a criterion of 0.1% for false alarm rate. However, more recently, in [9], the authors cited 0.6% as an acceptable FPR for their anomaly-based detection method for Hypertext Transport Protocol (HTTP) cyber-attacks. It is clear from the above results that all our models have FPR below 0.1%, surpassing both baselines. Even SVM as the worst model has an FPR of 0.042%, which is still far below 0.1%.

However, there is one important thing to consider here: our models are binary classification models, which means they only classify network traffics as either normal or anomalous. By combining all the anomalous traffics into a single class, we are mitigating the unbalance distribution between normal traffic and anomalous traffic. However, for a real-world IDS solution, it is important to develop a multi-class classification model so network administrators can take the right actions according to the type of threats.

Our results also point out the importance of training data to achieve a good performance. Prior to the mini-presentation session, I used a different dataset from the internet that only has 405 rows and two features: latency and throughput values. Using the same parameters as what we are using in this paper, k-nearest neighbor only gave an accuracy of 96.2687%, with the precision of 0.5, recall of 0.2, and f1-score of 0.29. The decision tree model also gave bad results with the accuracy of 97.0149%, precision of 0.6, recall of 0.6, and f1-score of 0.6. The details of these results are given on the mini_presentation.pptx slide deck.

It is also important to understand that all our models used classifier-based machine learning techniques. In order for these techniques to perform well, all the traffics in the dataset need to be labeled as either normal or anomalous for training the models. On the other hand, clustering machine learning techniques such as k-means can be adopted if labeled datasets are not available. Further study may need to be conducted to analyze the performance of clustering-based machine learning techniques for IDS.

References

[1]: A. Phadke, M. Kulkarni, P. Bhawalkar, and R. Bhattad, “A review of machine learning methodologies for network intrusion detection,” *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019.

[2]: D. A. Burgio, “Reduction of False Positives in Intrusion Detection Based on Extreme Learning Machine with Situation AwarenessExtreme Learning Machine with Situation Awareness,” thesis, 2019.

[3]: J. Miller, “From ids and IPS to Siem: What you should know: Bitlyft cybersecurity,” *BitLyft*, 15-Dec-2020. [Online]. Available: https://www.bitlyft.com/resources/from-ids-and-ips-to-siem-everything-you-need-to-know. [Accessed: 14-Dec-2021].

[4]: McHugh, J. (2000). Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Transactions on Information and System Security (TISSEC), 3(4), 262-294.

[5]: P. Dini and S. Saponara, “Analysis, design, and comparison of machine-learning techniques for networking intrusion detection,” *Designs*, vol. 5, no. 1, p. 9, 2021.

[6]: P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P.-N. Tan, “Data Mining for Network Intrusion Detection,” 2002.

[7]: P.-N. Tan, A. Karpatne, V. Kumar, and M. Steinbach, *Introduction to data mining*. Harlow: Pearson, 2020.

[8]: S. Hettich and S. D. Bay, “The UCI KDD Archive.” Irvine, CA, 1999.

[9]: Swarnkar, M. & Hubballi, N. (2016). OCPAD: One class Naive Bayes classifier for payload based anomaly detection. Expert Systems with Applications, 64, 330-339.

[10]: W. Stallings and L. Brown, *Computer security: Principles and practices*. Pearson, 2014.

[11]: Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and Deep Learning Approaches,” *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, 2020.