FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

DETECTION OF WEB ATTACKS VIA PART CLASSIFIER

Omar Iskndar AHMED

Master's Thesis

DEPARTMENT OF SOFTWARE ENGINEERING
Graduate School of Natural and Applied Sciences

JANUARY 2022

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering
Graduate School of Natural and Applied Sciences

Master's Thesis

DETECTION OF WEB ATTACKS VIA PART CLASSIFIER

Author
Omar Iskndar AHMED

Supervisor
Prof. Dr. Cihan VAROL

JANUARY 2022
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering
Graduate School of Natural and Applied Sciences

Master's Thesis

Title:	Detection of Web Attacks via PART Classifier
Author:	Omar Iskndar AHMED
Submission Date:	19 November 2021
Defense Date:	14 January 2022

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

Signature

Supervisor:	Prof. Dr. Cihan VAROL Sam Houston State University, Faculty of science & Engineering	Approved
Chair:	Doc. Dr. Murat KARABATAK Firat University, Faculty of Technology	Approved
Member:	Dr. Ogr Uyesi Ali ARI Inonu University, Faculty of Engineering	Approved

This thesis was approved by the Administrative Board of the Graduate School on

..... / ..... / 20 .....

Signature

Prof. Dr. Kürşat Esat ALYAMAÇ
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled " Detection of Web Attacks via PART Classifier" in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

14 January 2022

Omar Iskndar AHMED

PREFACE

This thesis aims to use PART classifier to create an intrusion detection system (IDS) that is able to classify network traffics based on CICIDS2017 dataset to detect web-attacks. Detecting web-attacks is important to protect users' sensitive information transferring between network devices. The difficulty of this is that the penetration tools are updated constantly, new ways are invented for attacking the networks. PART as an unused classifier for detecting web-attacks and protecting the traveling data has motivated me to write this thesis.

No person, institution, and organization contributed directly or indirectly to the preparation of this thesis.

Omar Iskndar AHMED
ELAZIG, 2022

TABLE OF CONTENTS

ABSTRACT

Detection of Web Attacks via PART Classifier

Omar Iskndar AHMED

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering
Graduate School of Natural and Applied Sciences
January 2022, Page: vi + 33

With the vast and continuous growth in both computers and communications fields, despite its facilitation of work at all levels, there are a number of new challenges society is facing. The most important of which is the security of sensitive data. With so many hackers wanting to steal sensitive information and exploit it for their own unethical purposes, new protection techniques have to be found. In recent years, Intrusion Detection System (IDS) technology has emerged as an effective option for protecting information within the network. This technology can distinguish between normal traffic and intrusion within the network. In this study, the PART-machine learning classifier algorithm was used to detect web attack attempts based on one of the most recent datasets CICIDS2017. The classifier achieved more than 99% accuracy. RandomForest, NaiveBayes, and BayesNet algorithms are also tested for comparison purposes.

Keywords: CICIDS2017 dataset, IDS, PART, Web attack, WEKA

ÖZET

PART Sınıflandırıcı ile Web Saldırılarının Tespiti

Omar Iskndar AHMED

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı
Fen Bilimleri Enstitüsü
Ocak 2022, Sayfa: vi + 33

Hem bilgisayar hem de iletişim alanlarındaki muazzam ve sürekli büyüme, her düzeyde çalışmayı kolaylaştırmasına rağmen, toplumun karşı karşıya olduğu bir dizi yeni zorluk var. Bunlardan en önemlisi hassas verilerin güvenliğidir. Hassas bilgileri çalmak ve kendi etik olmayan amaçları için kullanmak isteyen bu kadar çok bilgisayar korsanından ötürü yeni koruma tekniklerinin bulunması gerekiyor. Son yıllarda, Saldırı Tespit Sistemi (IDS) teknolojisi, ağ içindeki bilgileri korumak için etkili bir seçenek olarak ortaya çıkmıştır. Bu teknoloji, normal trafik ile ağ içindeki izinsiz girişleri ayırt edebilir. Bu çalışmada, en güncel veri kümelerinden biri olan CICIDS2017'ye dayalı olarak web saldırı girişimlerini tespit etmek için PART-makine öğrenimi sınıflandırıcı algoritması kullanılmıştır. Sınıflandırıcı %99'dan fazla doğruluk başarısını sağlamıştır. RandomForest, NaiveBayes ve BayesNet algoritmaları da karşılaştırma amacıyla test edilmiştir.

Anahtar Kelimeler: CICIDS2017 veri seti, IDS, PART, Web saldırısı, WEKA

LIST OF FIGURES

Page

LIST OF TABLES

Page

SYMBOLS AND ABBREVIATIONS

Abbreviations

CAE : CfsSubset Attribute Evaluator

CFS : Correlation Feature Selection

CSE : Classifier Subset Evaluator

EFS : Ensemble Feature Selection

FOS : Feature Occurrence Subsets

HIDS : Host-based Intrusion Detection System

IDS : Intrusion Detection System

IG : Information Gain

NIDS : Network-based Intrusion Detection System

NVD : National Vulnerability Database

XSS : Cross-Site Scripting

1. INTRODUCTION

Penetrations to the systems have continuously increased in the past years. Penetration methods change, but the goal remains access to sensitive data, as the most important reason for penetration. As mentioned in [1], The National Vulnerability Database (NVD) has released information on over 43,000 software vulnerabilities affecting more than 17,000 software applications since its inception in 1997. Attackers could use these vulnerabilities to penetrate web applications using different methods and attack types such as brute force, XSS, and SQL injection. These vulnerabilities can be exploited to get access to sensitive data such as username, password, and bank number, etc. A skilled hacker can easily evade a firewall and traditional security system to attack a web server, network's node (devices), or even the network itself [2].

After knowing these facts, it is important to protect our data and devices from penetrations. The first step to protecting data is to detect an attack, and this fact was the reason for the emergence of intrusion detection system (IDS) technology. This technology has the ability to distinguish between normal traffic and intrusion within the network. IDS is basically categorized as Network-Based (NIDS) and Host-Based (HIDS). NIDS is located in a predefined point within the network to check traffics of all devices connected to the network. It monitors the traffic that passes through the network, and whenever an attack or abnormal behavior is detected, an alert is sent to the administrator. HIDS is installed on an independent device, to check the incoming and outgoing traffic of the device only. HIDS alerts the device administrator whenever malicious activity is detected.

IDS has three detecting methods: signature-based IDS, anomaly-based IDS, and hybrid-based IDS. Signature-based IDS store pattern (signature) of known attacks and compare them with the incoming traffic packets to detect already known attacks. But it fails in detecting new malware attacks because their signatures are unknown. Anomaly-based IDS (AIDS) is proposed to detect unknown attacks. It uses machine learning algorithms to create a trustful model, then the coming packets are compared with the pre-created model to declare whether any attacks occur or not. The model can be created according to the application and hardware they are used for. Hybrid-based IDS combines signature-based IDS and anomaly-based IDS strengths to detect known and unknown attacks.

Machine learning algorithms are used to create effective AIDS models that can detect attacks. These models are trained and tested using different datasets and to measure their performance false alarm rates and accuracy are used. However, most of the used datasets are highly imbalanced in the terms of cybersecurity, these datasets are 98% classified as normal and only 2% classified as attacks. Two of the most used datasets are the DARPA that is generated in 1998 and the KDD cup 1999 that is generated in 1999. However, these two datasets contain several issues that make them a bad choice

for creating a modern AIDS model. These issues include A) old-fashioned attacks that are not used in the modern network, B) limited amount of data, and C) A large number of duplicated records. Therefore, the NSL-KDD dataset was proposed to overcome the limitations of the 1999 KDD Cup, but it was not without problems too.

In this study the "Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX" CSV file, one of the 8 files of the CIC-IDS2017 dataset, is selected for experiments. Although this dataset is the most updated one of the original KDD-Cup 99 data, it still contains some issues that have been covered in the preprocessing stage. Five features selection algorithms with five new subsets were created using an ensemble feature reduction method [2]. These ten subsets are used for training and testing the system. Then, PART, RandomForest, NaiveBayes, and BayesNet algorithms are used with 10-fold cross-validation to create different models. Algorithms are compared in the terms of accuracy, F1-score, and building time. Finally, the results are compared with the J48 algorithm results obtained by Kshirsagar, D. et al. [2].

Our contribution is to demonstrate the efficacy of the mentioned algorithms to detect web attacks. The rest of the thesis is organized as follows. Section 2 provides a review of the literature. Section 3 describes the used dataset and tested algorithms. Section 4 gives information about the methodology and experimental results. Finally, the conclusion is presented in Section 5.

2. LITERATURE REVIEW

In this section, the related works on intrusion detection systems (IDS) with the CIC-IDS2017 dataset are shortly mentioned.

A comparative model is proposed by Aksu, D. et al. [3] for detecting port scan attacks using deep learning and SVM machine learning classifiers. At first, 286,467 records (from the CIC-IDS2017 dataset) are taken which contain normal traffic and port scan attacks. Then, the data are normalized and divided into 67% as training data and 33% as test data. For the SVM algorithm, no features selection (FS) algorithm is used, and all features are taken into account. On the other hand, the deep learning model contains 7 hidden layers with a different number of neurons. As a result, the accuracy for SVM was 0.6979 (69.79%) and for deep learning, it was 0.9780 (97.80%).

Yulianto, A. et al. [4] proposed a framework to improve the performance of AdaBoost-based Intrusion Detection System (IDS) by using CIC-IDS2017 dataset and the following technologies: 1) Synthetic Minority Oversampling Technique (SMOTE), 2) Principal Component Analysis (PCA), and 3) Ensemble Feature Selection (EFS). The result shows an improvement to AdaBoost-based IDS. Specifically, the authors obtained an accuracy of 81.83%, precision of 81.83%, recall of 100%, and an F1 score of 90.01%.

Panwar, S. S. et al. [5] proposed a model to improve the accuracy of machine learning algorithms (ML) for detecting any intrusion. The researchers used Correlation Feature Selection (CFS), Classifier Subset Evaluator (CSE) with J48, Naive Bayes, and Decision Tree algorithms to estimate the accuracy of these subsets. The WEKA tool was used for testing. J48 algorithm with Correlation Feature Selection (CFS) with Naive Bayes has the highest accuracy (99.9951%) for Brute Force attack.

Stiawan, D. et al. [6] investigated the ability of Information Gain (IG) technology to specify relevant features for traffic classification. They used 20% of the CIC-IDS2017 dataset in this experiment. Then 70% of it is used as training data and 30% as testing data. After that, by using IG, specific features are selected and grouped using their weight. They used Random Forest, Random Tree, Bayes Net, Naive Bayes, and J48 classifiers. The results show that Random Forest has 99.86% accuracy and J48 has 99.87% accuracy with more time needed for execution.

Shailesh Panwar, S. et al. [7] used four feature selections algorithms (Classified Subset Evaluator with Naive Bayes (CSE-NB), Classified Subset Evaluator with Decision Tree (CSE-DT), Subset Evaluator Classified with J48 (CSE-J48), and CfsSubset Attribute Evaluator (CAE)) on the CIC-IDS2017 dataset. ML algorithms used were REPTree and OneR. The researcher calculated accuracy, specificity, sensitivity, and time to evaluate each algorithm. Results showed that for the infiltration attack OneR with Classifier Subset Evaluator with Naive Bayes has 99.9938% accuracy

and for the same attack REPTree with Classifier Subset Evaluator with Naive Bayes has 99.9986% accuracy.

Botnet Detection or malicious detection approach is proposed by Jabbar and Mohammed. [8] to detect any intrusion in network traffic using Machine Learning (ML) with two features selected to filter on the CIC-IDS2017 dataset. The two filters are Correlation Attribute Eval (CAE) and Principal Component (PC) used with six ML algorithms which are Random Forest, Naive Bayes, Multilayer Perceptron, IBK, JRip, and OneR. When the Correlation Attributes Eval filter is used with 9 attributes, the JRip ML algorithm scored the highest accuracy of 100% and Random Forest achieved 99.9974%. While with the Principal Component filter Random Forest and Naïve Bayes scored the highest accuracy of 99.9607%. They later used different features-distance measures to reduce selected features and processing time [9]. Feature-distance measures (Dice, Driver & Kroeber, Pearson Correlation, Cosine, and Overlap) are used to detect the highest 10 features for each one. Then, multilayer perceptron, JRip, IBK, and random forest ML algorithms are tested with a different number of features. The result shows that when eight to ten features are selected, Random Forest and JRip algorithms with both overlap and Driver & Kroeber have the highest accuracy of 100%. Overall, when fewer features (8) are selected, the framework works with higher efficiency.

Alsameraee and Ibrahem [10] addressed the imbalance issue of the CIC-IDS2017 dataset by using the Synthetic Minority Over-Sampling Technique (SMOTE). Later, a Balanced Multiclass Dataset algorithm (BMSD) is proposed. Finally, several machine learning algorithms are implemented with BMSD (balanced) and without BMCD (imbalanced) datasets. Balance dataset improved accuracy from 96.78% to 99.32% for the Random Forest algorithm.

Maseer, Z. K. et al. [11] evaluated the performance of the Anomaly-Based Intrusion Detection System (AIDS) by measuring the true positive and negative rates, accuracy, recall, precision, and F1-Score of 31 ML-AIDS models. Ten machine learning algorithms are divided into supervised (k-NN, SVM, DT, RF, ANN, NB, and CNN) and non-supervised (EM, SOM, and k-means clustering). Supervised k-NN algorithms achieved a 99.52% accuracy. Overall, the k-NN, DT, and NB had the highest accuracy when detecting web attacks.

Park and Park [12] said that although the anomaly-based IDS achieves a high accuracy rating, significant computational overhead is required in training and deployment. To address this issue, a simple ANN IDS model is proposed. This model uses Gradient Descent (GD) and Adam optimizer techniques along with the feed forward and the back propagation algorithms to minimize the overall computational overhead and obtain a high-performance level. The result shows that the ANN with the use of Adam optimizer builds a model that uses only a single hidden layer. Therefore, its computational overhead is less than that of the ANN with a GD optimizer based on the IDS model.

Perez, D. et al. [13] combined two feature learnings called Principal Component Analysis (PCA) and autoencoder with four known anomaly-based intrusion detection systems which are Local Outlier Factor (LOF), One-Class Support Vector Machine (OC-SVM), Isolation Forest (IF), and Robust Covariance (RC) to evaluate the effectiveness of those feature learnings on the mentioned systems. For these experiments, four datasets: UNSW-NB15, NSL-KDD, CIC-IDS-2017, and Kyoto were used. For the CIC-IDS2017 dataset, the best accuracy for PCA was obtained by LOF that scored 0.63% and for autoencoder, the best accuracy was scored by Isolation Forest (IF) with a score of 0.74%. The OC SVM approach using the latent space produced by the autoencoder shows the most improvement for the datasets.

Vinayakumar, R. et al. [14] proposed a deep neural network (DNN) to develop a flexible and effective intrusion detection system IDS that will be able to detect cyber-attacks. The proposed DNN model learns the abstract and high-dimensional feature learning of the IDS data by sending them through several hidden layers. The experiments that were performed on KDDCup 99, NSL-KDD, UNSW-NB15, Kyoto, WSN-DS, and CIC-IDS2017 datasets show that DNN can execute well compared to previous machine learning algorithms models. Finally, a hybrid DNNs framework called Scale-Hybrid-IDS-AlertNet (SHIA) was proposed which can work in real time in both network and host levels to alert the administrator for potential attacks. On the CIC-IDS2017, the DNN with one layer scored 0.960 as accuracy, 0.969 as precision, 0.960 as recall, and 0.962 as F1-score for multi-class classification.

Another framework for real-time detection in a high-speed internet environment was proposed by Zhang, H. et al. [15]. The framework was implemented by a distributed RF detection model based on the Spark engine (DRFBS) that is able to process a huge volume of data very quickly. Finally, the authors concluded that the DRFBS has a shorter detection time, scores higher accuracy, and can detect real-time intrusion in a high-speed internet environment. The model achieved 96.6% as F1-score.

Another model of the intrusion detection system (IDS) is called Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) for Software Defined Networking (SDN) was proposed by Tang, T. A. et al. [16]. This experiment was performed on both NSL-KDD and CICIDS2017 datasets. The proposed model achieved 89% accuracy for the NSL-KDD and 99% for the CIC-IDS2017 datasets. They also concluded that their model demonstrates powerful intrusion detection capabilities in SDN environments.

Another anomaly-based intrusion detection system (AIDS) is called Machine Learning based Intrusion Detection System (MLIDS) for the cloud environment, was proposed by Chiba, Z. et al. [17]. The MLIDS is able to protect both inside and outside intrusions in the cloud environment with high precision and low false alerts. To create this model, the Improved Genetic Algorithm (IGA) and the Simulated Annealing Algorithm (SAA) were combined to generate a hybrid optimization

framework (IGASAA). For the testing purpose, CICIDS2017, CIDDS-001, and NSL-KDD version 2015 datasets were used. The proposed model achieved 99.93% accuracy and 99.92% detection rate on the CICIDS2017 datasets.

Another experiment on the CIC-IDS2017 dataset was proposed by Alrowaily, M. et al. [18]. In this work, seven machine learning algorithms, Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, Multilayer perceptron (MLP), AdaBoost, and Quadratic Discriminant Analysis (QDA) were tested to evaluate their performance. The results showed the best performance by the KNN algorithm that achieved 99.55% accuracy, 99.53% precision, 99.55% recall, and 99.50% F1-score.

Ferrag, M. A. et al. [19] proposed a hierarchical IDS model based on the combination of three classifiers that are REP tree, JRip algorithm, and Forest PA called Rule and Decision tree-based iteration detection system (RDTIDS). REP Tree and JRip work in parallel and feed the Forest PA classifier. Then the proposed model was trained and tested using the CIC-IDS2017 and the BoT-IoT datasets. Then, the results were compared to other models such as Random Forest, REP tree, MLP, Naïve Bayes, JRip, and J48. The proposed model RDTIDS scored the highest accuracy rating of 96.665% for the CIC-IDS2017 and 96.995% for the BoT-IoT.

Adhao, R. B. et al. [20] showed the effect of choosing the best features on accuracy and building time. The CICIDS2017 dataset was used with the J48 algorithm. The researchers started their experiment by selecting all features and decreasing one each time and comparing the accuracy and build time of each experiment. After 82 experiments, starting from selecting 82 features to only one feature, the J48 algorithm achieved the best results in terms of accuracy and building time when 15 features were selected. They concluded that reducing the features to 15 instead of using all the features will increase the accuracy and speed. For 15 features, 96.36% accuracy and 16.58 seconds as building time were achieved. Where, when all features are used, 96.11% accuracy and 68.22 second building time were achieved. In terms of the building time, the reduced subset was much faster and could be a better choice when fast detection is needed, such as for a real-time application.

Table 2.1 summarizes the comparison between different literature studies mentioned in this section.

Table 2.1. Comparison of Literatures.

#Ref	Algorithm	Feature selection/technology	Accuracy
[3]	SVM	Not used	69.79%
	Deep learning	Not used	97.80%
[4]	AdaBoost	EFS	81.83%
[5]	J48	CFS	99.9951%
[6]	Random Forest	IG	99.86%
	J48	IG	99.87%
[7]	OneR	CSE-NB	99.9938%

Table 2.1.(Continued)

#Ref	Algorithm	Feature selection/technology	Accuracy
	REPTree	CSE-NB	99.9986%
[8]	Jrip	CAE	100%
	Ransom Forest	CAE	99.9974%
		PC	99.9607%
	Naïve Bayes	PC	99.9607%
[9]	Random Forest	Feature-distance measures (overlap)	100%
		Feature-distance measures (Driver & Kroeber)	
	JRip	Feature-distance measures (overlap)	100%
		Feature-distance measures (Driver & Kroeber)	
[10]	Random Forest	With BMSD	99.32%
		Without BMSD	96.78%
[11]	k-NN	Not used	99.52%
[13]	Local Outlier Factor (LOF)	Principal Component Analysis (PCA)	63%
	Isolation Forest (IF)	Autoencoder	74%
[14]	DNN	Not used	96%
[15]	DRFBS	Not used	96.6%
[16]	GRU-RNN	Not used	99%
[17]	IGASAA	Not used	99.93%
[18]	KNN	Not used	99.55%
[19]	RDTIDS	Not used	96.665%
[20]	J48	Deacease one each time	96.36%

3. THE CIC-IDS2017 DATASET AND MACHINE LEARNING CLASSIFIERS

In this section, we provide an overview of the CIC-IDS2017 dataset as well as machine learning classifiers and the Weka tool that have been used in this thesis. Web-attacks contained in this dataset such as brute force, SQL injection, and XSS are also described briefly.

3.1. CIC-IDS2017 dataset

The Canadian Institute of Cybersecurity produced the CIC-IDS2017 dataset by recording network traffic for five days. CIC-IDS2017 is the latest among its peers originated by the KDDCUP 99 dataset and has been widely used due to its possession of the necessary features and data that is suitable to the development of securing networks these days. It consists of 8 separates files. Each file contains normal traffic called (Benign) and abnormal traffic. The characteristics of the dataset are presented in Table 3.1 and a brief description of the 8 files is presented in Table 3.2. The "Thursday-WorkingHours-Morning WebAttacks.pcap_ISCX is the selected file for this study, because it contains the web-attacks Brute Force, Cross-Site Scripting (XSS) and SQL Injection. The benign class covered 168,186 records, 21 records are covered by SQL injection, 652 records are covered by XSS, and 1507 records are covered by brute force class in the preprocessing stage. A duplicated feature called "Fwd Header Length" has been removed. Also, "NaN" and "Infinity" values are replaced with 0 to enhance the performance.

Table 3.1. Characteristics of CIC-IDS2017 dataset.

Characteristic's name	Working Days
Name	CIC-IDS2017
Type	Multi-class dataset
Year of release	2017
Total number of distinct instances	2830540
Number of distinct classes	15
Number of features	79
Number of files	8

Table 3.2. A brief description of the 8 files of the CIC-IDS2017 dataset [21].

File's Name	Working Days	Class Found
Monday-WorkingHours.pcap_ISCX.csv	Monday	Benign
Tuesday-WorkingHours.pcap_ISCX.csv	Tuesday	Benign, SSH-Patator, FTP-Patator

File's Name	Working Days	Class Found
Wednesday-workingHours.pcap_ISCX.csv	Wednesday	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ ISCX.csv	Thursday	Benign, Brute Force, XSS, Sql Injection
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ ISCX.csv	Thursday	Benign, Infiltration
Friday-WorkingHours-Morning.pcap_ISCX.csv	Friday	Benign, Bot
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Friday	Benign, PortScan
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Friday	Benign, DDoS

3.1.1. Web-attack SQL injection

Structured Query Language (SQL) is a number of lines of code that is able to transfer data between the database and application. SQL is simple and easy to organize, insert, update and delete data from a database system. SQL standers have been published by ISO in 1987. Executing a SQL query is done typically by writing a code that is running line by line and retrieving a set of data in an organized manner. Where SQL injection is the process of changing SQL query or changing parameter's value to gain access to the database's data [22]. The attacker can insert new SQL keywords or operators to modify a query [23]. The SQL attack is listed No. 1 in the OWASP organization's Top 10 security risk report in 2017 [23]. If the attack is successful, the attacker can spoof user's identities, read sensitive data from a database, editing (insert, delete and update) data, make the data unavailable or destroy it as well as performing administration operations such as turning off the database management system (DBMS). This leads to the main security concepts of any system in the cybersecurity field which are authentication, authorization, confidentiality, and integrity. OWASP organization has defined these concepts as follow:

a. Authentication

If the written code for authentication is not good enough, an SQL injection attack can bypass the identity mechanism.

b. Authorization

If the sensitive data are stored in plain text (not encrypted), an SQL injection attack can break the authorization.

c. Confidentiality

Confidentiality is broken if the attacker gains access to a database that stores sensitive data.

d. Integrity

Since with an SQL injection an attacker can access the data or even delete them, this will disrupt the integrity.

The below SQL query shows a normal SQL query to retrieve specific data from a database.

Select ID, FirstName, Surname from Customers

The above is a normal SQL query to select the id, first name, and surname of all customers in the customer's Table.

An attacker can create a malicious SQL by adding a single quotation character in a text field as shown in Figure 3.1.

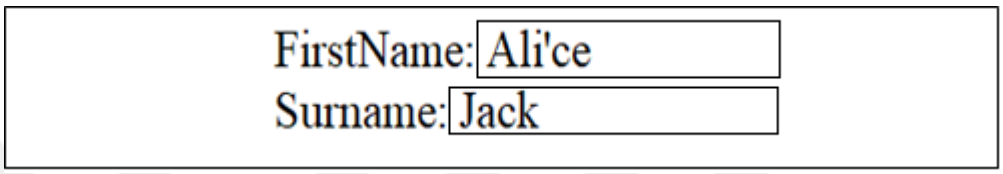

Figure 3.1. Adding quotation to textfield.

Now, for the Figure 3.1, the SQL query will become:

Select ID, FirstName, Surname from Customers where FirstName = 'Ali'ce' and Surname = 'Jack'

When the above query is executed, instead of retrieving the required data from the database, it will return an error like:

Incorrect syntax near li' as the database tried to execute "Ali"

The reason for this error is that the quotation character means escape in a SQL query. But this is not the worst thing that the attacker can do. By writing the following string in the textfield as shown in Figure 3.2, the attacker will delete all customers data from the database table.

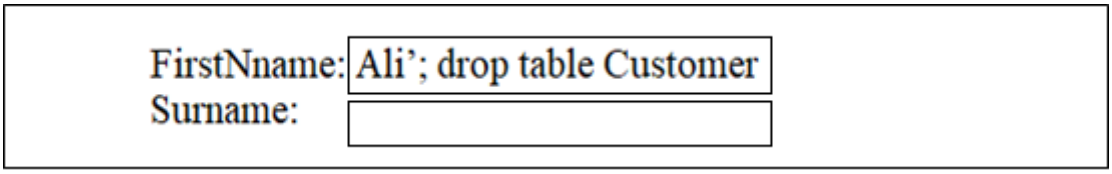

Figure 3.2. Delete all customer's data.

3.1.2. Web-attack Cross-Site Scripting (XSS)

The web-attack Cross-Site Scripting (XSS) is also a type of injection attack where a malicious user sends or executes a malicious script or code using web-applications to another user's device. Cross-site scripting flaws typically allow an attacker to impersonate a victim user, perform all of the user's actions, and access all of the user information. Also, if the attacker gains access to an administrator account or gets a higher privilege, he can control the whole system and do what

he wants [24]. The XSS attack is listed No. 7 in the OWASP organization's Top 10 security risk report in 2017 [23]. They also mentioned that two-thirds of web applications are vulnerable to XSS attacks [24]. XSS attacks simply take advantage of a vulnerable web application's lack of input validation to insert malicious code. After that, an authenticated user requests access to and use of the web application or clicks the URL sent by the attacker. The attacker can extract the victim's private information, such as a session ID, username, and password from the injected code. Finally, because the attacker acquired access to the victim's sensitive information, the attacker might get illegal access to the victim's session while it was still ongoing/active [23].

Cross-Site Script (XSS) attack has three main types:

a. Reflected XSS

This type is the most common one between XSS attacks types. When a susceptible application accepts malicious code and includes it in an HTTP response, this is known as reflected XSS. It all begins when the attacker injects malicious code into a Web-based application. The attacker then creates an URL link that is XSS-infected. Then, via e-mail or other means, give those URLs to the target. The URL link appears to be genuine, so there will be no doubt. Finally, if the target hits the link, the attacker will have access to all of the victim's personal information. In order to activate XSS with Reflected XSS, user involvement is required.

Figure 3.3 shows the process of reflected cross-site script attack.

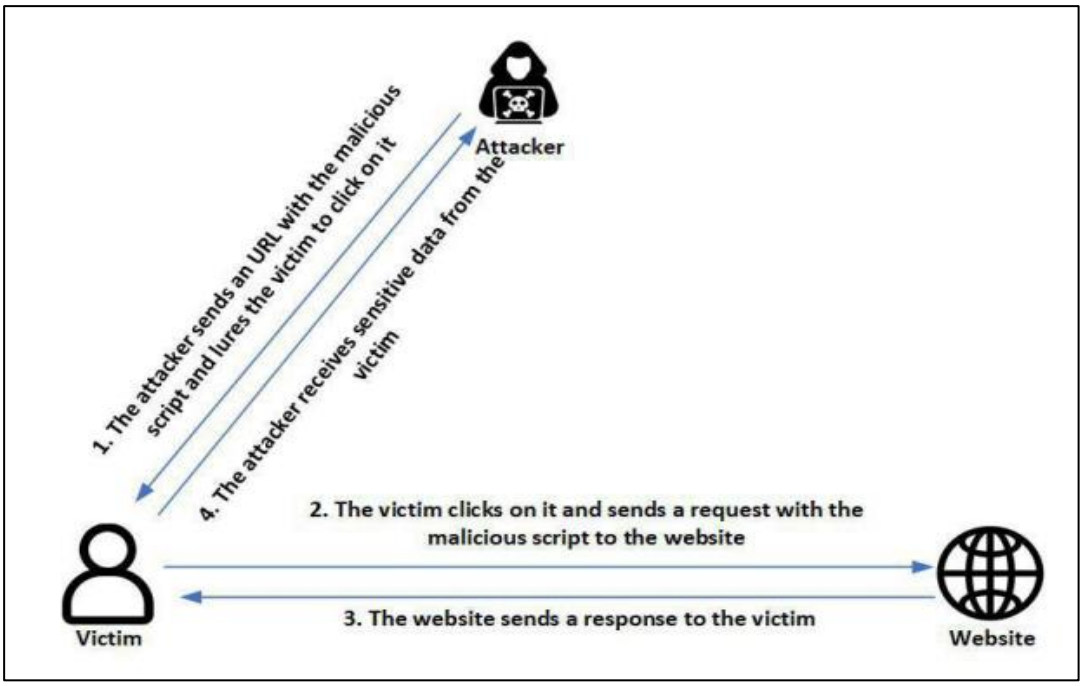

Figure 3.3. Reflected XSS attack [23].

b. Stored/Persistent XSS

Like the reflected XSS type, stored XSS accepts malicious code as well as executes and stores it in the web application. Therefore, every user that requests the link to the injected web application will be affected by this type of XSS attack. The attacker can even execute malware such as trojans by using stored XSS attacks. Because it stores malicious code in the server, this type is known as the most dangerous type of XSS.

Stored XSS process is shown in Figure 3.4.

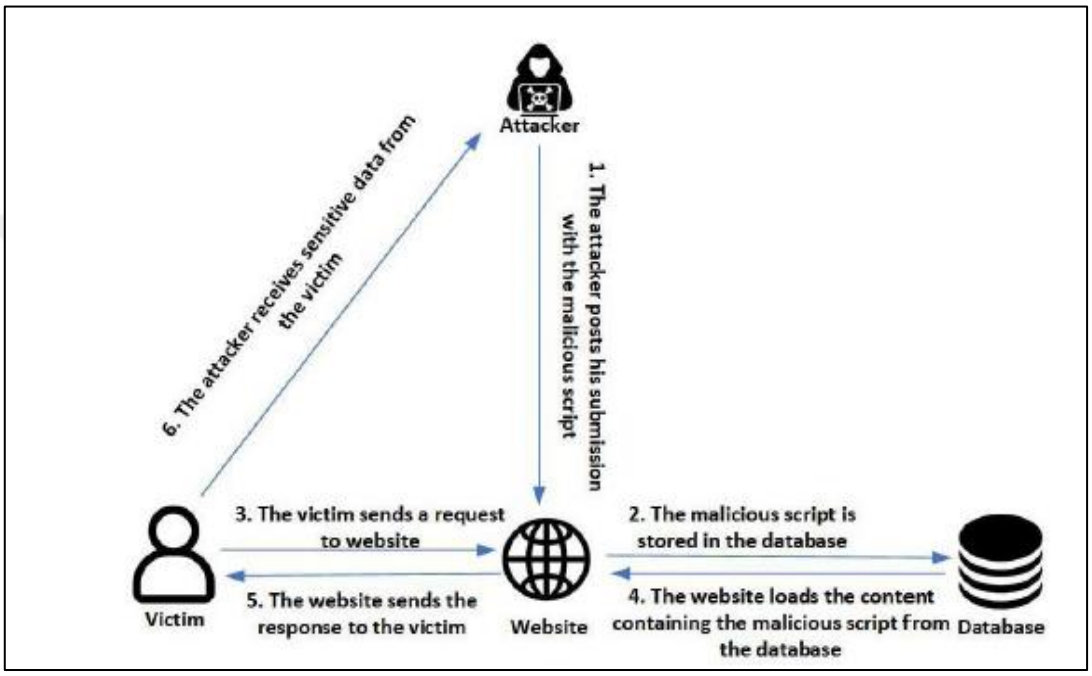

Figure 3.4. Stored XSS attack [23].

c. Dom-Based XSS

Both reflected XSS and Stored XSS communicate with the server and use the server as a trap to attack their victims. But Dom-based XSS works on the client-side and uses the objects that are stored in the browser when JavaScript code is executed. In this attack, the attacker can claim some data such as document cookies or document location. Dom-Based XSS process is shown in Figure 3.5.

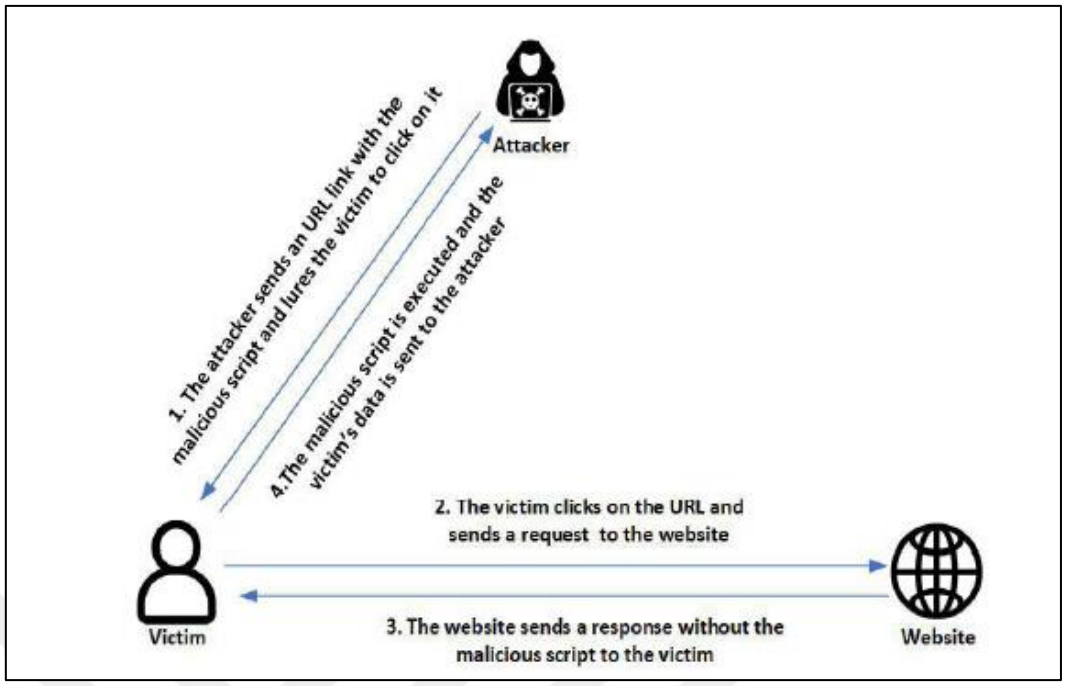

Figure 3.5. DOM-based XSS attack [23].

3.1.3. Brute Force Web-attack

The brute force is a common web attack where an attacker tries to gain access to a user account by entering all possible values to the login form, either using dictionary methods that contain a list of pre-arranged strings or a random sequence method that tries all possible strings in a sequence [24, 25]. This attack could be very effective if the user uses a common username and password that can be guessed easily. The top 3 common usernames used in WordPress are [25]:

a. admin: 84%.
b. administrator: 4%.
c. root: 4%.

And top 3 common passwords used in WordPress are [25]:

a. password: 14%.
b. admin: 10%.
c. 123456: 6%.

Figure 3.6 shows the most common usernames and Figure 3.7 shows the most common passwords.

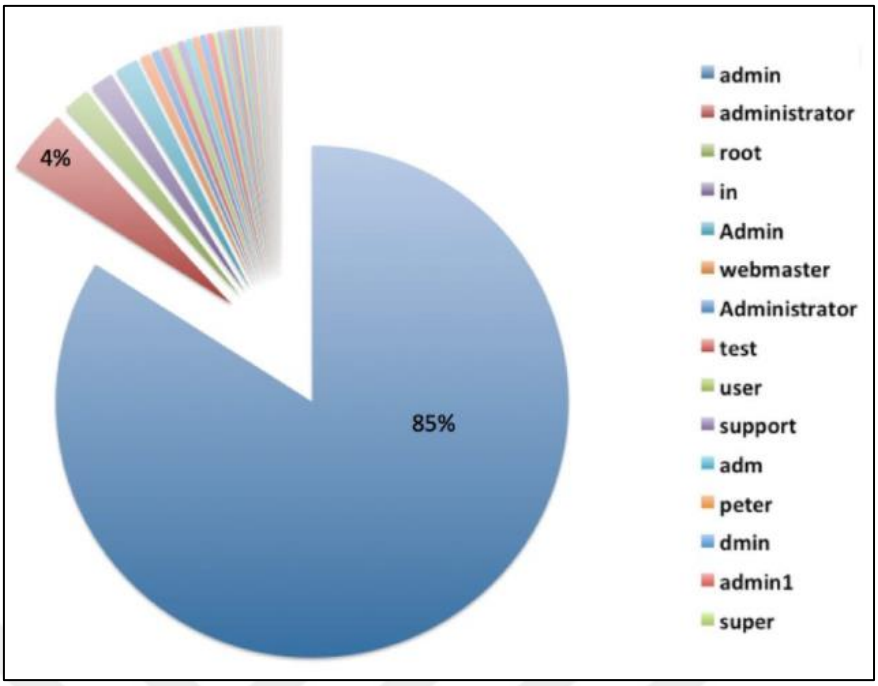

Figure 3.6. Common username used in WordPress [25].

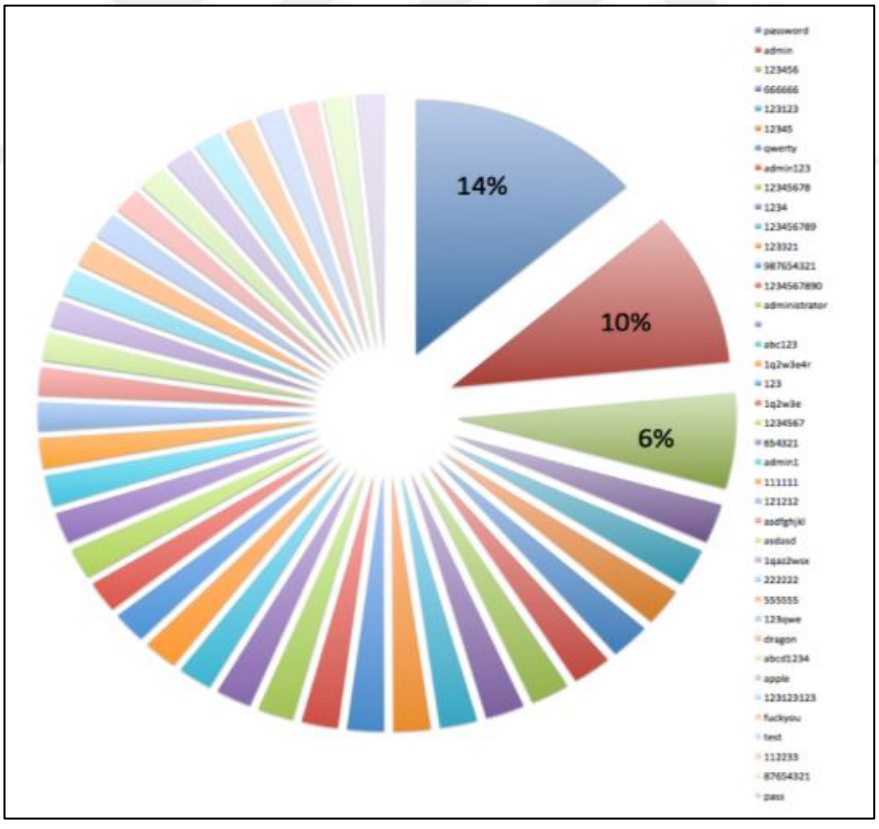

Figure 3.7. Common password used in WordPress [25].

3.2. PART Classifier

The PART algorithm is a combination of C4.5 and RIPPER algorithms [26]. While C4.5 creates a tree and transfers it to a rule-set, RIPPER uses the "separate and conquer" algorithm that provides a more direct method to generate one rule at a time. It first selects an instance and makes it a rule. Then, it removes covered instances from trained data, and it repeats the process to create new rules one by one for the remaining instances until no instances remain. Then, it uses reduced error pruning as a pruning method. This pruning method (Reduced error or Global optimization) will be applied to a rule if and only if it will increase the accuracy of the ruleset [26].

By repeatedly generating partial decision trees, the PART classifier becomes an effective and accurate technique for deriving learning one at a time. Unlike both C4.5 and RIPPER, the PART does not need to perform global optimization when generating a set of rules. This provides simplicity as the main advantage. Combining decision trees and separate and conquer technologies adds flexibility and speed [26]. Using separate and conquer techniques in PART and standard approach is different in the way of creating rules. Instead of wasting time to build a pruned decision tree to create a single rule and discard it, PART uses a "partial" decision tree. To create a partial tree, the construction and pruning operations are combined to produce a tree that cannot be pruned anymore. Then this tree is converted to a rule. Because PART avoids post-processing, it solves the very slow performance of the C4.5 algorithm for pathological datasets [26].

3.3. Random Forest Classifier

Random forest is one of the most popular classification algorithms used in intrusion detection systems (IDS). It is an ensemble of multiple decision tree classifiers that use bagging technology to generate subsets of data for each tree and classification and regression tree (CART) to build a tree that is pruning-free [27]. This means that some instances could be selected several times while some never selected [28]. Two of each three instances are selected to train the model, those instances are call in-bag instances, while the remaining ones are selected to test the performance of the model, those are called out-of-the-bag instances [28].

Error estimate of random forest classifier is called out-of-the-bag (OOB) error. The classifier produces multiple trees independently without any pruning and each node is split using a user-predefined number of attributes. The attributes selection is a random process [28]. By creating a pre-defined number of trees, random forest creates trees that have high variance and low bias [29]. The average of class assignment probabilities is calculated by all the created trees to take the final classification decision. As a result, new unlabeled data input is compared to all decision trees and each tree casts a vote for class membership. The membership class that receives the most votes will be selected as the final result [28].

3.4. Naïve Bayes Classifier

Naive Bayes is one of the most famous machine learning and data mining methods for classification. It calculates the chance that a new instance belongs to a specific class based on the assumption that all features are unrelated to each other, which means changing the value of one feature does not influence the value of any other feature used in this classifier [30, 31]. After calculating the probability of an instance for each class, the instance will be assigned to the class with the highest probability [32].

Naïve Bayes calculate the probability of an instance using the following formula:

$$P(H|E)=\frac{P(H) * P(E|H)}{P(E)} \tag{3.1}$$

Where:

P(H|E) is Probability of Hypothesis "H" given an event "E".

P(H) is Probability of Hypothesis "H" irrespective of any other event.

P(E|H) is Probability that event "E" will occur given the Hypothesis "H".

P(E) is. Probability that the event "E" will occur.

Figure 3.8 shows the workflow diagram of Naïve Bayes classifier.

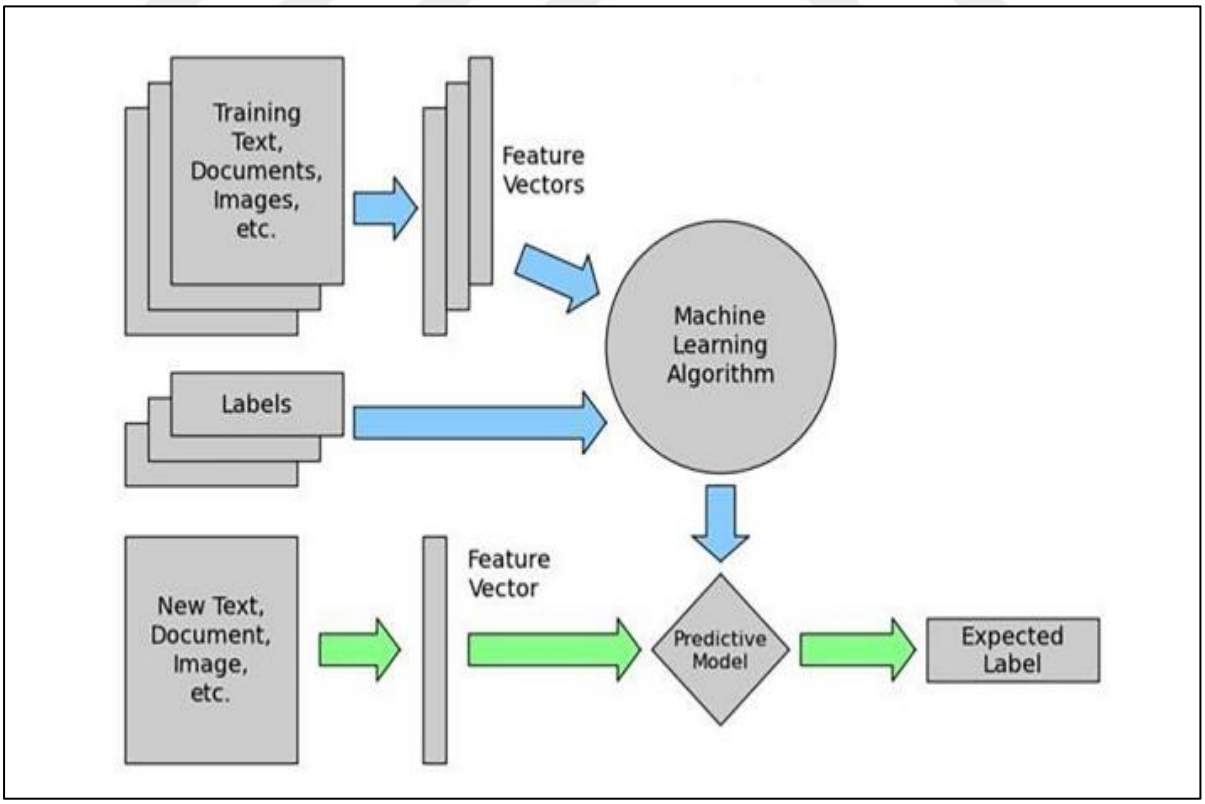

Figure 3.8. Naïve Bayes classifier workflow [32].

3.5. Bayes Net Classifier

Bayes Net or Bayes Network is a widely used method that builds a Bayesian network by calculating the conditional probability on each node, based on the Bayes theorem [33]. It's a member of the probabilistic graphical model family that uses a directed acyclic graph (DAG) to express a set of variables and their conditional dependencies. Bayes Net learns in two stages: network structure learning and probability table learning. This graph algorithm is used to express knowledge in an area that is ambiguous [34].

3.6. Weka Tool

WEKA stands for Waikato Environment for Knowledge Learning and is an open-source GUI program that has been developed by the University of Waikato in New Zealand. It facilitates a variety of machine learning and data mining operations on a dataset, including classification, regression, clustering, pre-processing, and attribute choices, as well as visualizing the results.

In this thesis, five feature (attributes) selection algorithms with four machines learning classifiers (PART, RandomForest, NaiveBayes, and BayesNet) are used to calculate the accuracy, F1-measure, and build time for each classifier on the CIC-IDS2017 dataset in Weka tool.

4. METHODOLOGY AND EXPERIMENTS

The methodology of our experiment is shown in Figure 4.1. After the preprocessing stage mentioned in the previous section. Five filter technologies (Information Gain (IG), Correlation (CR), Gain Ratio (GR), ReliefF, and Chi-square) are used with Ranker as a search method to automatically select 19 features for each filter technology. The selected features for each filter technology are shown in Table 4.1. Then, depending on the ensemble feature reduction method proposed in [2], the five technologies selected features are used to produce new subsets called Feature Occurrence Subsets (FOS). Features that were selected at least once are put in FOS1, features that were selected at least twice are put in FOS2, and so on until FOS5. The selected features for each subset are shown in Table 4.2. As the next step, the PART, RandomForest, BayesNet, and NaiveBayes algorithms are experimented with the five technologies as well as with the FOS subsets. After that, the results of the algorithms are compared in terms of accuracy, F1-score, and model building time. After executing the five algorithms using the Weka data mining tool, the best results are compared to the literature review.

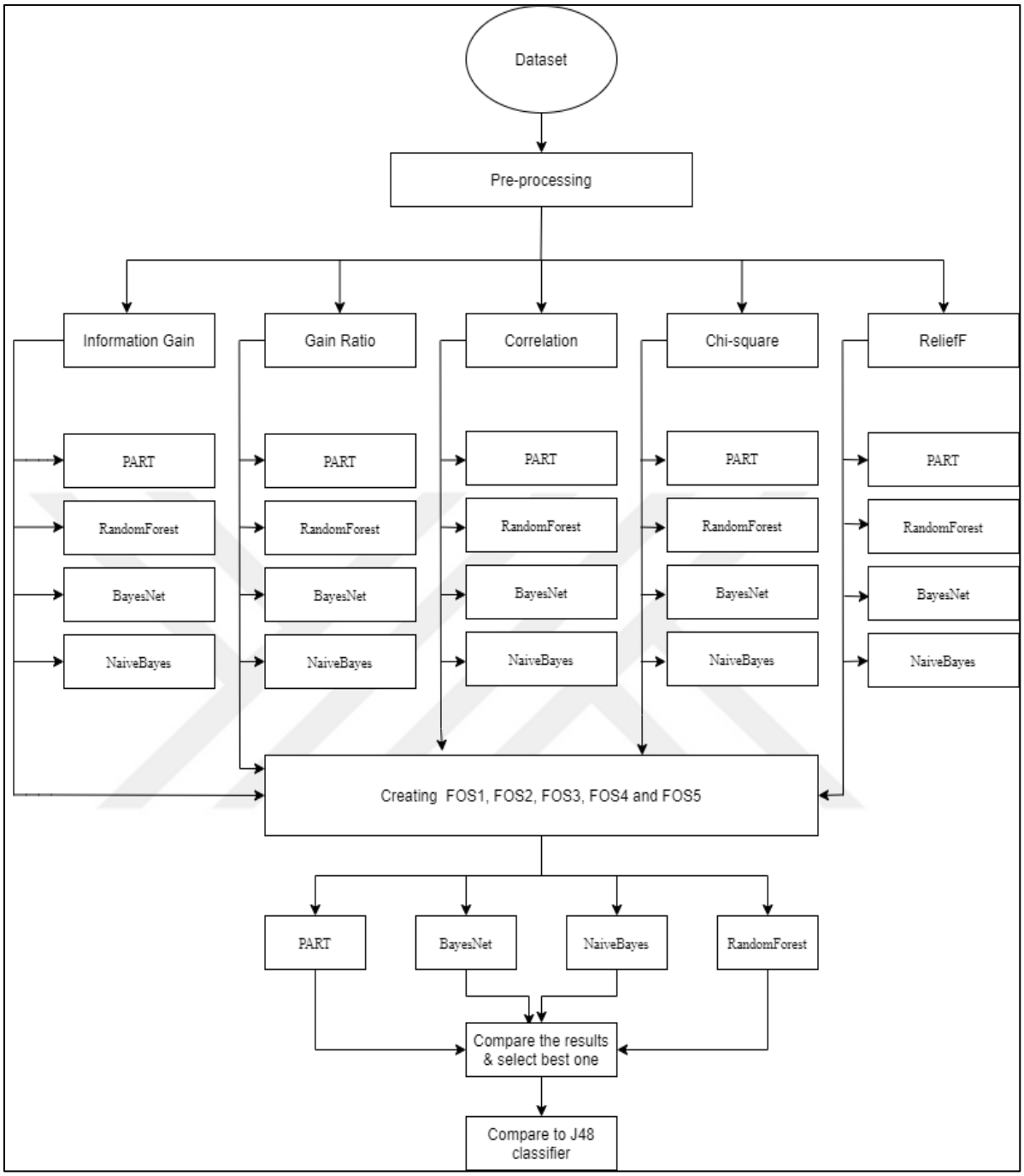

Figure 4.1. Flow Chart of Testing Procedure.

Table 4.1. Selected features by filter technologies.

Filter technology	Selected feature's addresses
IG	67, 25, 37, 35, 16, 22, 18, 17, 19, 23, 38, 24, 2, 21, 20, 40, 7, 9, 54
CR	67, 47, 66, 52, 69, 39, 12, 8, 23, 53, 41, 48, 1, 13, 55, 42, 40, 11, 9
GR	67, 25, 9, 54, 7, 63, 5, 23, 40, 1, 18, 47, 41, 66, 53, 2, 22, 24, 35
ReliefF	47, 48, 66, 49, 1, 69, 2, 26, 21, 31, 45, 19, 29, 52, 24, 12, 67, 14, 55
Chi-square	67, 25, 18, 37, 22, 16, 35, 23, 17, 19, 2, 20, 38, 21, 7, 24, 40, 5, 63

Table 4.2. Feature Occurrence Subset (FOS).

FOS	Selected feature's addresses
FOS1	1, 2, 5, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 29, 31, 35, 37, 38, 39, 40, 41, 42, 45, 47, 48, 49, 52, 53, 54, 55, 63, 66, 67, 69
FOS2	1, 2, 5, 7, 9, 12, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 35, 37, 38, 40, 41, 47, 48, 52, 53, 54, 55, 63, 66, 67, 69
FOS3	1, 2, 7, 9, 18, 19, 21, 22, 23, 24, 25, 35, 40, 47, 66, 67
FOS4	2, 23, 24, 40, 67
FOS5	67

The personal computer used to implement this experiment is Intel(R) Core (TM) i5-3230M CPU @ 2.60GHz as a processor with 8GB RAM. The accuracy and F1-Scores are calculated using the following formulas. Experiments were carried out with the 10 folds cross-validation on the CICIDS2017 dataset. The performance of each classifier was evaluated in terms of Accuracy, F1-Measure, and Building time (time needed to create the model).

4.1. Accuracy

Accuracy is a performance metric that expresses the percentage of correctly classified instances. It does not consider the true positives and true negatives separately. This is the fundamental reason why accuracy alone cannot be used to define model performance. Aside from accuracy, other performance metrics must be used.

$$\text{Accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}} \tag{4.1}$$

4.2. Precision

Precision, also known as positive predictive value, calculates the percentage of relevant rather than irrelevant findings. It's only a small percentage of significant recollections.

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{4.2}$$

4.3. Recall

Recall is the proportion of relevant instances remembered. The recall for the most important result is the sensitivity.

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{4.3}$$

4.4. F1-Measure

F1-Measure is a value that estimates the system's overall performance by combining precision and recall into a single number.

$$\text{F1-Score}=2*\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

where:

TP is the number of correct predictions that an instance is positive.

FP is the number of incorrect predictions that an instance is positive.

FN is the number of incorrect predictions that an instance is negative.

TN is the number of correct predictions that an instance is negative.

Figure 4.2 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using Information gain technology when the top 19 features are selected.

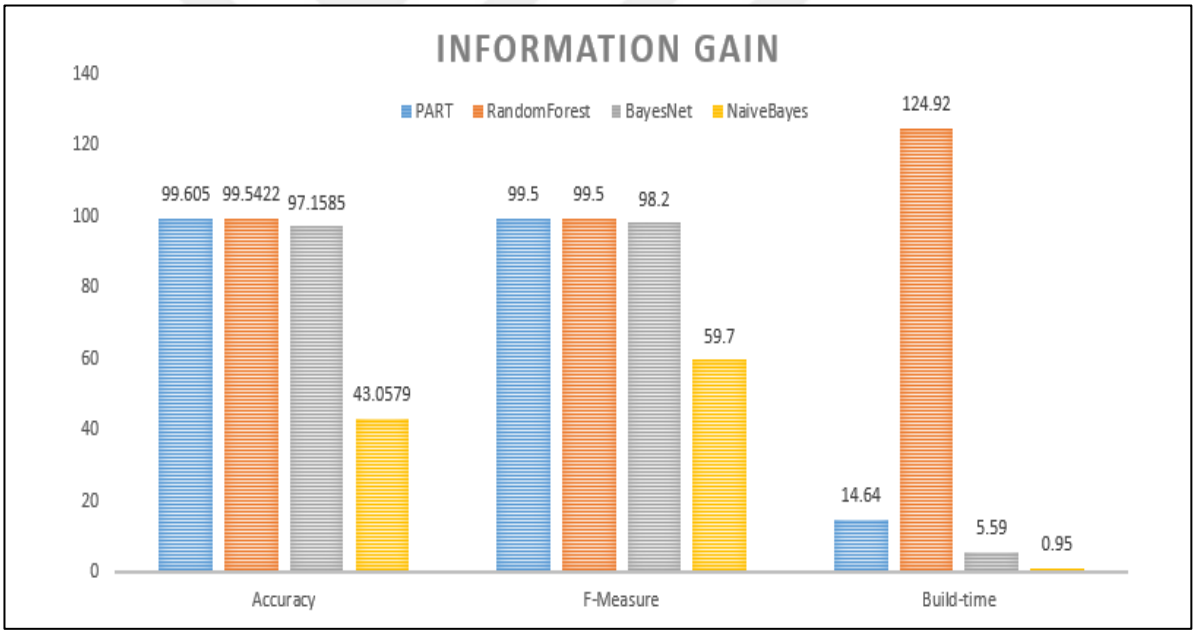

Figure 4.2. Comparison using Information Gain.

Figure 4.3 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using Gain Ratio technology when the top 19 features are selected.

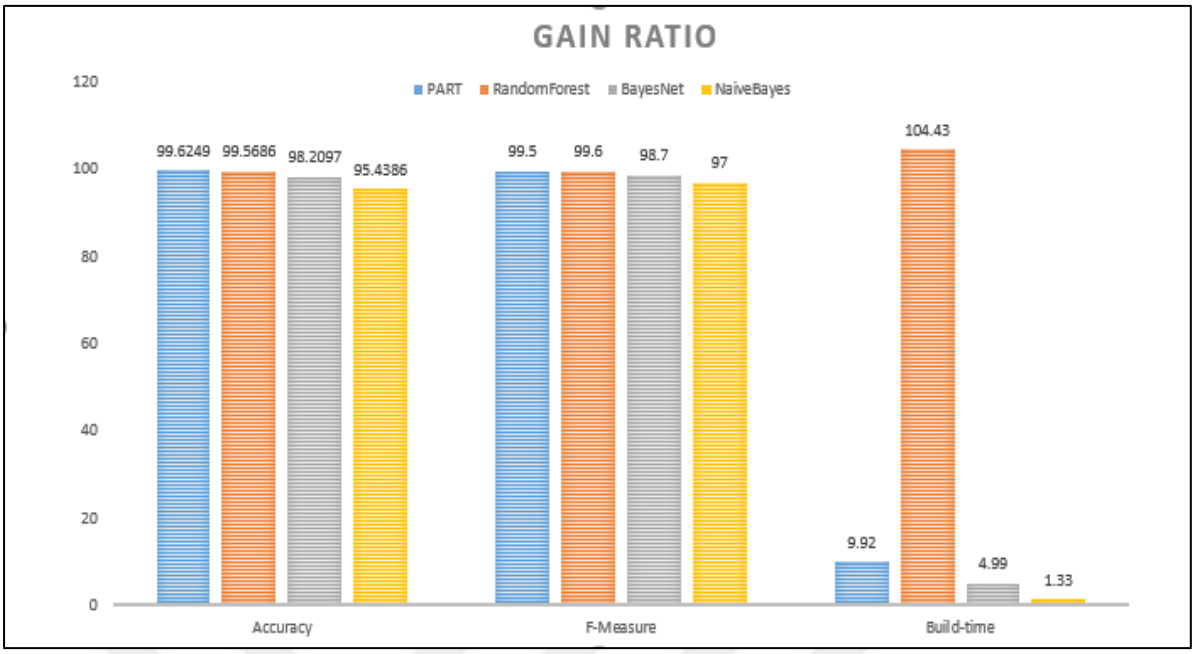

Figure 4.3. Comparison using Gain Ratio.

Figure 4.4 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using correlation technology when the top 19 features are selected.

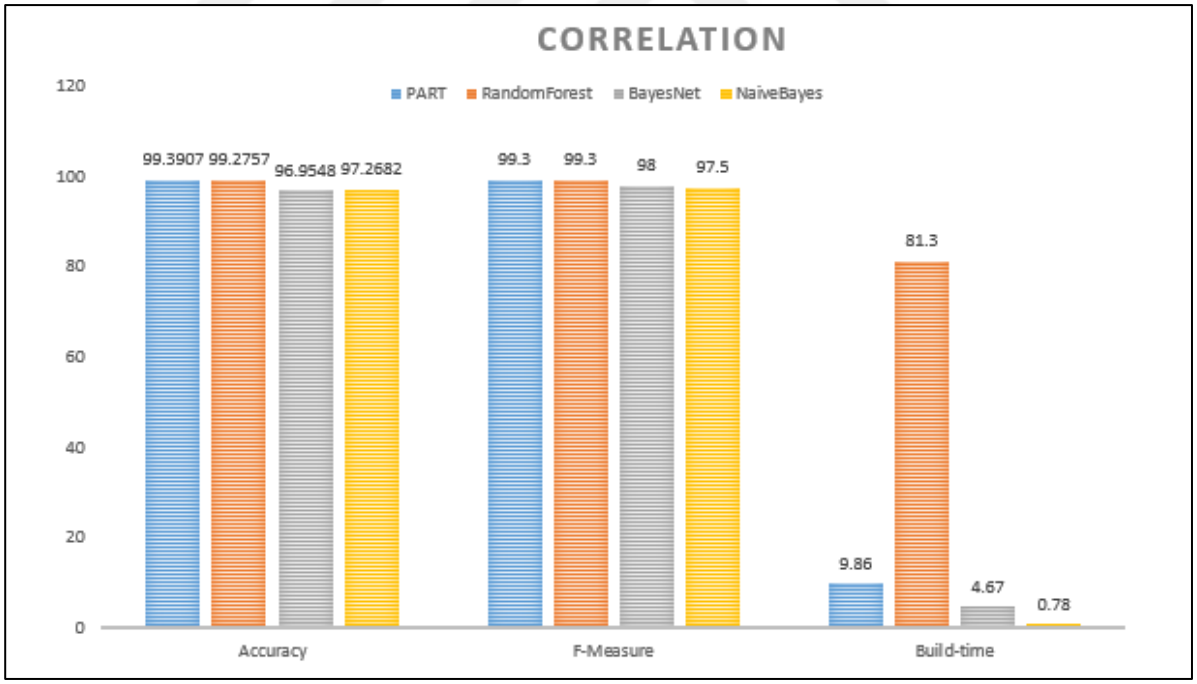

Figure 4.4. Comparison using Correlation.

Figure 4.5 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using Chi-square technology when the top 19 features are selected.

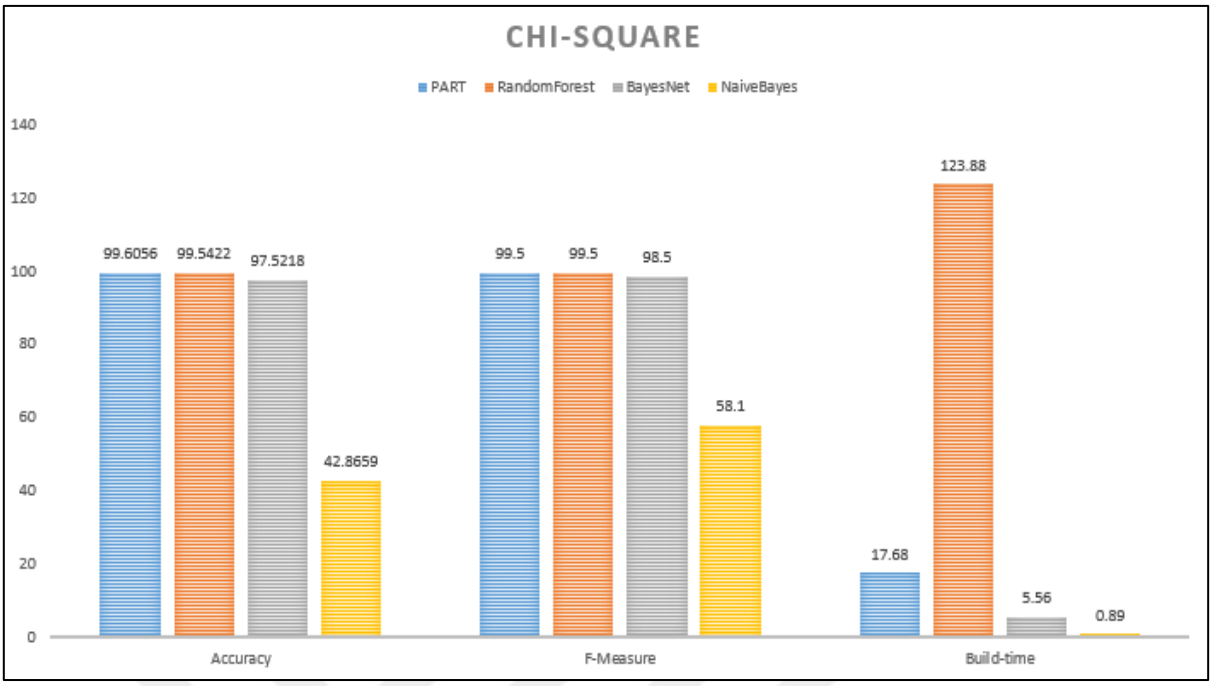

Figure 4.6. Comparison using Chi-square.

Figure 4.6 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using ReliefF technology when the top 19 features are selected.

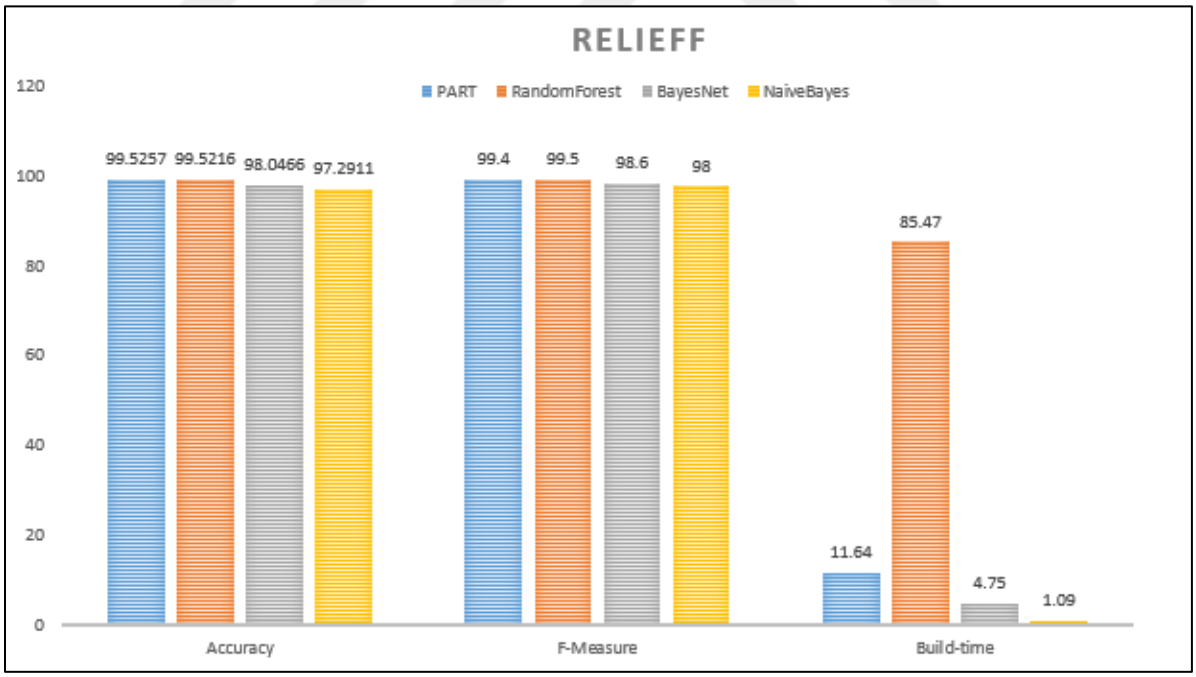

Figure 4.5. Comparison using ReliefF.

After completing our experiments on the five feature selection techniques using the PART, RandomForest, NaiveBayes, and BayesNet classifiers, we found that the PART classifier achieved the highest accuracy among other classifiers and needed less building time than Random Forest.

The same experiments are repeated on FOS1, FOS2, FOS3, FOS4, and FOS5. Figure 4.7 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using FOS1 that contain 42 features.

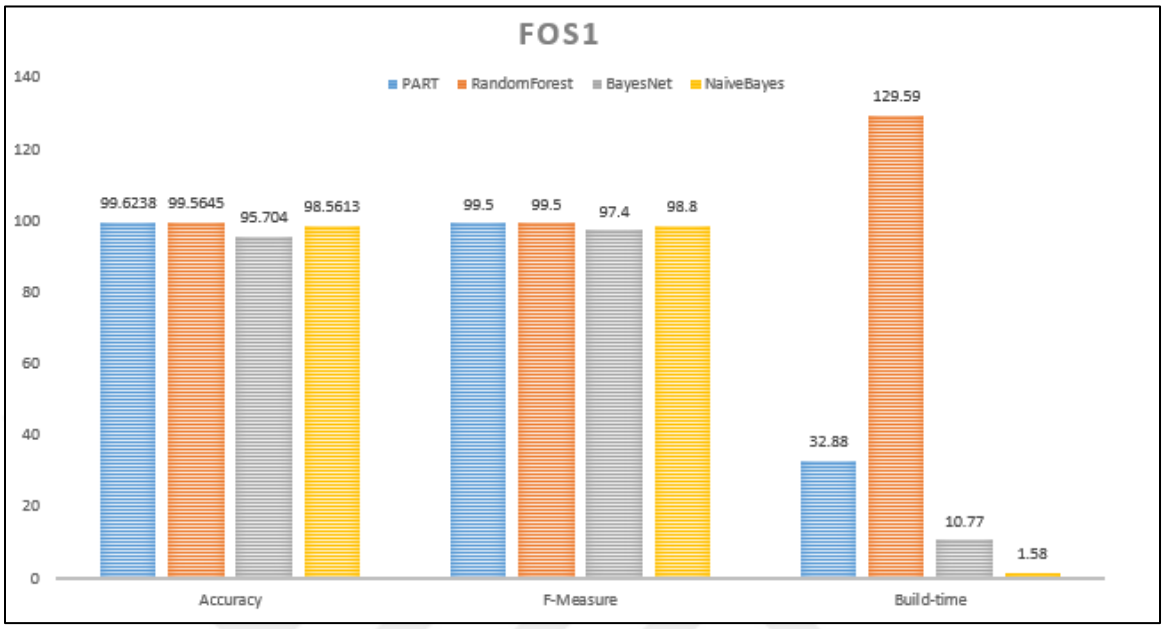

Figure 4.7. Comparison using FOS1.

Figure 4.8 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using FOS2 that contain 31 features.

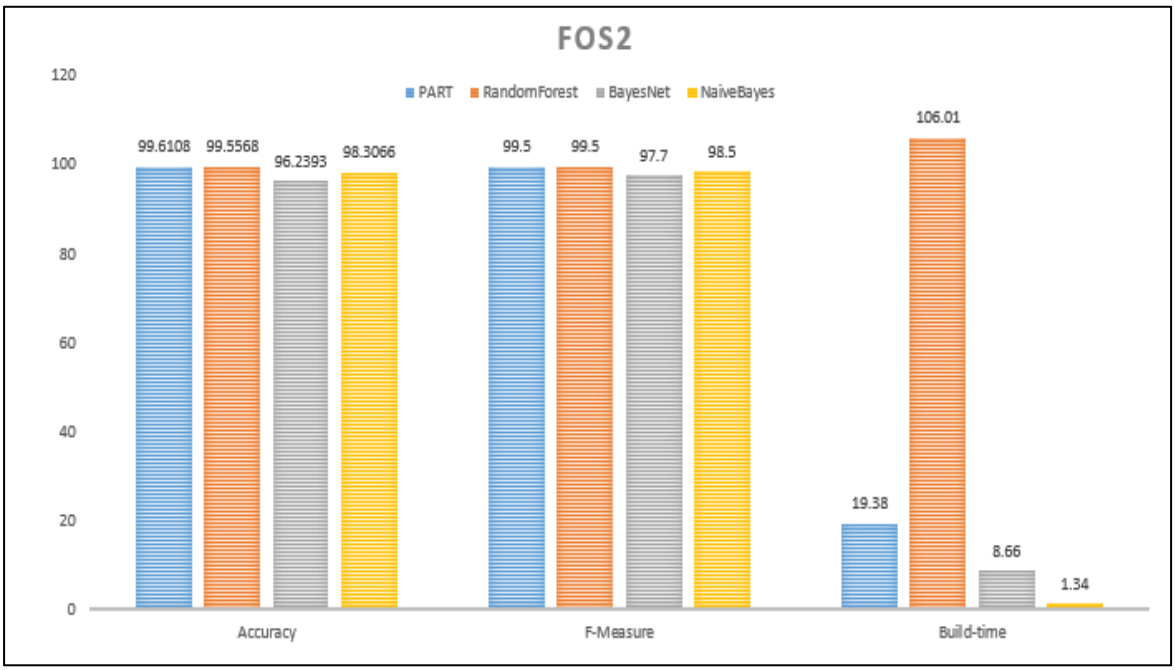

Figure 4.8. Comparison using FOS2.

Figure 4.9 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using FOS3 that contain 16 features.

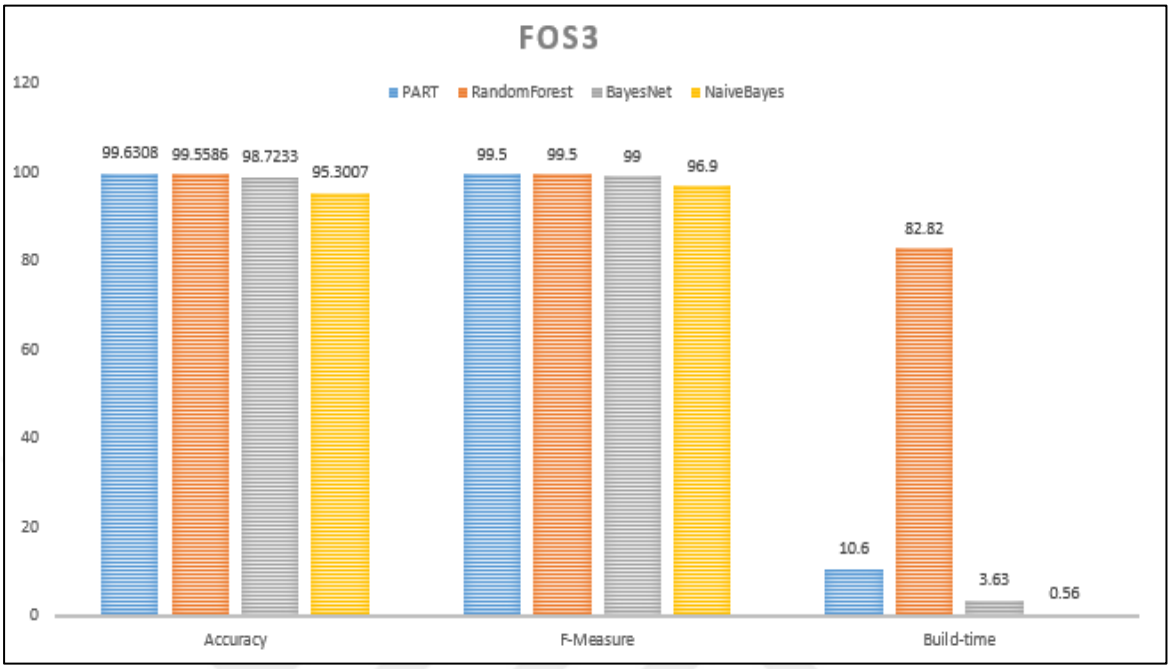

Figure 4.9. Comparison using FOS3.

Figure 4.10 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using FOS4 that contain 5 features.

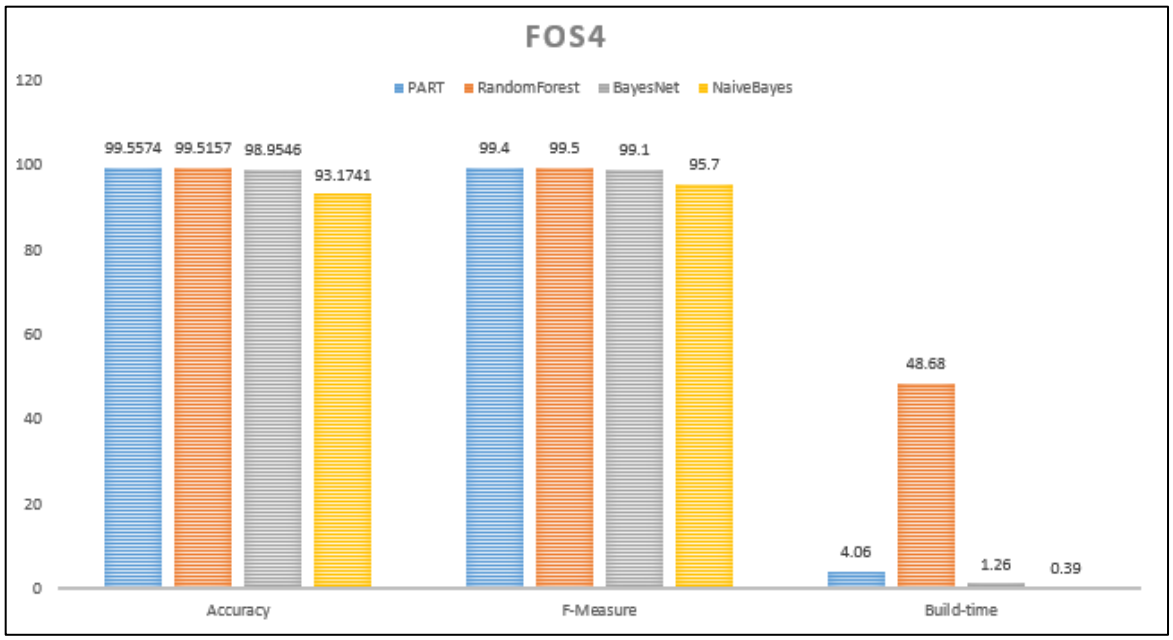

Figure 4.10. Comparison using FOS4.

Figure 4.11 shows the comparison between the PART, RandomForest, BayesNet, and NaiveBayes classifiers using FOS5 that contain 1 feature.

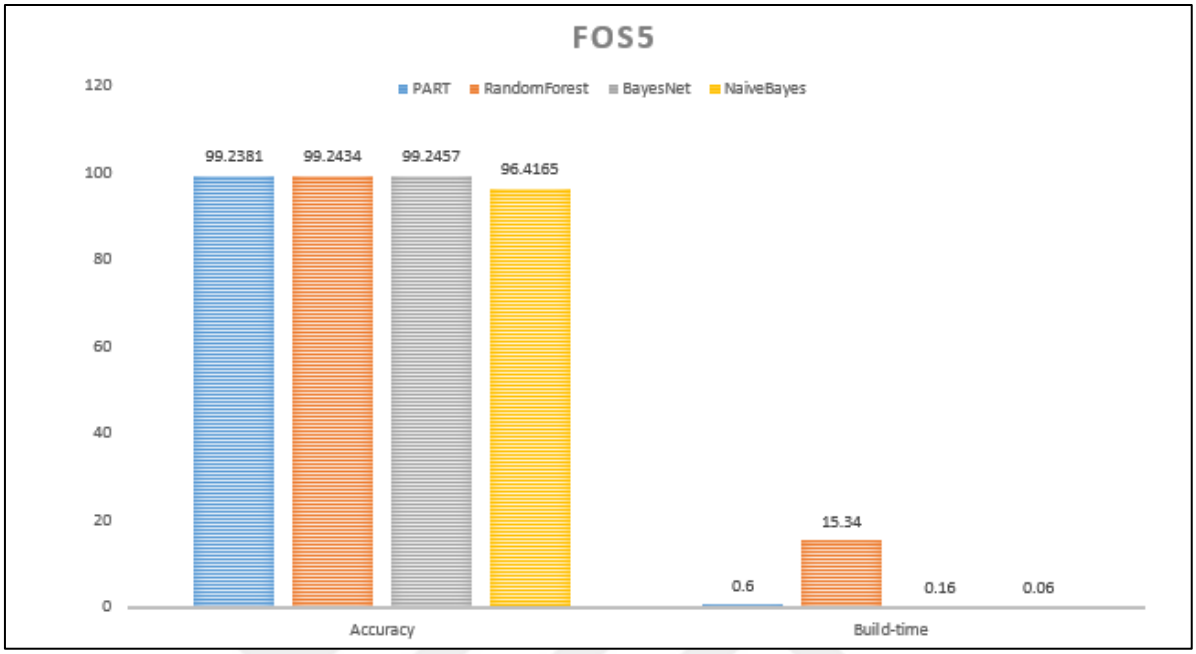

Figure 4.11. Comparison using FOS5.

Once again, previous experiments show the success of the PART's accuracy classifier over others when FOS1, FOS2, FOS3, and FOS4 subsets are used. Table 4.3 summarizes the results of the experiments. From the aforementioned table, it is clear that the PART classifier is superior to its peers in most of the experiments that were carried out in this thesis. RandomForrest is ranked the second best, slightly ahead of the PART classifier in terms of accuracy. However, there is a huge difference between them in terms of the time it takes for each of them to build their own model, and here it is easy to notice the remarkable superiority of PART Classifier. The third best accurate algorithm was BayesNet. Despite its clear superiority over PART and RandomForest in terms of building time, it was inferior to them in terms of accuracy. Naive Bayes, despite its clear superiority over others in terms of building time, it failed to achieve high accuracy especially when used with information gain and chi-square technologies, 43.0579% for IG and 42.8659% for chi-square was achieved. The PART classifier achieved its highest accuracy when the FOS3 subset is used, which consists of 16 features with an accuracy of 99.6308% percent.

Table 4.3. Comparison summarizes.

Technology	No. features	PART			Random Forest			NaiveBayes			BayesNet		
		Accuracy %	F1-score %	B. time (s)	Accuracy %	F1-score %	B. time (s)	Accuracy %	F1-score %	B. time (s)	Accuracy %	F1-score %	B. time (s)
IG	19	99.605	99.5	14.64	99.5422	99.5	124.92	43.0579	59.7	0.95	97.1585	98.2	5.59
GR	19	99.6249	99.5	9.92	99.5686	99.6	104.43	95.4386	97	1.33	98.2097	98.7	4.99
CR	19	99.3907	99.3	9.86	99.2757	99.3	81.3	97.2682	97.5	0.78	96.9548	98	4.67
Chi-square	19	99.6056	99.5	17.68	99.5422	99.5	123.88	42.8659	58.1	0.89	97.5218	98.5	5.56
ReliefF	19	99.5257	99.4	11.64	99.5216	99.5	85.47	97.2911	98	1.09	98.0466	98.6	4.75
FOS1	42	99.6238	99.5	32.88	99.5645	99.5	32.88	98.5613	98.8	1.58	95.704	97.4	10.77
FOS2	31	99.6108	99.5	19.38	99.5568	99.5	106.01	98.3066	98.5	1.34	96.2393	97.7	8.66
FOS3	16	99.6308	99.5	10.6	99.5586	99.5	82.82	95.3007	96.9	0.56	98.7233	99	3.63
FOS4	5	99.5574	99.4	4.06	99.5157	99.5	48.68	93.1741	95.7	0.39	98.9546	99.1	1.26
FOS5	1	99.2381	?	0.6	99.2434	?	15.34	96.4165	?	0.06	99.2457	?	0.16

Table 4.4 show the confusion matrix for PART classifier when highest accuracy is achieved with the FOS3 subset.

Table 4.4. Confusion Matrix.

		Predicted			
		Benign	**Brute force**	**XSS**	**Sql injection**
Actual	**Benign**	168176	9	1	0
	Brute force	6	1492	6	3
	XSS	13	587	51	1
	Sql injection	2	1	0	18

Now we will calculate the average Sensitivity and the average Specificity of the above matrix using the following formula:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.5}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4.6}$$

The weighted average sensitivity is equal to (0.996) rounded to 3 digits and the weighted average Specificity is equal to (0.99) rounded to 3 digits.

Compared with the J48 classifier [2], the PART classifier achieved better accuracy in most experiments but was slower in the build time as shown in Table 4.5.

Table 4.5. Comparison of PART and J48 classifier.

Technology	**No. features**	**PART**		**J48**	
		Accuracy (%)	**B.Time (s)**	**Accuracy (%)**	**B.Time (s)**
IG	19	99.605	14.64	99.6038	8.17
GR	19	99.6249	9.92	99.6173	8.91
CR	19	99.3907	9.86	99.3884	5.67
Chi-square	19	99.6056	17.68	99.6003	8.97
ReliefF	19	99.5257	11.64	99.5375	5.65
FOS1	42	99.6238	32.88	99.6167	18.87
FOS2	31	99.6108	19.38	99.6149	13.83
FOS3	16	99.6308	10.6	99.6191	7.34
FOS4	5	99.5574	4.06	99.5621	1.57
FOS5	1	99.2381	0.6	99.2463	0.29

In this comparison, the best results for both classifiers are being achieved when the FOS3 subset is used, J48 scored 99.6191% and PART scored 99.6308% as accuracy, leading to PART being more accurate but J48 being 7% faster.

If the average accuracy and build time of both classifiers are calculated, 99.54128% as accuracy with 13.126s as build time are achieved by the PART classifier, for J48 99.54064% as accuracy and 7.927s as build time were the results.

5. CONCLUSIONS

This thesis shows the ability of four machine learning algorithms (PART, RandomForest, NaiveBayes, and BayesNet) to detect web-attacks. The experiment is implemented on the CICIDS2017 dataset as the most updated dataset among its peers. The algorithms are compared in terms of accuracy, F1-score, and building time. The results show that the PART classifier achieved the highest accuracy with 99.6308%. Also, we see that the J48 classifier can be faster than the PART classifier and as explained earlier PART is simpler than J48. In the future, we will test the ability of PART classifier to detect other attacks such as BotNet, DoS and PortScan. Also, we will compare PART classifier performance, when different datasets are used. Finally, we want to use PART classifier with new feature selection algorithms to improve the result.

RECOMMENDATIONS

Finally, many important topics are suggested for future works:

1. Testing the ability of PART classifier with other attacks such as BotNet, DoS and PortScan.
2. Comparing PART classifier performance when different datasets are used.
3. Using PART classifier with new feature selection algorithms to get better result.

REFERENCES

[1] Amankwah, R., Chen, J., Kudjo, P. K., & Towey, D. (2020). An empirical comparison of commercial and open-source web vulnerability scanners. *Software: Practice and Experience*, *50*(9), 1842-1857.

[2] Kshirsagar, D., & Kumar, S. (2020). An ensemble feature reduction method for web-attack detection. *Journal of Discrete Mathematical Sciences and Cryptography*, *23*(1), 283-291.

[3] Aksu, D., & Aydin, M. A. (2018, December). Detecting port scan attempts with comparative analysis of deep learning and support vector machine algorithms. In *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)* (pp. 77-80). IEEE.

[4] Yulianto, A., Sukarno, P., & Suwastika, N. A. (2019, March). Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset. In *Journal of Physics: Conference Series* (Vol. 1192, No. 1, p. 012018). IOP Publishing.

[5] Panwar, S. S., Negi, P. S., Panwar, L. S., & Raiwani, Y. (2019). Implementation of machine learning algorithms on cicids-2017 dataset for intrusion detection using WEKA. *International Journal of Recent Technology and Engineering Regular Issue*, *8*(3), 2195-2207.

[6] Stiawan, D., Idris, M. Y. B., Bamhdi, A. M., & Budiarto, R. (2020). CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access*, *8*, 132911-132921.

[7] Singh Panwar, S., Raiwani, Y. P., & Panwar, L. S. (2019, March). Evaluation of network intrusion detection with features selection and machine learning algorithms on CICIDS-2017 dataset. In *International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019, Uttaranchal University, Dehradun, India*.

[8] Jabbar, A. F., & Mohammed, I. J. (2020, November). Development of an Optimized Botnet Detection Framework based on Filters of Features and Machine Learning Classifiers using CICIDS2017 Dataset. In *IOP Conference Series: Materials Science and Engineering* (Vol. 928, No. 3, p. 032027). IOP Publishing.

[9] Jabbar, A. F., & Mohammed, I. J. (2021). BotDetectorFW: an optimized botnet detection framework based on five features-distance measures supported by comparisons of four machine learning classifiers using CICIDS2017 dataset. *Indonesian Journal of Electrical Engineering and Computer Science*, *21*(1), 377-390.

[10] Alsameraee, A. A. A., & Ibrahem, M. K. (2020). Toward Constructing a Balanced Intrusion Detection Dataset: Toward Constructing a Balanced. *Samarra Journal of Pure and Applied Science*, *2*(3), 132-142.

[11] Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., & Foozy, C. F. M. (2021). Benchmarking of machine learning for anomaly-based intrusion detection systems in the CICIDS2017 dataset. *IEEE Access*, *9*, 22351-22370.

[12] Park, S., & Park, H. (2019, March). Ann based intrusion detection model. In *Workshops of the International Conference on Advanced Information Networking and Applications* (pp. 433-437). Springer, Cham.

[13] Pérez, D., Alonso, S., Morán, A., Prada, M. A., Fuertes, J. J., & Domínguez, M. (2019, May). Comparison of network intrusion detection performance using feature representation. In *International Conference on Engineering Applications of Neural Networks* (pp. 463-475). Springer, Cham.

[14] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access*, *7*, 41525-41550.

[15] Zhang, H., Dai, S., Li, Y., & Zhang, W. (2018, November). Real-time distributed-random-forest-based network intrusion detection system using Apache spark. In *2018 IEEE 37th international performance computing and communications conference (IPCCC)* (pp. 1-7). IEEE.

[16] Tang, T. A., McLernon, D., Mhamdi, L., Zaidi, S. A. R., & Ghogho, M. (2019). Intrusion detection in sdn-based networks: Deep recurrent neural network approach. In *Deep Learning Applications for Cyber Security* (pp. 175-195). Springer, Cham.

[17] Chiba, Z., Abghour, N., Moussaid, K., & Rida, M. (2019). Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms. *computers & security*, *86*, 291-317.

[18] Alrowaily, M., Alenezi, F., & Lu, Z. (2019, July). Effectiveness of machine learning based intrusion detection systems. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage* (pp. 277-288). Springer, Cham.

[19] Ferrag, M. A., Maglaras, L., Ahmim, A., Derdour, M., & Janicke, H. (2020). Rdtids: Rules and decision tree-based intrusion detection system for internet-of-things networks. *Future internet*, *12*(3), 44.

[20] Adhao, R. B., & Pachghare, V. K. (2019, December). Performance-Based Feature Selection Using Decision Tree. In *2019 International Conference on Innovative Trends and Advances in Engineering and Technology (ICITAET)* (pp. 135-138). IEEE.

[21] Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. *International Journal of Engineering & Technology*, *7*(3.24), 479-482.

[22] Mousa, A., Karabatak, M., & Mustafa, T. (2020, June). Database security threats and challenges. In *2020 8th International Symposium on Digital Forensics and Security (ISDFS)* (pp. 1-5). IEEE.

[23] Liu, M., Zhang, B., Chen, W., & Zhang, X. (2019). A survey of exploitation and detection methods of XSS vulnerabilities. *IEEE Access*, *7*, 182004-182016.

[24] Park, J., Kim, J., Gupta, B. B., & Park, N. (2021). Network Log-Based SSH Brute-Force Attack Detection Model. *CMC-COMPUTERS MATERIALS & CONTINUA*, *68*(1), 887-901.

[25] Perez, T. (2014). Understanding Denial of Service and Brute Force Attacks-WordPress, Joomla, Drupal, vBulletin.

[26] Frank, E., & Witten, I. H. (1998). Generating accurate rule sets without global optimization.

[27] Qu, Z., Li, H., Wang, Y., Zhang, J., Abu-Siada, A., & Yao, Y. (2020). Detection of electricity theft behavior based on improved synthetic minority oversampling technique and random forest classifier. *Energies*, *13*(8), 2039.

[28] Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, *114*, 24-31.

[29] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[30] Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*, *192*, 105361.

[31] Abraham, R., Simha, J. B., & Iyengar, S. S. (2006, December). A comparative analysis of discretization methods for Medical Datamining with Naïve Bayesian classifier. In *9th International Conference on Information Technology (ICIT'06)* (pp. 235-236). IEEE.

[32] Kolluri, J., & Razia, S. (2020). Text classification using Naïve Bayes classifier. *Materials Today: Proceedings*.

[33] Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2), 131-163.

[34] Alabdulwahab, S., & Moon, B. (2020). Feature selection methods simultaneously improve the detection accuracy and model building time of machine learning classifiers. *Symmetry*, *12*(9), 1424.

CURRICULUM VITAE

Omar Iskndar Ahmed

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

ACADEMIC ACTIVITIES

Paper:

1. **Ahmed, O. I., & Varol, C. (2021, June). Detection of Web Attacks via PART Classifier. In 2021 9th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-4). IEEE**.