National College of Ireland

A Comparative Evaluation of Machine Learning Models and EDA through Tableau Using CICIDS2017 Dataset

MSc Research Project
Data Analyticss

Abhay Singh Bangari
Student ID: X21153507

School of Computing
National College of Ireland

Supervisor: Cristina Hava Muntean

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Abhay Singh Bangari
Student ID:	X21153507
Programme:	Data Analyticss
Year:	2022
Module:	MSc Research Project
Supervisor:	Cristina Hava Muntean
Submission Due Date:	15/12/2022
Project Title:	A Comparative Evaluation of Machine Learning Models and EDA through Tableau Using CICIDS2017 Dataset
Word Count:	6532
Page Count:	27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	1st February 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Comparative Evaluation of Machine Learning Models and EDA through Tableau Using CICIDS2017 Dataset

Abhay Singh Bangari
X21153507

Abstract

Machine learning is utilized globally in network security, but computers need time to learn. Machine learning can identify many hacker attacks that humans cannot. Business intelligence and machine learning are being studied to strengthen network systems. The research topic is briefly covered in the study. Academic articles on the study topic are examined, and effective research methods are described. In this work, Python is used as a medium to build popular algorithms and Tableau for visualizations. Machine learning models like AdaBoost, XGBoost, Random Forest, Decision Tree, KNearest Neighbor, and Linear Discriminant Analysis. The Canadian Institute for Cybersecurity's CICIDS2017 dataset is used for in-depth analysis. Performance metrics for all the algorithms are computed, with the help of accuracy, F1-score, precision, and recall. The following investigation revealed that XGBoost is the better-performing algorithm. The random forest model is the best-performing model in terms of accuracy (98.38%), and F1-score (98.37%). Contrary to others, AdaBoost and linear discriminant analysis models have been proven to be less effective at preventing intrusions. On testing with different numbers of features in the random forest, it is discovered that the model with 35 features improves its performance.

1 Introduction

Machine Learning improves most technologies Milenkoski et al. (2015) Modi and Acha (2017). Many studies have been done to integrate Machine Learning into cyber security Viegas et al. (2016), Buczak and Guven (2015). Gmail employs Machine Learning to identify spam. Making a country's digital defense system safer is necessary, but ML can't help Coelho et al. (2017). Bayes Net, Multilayer Perceptron (MLP), C 4.5 Decision Tree, and Naive Bayes Algorithm classify IP. Internet traffic growth poses cyber security issues. First, data stream volume makes manual analysis unfeasible. Second, the pace of new threats is great; hence short-lived, adaptive patterns are common. Detecting and predicting evasive dangers becomes difficult. Tackling these issues takes time and money, and it is costly and difficult to hire domain specialists Žliobaitė et al. (2015), Ghosh et al. (2017). Machine Learning needs data. Because ML approaches need data to learn, it's crucial to comprehend the data before designing a model or applying algorithms.Ghosh et al. (2017), L'heureux et al. (2017). Machine Learning has benefits in cyber security subfields. Behavior modeling helps distinguish normal from abnormal behaviors. They

are dynamically discovering new and nuanced assaults in signatures-tradecraft. Machine Learning adapts better to changing threat domains than traditional methods. Machine learning methods like bias-variance and precision-versus help reduce false alarms. Sharma et al. (2016), Pervez and Farid (2014). Both business intelligence and machine learning are technology-driven processes that analyze data and present all the relevant, actionable information. These processes are directly responsible for assisting business executives, management staff, and other enterprise users in making more effective decisions. Hackers are gaining access to new forms of technology regularly, allowing them to launch more damaging attacks on various companies. One of the most critical areas that the attackers focus on is the "network security systems" of the enterprises, which are vital. Therefore, business intelligence is essential to provide significant benefits to firms so that these organizations may get advantages and keep their infrastructures strong. Simulated data is considered here, the need for business intelligence to satisfy the organization's work security and firewall management needs. The remainder of the paper is organized as follows. Section II of the report presents the literature review and highlights the most notable works in the field. Section III of the form explores various cybersecurity applications that use machine learning. Section IV focuses on the cybersecurity data sets for machine learning that includes network flow and packet-level data. Different ML techniques related to cyber security are detailed in section V, and finally, the paper concludes in section VI with several challenges in providing cyber-attacks.

1.1 Background

In today's networks, many different pieces of hardware are in use, each of which has its unique logging format. It is only possible to know the proper true health of a company in each market by collecting statistical data Villamarín and Diaz Pinzon (2017). Integrating data collection, storage systems, and knowledge management with analysis instruments for presenting "complicated internal & competitive information" to decision-makers and planners, a business intelligence system is a powerful decision-making tool. The BI tools may be accountable for enhancing data opportunities and transparency through scorecards, dashboards, graphs, and other displays. Administrations conducting company management in an online context need access to appropriate internal statistical data Kim et al. (2017). Data security in corporate networks is of the utmost importance. By constantly monitoring and improving the processes that drive the changes, business intelligence may be held accountable for safeguarding information security and network security. Additionally, businesses may use BI to keep tabs on their data's security and build a more secure network architecture and culture. Traditionally, IDS systems have been used to ascertain the IP addresses of hackers and identify any network vulnerabilities that may exist. The administrator changes the firewall's configuration and manually queries the QS logs for information on the data streams and service types. With the assistance of BI, businesses may better secure their networks by installing firewalls and other barriers to prevent unauthorized users from accessing sensitive data.

1.2 Research Aim

The study's overall objective is to uncover the significance of business intelligence to the safety of corporate networks and the myriad threats such networks face. . The investigation may try to shed light on the advantages and disadvantages of so-called "network

security systems" as well as how the systems' flaws are mitigated using integrated business information. This study aims to identify the critical component of network security and conduct an in-depth analysis of those components using several different machine-learning models. This will be accomplished using several other methods and tools, including machine learning, Python, and Tableau.

1.3 Objective

- To evaluate the impact of business intelligence on the "network security system."
- To identify the major network security challenges.
- To determine effective methods and techniques for network security by identifying a cyber-attack before it happens.
- Application of machine learning in cyber security.
- Understand the pattern of the attackers.

1.4 Research Question and Statement

- A comparison study of the performance of XGBoost with other machine learning models used to predict malicious attacks in network security systems.
- What is the distribution of each cyber-attack on a particular day of the week?
- What are the different approaches to predicting and avoiding network attacks using machine learning algorithms?
- Application of machine learning in cyber-security.
- To find out whether the efficiency and accuracy of the highest performing algorithm out of all the algorithms used increases or decreases with different numbers of features.

The remaining sections of the paper are structured as follows: In Section 2, we offer exciting studies. The methodology and experimental design for this investigation are briefly discussed in Sections 3 and 4, respectively. I've provided additional information on the implementation and experiments conducted in Section 5. Section 6 gives information about the results or output obtained. Sections 7 and 8 conclude the research study with suggestions and the final findings achieved during the analysis.

2 Related Work

2.1 Detailed Analysis of Imbalanced Data for a Wireless Network

Class imbalance is a known issue in NSL-KDD, as outlined by Rodda and Erothi Rodda and Erothi (2016). Researchers have tried out four different classification methods (Naive Bayes, Bayes Network, J48, and Random Forest) and found that none could accurately categorize a class with a sparse distribution.

2.2 Interpreting Performances of Several Machine Learning Algorithms

Belavagi and Muniyal (2016)Research published in 2016 evaluates the efficiency of several machine-learning techniques by applying them to a dataset known as NSL-KDD, which has 42 attributes for each record. RF is the algorithm that performs the best, with an accuracy of 99%. This study needs the implementation of the classifiers for the multi-class classification and considers all the attributes instead of the most important ones. In the future, it is suggested to use the classifiers for multi-class types (the categorization of numerous assaults) and with essential attributes.

2.3 Comparision of Machine Learning Models on Three Different Datasets

The authors built an ML plugin for Snort in a prior work from 2017 that used the Weka library. The plugin was run in parallel with Snort's analyzer to identify attacks that were previously unknown or had been changed. This not only helped minimize the number of false alarms generated by Snort's analyzer but also detected previously undisclosed hostile traffic. The ML models were trained on three distinct datasets before being tested on newly generated traffic. SVM, decision trees, fuzzy logic, BayesNet, and Naive Bayes, in addition to hybrid versions of these algorithms, were the ones that were examined. The hybrid implementations of the SVM method yielded the greatest overall performance in this test. In the future, it is recommended that their research include other detection systems, other hybrid machine learning algorithms, and additional fine-tuning of parameters. Gustavsson (2019)

2.4 Generation of a New Dataset for the Detection of Intrusion in the Network Systems

The CICIDS2017 dataset was evaluated by its producers with Scikit-Learn ML algorithms. The CICFlowmeter extracted the first 80 unique characteristics needed for TCP/UDP flow. They accurately identified the traffic with a success rate of 98% by using Random Forest to determine the most critical criteria for detecting each assault. The standard deviation of flow inter-arrival time and the standard deviation of packet length in the opposite direction were very helpful for identifying DDoS activity. RF (98%), ID3 (98%) (a kind of Decision Tree), KNN (96%), and QDA (97%), which is the highest accuracySharafaldin et al. (2018). Thirteen computers were employed for this use. There are much accessible that could be used instead, and newer threats might also be considered in the future.

2.5 Evaluating CSIC 2010 HTTP Dataset Using Machine Learning Technique to Improve Detection Rates

To identify cyber-attacks directed at online applications, Nguyen and Franke (2012) suggested an innovative approach. This study evaluates this strategy compared to the machine learning algorithms AdaBoost, Naive Bayes, Part, and J48. This model is evaluated using the CSIC 2010 HTTP dataset. Solutions that facilitate client-server communication

over HTTP are the primary research emphasis here. The author asserts that his methodology may improve detection rates without simultaneously increasing false positives. The J48 technique is very efficient for this issue, yielding a true-positive value of 0.04.

2.6 Using Deep Learning along with Deep Q Networks and Multiple GPUs for Intrusion Detection

In a 2018 master's project, researchers applied deep learning using the Deep Q Network algorithm to the CICIDS2017 dataset, achieving a 92% success rate. Furthermore, they claim that their approach may identify previously unseen assaults Janagam and Hossen (2018). Using more than one graphics processing unit (GPU) may help speed up the running time. In addition, other methods might be utilized, such as neural networks, fuzzy logic, etc.

2.7 Feature Selection Technique Using Discrete Differential Evolution and Decision Tree

Discrete Differential Evolution (DDE) and the C4.5 machine learning method are discussed in the study Popoola and Adewumi (2017). The suggested method achieves 99.92% with only 16 valuable features in classification accuracy. The models developed in this work may be tested on other authors' available datasets. These datasets will include the most recent attack and characteristics not found in the dataset analyzed in this study.

2.8 NSL-KDD Dataset For SVM-Based Feature Selection Technique

Pervez and Farid Pervez and Farid (2014) developed an SVM-based feature selection technique that achieved 91% accuracy with three features and 99% accuracy with 41 features across all training sets; however, when tested on a dataset it had never seen, its classification accuracy dropped to 82.37%.

2.9 Graphed-Based Feature Clustering for Bonet Detection

Garg and Kumar Chowdhury et al. (2017)examined several criteria ranking and sorting procedures. Two or three feature selection approaches were combined using Boolean AND to see how well they performed. When employing 15 features and the IBK classifier, the pairing of symmetry and gain ratio for feature extraction had the most fantastic accuracy out of the ten methods examined. However, it is still being determined why and how the dataset was randomly picked, which means that the outcome cannot be repeated. Topological features of nodes in a network were provided as a novel approach to botnet identification in a paper by Chowdhury et al. Chowdhury et al. (2017). This technique may be utilized to discover anomalies using a small number of nodes. The suggested model is an unsupervised system that relies heavily on clustering, specifically a self-organizing map (SOM). Specifically, the CTU-13 datasets, a massive dataset that includes the bot-labeled nodes. The suggested technique can identify a bout with acceptable accuracy by scanning just a small number of nodes, as shown by comparison with the Support Vector Machine (SVM) method, which is also used to detect the same.

2.10 Machine Learning for Adaptive Detection System

Naive Bayes is used for feature selection, and principal component analysis (PCA) is provided as a framework for creating a network intrusion detection system Soni and Bhushan (2019). For this study, we recommend using the intrusion detection dataset from KDDCup 1999. The findings demonstrate the superiority of the suggested technique over decision tree and neural network-based approaches in terms of detection rates, the time required, and overall cost. An accuracy of 94% has been achieved with this model.

2.11 Application of Machine Learning Cyber-Security

[1]

- As cybersecurity is essential in determining if cyber threats have penetrated the network, The most challenging aspect of cybersecurity is deciding whether the connection requests into the system are doing any malicious activity, such as transmitting or receiving data. Machine learning may greatly assist experts in this regard, allowing them to identify cyber dangers more quickly and accurately. Cyber threat identification enabled by AI may also be utilized for call monitoring and system surveillance.

- Antivirus software should be installed on all computers before they are used, as this will ensure that they are protected from any potential malware that may be spread across the network. No virus can evade detection by antivirus software incorporating machine learning to identify and warn users of potential threats.

- Considering the widespread usage of this technology by hackers, machine learning may be employed to pinpoint the locations of any resulting cybersecurity flaws. Businesses should also use machine learning for cybersecurity objectives. This method of protecting against cyberattacks has the potential to become the industry standard.

- By analyzing historical datasets of cyberattacks, machine learning may help identify which parts of the network are most frequently targeted by attackers. This can be used to assign a score to the assault in a specific region of the network.

2.12 Challenges Faced in Preventing Cyber Attacks

- Locking the files on a victim's device and then demanding payment to unlock them is a common tactic used by ransomware. Successfully settling a debt allows the surviving party to reclaim all network privileges. Data specialists, cyber security experts, IT personnel, and corporate leaders all agree: ransomware is the worst. The prevalence of ransomware attacks is growing in the cybercrime world. To protect their businesses, organizational leaders and IT managers need a solid strategy for dealing with Cyberattacks Zimba et al. (2018). Recovery of data and service restoration from businesses and customers, as well as notification of violations of the Notifiable Data Theft Program, need extensive planning and preparation.

[1]For More Information: `https://www.analyticsinsight.net/top-10-applications-of-machine-learning-in-cybersecurity/`

- When compared to other technical developments during the same period, blockchain technology stands above the others. It is the first time in human history that a native digital medium exists to exchange value between individuals. Cryptocurrencies like Bitcoin rely on the blockchain mechanism. Blockchain helps to conduct a transaction or business with another party or parties without the requirement for a reliable third party, which is a massive global platform Narang et al. (2014). It is not easy to foresee blockchain systems' benefits to cyber security.

- The fundamental benefit of AI in our method of information defense is that it allows us to safeguard and protect a network before the malware assault even begins. Artificial intelligence responds instantly to hostile assaults, preventing further damage to a company. IT business executives and information security risk management groups see AI as a future protective control to help our organization keep on top of the cyber security growth curve.

2.13 Conclusion

The studies/related works discussed above have some limitations, and in this research study, all the challenges faced in them are handled properly. Some of the challenges and how they were tackled are mentioned ahead. In some of the existing works, the datasets used are outdated and have a limited number of features; for example, the NLS-KDD dataset has 42 attributes UNSW-NB15 has 49 attributes, whereas to tackle this issue dataset (CICIDS 2017) is used that has about 80 features which improved quality of the research work. Random forest was widely used in most of the research works. In this study, additional classification and machine learning methods are also incorporated that could handle high-class datasets to understand better and carry out a model comparison. Lack of business intelligence is also covered in this study. Tableau software is used to have in-depth knowledge related to the dataset that would help the researchers understand the hidden information so that they can work in the right direction.

3 Methodology

This section will discuss the study approach, known as "knowledge discovery in database" (KDD). The following is a rundown of the procedures that make up the methodology: [2]

The area of Knowledge Discovery in Databases, or KDD, was initially formed in 1989 with the purpose of comprehensive data searching. After then, "data mining" was created to present and analyze data for decision-makers. Knowledge discovery and development (KDD) employs data mining to efficiently find patterns and structures in data.

3.0.1 Stages of KDD

[3]

- Data Selection.

[2]KDD Chart `https://www.geeksforgeeks.org/kdd-process-in-data-mining/`

[3]Stages Explained in Detail `https://www.geeksforgeeks.org/kdd-process-in-data-mining/`

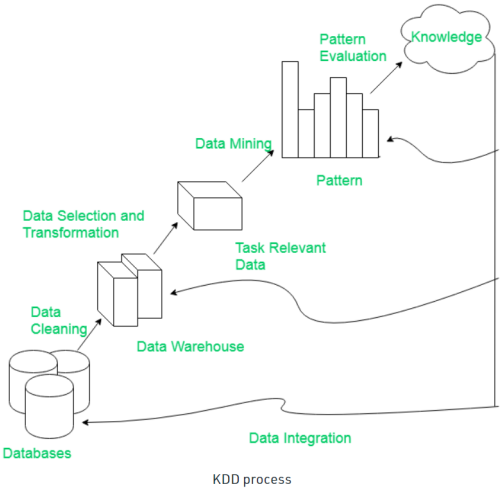

Figure 1: Flow of KDD Methodology

- Data Pre-processing.
- Data Transformation.
- Data-Mining.
- Data Interpretation and Evaluation

4 Design Specification

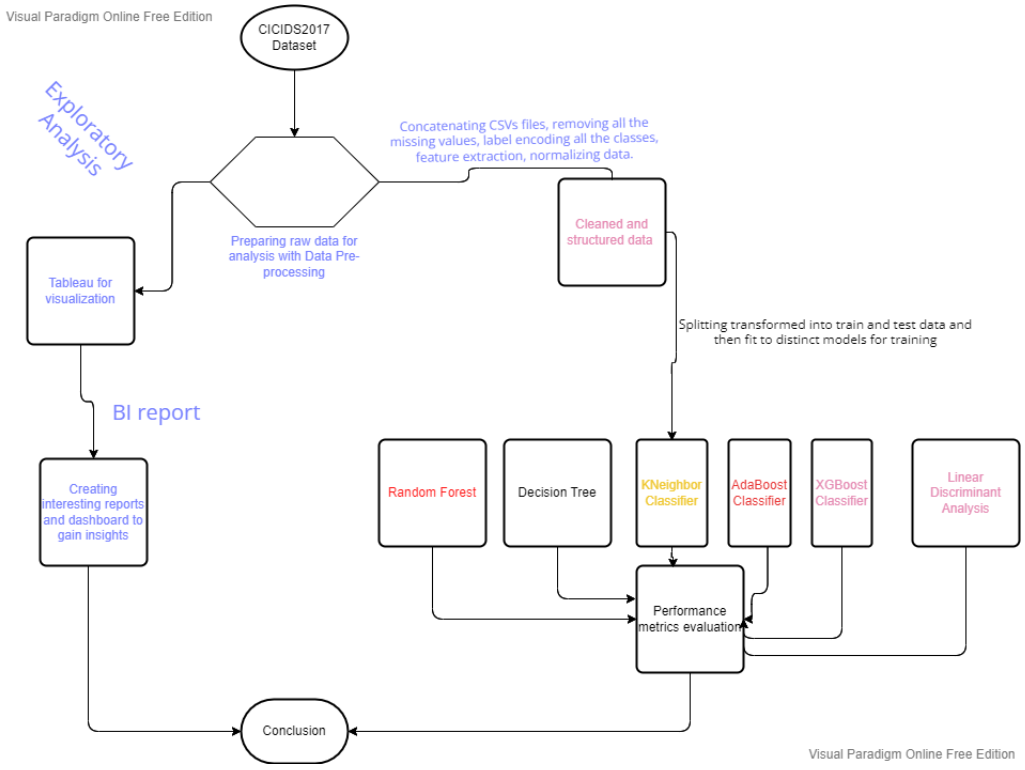

Figure 2: Design

The image above shows the flow that has been followed throughout the research to achieve all the objectives. Firstly, the dataset is taken from the public domain. It consists of 8 CSV format files, of which only five were used for the analysis. After gathering the data, pre-processing was performed where missing values were removed, and label encoding was done. Apart from these two, the data was normalized using standard scalar, and features were extracted using selectkbest. The next stage is data exploratory, which is done on Tableau. After removing all the essential insights and pre-processing the data, it is time to dive into the modeling. Several models were built and implemented, such as AdaBoost, random forest, xgboost, linear discriminant analysis, k-neighbor classifier, and decision tree. In the end, all the results obtained were compared, and interpretations were made. The entire experiment was carried out in a way that fulfilled our objectives.

5 Implementation

5.1 Dataset

File Name	Type of Traffic	Number of Record
Monday-WorkingHours.pcap_ISCX.csv	Benign	529,918
Tuesday-WorkingHours.pcap_ISCX.csv	Benign	432,074
	SSH-Patator	5,897
	FTP-Patator	7,938
Wednesday-WorkingHours.pcap_ISCX.csv	Benign	440,031
	DoS Hulk	231,073
	DoS GoldenEye	10,293
	DoS Slowloris	5,796
	DoS Slowhttptest	5,499
	Heartbleed	11
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Benign	168,186
	Web Attack-Brute Force	1,507
	Web Attack-Sql Injection	21
	Web Attack-XSS	652
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv	Benign	288,566
	Infiltration	36
Friday-WorkingHours-Morning.pcap_ISCX.csv	Benign	189,067
	Bot	1,966
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Benign	127,537
	Portscan	158,930
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Benign	97,718
	DdoS	128,027
Total Instance/ Record		**2,830,743**

Figure 3: CICIDS2017

This dataset from Canadian Institute for Cyber-Security[4] is used for this analysis. Each session in MachineLearningCSV is recorded in its comma-separated value (CSV) file. There are eight (8) total sessions. The file includes "attacks." More information on attack traffic is provided in the second section of Figure 2. There are 14 separate attacks in this data collection, as opposed to the benign and regular varieties of traffic. In this paper, the authors consider intricate details that stand in for complicated assaults on a contemporary network's traffic characteristics. For instance, detecting infiltration and bot attack types needs features like sub-flow forward bytes and total length forward package, neither of which are present in NSL-KDD. Attacks like DDoS, DoS Hulk, DoE

[4]Link to Dataset: `http://www.unb.ca/cic/datasets/IDS2017.html`

GoldenEye, and Heartbleed attacks can only be detected using the Bwd Packet Length Standard function. Web-Attack, SSH-Patator, and FTP-Patator assaults cannot be seen without the Init Win Fwd Bytes component. In contrast, regular traffic requires the Min Bwd Package Length and Fwd Average Package Length characteristics [58]. Figure 2 shows the more sophisticated forms of attacks seen in CICIDS-2017. The goal of using the CICIDS-2017 dataset in the experiments is to have a dataset that is a near approximation of current real-world network traffic.Sharafaldin et al. (2018)

5.2 Data-Preprossessing And Data Analysis

The very first step is to import all the necessary libraries. Majorly, sklearn is implemented throughout the analysis. For mathematical computation, the NumPy library is used. Also, the Pandas library is executed during the study. Pandas is a library that provides high-level data structures and an extensive range of analytical capabilities. Pandas, Numpy, XGBoost, Random Forest, Decision Tree, AdaBoost, QDA, LDA, GaussianNB, KNN, and sklearn are the primary drivers of the implementation. The SweetViz library is used too. Sklearn is used primarily for pre-processing.
The steps followed are as follows:

- Loading data
- Exploratory data analysis

In the MachineLearningCVE file, all the files are in CSV format, so Pandas is the most preferred library to load the dataset. The index is ignored while loading the data into the data frame. Five datasets were loaded separately and combined using the concatenation function, a pandas function. Next, it is beneficial to know the missing values in your dataset. So a function named missing_data was defined that will count the number of missing values along with the percentage of those missing values related and returns a data frame containing the total of missing values and the percentage of the same for features that have missing inputs. Please refer to Figure4 as it shows Flow Packets/s and Flow Bytes/s have 2167 missing values, respectively. After detecting the missing values, you either remove them or fill them with an appropriate value. The method used in this research to deal with missing values is deleting them from the dataset because eliminating the rows containing the missing values would not make much difference as there are, in total, 2064641 rows. After performing this step, the total number of rows in the dataset is 2062474. The cleaned data frame is now converted to a new CSV file, which is used for machine modeling in later sections. To summarize the newly formed data, the SweetViz library is used, which will summarize the entire dataset, and an HTML of the same is generated. [5]

Figure5 below shows all the target labels, and corresponding value counts for each available label in the data. Target labels present different classes of network attacks. The BENIGN class has the highest count in the data, which is 1666714.

[5]Drive link for the summary report: `https://drive.google.com/file/d/1tILTxAfQHAt2rk-1sC11ufYIT6bBRD1Z/view?usp=sharing`

	Total	Percent
Flow Packets/s	2167	0.104958
Flow Bytes/s	2167	0.104958

Figure 4: Total Numbers of Missing Values and Percentage of Missing Value

```
BENIGN                      1666714
DoS Hulk                     230124
DDoS                         128025
DoS GoldenEye                 10293
FTP-Patator                    7935
SSH-Patator                    5897
DoS slowloris                  5796
DoS Slowhttptest               5499
Web Attack Brute Force         1507
Web Attack XSS                  652
Web Attack  Sql Injection        21
Heartbleed                       11
Name:  Label, dtype: int64
```

Figure 5: Taget Label and Value Count

Looking at figure6, it is clear that the data is imbalanced, and this issue must be dealt with. In our case, balancing the data is not feasible as downscaling would cause a significant loss of the data, and for upscaling the data, we must upscale all 11 criteria to 16.6 lakh, which is not the right approach. Using the data without balancing would not cause any harm to the performance as the gap between accuracy and F1-score is not high.

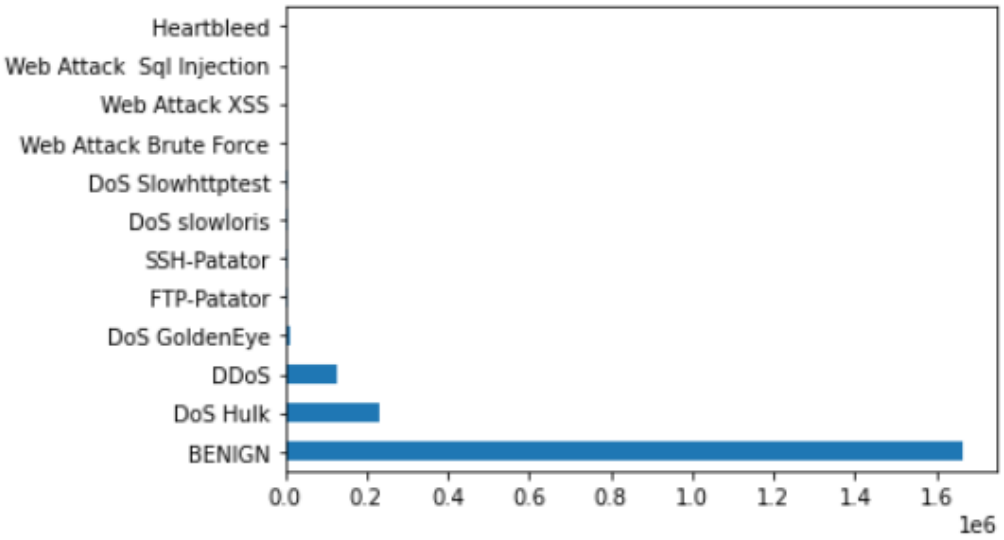

Figure 6: Imbalanced Data

Since the target labels are all categorical, they should be converted to numerical data so that machine learning models can understand the data efficiently and extract essential and hidden insights. For this purpose label encoder is used from the sklearn. All the labels are assigned unique numerical values.
Selecting relevant features is the most crucial step before getting into model training with the data. With only the essential elements, high-quality model training can be achieved. It helps reduce the disturbance and improve the overall accuracy of the model. Additionally, it makes the model activity fast. Hence, prediction can be made quickly if model training is fast. SelectKBest is a feature selection with chi2 as a score function, as it is a classification issue. It scores the features that help keep only the relevant parts in our dataset. SelectKBest feature selector was applied to the whole dataset, and later splitting was performed to attain train and test datasets. The conventional approach is to make the feature selection after splitting as there can be information leakage, if done before, from the Test-Set. The contradictory argument used in our study is that only the Training Set picked from the entire dataset is utilized for Feature Selection. The feature selection or feature necessary score orders are expected to be dynamically modified with a change in the random_state of the Train_Test_Split. Additionally, it is undesirable if the feature selection for any given work varies because this prevents the application of a generalization of feature importance. Second, because the entire historical data is not examined, if only the Training Set is utilized for feature selection, the test group may contain specific cases that defy or contradict the feature selection made exclusively on the Training Set. For the initial stage, 30 features were extracted for training all the models. Before diving into the modeling, splitting the entire dataset into training and testing data is essential. Training data is used for model training, and test data is used for testing the model's performance. Here, the data is split in the ratio of 75:25, respectively. Simultaneously standardizing the data is done using a standard scaler from the sklearn library. The standard scaler is a technique for normalizing data so that the converted feature has a 0 mean and one standard deviation. The converted features tell us how far the original part is from the feature's mean value, often known as a z-score in statistics.

Our data is ready to be fed to the model for training purposes.

5.2.1 Tableau for Visualizations

For visualization, Tableau is the preferred application. Tableau is among the best-known BI tools out there. It's a dashboard app that makes it easy to compile summaries and draws insights from data via graphics and other forms of graphical representation. All the visualization generated on the tableau is discussed in the evaluation section.

5.3 Machine Learning Models

Once the entire dataset is split into the training and test datasets, the next step is to fit that dataset into the model for training purposes. Machine learning models like AdaBoost, XGBoost Classifier, Random Forest Classifier, Linear Discriminant Analysis, and Decision Tree were implemented one after the other, and their performance and other performance metrics were evaluated. Majorly sklearn library is used for importing the models.

In the end, the machine learning model with the highest accuracy is further tested with different features. Initially, all the models were tested on the top 30 components, and now the best model will be tested on 15, 22, and 35 parts and compared with each other on performance level. Below Each machine learning model is thoroughly defined.

5.3.1 Adaboost Model

The AdaBoost method tries to manipulate training samples to produce different hypotheses. It operates on the weighting concept, with each model correcting the previous model's mistake. This technique records a probability distribution value on the training data and repeatedly generates a dimensional training dataset by sampling with variation based on this value. Following that, the learning algorithm is used to produce a classifier. The error rate of the classifier is calculated using the training sets. The weights are then assigned based on the error value, with a higher weight given to misclassified data points. In this manner, the mistake will be considered in subsequent models.

5.3.2 XGBoost Model

Machine-learning rivals highly regard this model. These days, XGboost software is helpful for linguistic data analysis and a gradient in giving a framework for boosting. Its applications include issue classification and regression, which is why this program performs well in classifying binary problems. Predictions of many regressions are made hierarchically in this model, and weighted outcomes are retrieved simply by weighing incorrectly labeled instances more strongly.

5.3.3 Random Forest

To ensure a successful implementation of tree classifiers like Random Forest, it is recommended that the number of trees is set in advance. There is one decision tree for each "tree." Randomly chosen characteristics from the dataset are used to populate each tree. Because of this, the random tree classifier may be seen as a set of discrete decision trees. To make a comprehensive prediction, we must migrate the results of many decision trees

and select the predicted class with the highest number of votes. Since it can process data with several types, the classification by random forest is employed for validation.

5.3.4 Decision Tree

Decision trees are tree diagrams in which the inner nodes stand for characteristics, the branch nodes for evaluation rules, and the outside nodes for results. When building a decision tree, the root node is always at the top. The system learns to divide data depending on many attributes.

5.3.5 KNearestNeighbor

kNN is a supervised learner that includes both classification and regression. Supervised machine-learning algorithms may be broken into two types, depending on the kind of target attribute they can forecast. Classification involves categorical prediction.

5.3.6 Linear Discriminant Analysis

LDA is a machine-learning classifier that calculates the average and standard deviation for class-labeled input characteristics.

5.4 Metrics Used

Every machine-learning pipeline has performance metrics as part of it. They tell you how far you've come and give you a number for it. Whether linear regression or a SOTA technique like BERT, all machine learning models need a method for evaluating how well they perform. A few terminologies are used in each of the equations below; their complete forms are mentioned below.

where TP = true positives, TN = true negatives, FP = false positives, and FN = false negatives.

5.4.1 Accuracy

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

One way to judge classification models is based on how efficiently they function. Informally, accuracy is how many of our model's predictions turned out to be true.

5.4.2 F1-Score

[6] F-Measure allows you to integrate accuracy & recall into a single measure that includes both. Both accuracy and recollection, by themselves, provide only part of the narrative. We might have good precision with poor memory or poor precision with outstanding recall. The F-measure allows you to convey both issues with a single numerical result. Once accuracy and recall for a binary or multi-class task have been obtained, the two scores may be combined to calculate the F-measure.

[6] F1-Score: https://developers.google.com/machine-learning/crash-course/classification/accuracy

5.4.3 Precision

[7]

$$\text{Precision} = \frac{\text{True Positive}}{\text{TruePositive} + \text{FalsePositive}} \tag{2}$$

A classification model's capacity to detect only relevant data items Precision is defined mathematically as the number of true positives divided by the sum of the numbers of TP and FP.

5.4.4 Recall

[8]

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \tag{3}$$

A model's capacity to discover all relevant examples within a data collection. We define recall mathematically as the number of true positives plus the number of false positives.

6 Evaluation

6.1 Visualization

Tableau is a business intelligence & data visualization application that enables users to make sense of the information they have access to via interactive diagrams, charts, and graphs. Use visual aids such as graphs, charts, plots, and so on to show the results. Each visualization is discussed thoroughly, and vital interpretations are mentioned in the next section.

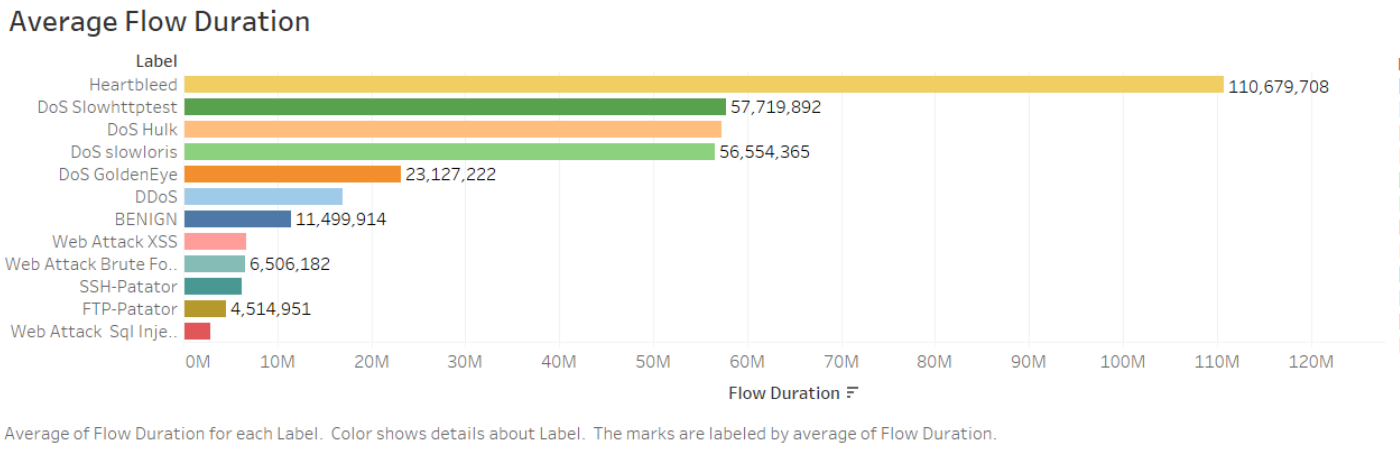

Figure 7: Average Flow Duration Per Attack

[7]Precision: `https://builtin.com/data-science/precision-and-recall`
[8]Precision: `https://builtin.com/data-science/precision-and-recall`

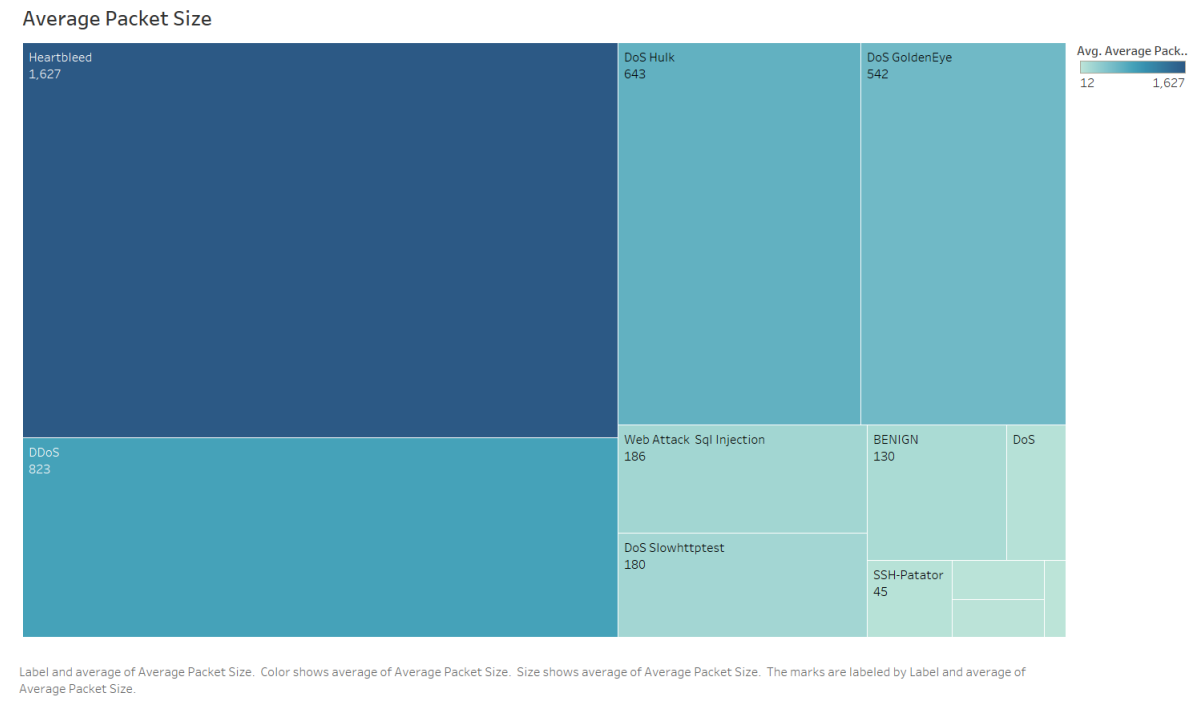

Figure 8: Average Packet Size

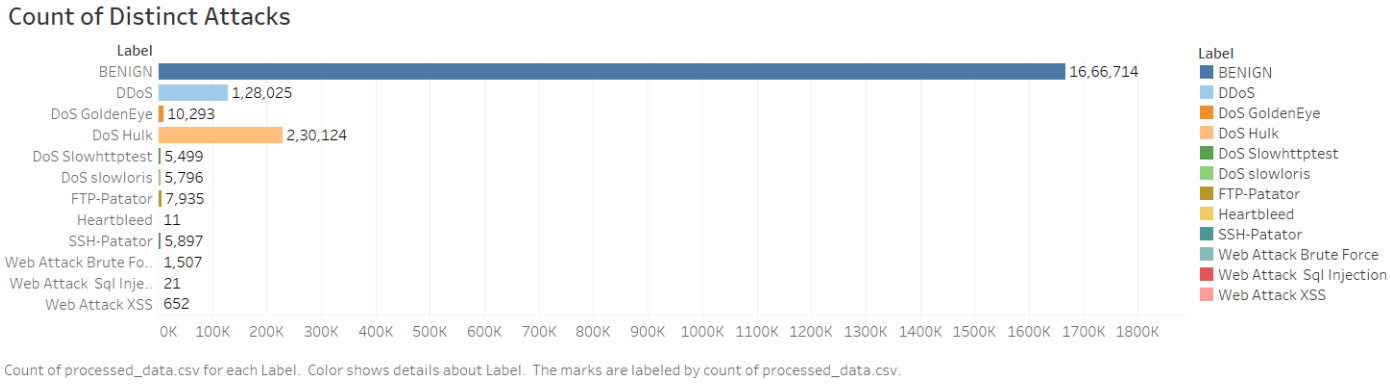

Figure 9: Frequency of Distinct Attacks

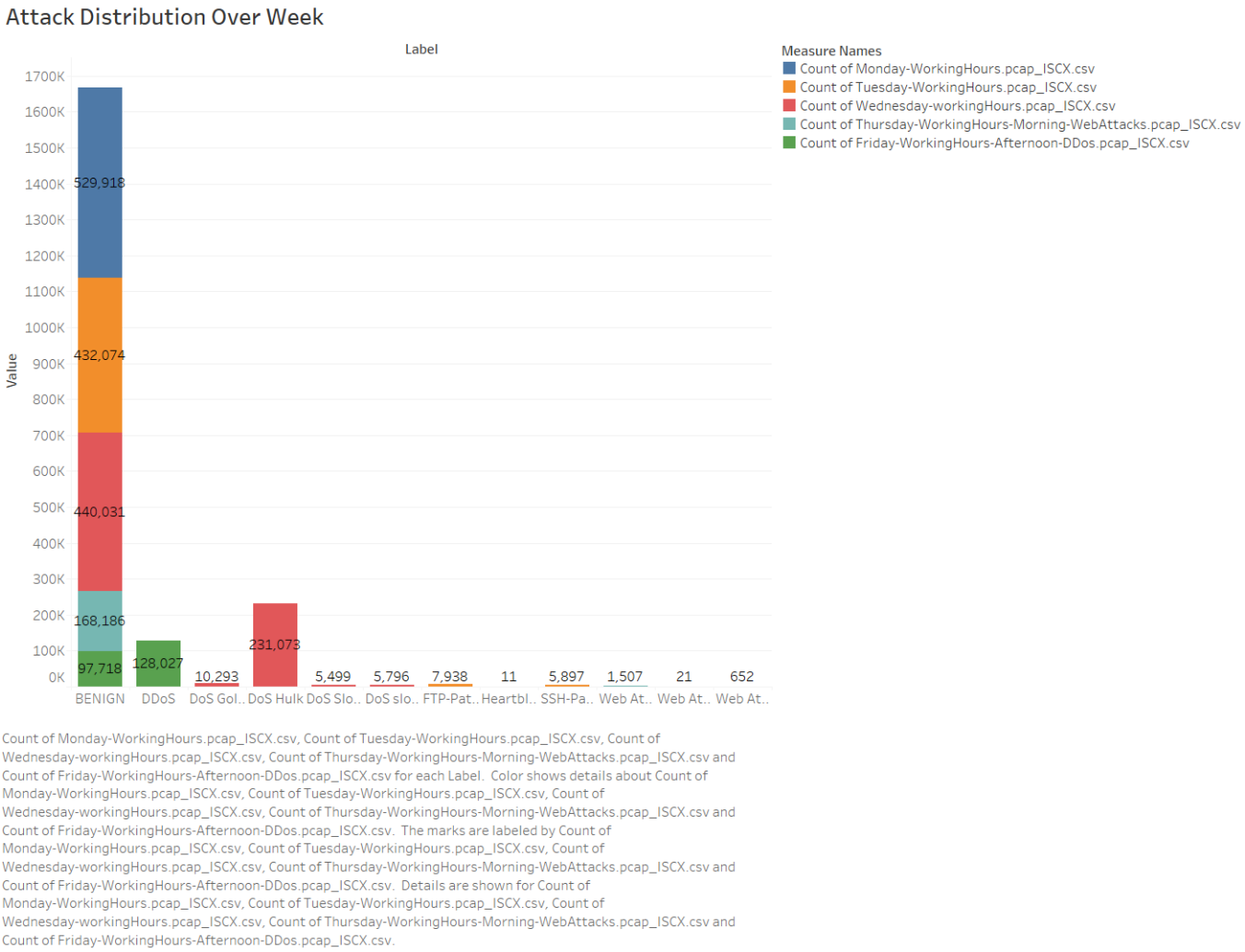

Figure 10: Attacks Distributions Over Week

Finally, the dashboard shown in Figure 11 is generated where all the previously mentioned chats are clubbed in one frame. This dashboard is interactive and can be published online so that others working from around the world can access it easily and quickly. The link mentioned in the footnote will navigate you to the dashboard published on the tableau public online platform.

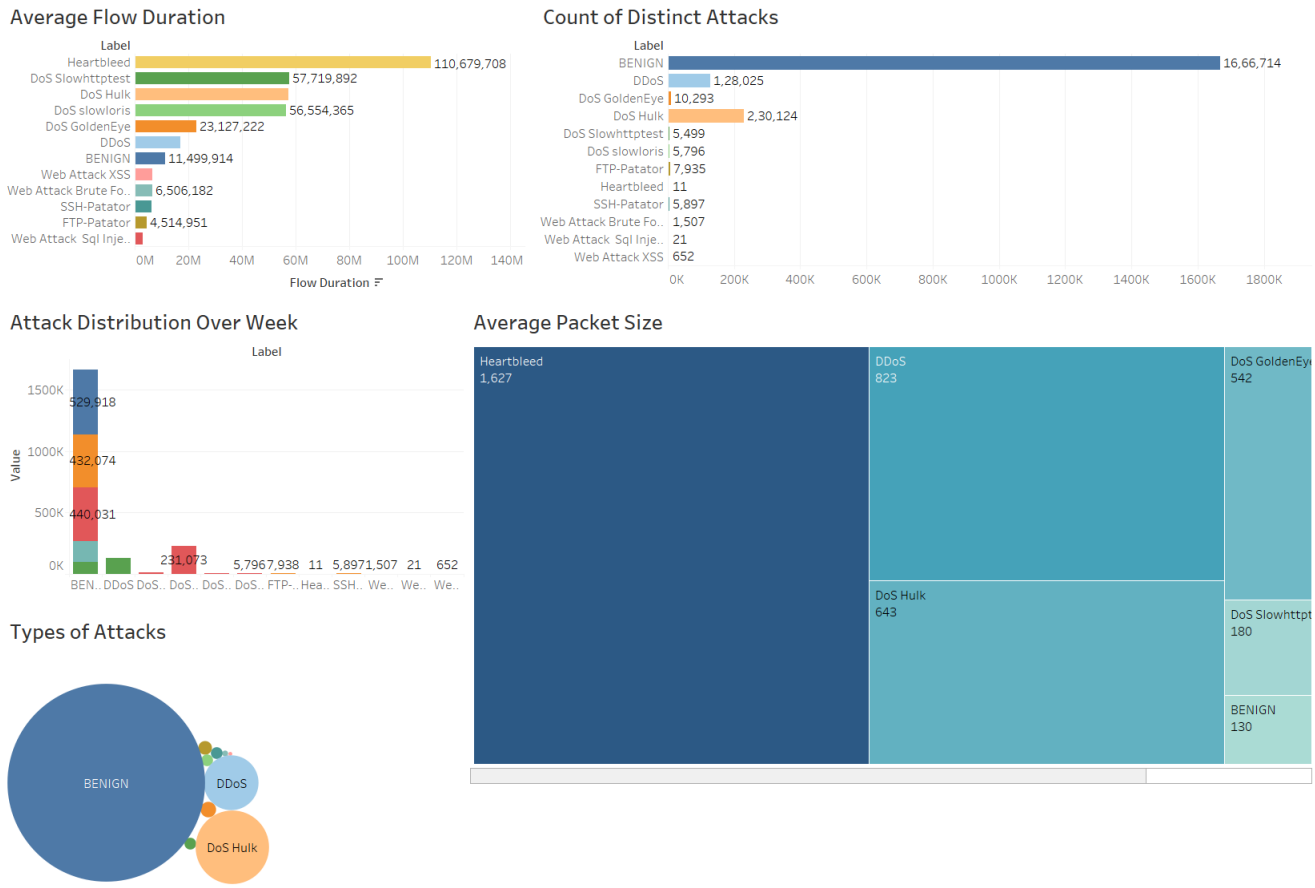

Figure 11: Tableau Dashboard
Figure 11

[9]

6.2 Evaluation of Model's Output

6.2.1 AdaBoost Model

The model's accuracy and other metrics related to this model can be seen in Figure 15. According to the image, AdaBoost has achieved 85.96% accuracy, F1-score is 86.15%, and metrics like precision and recall are 89.08% and 85.95%, respectively. (Figure 12) is a direct comparison of the actual value to the values predicted by the trained model. Out of the 11 outputs, the Adaboost algorithm has made two wrong predictions.

6.2.2 XGBoost Model

The accuracy achieved by this model is 91.87%, where F1-score, precision, and recall are 92.61%, 93.60%, and 91.87%, respectively. Refer to Figure 15 to see the graphical representation of the same. Figure 12 Out of 25 outputs XGBoost classifier has made only two wrong predictions.

[9]Tableau Dashboard `https://public.tableau.com/app/profile/abhay.singh.bangari/viz/BusinessDashboard_16710555971690/Dashboard1?publish=yes`

	Actual	Predictions
0	BENIGN	BENIGN
1	BENIGN	BENIGN
2	DoS Hulk	BENIGN
3	BENIGN	BENIGN
4	BENIGN	DoS GoldenEye
5	DoS Slowhttptest	BENIGN
6	BENIGN	BENIGN
7	BENIGN	BENIGN
8	BENIGN	BENIGN
9	BENIGN	BENIGN
10	BENIGN	BENIGN
11	BENIGN	BENIGN

(a) Adaboost comparision

	Actual	Predictions
0	BENIGN	BENIGN
1	BENIGN	BENIGN
2	DoS Hulk	BENIGN
3	BENIGN	BENIGN
4	BENIGN	BENIGN
5	DoS Slowhttptest	DoS Slowhttptest
6	BENIGN	BENIGN
7	BENIGN	BENIGN
8	BENIGN	BENIGN
9	BENIGN	BENIGN
10	BENIGN	BENIGN
11	BENIGN	BENIGN
12	BENIGN	BENIGN
13	BENIGN	BENIGN
14	BENIGN	BENIGN
15	BENIGN	BENIGN
16	BENIGN	BENIGN
17	BENIGN	BENIGN
18	BENIGN	BENIGN
19	DoS Hulk	DoS Hulk
20	DoS Hulk	BENIGN
21	BENIGN	BENIGN
22	BENIGN	BENIGN
23	BENIGN	BENIGN
24	DoS Hulk	DoS Hulk
25	DDoS	DDoS

(b) XGB comparision

Figure 12: Actual Vs. Predicted Value Comparision Table for AdaBoost, and XGB Figure 12

6.2.3 K-Nearest Neighbor Model

As per the graphical representation of the classification report Figure 15accuracy achieved is 97.92%. The outputs of other metrics are also very similar to the accuracy. F1-score(97.89%), precision(97.87%), recall(97.92%). Figure 13This image shows a prediction output straight from after the code was run. Here, the KNN model predicts only one value wrong out of the 25 mentioned cases in

6.2.4 Random Forest

According to the picture, Random Forest has an accuracy of 98.38%, an F1-score of 98.37.15%, and measures like precision and recall of 98.44% and 98.37%, respectively. Figure13 represents that the algorithm predicted all the values correctly when the top 25 values were taken.

	Actual	Predictions
0	BENIGN	BENIGN
1	BENIGN	BENIGN
2	DoS Hulk	BENIGN
3	BENIGN	BENIGN
4	BENIGN	BENIGN
5	DoS Slowhttptest	DoS Slowhttptest
6	BENIGN	BENIGN
7	BENIGN	BENIGN
8	BENIGN	BENIGN
9	BENIGN	BENIGN
10	BENIGN	BENIGN
11	BENIGN	BENIGN
12	BENIGN	BENIGN
13	BENIGN	BENIGN
14	BENIGN	BENIGN
15	BENIGN	BENIGN
16	BENIGN	BENIGN
17	BENIGN	BENIGN
18	BENIGN	BENIGN
19	DoS Hulk	DoS Hulk
20	DoS Hulk	BENIGN
21	BENIGN	BENIGN
22	BENIGN	BENIGN
23	BENIGN	BENIGN
24	DoS Hulk	DoS Hulk
25	DDoS	DDoS

(a) KNN comparision

	Actual	Predictions
0	BENIGN	BENIGN
1	BENIGN	BENIGN
2	DoS Hulk	DoS Hulk
3	BENIGN	BENIGN
4	BENIGN	BENIGN
5	DoS Slowhttptest	DoS Slowhttptest
6	BENIGN	BENIGN
7	BENIGN	BENIGN
8	BENIGN	BENIGN
9	BENIGN	BENIGN
10	BENIGN	BENIGN
11	BENIGN	BENIGN
12	BENIGN	BENIGN
13	BENIGN	BENIGN
14	BENIGN	BENIGN
15	BENIGN	BENIGN
16	BENIGN	BENIGN
17	BENIGN	BENIGN
18	BENIGN	BENIGN
19	DoS Hulk	DoS Hulk
20	DoS Hulk	DoS Hulk
21	BENIGN	BENIGN
22	BENIGN	BENIGN
23	BENIGN	BENIGN
24	DoS Hulk	DoS Hulk
25	DDoS	DDoS

(b) Random Forest comparision

Figure 13: Actual Vs. Predicted Value Comparision Table for AdaBoost and XGB

6.2.5 Decision Tree Model

This model achieves an accuracy of 98.33 percent, while its F1-score, precision, and recall are respectively 98.33 percent, 98.39 percent, and 98.33 percent. Decision Tree has performed as well as random forest by predicting all top 25 cases correctly.14b

6.2.6 Linear Discriminant Analysis

This model achieves an accuracy of 89.29 percent, while its F1-score, precision, and recall are respectively 88.87 percent, 89.20 percent, and 89.28 percent. This Model performed well but not compared to the random forest and decision tree, predicting 7 cases wrong out of 25.14b

	Actual	Predictions
0	BENIGN	BENIGN
1	BENIGN	BENIGN
2	DoS Hulk	DoS Hulk
3	BENIGN	BENIGN
4	BENIGN	BENIGN
5	DoS Slowhttptest	DoS Slowhttptest
6	BENIGN	BENIGN
7	BENIGN	BENIGN
8	BENIGN	BENIGN
9	BENIGN	BENIGN
10	BENIGN	BENIGN
11	BENIGN	BENIGN
12	BENIGN	BENIGN
13	BENIGN	BENIGN
14	BENIGN	BENIGN
15	BENIGN	BENIGN
16	BENIGN	BENIGN
17	BENIGN	BENIGN
18	BENIGN	BENIGN
19	DoS Hulk	DoS Hulk
20	DoS Hulk	DoS Hulk
21	BENIGN	BENIGN
22	BENIGN	BENIGN
23	BENIGN	BENIGN
24	DoS Hulk	DoS Hulk
25	DDoS	DDoS

(a) Decision Tree comparision Figure 14b

	Actual	Predictions
0	BENIGN	BENIGN
1	BENIGN	BENIGN
2	DoS Hulk	DoS Hulk
3	BENIGN	BENIGN
4	BENIGN	BENIGN
5	DoS Slowhttptest	DoS Slowhttptest
6	BENIGN	BENIGN
7	BENIGN	BENIGN
8	BENIGN	BENIGN
9	BENIGN	BENIGN
10	BENIGN	BENIGN
11	BENIGN	BENIGN
12	BENIGN	BENIGN
13	BENIGN	BENIGN
14	BENIGN	BENIGN
15	BENIGN	BENIGN
16	BENIGN	BENIGN
17	BENIGN	BENIGN
18	BENIGN	BENIGN
19	DoS Hulk	DoS Hulk
20	DoS Hulk	DoS Hulk
21	BENIGN	BENIGN
22	BENIGN	BENIGN
23	BENIGN	BENIGN
24	DoS Hulk	DoS Hulk
25	DDoS	DDoS

(b) Decision Tree comparision Figure 14b

Figure 14: Actual Vs. Predicted Value Comparision Table for Decision Tree, and LDA
The figure below is a detailed representation of all the metrics that helps in evaluating the performance of all the machine learning models. It shows the accuracy, F1-score, precision, and recall in one graph and is sorted in increasing order. Figure15

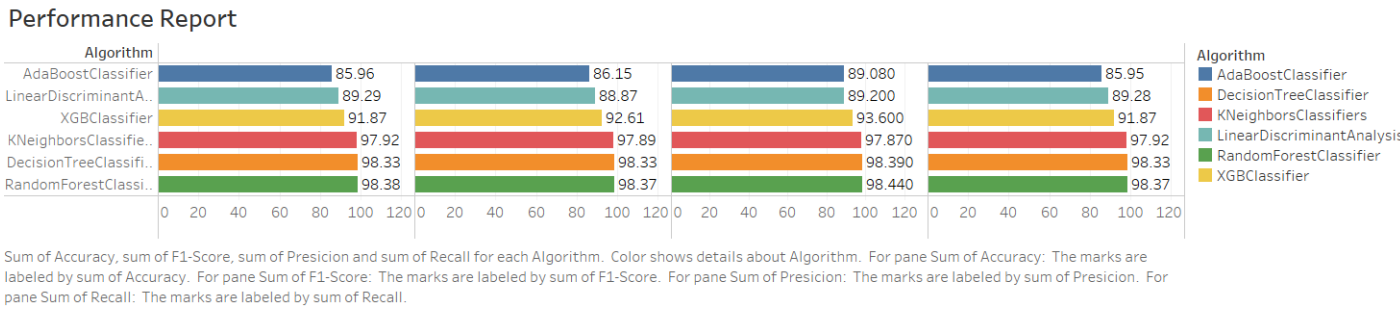

Figure 15: Report

6.3 Random Forest with 15, 20, 30, and 35 Features Metrics

Accuracy: 97.24%
F1-Score: 97.09%

	precision	recall	f1-score	support
0	0.97	1.00	0.98	416269
1	0.99	0.99	0.99	32129
2	0.99	0.98	0.98	2630
3	0.99	0.79	0.88	57757
4	0.96	0.98	0.97	1360
5	1.00	0.98	0.99	1485
6	1.00	0.98	0.99	1943
7	1.00	0.67	0.80	3
8	0.99	0.50	0.66	1475
9	0.00	0.00	0.00	7
10	0.73	0.63	0.67	401
11	0.43	0.33	0.37	160
accuracy			0.97	515619
macro avg	0.84	0.74	0.77	515619
weighted avg	0.97	0.97	0.97	515619

Figure 16: 15 Features Classification Report

Accuracy: 97.39%
F1-Score: 97.26%

	precision	recall	f1-score	support
0	0.97	1.00	0.98	416269
1	0.99	0.99	0.99	32129
2	0.99	0.98	0.98	2630
3	0.98	0.81	0.88	57757
4	0.96	0.98	0.97	1360
5	1.00	0.98	0.99	1485
6	1.00	0.98	0.99	1943
7	1.00	1.00	1.00	3
8	0.99	0.50	0.67	1475
9	0.00	0.00	0.00	7
10	0.72	0.72	0.72	401
11	0.39	0.25	0.31	160
accuracy			0.97	515619
macro avg	0.83	0.77	0.79	515619
weighted avg	0.97	0.97	0.97	515619

Figure 17: 20 Features Classification Report

```
print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	416269
1	1.00	1.00	1.00	32129
2	0.99	0.99	0.99	2630
3	0.91	0.98	0.94	57757
4	0.96	0.98	0.97	1360
5	1.00	0.99	0.99	1485
6	1.00	0.98	0.99	1943
7	1.00	1.00	1.00	3
8	1.00	0.50	0.67	1475
9	0.00	0.00	0.00	7
10	0.71	0.73	0.72	401
11	0.39	0.22	0.28	160
accuracy			0.98	515619
macro avg	0.83	0.78	0.80	515619
weighted avg	0.98	0.98	0.98	515619

Figure 18: 30 Features Classification Report

Accuracy: 99.90%
F1-Score: 99.89%

	precision	recall	f1-score	support
0	1.00	1.00	1.00	416269
1	1.00	1.00	1.00	32129
2	1.00	1.00	1.00	2630
3	1.00	1.00	1.00	57757
4	0.99	0.99	0.99	1360
5	1.00	0.99	0.99	1485
6	1.00	1.00	1.00	1943
7	1.00	1.00	1.00	3
8	1.00	0.99	1.00	1475
9	0.00	0.00	0.00	7
10	0.73	0.91	0.81	401
11	0.45	0.14	0.22	160
accuracy			1.00	515619
macro avg	0.85	0.84	0.83	515619
weighted avg	1.00	1.00	1.00	515619

Figure 19: 35 Features Classification Report

6.3.1 Limitations of the Experiment

The very first challenge was to find the dataset. Many attempts were made to fetch the data from an enterprise, but they failed. That is why generic data is used from the Canadian Institute for Cyber-Security; the dataset is available for research. There are other generic datasets available like NSL-KDD, CSE-CIC-IDS2018 (this one is similar to the CICIDS2017 dataset; it is the one used in this study, but the size of the file is 400 GB, which is too big), and ISCX-SlowDoS-2016. The majority of the datasets are outdated and do not have sufficient details to carry out good research. Before Tableau, Google Studio and Power BI were the primary choices, but due to the large dataset, these could not handle it. In the CICIDS2017 dataset, there are 8 CSV files, but only five were taken because the data files are hefty, which makes analysis difficult. Models like XBGBoost, Decision Tree, Naive Bayes, and SVM might have good accuracy, but their execution time is extremely high, which makes the analysis extremely slow. On the other hand, models like AdaBoost and Linear Discriminant Analysis do not have better accuracy than those discussed above, but their execution time is less.

7 Discussion

As all the virtual experiments have been carried out, let's interpret those findings deeply. All the study work discussed in the related work section discusses different algorithms and feature selection to detect network intrusion. Still, it rarely has any author-incorporated business intelligence as a part of its research. However, in this study, essential insights are extracted from the data, which will help an organization protect its network from intrusion. Conveying and explaining the details in graphs is time-saving, and decisions can be made quickly and easily understand the whole scenario. So business intelligence adds high value to this research study. Figure7 & Figure 8 display the business aspect through the analysis of the flow of duration and the packet size related to the present attacks in the data. The business aspect may also be examined by examining the flow of bytes in packets and the duration of the flow of packets. According to the graphs, the package length and size, as well as the time of the flow, were at their peaks during the Heartbleed attack. During the Heartbleed attack, it can be seen that its flow duration is the highest, which is 110,679,708. **HeartBleed having the largest average flow duration going, requires more time to deliver a packet from one end to another. The packet**

size is also big as per Figure 8; hence it will not be preferred to attack a network. Attacks like Dos Slowhttptest and Dos Hulk have large average flow duration, and attackers might not prefer longer routes to intrude a network.
Figure 8The highest score for average packet size achieved by a Heartbleed attack is 1627, and the least is scored by FTP-Predator, which is 12. **A greater amount of packets can be sent if the packet size is small compared to the large attacking packets. Dos slow loris, FTP predators, SSH-pastor, etc. are small packet attacks; therefore, they are preferred over large packet attacks such as HeartBleed, DDos, Dos Hulk, etc., but other factors matter too before deciding which attack is more suitable.**
Figure 9The color in the chart denotes the Example 15 type of attack. Twelve attacks occurred throughout the week, from Monday to Friday. Benign, DoSHulk, DDoS, DoS GoldenEye, FTP-Patator, SSH-Patator, DoS SlowLoris, DoS Slowhttptest, Web Attack Brute Force, Web Attack XSS, Web Attack, SQL Injection, and Heartbleed When it comes to threats on networks, it is necessary first to analyze the data to determine which types of attacks are more common and how often they occur to establish a pattern. Based on the information shown in Figure 9. BENIGN is the kind of attack that occurs the most often in the data, i.e., 16,66,714, followed by DoS Hulk(2,30,124) and DDoS(1,28.025), and least is WebAttack XSS, i.e., 622. The other types of attacks are very uncommon. **BENIGN indicates that there is no attack in the network, which is a sign that the traffic on the network is entirely clean and safe. Even though DosSlowloris has a small packet size, the flow duration is large, so attackers would not choose this to infiltrate the network.**
Figure 10 shows the frequency of attacks from Monday to Friday. The benign attack is by far the most predominant. It is observed that the maximum number of attacks occurred on Wednesday. Benign again being the highest 440,031, Dos Hulk(231,073), DDos (128,027), Dos Golden(10,293), DoS Slowloris(5,796), Dos Slowhttptest(5,499), and lowest is HeartBleed(11). On the other hand, the least number of attacks that took place on Thursday was Benign(168,186), the highest of all, and WebAttack SQL Injection was the lowest (21). **Special care needs to be given on Wednesdays because attackers are more active on the same day. The reason is not available, but one reason could be that on Thursdays, the organization updates its firewalls and deals with previous bugs, so the best time for hackers could be Wednesday. But this is just an interpretation.**

8 Conclusion and Future work

After carrying out the experiments, it is found that random forest is the best-performing algorithm with an accuracy of 98.38% and an F1-score of 98.37%. In contrast, the AdaBoost Classifier algorithm scores the least accuracy, i.e., 85.96%. In this study, the XGBoost Classifier model(91.87%) is compared to 5 other machine models, and three models perform better. Random Forest, Decision Tree(98.33%), and KNN(97.92%) are the three better-performing models. Models like XBGBoost, Decision Tree, Naive Bayes, and SVM might have good accuracy, but their execution time is extremely high, which makes the analysis extremely slow. On the other hand, models like AdaBoost and Linear Discriminant Analysis do not have better accuracy than those discussed above, but their execution time is less. It is also found that the model's performance reduces as we choose

fewer features than we decided at the beginning, i.e., 30 parts, and improves as we select more features. Random Forest achieved an accuracy of 99.90% with 35 top-selected features using the feature selection technique. In all cases, the accuracy and F1-score gap are not much, which signifies that the data is free from balancing classes and over-fitting problems. Tableau is utilized for visualization. Detailed and interactive charts and bars are generated. In the end, the complete dashboard is also made with the help of the tableau. All the results indicated that the benign class is more prevalent in all the data files, which means that organizations will encounter fewer cyber-attacks. Looking at the bar chart that shows the distribution of attacks per week, Monday is one of the days that experience the most attacks compared to the rest of the week. All the related works mentioned in the research are considered to achieve all the essential objectives.

No deep learning techniques were implemented in this research, so in the future, the same analysis can be carried out using deep learning models. In this research, the ensemble learning techniques using decision trees outperformed the other models. In the future, it would be interesting to boost the other models with better feature selection techniques like information gain, adapting time series approaches by adding a lag feature, moving average, and so on. This problem has potential cyclicity and seasonality, which needs exploring. If that is looked into, specific unexplained patterns, the splitting of the problem into cases, trends, etc., can be predicted better. The data is extensive, so sharing it is a big challenge. Therefore, the next step could be to upload the data to the cloud to be fetched from anywhere for analysis. Also, Tableau can directly bring data from the cloud, saving time.

References

Belavagi, M. C. and Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection, *Procedia Computer Science* **89**: 117–123.

Buczak, A. L. and Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection, *IEEE Communications surveys & tutorials* **18**(2): 1153–1176.

Chowdhury, S., Khanzadeh, M., Akula, R., Zhang, F., Zhang, S., Medal, H., Marufuzzaman, M. and Bian, L. (2017). Botnet detection using graph-based feature clustering, *Journal of Big Data* **4**(1): 1–23.

Coelho, I. M., Coelho, V. N., Luz, E. J. d. S., Ochi, L. S., Guimarães, F. G. and Rios, E. (2017). A gpu deep learning metaheuristic based model for time series forecasting, *Applied Energy* **201**: 412–418.

Ghosh, U., Chatterjee, P., Tosh, D., Shetty, S., Xiong, K. and Kamhoua, C. (2017). An sdn based framework for guaranteeing security and performance in information-centric cloud networks, *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, IEEE, pp. 749–752.

Gustavsson, V. (2019). Machine learning for a network-based intrusion detection system: an application using zeek and the cicids2017 dataset.

Janagam, A. and Hossen, S. (2018). Analysis of network intrusion detection system with machine learning algorithms (deep reinforcement learning algorithm).

Kim, S. H., Jang, S. Y. and Yang, K. H. (2017). Analysis of the determinants of software-as-a-service adoption in small businesses: Risks, benefits, and organizational and environmental factors, *Journal of Small Business Management* **55**(2): 303–325.

L'heureux, A., Grolinger, K., Elyamany, H. F. and Capretz, M. A. (2017). Machine learning with big data: Challenges and approaches, *Ieee Access* **5**: 7776–7797.

Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A. and Payne, B. (2015). Evaluating computer intrusion detection systems: A survey of common practices, *ACM Computing Surveys* **48**: 12:1–.

Modi, C. N. and Acha, K. (2017). Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: A comprehensive review, *J. Supercomput.* **73**(3): 1192–1234.

Narang, P., Ray, S., Hota, C. and Venkatakrishnan, V. (2014). Peershark: detecting peer-to-peer botnets by tracking conversations, *2014 IEEE Security and Privacy Workshops*, IEEE, pp. 108–115.

Nguyen, H. T. and Franke, K. (2012). Adaptive intrusion detection system via online machine learning, *2012 12th international conference on hybrid intelligent systems (HIS)*, IEEE, pp. 271–277.

Pervez, M. S. and Farid, D. M. (2014). Feature selection and intrusion classification in nsl-kdd cup 99 dataset employing svms, *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*, IEEE, pp. 1–6.

Popoola, E. and Adewumi, A. O. (2017). Efficient feature selection technique for network intrusion detection system using discrete differential evolution and decision., *Int. J. Netw. Secur.* **19**(5): 660–669.

Rodda, S. and Erothi, U. S. R. (2016). Class imbalance problem in the network intrusion detection systems, *2016 international conference on electrical, electronics, and optimization techniques (ICEEOT)*, Ieee, pp. 2685–2688.

Sharafaldin, I., Lashkari, A. H. and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization., *ICISSp* **1**: 108–116.

Sharma, R. K., Kalita, H. K. and Borah, P. (2016). Analysis of machine learning techniques based intrusion detection systems, *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, Springer, pp. 485–493.

Soni, S. and Bhushan, B. (2019). Use of machine learning algorithms for designing efficient cyber security solutions, *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, Vol. 1, IEEE, pp. 1496–1501.

Viegas, E., Santin, A. O., Franca, A., Jasinski, R., Pedroni, V. A. and Oliveira, L. S. (2016). Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems, *IEEE Transactions on Computers* **66**(1): 163–177.

Villamarín, J. M. and Diaz Pinzon, B. (2017). Key success factors to business intelligence solution implementation, *Journal of Intelligence Studies in Business* **7**(1): 48–69.

Zimba, A., Wang, Z. and Chen, H. (2018). Multi-stage crypto ransomware attacks: A new emerging cyber threat to critical infrastructure and industrial control systems, *Ict Express* **4**(1): 14–18.

Žliobaitė, I., Bifet, A., Read, J., Pfahringer, B. and Holmes, G. (2015). Evaluation methods and decision theory for classification of streaming data with temporal dependence, *Machine Learning* **98**(3): 455–482.