JCBI
Journal of Computing & Biomedical Informatics Volume 03 Issue 02
ISSN: 2710 - 1606 2022

Research Article
https://doi.org/10.56979/302/2022/52

Cyber Threats Prediction Model using Advanced Data Science Approaches

Muhammad Arslan Ajmal[1], Muhammad Imran[1*,] Muhammad Asif Raza[1], and Ali Raza[2]

[1]Department of Computer Science, BZU, Multan, Pakistan.
[2]Department of Computer Science, Institute of Southern Punjab, Multan, Pakistan.
[*]Corresponding Author: Muhammad Imran. Email: m.imran@bzu.edu.pk

Received: April 19, 2022 **Accepted:** September 01, 2022 **Published:** September 27, 2022.

Abstract: In the era of technology and the widespread use of the internet, internet users' data and personal information are more at risk. Among various cyber-attacks, DDOS is one of the most dangerous cyber-attacks, which uses single or multiple victims for the unavailability of resources on a small and large scale. The amount and intensity of cyber-attacks are also increasing gradually with increasing internet usage. So, defensive strategies are also built with time to protect a network and network devices from many breaches and attacks attempted by many cyber terrorists. Instead of traditional defense mechanisms, data science makes it impressive and easy to predict and detect cyber-attacks. This study proposed a data science-based prediction model using a substantial dataset CICDDOS2019. In this research, different models of Machine Learning, e.g., Decision Tree, Random Forest, SVM, and Naïve Bayes, are applied after making this dataset clean and considering the best relevant features for getting maximum accuracy to detect and predict the cyber threats.

Keywords: Machine Learning; Network Security; Cyber Threads; Data Science; Prediction Model.

1. Introduction

As computer-based systems connected to the internet are progressively widespread, the organization "International Telecommunication Union" (ITU) accounted for many internet users worldwide who constantly use Internet services. For example, e-commerce, e-banking, entertainment, and education have been extended by billions. The risk of data breaches has grown as the number of virtual networks and internet-connected devices have risen. Cyber terrorists are very active in stealing and misusing active internet-based device users' data, information, and credentials. Massive quantities of confidential user information are vulnerable to various assaults, both internal and external. Cyberattacks have grown more sophisticated as algorithms have become more sophisticated due to technical advancements [1]. Because the number of cyber-attacks is increasing faster than the number of effective defenses against them, organizations must increase their investment in cyber security. Some recent techniques for defensive cyber-attacks involve machine learning approaches, policy-driven approaches, and dynamic, rule-based approaches [2]. The term "cybersecurity" is used to protect data and information over the internet.

Cybersecurity is a collection of technological innovations designed to prevent cyberattacks, damage, and unauthorized access [3] to computers, networks, applications, and data. Cyber security has seen substantial developments in recent years, both in technology and how it operates in the computing environment. This transformation is being facilitated by data science (DS). The usage of "Artificial Intelligence" (AI) components such as machine learning (ML) may greatly assist in the finding of new knowledge from large datasets. Data science is essential in a new scientific prototype [4] [5], which uses machine learning to revolutionize cybersecurity. Traditional firewall systems are not enough to ensure the protection of the open ports of any internet-based communication system. Usually, an IDS (Intrusion Detection System) is

prescribed to classify and detect various cyber-attacks. Based on data analysis, two categories, Network-Based and Host-Based IDS, are discussed later. Applying these advancements in cybersecurity detection and predicting cyber-attacks has become efficient, especially for new and zero-day attacks. So, this research coherently focused on DDOS-type attacks and applies data science approaches like data analysis, normalization, data standardization, data exploration, feature extraction, data cleaning, and training and testing of ML models using the latest DDOS-based "CICDDOS2019" dataset.

1.1. Motivation

Various cyber intrusions frequently threaten sensitive information stand. With the advancement of technology, cyber terrorists are becoming more active, employing various inventive methods to breach network security and misuse secret and sensitive information [62]. Security analysts bring up different solutions to overcome modern cyber-attack approaches, but these approaches are not enough due to the exponential growth of data and cyber threats. Extra resources and the latest strategies need to be used and implemented to escape from cyber-attacks. Cyber-security researchers are going around to preempt and catch up with the latest threats.

According to the research (2016), these three industries, government, retail, and technology, are targeted in nearly 95 percent of cyber breaches. These managements are aimed at two main reasons. The first is that these companies store a massive amount of personal credentials. According to a study organized at the University of Maryland [6], a cyber intervention appears every 39 seconds. According to the data released by Juniper Research [7], by 2024, the average value of a data crack will exceed 5 trillion dollars, and malware affected the healthcare department by 75% over the last seven years. "According to the Q2 2018 Threat Report, Nexus guard's quarterly article, the average number of DDoS attacks raised more than 26Gbps, increasing by 500%". By 2021, it is also assumed that 6 trillion dollars will be spent worldwide on cybersecurity [8].

A good intention has been dedicated to exploring IDS systems that depend on a machine learning approach for the evolution of a dataset. Classifying events (malicious or benign) is the primary prerequisite for this analysis, based on the labeled datasets [9]. The disproportion of datasets is a significant concern throughout the dataset creation stage. It is the data utilized to train the machine learning model that is causing the bias. Even though machine learning research is extensive, only a tiny proportion of publications examine the specifics of the data utilized in their investigations. For researchers, developing genuine and sophisticated models takes precedence over observing trends in datasets. However, it is still true: almost all large datasets created using machine learning algorithms are biased [10]. Researchers are beginning to recognize the need for proportional datasets in machine learning to reduce bias. When malicious samples are more minor than benign samples, there is a greater diversity of classes—overloaded data results in low encountering rates of lesser classes where datasets consist of more occurrences associated with the ethical conduct of a class than the attack class. Data destruction or overfitting are two examples of how this lowers achievement. All courses will be identified appropriately when trainees choose to be inclined toward the bulk class [11] [12]. Hence, the motivation for this work is to develop approaches in which the effect of bias in datasets may be reduced when discussing efficiency.

1.2. Problem Statement

Historically, cybersecurity solutions have been static and signature-based, relying on pattern recognition to identify a match between a previously recorded assault or malware and a new threat [13]. As a result, it must be updated regularly to include new signatures in the product database. As a result, zero-day assaults are impossible to detect or avoid. Additionally, conventional methods are very binary and provide few benefits over predictive models that may forecast the likelihood of assaults or hazardous behaviors dependent on data analysis methods. Access to a large amount of data also makes it possible to resolve challenging and complicated security problems. According to big data and data mining, the more data gathered, the more precise and accurate the analysis [13]. In its broadest sense, data science is the practice of applying a scientific approach to extract information from data and find new knowledge. By leveraging new technical advantages in storage, computation, and behavioral analytics, data science may

help develop novel cybersecurity solutions [14]. Consequently, Data Science is critical to cybersecurity because it relies on data and high-performance computing to combat cybercrime and safeguard consumers. A good data science project requires an efficient method that tackles all issues and budgets adequately for resources [15].

2. Related work

Researchers have developed novel methods for detecting assaults in recent years. They addressed their targeted study gaps by using various data sources, detection methods, and procedures. This section will discuss the latest developments in this area. As attack detection techniques have developed in response to the sophistication of contemporary attack creation.

2.1. Feature Extraction based approaches

Haider et al. [82] created a method for detecting anomalies in hosts by using the ADFA-LD [83] dataset. This collection contains low-footprint assaults that combine and complicate the separation of normal and aberrant host data. This subset of ADFA-LD is representative of the contemporary cyber threat environment. The study team noticed that one of the shortcomings of current methods for identifying abnormalities in the ADFALD dataset is their inability to extract characteristics corresponding to system calls. The researchers suggested a worldwide statistical feature collection method for integer data on character data zero watermarks to collect trustworthy and concealed characteristics from system calls [84]. Three machine learning methods were employed to train and evaluate the host-based ADFALD dataset: KNN, SVM with linear kernel, and RBF kernel. Because the data from low-footprint assaults cannot be separated from the usual data, an SVM with RBF kernels was utilized to achieve effective non-linear separation. The kernel's primary objective was to convert a data point into a new feature space from which normal and low-footprint attack data could be separated using SVM. The researchers utilized the nonparametric and computationally efficient KNN model for training a profile of typical system call behavior and then assessed it by calculating the divergence of a test system call from the usual profile. One of the model's primary disadvantages is that it gets considerably slower when the amount of data proliferates. Numerous tests were conducted with various k and Euclidean distance values to determine the degree of resemblance between two data points.

Yuxin et al. [85] suggested a semantic analysis-based method for behavior-based detection. The researchers provided an executable in assembly code and used it to create a control flow graph. The system call execution route is extracted and combined from a control flow graph to create a system call stream from an executable. The authors classified the system as call sequences using the decision tree method. Decision trees are used inductively to learn and are an approximation technique for discrete-valued functions. A decision tree's core nodes contain attributes, while the leaf nodes provide class labels; the route from root to leaf determines the classification rule. It constructs such a tree by determining the optimal set attribute for each instance and classifying the training data. Our suggested method achieved a better accuracy with a reduced false positive rate compared to dynamic detection.

It was shown by Aghaei et al. [86] that a semi-supervised one-class learning algorithm could be used with a feature extraction technique based on Principal Component Analysis to create a host-based intrusion detection system (PCA). To estimate the target class density, the researchers combined the target class probability function with an artificial class density function, which they characterized as a one-class classification. The PCA [87] based Eigen traces method was used to extract the representative features. Radial Basis Function neural networks and Random Forest were used to train and assess the proposed method. The ADFA LD benchmark dataset [88] simulated and assessed the experiments. When it comes to identifying bogus system calls, they have demonstrated outstanding accuracy, precision, and recall.

Xie et al. [89] developed a method for extracting frequency-based features from system call traces to identify the abnormality. They addressed the research issue of the longer training time associated with short sequence-based intrusion detection systems. The researchers evaluated their suggested framework using the ADFA-LD12 dataset. The ADFA-LD dataset was created on a Linux computer running kernel 2.6.38, supporting 325 distinct system calls. Each system call in a trace is assigned a call number between 1 and 325. The authors devised a method for determining the frequency of an individual system call inside

a trace. The frequency-based method transforms system call traces of varying lengths into frequency vectors of equal size. Due to the sparse nature of frequency vectors, PCA was employed to decrease the dimension, reducing the model's calculation cost. The researchers detected the abnormality using two frequency-based machine learning algorithms, k-Nearest Neighbor (KNN) [89, 90, 91] and k-Means Clustering (KMC) [92]. Xie and colleagues evaluated the effectiveness of several types of learning models for detecting attacks. In terms of accuracy and computing time, the KMC method beat the KNN algorithm.

Xie et al. expanded the study in [93] by applying a single class SVM on the ADFA-LD dataset using brief sequences of system calls. The repeated short sequences were omitted in this method to distinguish the anomalies from the usual profile easily. By keeping the computing cost low, the researchers achieved an acceptable level of performance. Additionally, they recommended enhancing their current approach by giving more weight to brief sequences that increase the separability of normal and aberrant activity.

2.2. DDOS Attack based Approaches

Radial Basis Function (RBF) neural networks are a new kind of neural network that the developers of [95] use to identify DDoS assaults based on packet characteristics. The method may be used for routers at the very edge of a victim network. They used seven feature vectors to activate an RBF neural network at each successive time step. The RBF neural network classifies input into two groups: standard and attack types. The Filtering and Attack Alarm Modules get the attack packets' source IP addresses if the incoming traffic is recognized as attack traffic. On the other hand, if the traffic is considered normal, it will be forwarded to its destination. [96] describes a data-mining approach for identifying DDoS assaults. The authors used an FCM cluster technique and an a priori association strategy to extract network traffic and network packet protocol status models and construct the detection model's threshold. The authors used decision trees and grey relational analysis [97] to identify DDoS assaults. Fifteen criteria may be used to classify an attack, including monitoring the incoming/outgoing packet/bytes rate and compiling the TCP SYN and ACK flag rates to describe traffic flows. These features were tested to see whether the normal traffic flow could be detected using the decision tree technique.

Different DDoS attack techniques have been proposed in the literature during the last ten years. The protocol level at which a DDoS flooding attack occurs has been studied in more depth, and it has been split into two kinds [98].

TCP, UDP, ICMP, DNS, and ICMP protocol packets are often used to launch DDoS flooding assaults on a network or transport.

DDoS floods target the application level by depleting resources such as sockets, RAM, CPU, and bandwidth to interrupt legal user services. It is more challenging to detect application-level intrusions than volumetric ones since they seem regular traffic. Early mitigation of assaults near their source is a significant issue in fighting DDoS attacks; nevertheless, a complete solution that meets these requirements has yet to be implemented [63, 98, 99].

Using a Hypertext Transfer Protocol (HTTP) technique, data sampling-based flood attack detection on web servers was created [100] using a Hypertext Transfer Protocol (HTTP) technique. The number of requests coming from the application layer and the total number of packets with no payload was used to assess if the traffic under investigation was regular or a victim of a DDoS assault. This means that a 20% sample rate had a detection rate of between 80% and 88%, according to the research findings. However, despite significant advancements, the suggested method is not suitable for use in automated detection systems yet.

When it comes to DDoS assaults and Flash Occurrences, D-FACE is a robust collaborative system [101] that utilizes metrics from GE and GID (FEs). When thousands of legitimate people try to simultaneously access a single computing resource, such as a website, a FE is like a volumetric DDoS attack. According to the findings, D-FACE can detect DDoS assaults as well as FEs. While the research makes significant advancements in the area, the validation relied on dated datasets. The suggested type of collaboration further limits the industrial application of the solution since it needs a high degree of ISP involvement [75].

SkyShield uses the divergence between two Sketches to identify abnormalities produced by attackers during the detection phase. The mitigation phase protects users via filtering, whitelisting, blacklisting, and CAPTCHA. The system was assessed based on custom datasets. Because Of the concentration of SkyShield

on the Application Layer, especially the HTTP protocol, that system was susceptible to flooding at both network and transport layers.

Umbrella [102] creates a tiered defensive architecture capable of mitigating a broad range of DDoS assaults. (e authors presented a method centered entirely on victim detection and protection. (e system was assessed in terms of traffic management using the designed testbed. (e authors assert that the system can defend itself against large-scale assaults. This technique, however, is extensively employed in business and is ineffective against massive DDoS assaults.

A semi-supervised machine learning method was recently developed to classify DDoS assaults. The CICIDS2017 dataset was utilized to assess system performance measures in this method [103]. While the study covers current DoS vectors, the method's online performance has not been tested.

3. Proposed architecture

To identify and forecast DDOS cyber-attacks on network-based systems, the suggested architecture is a complete step-by-step approach that begins with dataset selection and preprocessing and ends with modeling and assessing machine learning-based models. The whole scenario for the proposed architecture is described in this section.

As shown in Fig. 1, the proposed architecture comprises different steps. These are described as follows.

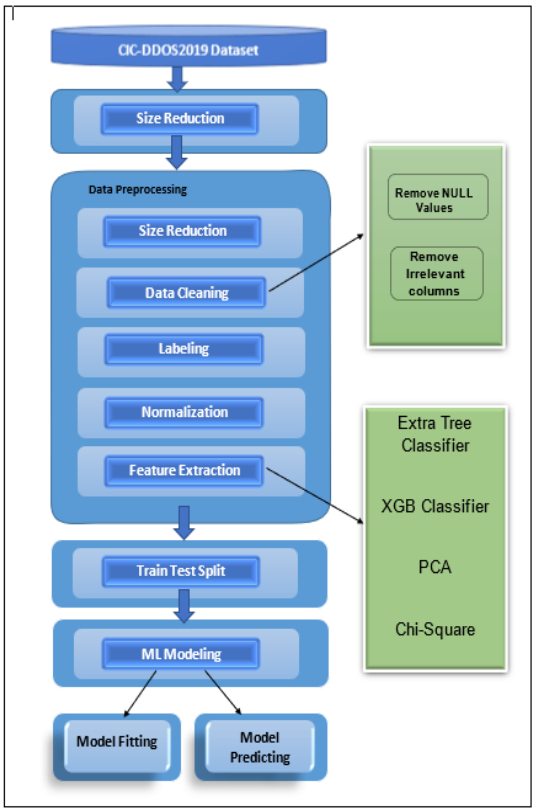

Figure 1. Proposed Architecture

The proposed model is divided into the following steps according to the data-science process.

3.1. Selected Dataset

The dataset used to create this model is CICDDoS2019, which provides benign and up-to-date examples of 12 famous DDoS assaults. The latest dataset contains details of different kinds of DDOS-type attacks. These include DNS, LDAP, NTP, SSDP, SYN MSSQL, SNMP, NetBIOS, UDP, UDP-Lag, WebDDoS, and TFTP. This dataset is in (.csv) format and was generated from raw data gathered on each computer, including network traffic (Pcaps) and event logs (Windows and Ubuntu event logs). Eighty-nine traffic characteristics were extracted from the raw data using the CICFlowMeter-V3 and stored as a CSV file per machine [79].

3.2. Selected Algorithms

The methods for detecting various kinds of DDoS Cyber threats are chosen based on their computational complexity. This criterion is critical for selecting a low-complexity algorithm.

3.2.1. Decision Tree

In general, decision tree analysis is a kind of predictive modeling that may be used in various situations. Decision trees may be built using an algorithmic method that segments the information according to several criteria. The most powerful algorithms in the field of supervised algorithms are decision trees.

They are suitable for classification as well as regression problems. In this research, 80% of data was used to train the Decision Tree and fit this model with a standardized dataset for classification and regression of different DDOS attacks.

3.2.2. Random Forest

A Radom Forest comprises many decision-making trees that collaborate and teach one another during the bagging process. The bagging method is predicated on the premise that combining several models maximizes efficiency, even though Radom Forest may be utilized for classification and regression problems. A two-three decision tree is shown in this picture [68, 104]. The outputs from the two trees are merged using the Random Forest's ensemble method to get the result.

A Random Forest contains almost the same hyperparameters as a Decision Tree, plus those of a bagging classifier that regulates the ensemble of trees. When splitting a node, a random subset of features tries to find the best feature rather than discovering the most significant feature [67, 104].

3.2.3. SVM

Support vector machines (SVMs) are supervised machine learning method that is both powerful and versatile. They are used for both classification and regression. However, they are often employed in categorization issues. SVMs were introduced in the 1960s but were improved in 1990. SVMs are implemented differently than other machine learning algorithms. They have been trendy in recent years because of their capacity to handle many continuous and categorical variables.

An SVM model essentially represents several classes in multidimensional space through a hyperplane. SVM will create the hyperplane iteratively to reduce the error. The purpose of SVM is to classify datasets to identify the greatest marginal hyperplane.

Margin can be positive or negative. Here it is come to know that SVM is to classify datasets to identify the MMH.

3.2.4. Naive Bayes

Naive Bayes is one of the essential supervised learning-based classifications, based on the famous probability theorem "Bayes Theorem." This theorem is used for conditional probability by using the following formula.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{1}$$

$$P(A|B) = \frac{P(A).P(B|A)}{P(B)} \tag{2}$$

1. The fundamental premise of Naive Bayes is that
2. each feature must be either independent or non-correlative.
3. Equivalent contribution to the outcome

So, the Naive Bayes is used for supervised classification with available features, and since we have multiple features in this research, the formula with multiple feature variables will be as follows.

$$P(y|x_1, \ldots., x_n) = \frac{P(x_1|y)P(x_2|y) \ldots \ldots P(x_n|y)P(y)}{P(x_1)P(x_2) \ldots \ldots P(x_n)} \tag{3}$$

4. Implementation

The above-discussed model steps are described here to explain how those steps are done for the proposed model.

4.1. Dataset Exploration

This downloaded [105] dataset consists of ten CSV files containing various types of attacks, a certain proportion of each file is included in this model.

As shown in Table 1, after concatenating all of these dataset files, the massive dataset employed in this model took shape (1678441, 88).

Table 1. Percentage Taken of Dataset Files

Attack	Percentage
NTP	10%
DNS	20%
LDAP	30%
MSSQL	20%
UDP	10%
UDP-LAG	20%
NETBIOS	15%
SSDP	20%
Syn	13%
TFTP	15%

1. The total number of records (rows) is 1678441
2. The number of features (columns) is 88.
3. RangeIndex: 1678441 entries, 0 to 1678440
4. Data columns (total 88 columns):

The used dataset contains the following amount of different attacks and benign traffic as mentioned in table 2.

Table 2. All Types of Attacks

Attack Type	Amount
DrDoS_LDAP	314355
DrDoS_SSDP	209657
DrDoS_MSSQL	209495
DrDoS_DNS	209294
DrDoS_NetBIOS	157170
TFTP	157118
Syn	136262
DrDoS_UDP	104758
DrDoS_NTP	103505
UDP-lag	73306
BENIGN	3435
WebDDoS	86

4.2. Dataset Pre-processing

The initial stage in implementation is to do preprocessing on our datasets. Preprocessing is preparing the data for model training. Data Science's primary focus is to prepare data or clean datasets before applying any model for training and testing. In this work, four-step has been done in this preprocessing phase.

4.2.1. Reduce Size

In this first step, we reduce the size of the above dataset, which is extracted from different types of CIC-DDOS2019 downloaded datasets. 40% of the dataset is taken for different processes due to the limitation of hardware resources.

After this reduction of size, the shape of the dataset is: (671376, 88).

4.2.2. Data Cleaning

In this step, data is been cleaned through the following steps.

4.2.2.1. Remove null values

First of all, changed incorrect values such as "inf" to "nan" values to easily remove null values. The utilized dataset has a significant number (45844) of null values, which contributes to the suggested model's poor performance in predicting DDOS attacks. "dropna()" method is used to remove all the null values.

4.2.2.2. Removal of Irrelevant Columns

In this step, all the columns with incomplete and less information and more than 70% of null values are removed. After removing irrelevant columns and null values, the shape of the used dataset is : (648454, 26).

4.2.3. Label Categorical values

It is not very easy to process textual data by ML models. So, the “LabelEncoder” library of “Scikit Learn. Preprocessing” is used to label categorical values into numerical ones.

4.2.4. Normalization

The numerical range of the feature data varies, which causes the training process to be skewed toward significant values. We normalize random features to the normal distribution as follows:

$$sx_i = \frac{f_i - \mu_i}{\sigma_i} \tag{4}$$

where σ_i is the standard deviation and mean of the feature, respectively. We normalize the data for fixed value features using min-max scaling as follows:

$$x_i = \frac{f_i - f_{min}}{f_{max} - f_{min}} \tag{5}$$

4.2.5. Feature Extraction

Feature extraction of the used dataset is done through different following methods

4.2.5.1. Extra Tree Classifier

Through ExtraTreeClassifier 20 best features are extracted shown below in figure 2:

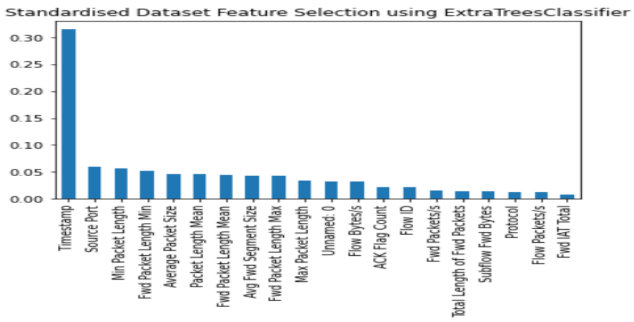

Figure 2. Extra Tree Classifier

4.2.5.2. XGBClassifier

Boosting algorithm "XGBClassifier" also done for best feature selection following is the result of XGBClassifier as shown in figure 3.

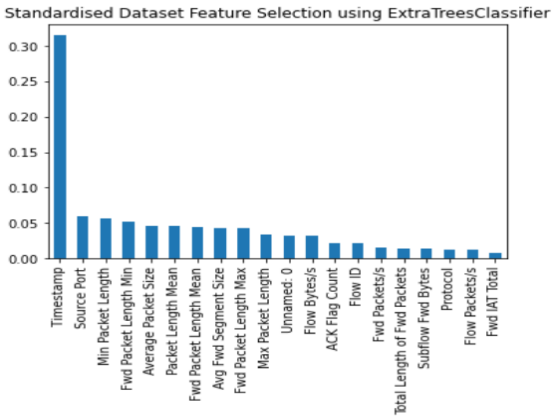

Figure 3. XGB Classifier

4.2.5.3. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an unsupervised learning method for reducing dimensionality in machine learning. It is a statistical process that uses an orthogonal transformation to convert correlated data into a set of linearly uncorrelated data. The newly revised qualities are the Principal Components. It is one of the most extensively used tools for exploratory data analysis and predictive modeling. It is a technique for removing strong patterns from a dataset by lowering variances.

Dimensionality reduction of the above dataset is made by PCA, which shows in Figure 4

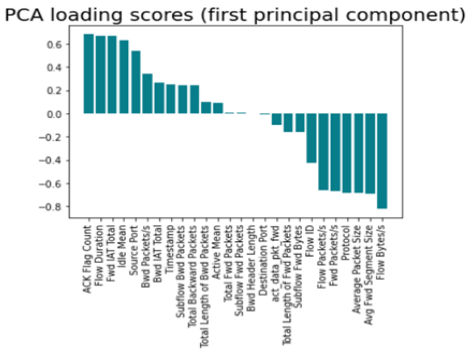

Figure 4. PCA

4.2.5.4. Chi-Square Feature Selection

The Best Features are selected after applying the "chi-square" method to our reduced dataset. Following are the selected 20 best features through "chi-square."

Selected best 20:

['Flow ID', ' Source Port', ' Protocol', ' Timestamp', ' Flow Duration', ' Total Backward Packets', 'Total Length of Fwd Packets', 'Flow Bytes/s', ' Flow Packets/s', 'Fwd IAT Total', 'Bwd IAT Total', 'Fwd Packets/s', ' Bwd Packets/s', ' ACK Flag Count', ' Average Packet Size', ' Avg Fwd Segment Size', ' Subflow Fwd Bytes', ' Subflow Bwd Packets', ' act_data_pkt_fwd', 'Idle Mean']

4.2.5.5. Manualis Features Reduction

In this step of manual features reduction, features/columns of the used dataset (CICDDOS2019) are dropped or deleted manually to extract the best feature of this dataset. This is done on the bases of the following parameters.

1. Remove the columns which have all the entries zero or null.
2. Remove columns containing more than 70% of data null.

3. Remove extra columns like (Fwd Packet Length Max, Bwd Packet Length Min, etc.). Because the total amount is given off that type of column, if total is given max, min, std not needed. It could be found in python if required. If the total amount has not been given of any column type like (Active Max, Idle Min, etc.) is also dropped because the column which has to mean value is enough.
4. Remove that columns which have negative values and infinity values.

4.2.6. Train Test Split

In this step, data is split into the training and testing part. The ratio on which required data is split is 80: 20. Eighty percent of the data is utilized for training the models, whereas 20 percent is used to test the models.

4.2.7. ML Modeling

After splitting the data into training and testing parts, in this step, different Machine Learning models are trained like "Random Forest," "Decision Tree," "SVM (Support Vector Machine)," and "Nave Bayes."

5. Results and Evaluation

All the results in this research are based on three parameters: result after chi-square, result without chi-square and result after manual reduction.

5.1. System Specification

The system used for above results contains the following specifications as shown in table 3.

Table 3. System Specification

Core i5, 5th Generation	
System Processor	AMD PRO A8-8600, 10 Compute Cores 4C+6G 1.6GHz
RAM	4.00GB (3.76GB useable)
SSD	128Gb
HDD	360GB
Operating System	Windows 10 Pro
Tool	Jupyter Notebook (Anaconda Navigator base Environment)
Language	Python

5.2. Accuracy Score

The accuracy level of each method is described below in table 4.

Table 4. Accuracy

Models	**After chi-square**	**Without chi-square**	**Manual Reduction**
Random Forest	99.32749259624877	99.82930240210595	99.87130932828633
Decision Tree	91.4640918065153	99.80719397828233	99.8441015782481
SVM	92.39521635406383	74.79228364593617	66.04580623602293
Naïve Bayes	66.84199160908193	88.77868953603158	87.70480425674111

6. Conclusion

The overall conclusion of the research is that a model based on Data Science and Machine Learning is considerably more appropriate and successful in cyber security, particularly in the prediction of cyberattacks. This prediction is more accurate and automated, and it has made a significant contribution to the area of information security. Following this investigation, it has been discovered that data pre-processing is the fundamental and most crucial phase in developing any model based on data science methodologies. Because open-source datasets include data in large quantities and contain a large amount of raw data, it is

simple to utilize any dataset once it has been pre-processed to match any ML model, such as the one used in this investigation. As a result, the research ends in this chapter. It provides a discussion of the results and suggestions for further research and development. In that it gives a high-level summary of the technique and directs new researchers on the proper path, the chapter is essential.

7. Discussion

After all the above steps of preprocessing the dataset, we know that the accuracy of the different models used in this research depends a lot on feature selection and informative data. It has been seen that when we manually reduce the feature of a dataset based on null and negative values and train and test all the models, the accuracy is increased, but this is very time consuming, and hard to select which feature is relevant and informative and which is not. So here is a tradeoff: by increasing accuracy, we require more time and work.

Through different present feature selection and extraction algorithms and methods, it is easy to select and extract features or columns, but the model's accuracy is compromised. By seeing the above table of accuracy, it is observed. Currently, this research focuses on a specific category of cyber-attack and trains and predicts specific models. However, the future direction is to work on broader cyber-attack categories. A model should be built for the prediction of many categories of cyber-attacks. The extraction and selection of features will be made automatically, but the model's accuracy should not be compromised.

References

1. Vinayakumar, A. (2019). Vinayakumar R., Alazab M., Soman K., Poornachandran P., Al-Nemrat A., Venkatraman S. Deep learning approach for intelligent intrusion detection system, IEEE Access, 7, 41525-41550.
2. Liu, M., Xue, Z., Xu, X., Zhong, C., & Chen, J. (2018). Host-based intrusion detection system with system calls: Review and future trends. ACM Computing Surveys (CSUR), 51(5), 1-36.
3. Aftergood, S. (2017). Cybersecurity: The cold war online. Nature, 547(7661), 30-31.
4. Tansley, Stewart, and Kristin Michele Tolle. *The fourth paradigm: data-intensive scientific discovery*. Ed. Anthony JG Hey. Vol. 1. Redmond, WA: Microsoft research, 2009.
5. Cukier, K. (2013). Data, data everywhere: a special report on managing information. Economist.
6. [online]. https://eng.umd.edu/news/story/study-hackers-attack-every-39-seconds
7. "Cybersecurity Breaches to Increase Nearly 70% Over the Next 5 years," [Online]. Available: https://www.juniperresearch.com/press/business-losses-cybercrime-data-breaches.
8. "15 Alarming Cyber Security Facts and Stats," [Online]. Available: https://www.cybintsolutions.com/cyber-security-facts-stats/.
9. Laskov, P., Düssel, P., Schäfer, C., & Rieck, K. (2005, September). Learning intrusion detection: supervised or unsupervised?. In International Conference on Image Analysis and Processing (pp. 50-57). Springer, Berlin, Heidelberg.
10. P. Krishnamurthy, "Understanding Data Bias Towards Data Science," [Online]. Available: https://towardsdatascience.com/survey-d4f168791e57.
11. Karatas, G., Demir, O., & Sahingoz, O. K. (2020). Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. IEEE Access, 8, 32150-32162.
12. Subba, B., Biswas, S., & Karmakar, S. (2016, March). A neural network based system for intrusion detection and attack classification. In 2016 Twenty Second National Conference on Communication (NCC) (pp. 1-6). IEEE.
13. R. Mastrogiacomo, "The conflict between data science and cybersecurity," [Online]. Available: https://www.information-management.com/opinion/the-conflict-between-data-science-andcybersecurity.
14. D. L. Pegna, "Creating cybersecurity that thinks," [Online]. Available: https://www.computerworld.com/article/2881551/creating-cyber-security-that-thinks.html.
15. Foroughi, F., & Luksch, P. (2018). Data science methodology for cybersecurity projects. arXiv preprint arXiv:1803.04219.
16. National Research Council. (2007). Toward safer and more secure cyberspace. National Academies Press.
17. Craigen, D., Diakun-Thibault, N., & Purse, R. (2014). Defining cybersecurity. Technology Innovation Management Review, 4(10).
18. Aftergood, S. (2017). Cybersecurity: The cold war online. Nature, 547(7661), 30-31.
19. Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. Journal of Computer and System Sciences, 80(5), 973-993.
20. Mukkamala, S., Sung, A., & Abraham, A. (2005). Cyber security challenges: Designing efficient intrusion detection systems and antivirus tools. Vemuri, V. Rao, Enhancing Computer Security with Smart Technology.(Auerbach, 2006), 125-163.
21. A. C. Champion, "Threats and Attacks," in CSE 4471: Information Security;
22. E. 2. Fischer, "Cybersecurity issues and challenges: In brief; https://sgp.fas.org › crs › misc (2016)
23. Sun, N., Zhang, J., Rimba, P., Gao, S., Zhang, L. Y., & Xiang, Y. (2018). Data-driven cybersecurity incident prediction: A survey. IEEE communications surveys & tutorials, 21(2), 1744-1772.
24. Jafarian, J. H., Al-Shaer, E., & Duan, Q. (2015). An effective address mutation approach for disrupting reconnaissance attacks. IEEE Transactions on Information Forensics and Security, 10(12), 2562-2577.
25. Zargar, G. R., & Kabiri, P. (2009, July). Identification of effective network features for probing attack detection. In 2009 First International Conference on Networked Digital Technologies (pp. 392-397). IEEE.
26. P. a. R. D. 1. ,. 2. p.-8. Ram, "Satan: double-edged sword. Computer".
27. Gadge, J., & Patil, A. A. (2008, December). Port scan detection. In 2008 16th IEEE international conference on networks (pp. 1-6). IEEE.
28. Dave Hartley, "What Is SQL Injection?," in SQL Injection Attacks and Defense. https://www.oreilly.com › view › xhtml › CHP001; Justin Clarke. Released June 2009. Publisher(s): Syngress. ISBN: 9781597499736
29. Su, Z., & Wassermann, G. (2006). The essence of command injection attacks in web applications. Acm Sigplan Notices, 41(1), 372-382.
30. Threepak, T., & Watcharapupong, A. (2014, February). Web attack detection using entropy-based analysis. In The International Conference on Information Networking 2014 (ICOIN2014) (pp. 244-247). IEEE.
31. Kapetanovic, D., Zheng, G., & Rusek, F. (2015). Physical layer security for massive MIMO: An overview on passive eavesdropping and active attacks. IEEE Communications Magazine, 53(6), 21-27.
32. ". M. B. Salem and S. J. Stolfo, "Masquerade attack detection using a search behavior" DOI:10.1007/978-3-642-23644-0_10 Source DBLP Conference: Recent Advances in Intrusion Detection - 14th International Symposium, RAID 2011, Menlo Park, CA, USA, September 20-21, 2011. Proceedings.
33. Nikiforakis, N., Meert, W., Younan, Y., Johns, M., & Joosen, W. (2011, February). SessionShield: Lightweight protection against session hijacking. In International Symposium on Engineering Secure Software and Systems (pp. 87-100). Springer, Berlin, Heidelberg.
34. McIntosh, T., Jang-Jaccard, J., Watters, P., & Susnjak, T. (2019, December). The inadequacy of entropy-based ransomware detection. In International conference on neural information processing (pp. 181-189). Springer, Cham.

35. Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. Journal of Computer and System Sciences, 80(5), 973-993.
36. Alazab, M., Venkatraman, S., Watters, P., & Alazab, M. (2010). Zero-day malware detection based on supervised learning algorithms of API call signatures.
37. Bilge, L., & Dumitraş, T. (2012, October). Before we knew it: an empirical study of zero-day attacks in the real world. In Proceedings of the 2012 ACM conference on Computer and communications security (pp. 833-844).
38. Alam, M. F., Singla, P., & Phursule, R. N. (2022, April). Design of Detection System using Deep Learning Algorithm for Attack on Network. In 2022 IEEE 7th International conference for Convergence in Technology (I2CT) (pp. 1-5). IEEE.
39. Masood, R., & Anwar, Z. (2011, December). Swam: Stuxnet worm analysis in metasploit. In 2011 Frontiers of Information Technology (pp. 142-147). IEEE.
40. Tu, T. D., Guang, C., Xiaojun, G., & Wubin, P. (2014, July). Webshell detection techniques in web applications. In Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.
41. Moustafa, N., & Slay, J. (2015, November). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.
42. Mahal, J. A., & Clancy, T. C. (2018, October). Analysis of Pilot-Spoofing Attack in MISO-OFDM System Over Correlated Fading Channel. In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM) (pp. 341-346). IEEE.
43. Zhu, R., Shu, T., & Fu, H. (2017, October). Empirical statistical inference attack against PHY-layer key extraction in real environments. In MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM) (pp. 46-51). IEEE.
44. Nakhila, O., Attiah, A., Jin, Y., & Zou, C. (2015, October). Parallel active dictionary attack on WPA2-PSK Wi-Fi networks. In MILCOM 2015-2015 IEEE Military Communications Conference (pp. 665-670). IEEE.
45. Erdin, E., Aksu, H., Uluagac, S., Vai, M., & Akkaya, K. (2018, October). OS independent and hardware-assisted insider threat detection and prevention framework. In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM) (pp. 926-932). IEEE.
46. Ficke, E., Schweitzer, K. M., Bateman, R. M., & Xu, S. (2018, October). Characterizing the effectiveness of network-based intrusion detection systems. In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM) (pp. 76-81). IEEE.
47. Imamverdiyev, Y., & Abdullayeva, F. (2018). Deep learning method for denial of service attack detection based on restricted boltzmann machine. Big data, 6(2), 159-169.
48. Loukas, G., & Öke, G. (2010). Protection against denial of service attacks: A survey. The Computer Journal, 53(7), 1020-1037.
49. Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. Big data, 1(1), 51-59.
50. D. L. Pegna, "CYBERSECURITY AND DATA SCIENCE," [Online]. Available: https://www.computerworld.com/article/2881551/creating-cyber-security-that-thinks.html.
51. Cao, L., 2017. Data science: challenges and directions. Communications of the ACM, 60(8), pp.59-68. Cao, Longbing. "Data science: challenges and directions." *Communications of the ACM* 60.8 (2017): 59-68.
52. Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: an overview from machine learning perspective. Journal of Big data, 7(1), 1-29.
53. Saxe, J., & Sanders, H. (2018). Malware data science: attack detection and attribution. No Starch Press.
54. McIntosh, T. R., Jang-Jaccard, J., & Watters, P. A. (2018, December). Large scale behavioral analysis of ransomware attacks. In International Conference on Neural Information Processing (pp. 217-229). Springer, Cham.
55. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1), 1-22.
56. Anwar, S., Mohamad Zain, J., Zolkipli, M. F., Inayat, Z., Khan, S., Anthony, B., & Chang, V. (2017). From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. Algorithms, 10(2), 39.
57. Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M., & Karimipour, H. (2019). Cyber intrusion detection by combined feature selection algorithm. Journal of information security and applications, 44, 80-88.
58. Tapiador, J. E., Orfila, A., Ribagorda, A., & Ramos, B. (2013). Key-recovery attacks on KIDS, a keyed anomaly detection system. IEEE Transactions on Dependable and Secure Computing, 12(3), 312-325.
59. Tavallaee, M., Stakhanova, N., & Ghorbani, A. A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40(5), 516-524.
60. Brahmi, I., Brahmi, H., & Yahia, S. B. (2015, May). A multi-agents intrusion detection system using ontology and clustering techniques. In IFIP International Conference on Computer Science and its Applications (pp. 381-393). Springer, Cham.
61. Qu, X., Yang, L., Guo, K., Ma, L., Sun, M., Ke, M., & Li, M. (2021). A survey on the development of self-organizing maps for unsupervised intrusion detection. Mobile networks and applications, 26(2), 808-829.
62. Liao, Hung-Jen, et al. "Intrusion detection system: A comprehensive review." *Journal of Network and Computer Applications* 36.1 (2013): 16-24.
63. Alazab, Ammar, et al. "Using feature selection for intrusion detection system." *2012 international symposium on communications and information technologies (ISCIT)*. IEEE, 2012.
64. Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R., & Bellekens, X. (2018). A taxonomy and survey of intrusion detection system design techniques, network threats and datasets.
65. Liu, M., Xue, Z., Xu, X., Zhong, C., & Chen, J. (2018). Host-based intrusion detection system with system calls: Review and future trends. ACM Computing Surveys (CSUR), 51(5), 1-36.
66. Roesch, M. (1999, November). Snort: Lightweight intrusion detection for networks. In Lisa (Vol. 99, No. 1, pp. 229-238).

67. OSSEC, ""OSSEC - World's Most Widely Used Host Intrusion Detection," [Online]. Available: https://www.ossec.net/.
68. G.V.a.C.Kruegel, Host-based Intrusion Detection, in Handbook of Information Security, 2006; https://sites.cs.ucsb.edu › ~vigna › publication
69. Taj, R. (2020). A Machine Learning Framework for Host Based Intrusion Detection using System Call Abstraction.
70. Viegas, E., Santin, A. O., Franca, A., Jasinski, R., Pedroni, V. A., & Oliveira, L. S. (2016). Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems. IEEE Transactions on Computers, 66(1), 163-177.
71. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. Ieee access, 6, 35365-35381.
72. Dutt, I., Borah, S., Maitra, I. K., Bhowmik, K., Maity, A., & Das, S. (2018). Real-time hybrid intrusion detection system using machine learning techniques. In Advances in Communication, Devices and Networking (pp. 885-894). Springer, Singapore.
73. Ragsdale, D. J., Carver, C. A., Humphries, J. W., & Pooch, U. W. (2000, October). Adaptation techniques for intrusion detection and intrusion response systems. In Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0 (Vol. 4, pp. 2344-2349). IEEE.
74. Criscuolo, P. J. (2000). Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and stacheldraht ciac-2319. California Univ Livermore Radiation Lab.
75. A. ,. 2. Inc., "NETSCOUT Arbor's 13th Annual Worldwide Infrastructure Security Report. Technical Report .".
76. Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE communications surveys & tutorials, 15(4), 2046-2069.
77. Silva, S. S., Silva, R. M., Pinto, R. C., & Salles, R. M. (2013). Botnets: A survey. Computer Networks, 57(2), 378-403.
78. Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. Future generation computer systems, 82, 395-411.
79. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In 2019 International Carnahan Conference on Security Technology (ICCST) (pp. 1-8). IEEE.
80. Dickinson, J., Dickinson, S., Bellis, R., & Mankin, A. D. Wessels," DNS Transport over TCP-Implementation Requirements. RFC 7766, DOI 10.17487/RFC7766, March 2016,< http://www. rfc-editor. org/info/rfc7766.
81. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In 2019 International Carnahan Conference on Security Technology (ICCST) (pp. 1-8). IEEE.
82. Haider, W., Hu, J., & Xie, M. (2015, June). Towards reliable data feature retrieval and decision engine in host-based anomaly detection systems. In 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA) (pp. 513-517). IEEE.
83. Haider, W., Hu, J., Xie, Y., Yu, X., & Wu, Q. (2017). Detecting anomalous behavior in cloud servers by nested-arc hidden semi-Markov model with state summarization. IEEE Transactions on Big Data, 5(3), 305-316.
84. Haider, W., Hu, J., Yu, X., & Xie, Y. (2015, November). Integer data zero-watermark assisted system calls abstraction and normalization for host based anomaly detection systems. In 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing (pp. 349-355). IEEE.
85. Yuxin, D., Xuebing, Y., Di, Z., Li, D., & Zhanchao, A. (2011). Feature representation and selection in malicious code detection methods based on static system calls. Computers & Security, 30(6-7), 514-524.
86. Aghaei, E., & Serpen, G. (2019). Host-based anomaly detection using Eigentraces feature extraction and one-class classification on system call trace data. arXiv preprint arXiv:1911.11284.
87. Mighan, S. N., & Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. International Journal of Information Security, 20(3), 387-403.
88. Creech, G., & Hu, J. (2013, April). Generation of a new IDS test dataset: Time to retire the KDD collection. In 2013 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 4487-4492). IEEE.
89. Xie, M., Hu, J., & Slay, J. (2014, August). Evaluating host-based anomaly detection systems: Application of the one-class SVM algorithm to ADFA-LD. In 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) (pp. 978-982). IEEE.
90. Xie, M., Han, S., & Tian, B. (2011, November). Highly efficient distance-based anomaly detection through univariate with PCA in wireless sensor networks. In 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (pp. 564-571). IEEE.
91. Xie, M., Hu, J., Han, S., & Chen, H. H. (2012). Scalable hypergrid k-NN-based online anomaly detection in wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems, 24(8), 1661-1670.
92. Mahmood, A. N., Hu, J., Tari, Z., & Leckie, C. (2010). Critical infrastructure protection: Resource efficient sampling to improve detection of less frequent patterns in network traffic. Journal of Network and Computer Applications, 33(4), 491-502.
93. Xie, M., Hu, J., & Slay, J. (2014, August). Evaluating host-based anomaly detection systems: Application of the one-class SVM algorithm to ADFA-LD. In 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) (pp. 978-982). IEEE.
94. Osanaiye, O., Choo, K. K. R., & Dlodlo, M. (2016). Analysing feature selection and classification techniques for DDoS detection in cloud. Proceedings of Southern Africa Telecommunication.
95. Velliangiri, S., & Premalatha, J. (2019). Intrusion detection of distributed denial of service attack in cloud. Cluster Computing, 22(5), 10615-10623.

96. Devi, M. D., Priya, V. G., Sarumathy, S., & Sujatha, M. R. Preventing DDoS Attack Using Fuzzy C Mean Clustering.
97. Wu, Y. C., Tseng, H. R., Yang, W., & Jan, R. H. (2011). DDoS detection and traceback with decision tree and grey relational analysis. International Journal of Ad Hoc and Ubiquitous Computing, 7(2), 121-136.
98. Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE communications surveys & tutorials, 15(4), 2046-2069.
99. Chang, R. K. (2002). Defending against flooding-based distributed denial-of-service attacks: A tutorial. IEEE communications magazine, 40(10), 42-51.
100. Jazi, H. H., Gonzalez, H., Stakhanova, N., & Ghorbani, A. A. (2017). Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. Computer Networks, 121, 25-36.
101. Behal, S., Kumar, K., & Sachdeva, M. (2018). D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events. Journal of Network and Computer Applications, 111, 49-63.
102. Liu, Z., Cao, Y., Zhu, M., & Ge, W. (2018). Umbrella: Enabling ISPs to offer readily deployable and privacy-preserving DDoS prevention services. IEEE Transactions on Information Forensics and Security, 14(4), 1098-1108.
103. Aamir, M., & Zaidi, S. M. A. (2021). Clustering based semi-supervised machine learning for DDoS attack classification. Journal of King Saud University-Computer and Information Sciences, 33(4), 436-446.
104. Donges, N. (2018). The random forest algorithm. Towards data science, 22.
105. "Canadian Institute for Cybersecurity," [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html.