Deploying and analyzing classification algorithms for Intrusion Detection

A project report,
submitted in partial fulfillment of the requirements for the B.Tech project

by

Himanshu Pandey (2018IMT-038)

Under the Supervision of

Dr. Saumya Bhadauria

विश्वजीवनामृतं ज्ञानम्

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR-474 015
2021

Thesis Certificate

I hereby certify that the work, which is being presented in the report/thesis, entitled Deploying and analyzing classification algorithms for Intrusion Detection, is in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in Information Technology and submitted to the institution is an authentic record of my work carried out during the period June-2021 to August-2021 under the supervision of Dr. Saumya Bhadauria. I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Signature: Himanshu

Name: Himanshu Pandey

Roll. No: 2018IMT-038

Date: 11-August-2021

To the best of my knowledge, the statements made by Mr. Himanshu Pandey (Roll No. 2018IMT-038) in the above thesis certificate are correct.

-____

Dr. Saumya Bhadauria

Date____________

Candidate's Declaration

I hereby certify that I have properly checked and verified all the items as prescribed in the checklist and ensure that my thesis is in the proper format as specified in the guideline for thesis preparation.

I declare that the work containing in this report is my work. I understand that plagiarism is defined as any one or a combination of the following:

(1) To steal and pass off (the ideas or words of another) as one's own

(2) To use (another's production) without crediting the source

(3) To commit literary theft

(4) To present a new and original idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else's work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy, as well as a close resemblance to some else's work, constitutes plagiarism.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Signature:

Name: Himanshu Pandey

Roll. No: 2018IMT-038

Date: 11-August-2021

Abstract

Hacking has grown in popularity in recent years, increasing cyber-attack quantity and variety. Malware, denial of service attacks, phishing, and social engineering are all examples of computer network threats. Antiviruses and firewalls are no longer sufficient for effective cybersecurity. To prevent these dangers, you can't only rely on antivirus and firewalls: you need multiple levels of defense. With the capacity to monitor packets from OSI layer 2 (Datalink) through layer 7 (Application), network-based Intrusion Detection Systems (IDSs) offer a supplementary technique of increasing security. IDSs that use anomaly detection can detect unknown assaults, but they are less accurate, resulting in a high amount of false alarms. Machine learning methods are investigated in this thesis to develop IDSs that can be deployed in real-world computer networks. To begin, a three-step optimization strategy is given to increase detection quality: 1) data augmentation to rebalance the dataset, 2) model performance optimization, and 3) ensemble learning to integrate the findings of the best models. This method has the disadvantage of requiring labeled datasets, which are rarely available in real-life situations. As a result, transfer learning is explored to train machine learning models on huge labeled datasets and subsequently finetune them on innocuous network traffic. This method also has problems because the models are trained on previously known assaults and so do not do anomaly detection. As a result, an unsupervised learning-based method is provided. It takes advantage of network prototyping. When anomalies are discovered, they are grouped into attacks or disregarded if they are isolated. Finally, network congestion detection is investigated. The bandwidth usage of various links is forecasted to prevent problems from arising.

Dedication

Dedicated to my mentor, Dr. Saumya Bhadauria for her advice, patience, and faith in my abilities.

Acknowledgments

I am highly indebted to Dr. Saumya Bhadauria and obliged for giving me the autonomy of functioning and experimenting with ideas. I would like to take this opportunity to express my profound gratitude to her not only for her academic guidance but also for her interest in my report and constant support coupled with confidence-boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within me. The nurturing and blossoming of the present work is mainly due to her valuable guidance, suggestions, astute judgment, constructive criticism, and an eye for perfection. My mentor always answered a myriad of my doubts with smiling graciousness and prodigious patience, never letting me feel like a novice by always lending an ear to my views, appreciating and improving them, and by giving me a free hand in my report. It's only because of her overwhelming interest and helpful attitude, the present work has attained the stage it has.

Finally, I am grateful to our Institution and colleagues whose constant encouragement served to renew my spirit, refocus my attention and energy, and helped me in carrying out this work.

Himanshu Pandey

2018IMT-038

Contents

1. Abstract 3
2. Introduction 7
3. Motivation 7
4. Literature Survey 8
    4.1. Background 8
    4.2. Key related research and Research Gaps 9
5. Thesis Objectives and deliverables 12
6. System Architecture/ Methodology 12
    6.1. Mechanism/Algorithm 13
7. Progress Made so far 15
8. Tasks to be completed 19
9. Gnatt Chart 20
10. References 20

List of Figures

1. Gnatt Chart 20

Tables

1. Confusion Matrix 16
2. Conclusion after running the given models 17
3. Conclusion after running the given models post feature engineering 19

2 Introduction

Computers have become an inextricable aspect of our lives. They are widely employed in all government agencies, businesses, private organizations, hospitals, and private residences. Because of their importance in our lives, protecting them from invasions is a huge task for us. Intrusions are still the most persistent hazards in the cyber world, despite significant advancements in protection technologies. Many ways for analyzing intrusions have been developed to date.This process and the design of the machine learning models are usually managed by a framework such as scikit-learn [1], Tensorflow [2], PyTorch[3], Matlab or Weka [4].

3 Motivation

With time, many various computer attacks are stretching their arms intending to harm the targeted system. For a company, antiviruses or firewalls are no longer sufficient to ensure the security of the systems of the company. Now we need multiple layers of security to ensure truly secured systems among which one of the most important layers is provided by the Intrusion Detection System (IDS) which is designed to protect its target against any potential attack through continuous monitoring of the system. In our thesis, we are going to deploy machine learning models which can detect known attacks through supervised intrusion detection and unknown attacks through unsupervised intrusion detection. Further, the thesis model will be deployed to predict bandwidth utilization and at last, all the machine learning techniques will be compared in detecting intrusions so that we can come up with better and efficient algorithms and others can have a knowledge to which algorithm is to be used in case of intrusion detection.

4 Literature Survey

4.1 Background

During the 1970s, the growth of computer networks created new problems related to monitoring user activities and access. Intrusion Detection Systems have signature-based detection (or “misuse detection”) and anomaly detection. In signature-based detection, the data monitored by the IDS is compared to known patterns of attacks and it can only detect known attacks while anomaly detection builds a model of the normal behavior of the system and then looks for deviations in the monitored data. Anomaly detection has a wider aspect but it can also detect unknown attacks which can generate irrelevant alarms.

Many machine learning techniques have come into the picture to detect anomalies. Some of them can rely on algorithms with the ability to learn directly from the data, without being explicitly programmed. Despite so many advantages, these algorithms are rarely deployed just because of a higher false positivity rate. Indeed, even a false positive rate of 1% can create so many false alarms on a high traffic network that they become impossible for an administrator to process[5].

4.2 Key related research and Research Gaps

Another crossover intrusion detection technique described by Gisung Kim et al. [6] gradually joins misuse location and peculiarity identification in a deteriorated structure.

To begin, a decision tree was utilized to develop a misuse detection model, which was then used to break down the normal training data into smaller groups. Then, in each deconstructed region, a one-class support vector machine (1-class SVM) was employed to create an anomaly detection model. During the integration, the anomaly detection model might leverage the known attack information to improve its capabilities while creating normal behavior profiles. This is the first time a misuse detection model has been used to improve the ability of an anomaly detection model. The C4.5 decision tree does not create a cluster, which can reduce the system's profiling ability and accuracy.

Shi-Jinn Horng et al. [7] devised a system for intrusion detection that incorporates a clustering technique, a basic feature selection algorithm, and the Support Vector Machine (SVM). This paper suggested an SVM-based network intrusion detection system with BIRCH hierarchical clustering for data pre-processing, in addition to a basic feature selection technique. In place of the original huge dataset, BIRCH hierarchical clustering gives a well-qualified and reduced dataset for SVM training. In addition to saving time during training, the generated classifiers outperformed SVM classifiers trained on the previously duplicated dataset. In terms of accuracy, however, the proposed method could achieve the highest score of 95.72 percent. In comparison to the other NIDS, this technique performs better in terms of accuracy (Network-based IDS). Only Dos and Probe attacks are detected, not U2L or R2L attacks.

Mrutyunjaya Panda et al. [8] introduced hybrid intelligent decision technologies that use data filtering in combination with directed learning methods and a classifier to produce better-classified judgments to detect network assaults. The Naive Bayes model is highly appealing because of its purity, elegance, robustness, and effectiveness, as shown by the findings. Decision trees, on the other hand, have demonstrated their effectiveness in both generalizing and detecting new assaults. The findings reveal that there is no single optimum algorithm that can consistently outperform others in all cases. There may be some reliance on the data's properties in some circumstances. To make better decisions, a domain expert or expert system may use the categorization findings to choose a suitable algorithm.

Juan Wang et. al. [9] presented a decision tree-based intrusion detection system. The information gain ratio is utilized instead of information gain when creating incursion rules. The findings of the experiment reveal that the C4.5 decision tree is viable, effective, and accurate. His research demonstrates that the C4.5 decision tree is a viable technique for implementing decision trees, with about 90% classifier accuracy. However, the mistake rate remains the same with this method.

Hong Kuan Sok et al. [10] provide work on feature reduction using the ADTree algorithm. ADTree also performs well in classification. Furthermore, its easy-to-understand decision rules enable the user to find factors that lead to a higher classification. This knowledge base makes it easier to create support vectors with a reduced dimension for a better classifier. The experiment backs up the idea of using this algorithm as a categorization and knowledge-finding tool. Due to the fewer procedures required to accomplish the categorization, the process has been simplified and the speed has increased dramatically.

Tavallaee et al. [11] submitted a study on the KDD CUP 99 Data Set, and after analyzing the full KDD dataset, it was discovered that there were two major flaws in the data set that impacted the performance of the assessed systems, resulting in a poor evaluation of anomaly detection algorithms. NSL-KDD, which comprises selected records from the KDD data set, was presented as a solution to the problems. Even though the proposed data set has some flaws and may not be a perfect representation of existing networks, they believe that it can still be used as a useful benchmark to help researchers compare different intrusion detection systems due to the lack of public data sets for network-based IDSs.

F. Amiri et al. [12] proposed the Feature Selection approach to improving the performance of existing classifiers by removing irrelevant information. In addition, PLSSVM, an enhanced Partial Least Squares Support Vector Machine, has been introduced. In this study, a linear and non-linear measure for feature selection during the pre-processing phase was investigated. PLSSVM classified normal and probing assaults data with an accuracy of 95.69 percent and 86.46 percent, respectively. By using linear correlation-based feature selection (LCFS), forward feature selection (FFSA), and modified mutual information, the effect of adjusting feature goodness measure and evaluation function has been examined in this work. Experiments on the KDDcup99 dataset show that feature selection techniques can enhance classification accuracy significantly. PLSSVM, on the other hand, missed a large number of dynamic attacks, such as DoS and U2R attacks, which behaved similarly to normal behavior (78.76 percent and 30.7 percent, respectively).

An alternating decision tree with boosting is proposed by Yonav Freund et al. [13]. The new learning algorithm combines decision trees and boosting. They compared the alternating decision tree to the C5.0 method in their article. On smaller datasets, ADtree quickly fits the data, and after 50 iterations, ADtree has a relatively modest error, whereas the stump boost's error remains substantial even after 200 iterations. This is an instance where ADtree's big capacity is advantageous. When comparing the size of classifiers, the ADtree classifiers are substantially smaller than those generated by C5.0 by boosting in all but three cases.

5 Thesis Objectives and deliverables

- Techniques for Supervised Intrusion Detection (Using a decision tree, random forest, extra tree, extra trees, k-Nearest Neighbors (KNN), SVM, logistic regression, Naive Bayes, gradient boosting, ensemble learning and multilayer perceptron (MLP), and transfer learning).
- Techniques for Unsupervised Intrusion Detection (Using ensemble learning, Autoencoder, MLP, LSTM, and GAN).
- Predicting Bandwidth Utilization (Using the Autoregressive Integrated Moving Average (ARIMA), the Multilayer Perceptron (MLP), and the Long Short-Term Memory network (LSTM) algorithms).
- To check the performance of Machine Learning techniques in detecting intrusions by comparing their sensitivity, specificity, and accuracy. (Applying detection rate, Accuracy, Precision, F1-score and ROC Curve).

6 System Architecture/ Methodology

The proposed strategy is to split the project into two sections. In the first phase, three attribute assessment methods are used to choose features from the retrieved feature set. The second phase trains machine learning classifiers with the selected feature and compares them by estimating their efficiency.

We downloaded the KDD Cup dataset for the supervised Intrusion Detection System. Intrusion detection should be possible with the algorithms. It is accomplished by identifying the characteristics that are responsible for their actions. In the presence of better features, classifiers will be better trained, which will improve their performance in predicting incursions. There are three stages to the methodology:

Feature normalization is done in Phase I by rescaling one or more attributes to have a mean of 0 and a standard deviation of 1.

Machine Learning classifiers such as Naive Bayes, Logistic Regression, SVM, Decision Tree, Random Forest, and GBDT / XGBoost are utilized in Phase II for training purposes.

We used Clustering features, PCA transformed features, and Feature engineering employing existing features in Phase III to build some extra features to our dataset.

Mechanism/Algorithm

Dataset Collection

To far, we have used KDD Cup 9, NSL-KDD datasets collected from Kaggle in our study.

Classification

To determine the data item's class, classification employs supervised learning. Machine Learning algorithms are programmed to distinguish between the various classes. Before using the classifiers, the features are standardized. The following classifiers are used to train and classify incursions in the current study, and they all belong to various fields of classification techniques.

Naive Bayes

It's a classification approach that uses probabilistic data. The classification is based on Bayes' theorem and the premise of conditional independence of attributes, which means that the existence of one feature does not affect the presence of others.

Bayes Theorem

$$P(A/B) \ = P(B/A)P(A)/P(B)$$

where:

P(A/B)=Probability of happening of A given that B has occurred.

P(B/A)=Probability of happening of B given that A has occurred.

P(A)= Probability of happening of A.

P(B)= Probability of happening of B.

Logistic Regression

Logistic regression models the probability of the default class. Logistic regression is named for the function used at the core of the method, the logistic function which is given as :

$$y = 1 / (1 \ + \ e^{-x})$$

SVM

For two-group classification issues, a support vector machine (SVM) is a supervised machine learning model that uses classification techniques. The hyperplane (which in two dimensions is simply a line) that best separates the tags is produced by a support vector machine using these data points.

Decision Tree

A Decision Tree Classifier is a non-linear ML classifier that makes judgments and classifies points into distinct categories using numerous lines/planes/hyperplanes, similar to an if-else statement.

Random Forest

It pertains to the ensemble classification approach. A decision tree is the basic unit of a random forest. The classification rules are used to create a decision tree, which is represented as a flowchart. It is constructed in such a way that the internal nodes represent a feature test, the branches represent the test result, and the leaf nodes represent class labels. Using the training data, Random Forest creates many decision trees and then calculates the mode of the output produced or the class predicted by the trees.

GBDT / XGBoost

The main advantage of XGB over gradient boosting machines is that it has several hyper-parameters that can be tweaked. Missing values are handled automatically by XGBoost. Parallelization, distributed computing, cache optimization, and other intuitive features are included.

7 Progress Made so far

Until I fulfil the first goal of the thesis, which is supervised learning algorithms to detect intrusions, as seen in the Gnatt chart. MIT Lincoln Labs prepared the dataset on which we worked. There are 4,94,021 data points and 42 characteristics in the dataset. After the dataset has been imported, it is cleaned to remove/impute NULL values and duplicates.

After that, we used Exploratory Data Analysis (EDA) to discover what the data could tell us in addition to the formal modelling. Python packages such as matplotlib, pandas, and seaborn are used in EDA.

Observations from EDA:

- Data points as “normal” (good connections):60.33%
- Bad connections, class “Neptune” : 35.594 % and “back”: 0.665 %
- Classes “rootkit.”, “load_module.”, “ftp_write.”, “multi-hop.”, “phf.”, “Perl.” and “spy.” have the least no. of data points.

The performances of the algorithms were measured using Confusion Matrix, precision, recall, and weighted f1-score.

Confusion Matrix:

It's a table that shows the classifier's performance on real-world data. The number of False Positives, True Positives, False Negatives, and True Negatives acquired after classification is shown in the confusion matrix.

Table 1: Confusion Matrix

	Predicted Class (Negative)	Predicted Class (Positive)
Actual Class (Negative)	True Negative(TN)	False Postive(FP)
Actual Class (Positive)	False Negative(FN)	True Positive(TN)

Using them, we calculate three quantities to determine the performance of the classifier:

- Sensitivity (True Positive Rate): It is the ratio of the number of correctly classified intrusions to the total number of intrusions in the dataset.
- Specificity (True Negative Rate): It is the ratio of the number of true negatives to the total number of actual negatives.

- Accuracy: It is the ratio of the number of correctly classified files to the total number of files in the dataset.

In our models, we employed TPR for sensitivity and TNR for specificity.

Vectorizing categorical data using One-hot encoding into service, flag, and protocol was the next stage.

The purpose of data standardisation was to rescale one or more attributes to have a mean of 0 and a standard deviation of 1.

The table below shows the results of various classification algorithms.

Table2:Conclusion after running the given models

Model	Train f1-score	Train TDR	Train FPR	Test f1-score	Test TPR	Test FPR
Naive Bayes	0.9671	99.40%	5.13%	0.9679	99.34%	4.91%
Logistic Regression	0.9813	99.81%	2.95%	0.9819	99.81%	2.76%
Support Vector Machine	0.9967	99.87%	0.48%	0.9966	99.87%	0.43%
Decision Tree - 1	0.9997	99.96%	0.0%	0.9986	99.90%	0.13%
Random Forest - 1	0.9999	99.98%	0.0%	0.9992	99.98%	0.13%
XG Boost - 1	0.9999	100.0%	0.0%	0.9994	99.98%	0.083%

We will employ these three classifiers ahead of the existing and feature engineered data because DT, RF, and XGBoost had the best performance.

Feature engineering :

Clustering features (using MiniBatchKmeans), PCA features, and constructing new features from the data (such as adding two current features and deleting two existing features) are used.

Conclusion after applying feature engineering:

Table3:Conclusion after running the given models post feature engineering

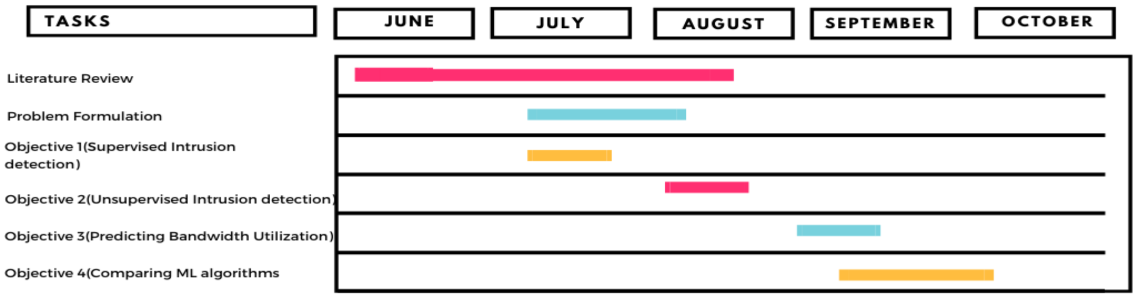
Model	Train f1-score	Train TDR	Train FPR	Test f1-score	Test TPR	Test FPR
Naive Bayes	0.9671	99.40%	5.13%	0.9679	99.34%	4.91%
Logistic Regression	0.9813	99.81%	2.95%	0.9819	99.81%	2.76%
Support Vector Machine	0.9967	99.87%	0.48%	0.9966	99.87%	0.43%
Decision Tree - 1	0.9997	99.96%	0.0%	0.9986	99.90%	0.13%
Random Forest - 1	0.9999	99.98%	0.0%	0.9992	99.98%	0.13%
XG Boost - 1	0.9999	100.0%	0.0%	0.9994	99.98%	0.083%
Decision	0.9998	99.97%	0.0%	0.9994	99.98%	0.083%

Tree - 2						
Random Forest -2	0.9999	99.99%	0.0%	0.9990	99.99%	0.15%
XG Boost - 2	0.9999	99.99%	0.0%	0.9994	99.98%	0.083%

8 Tasks to be completed:

- Techniques for Unsupervised Intrusion Detection (Using ensemble learning, Autoencoder [14], MLP, LSTM, and GAN)
- Predicting Bandwidth Utilization (Using the Autoregressive Integrated Moving Average (ARIMA), the Multilayer Perceptron (MLP) and the Long Short-Term Memory network (LSTM) algorithms) [15]
- To check the performance of Machine Learning techniques in detecting intrusions by comparing their sensitivity, specificity, and accuracy. (Applying detection rate, Accuracy, Precision, F1-score and ROC Curve) [16]

9 Gnatt Chart



Fig[1].Gnatt Chart

10 References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Ma chine Learning in Python ,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V.

Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “Ten sorflow: Large-scale machine learning on heterogeneous distributed systems,” arXiv:1603.04467 [cs], 3 2016. arXiv: 1603.04467

[3] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, and Z. Lin, “Automatic differentiation in pytorch,” p. 4

[4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Wit ten, “The weka data mining software: an update,” ACM SIGKDD Explorations Newsletter, vol. 11, p. 10, 11 2009.

[5] S. Axelsson, “The base-rate fallacy and the difficulty of intrusion detection,” ACM Trans. Inf. Syst. Secur., vol. 3, p. 186–205, Aug. 2000.

[6] Gisung Kim and Seungmin Lee (2014), A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection With Misuse Detection, ELSEVIER, Expert Systems with Applications vol. 41 pp. 1690 – 1700.

[7] Shi-Jinn Horng and Ming-Yang Su (2011), “Novel Intrusion Detection System Based On Hierarchical Clustering and Support Vector Machines”, ELSEVIER, Expert Systems with Applications. pp. 38 306-313.

[8] Mrutyunjaya Panda and Manas Ranjan Patra, “A Comparative Study Of Data Mining Algorithms For Network Intrusion Detection”, First International Conference on Emerging Trends in Engineering and Technology, pp 504-507, IEEE, 2008.

[9] Juan Wang, Qiren Yang, Dasen Ren, “An intrusion detection algorithm based on decision tree technology”, In the Proc. of IEEE Asia-Pacific Conference on Information Processing, 2009. [10] Hong Kuan Sok et.al, “Using the ADTree for Feature Reduction through Knowledge Discovery” Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International ,pp1040 – 1044.

[11] Tavallaee M, Bagheri E, Lu W, Ghorbani A. “A detailed analysis of the KDD CUP 99 data set”, 2009 IEEE Symposium on Computational intelligence for security and defense applications, 2009,pp 1-6.

[12] F. Amiri, M. Yousefi, C. Lucas, A. Shakery and N. Yazdani, “Mutual Information-Based Feature Selection for Intrusion Detection Systems”, Journal of Network and Computer Applications, Vol. 34, 2011, pp.1184–1199.

[13] Yonav Freund et.al, “The Alternating Decision Tree Algorithm”, ICML '99 Proceedings of the Sixteenth International Conference on Machine Learning, pp 124- 133.

[14] A. Ng, “Sparse autoencoder,” CS294A Lecture notes, vol. 72, no. 2011, p. 1–19, 2011.

[15] M. Labonne, C. Chatzinakis, and A. Olivereau 2020.

[16] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skorić, “Measuring intrusion detection capability: An information-theoretic approach,” in Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS ’06, (New York, NY, USA), pp. 90–101, ACM, 2006.