Malicious URL Detection Using Machine Learning

by

Abdul Aleem Syed

B.Sc., University of Sindh, Pakistan, 2011

A Report Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering

Malicious URL Detection Using Machine Learning

by

Abdul Aleem Syed

B.Sc., University of Sindh, Pakistan, 2011

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor

(Department of Electrical and Computer Engineering)

Dr. T. Ilamparithi, Departmental Member

(Department of Electrical and Computer Engineering)

Abstract

The detection of malicious Uniform Resource Locators (URLs) is important for network and cyber security. The Internet has long been a platform for online criminal activity. In this project, supervised Machine Learning (ML) is employed to identify and detect malicious URLs. The ISCX-URL-2016 dataset from the Canadian Institute for Cyber Security is employed for evaluation purposes. This dataset contains 79 features with four classes of URLs, namely spam, malware, phishing, and benign.

The Waikato Environment for Knowledge Analysis (WEKA) tool is used to test and train the ML classifiers. To compare the results, *k*-fold cross-validation is used with $k = 5$ and $k = 10$. Principal Component Analysis (PCA) is employed for dimensionality reduction of the dataset and the important features selected based on the eigenvalues. The best 10 and 25 features were selected using PCA and the classifiers were trained using 5-fold and 10-fold cross-validation. The classifiers were also trained using all 79 features. The ML classifiers evaluated are Random Forest (RF), Decision Tree, K-Nearest Neighbors (KNN), Bayesian Network (BayesNet), and Simple Logistic. The performance metrics accuracy, precision, recall, f-measure, and execution time are considered. The RF classifier resulted in the highest accuracy at 98.7% with 79 features. However, in terms of execution time, KNN outperformed RF with 0.06 s for 79 features and 98.3% accuracy, which is only second to RF. In general, the results obtained show that KNN provides the best overall performance.

Contents

List of Tables

List of Figures

Abbreviations

URL	……	Uniform Resource Locator
ML	……	Machine Learning
WEKA	……	Waikato Environment for Knowledge Analysis
PCA	……	Principal Component Analysis
RF	……	Random Forest
KNN	……	K-Nearest Neighbors
BayesNet	……	Bayesian Network
HTTP	……	Hyper Text Transfer Protocol
IP	……	Internet Protocol
COVID-19	……	Coronavirus Disease 2019
DNS	……	Domain Name System
CSS	……	Cascading Style Sheets
AI	……	Artificial Intelligence
RAT	……	Remote Access Trojan
GUI	……	Graphical User Interface
API	……	Application Programming Interface
TPR	……	True Positive Rate
FPR	……	False Positive Rate

Acknowledgement

First, I am very thankful to Almighty Allah for his countless blessings upon me. I also thank my parents, my wife, and my sons for their continuous love, support, and encouragement.

I am also very thankful to Dr. T. Aaron Gulliver for accepting me into his research group as a master's student. His support and guidance are highly appreciated.

My heartfelt thanks to Dr. T. Ilamparithi for being my committee member.

I also want to thank my uncle Syed Ghulam Rasool Shah for helping and supporting me at every stage of my life.

I would also like to acknowledge my good friend Salahuddin Jokhio for valuable discussions related to this project and Atique Shaikh for helping me to apply to the University of Victoria. Furthermore, I am thankful to Sadiq Ali, Adnan Janwari, Jemma Kosalko, Muhammad Naveed, and Yasir Shah for their sincere help and motivation during the COVID-19 pandemic.

Dedication

To my Murshid Sufi Sain Ali Anwar Bhatti Naqshbandi Saifi. Thank you for your prayers, love, support, and guidance, and thanks again for believing in me.

Chapter 1 Introduction

Technology is advancing at a rapid pace and the Internet is undergoing a transformation due to these advances. The use of the Internet in social and business areas is growing at a high rate, increasing the potential for cybercrime. As connectivity and the number of users grows, so does the number of attackers. This also means the victims can be governments, industry, and individuals. Predicting future threats and their nature is a complex and difficult endeavor. In recent years, it has become more important to protect computers and devices from cyber-attacks because of the increased number of ways to attack computers.

A common cyber-attack is using a virus to infiltrate a targeted machine. A virus is a malicious program or code that changes the way computers work and spreads from one machine to another [8] [9]. There have been significant developments in both attacks and solutions to protect against them in recent years. A computer can be infiltrated through accessing malicious URLs which have content that is abnormal such as viruses, ransomware, and keyloggers. In 2021, there were a total of 12.8 million URLs detected as malicious [1][2]. Malicious URLs have been identified as a significant cyber security threat. These URLs are typically spread by email, Twitter, Facebook, popups, and website advertisements. They make it possible for unsuspecting users to become victims of attacks. When victims click on such URLs, the contained malware files are downloaded to their computers and attackers are able to steal information, personal accounts, and data, and gain control of the computers. Over 40,000 malicious URLs are generated every day resulting in a cost of $17,700 per minute [3]. In 2020, more than 80% of computers were attacked [3]. Every year, there is a loss of billions of dollars around the world due to this crisis [4].

A prevention strategy is required to detect and react to the threats from malicious URLs. Typically, detection is via blacklists. In reality, maintaining a comprehensive blacklist is unrealistic due to the billions of URLs encountered daily by cyber security engineers [3]. Malicious URLs are collected ahead of time by security engineers and end users. These URLs are blacklisted in security software and desktop applications like antivirus and anti-spyware. This is a simple way to detect malicious URLs. To address this problem on a large scale, Machine Learning (ML) techniques can be employed.

1.1 Motivation

In 2020, the COVID-19 pandemic caused most office work to be shifted to remote platforms through the internet. Malicious URLs are being used by cybercriminals to take advantage of this situation [42]. URLs can contain malware and spyware. Spam emails can also be used to deceive users into clicking on malicious URLs. Some URLs may be authentic while others are used for phishing and spam attacks. ML is one of the most rapidly expanding and effective areas of technology in the modern world. With the use of existing data, analysis using ML can help identify future outcomes. This provides opportunities to predict important things like the weather and game results [9]. As the use of technology grows, it is more important to protect it as it is connected to our livelihoods. With the power of prediction and connection, we can better combat threats to health and security.

1.2 Related Work

Malicious URLs are an urgent threat to computers and are growing at a rapid pace. Many studies for detecting and identifying malicious URLs have been proposed in recent years to detect and prevent malicious URL attacks [9]. Malicious URLs are easily shared these days due to social hacking and social networking. To avoid these risks, there should be a mechanism in place that can automatically detect harmful URLs before a user clicks on them and a warning should pop up on the machine indicating potential threats. In general, such URLs are blacklisted in utility software, however, in the event of a new URL, utility software is unable to identify if it is harmful. As a result, there is a need for a system that can detect a new malicious URL. Machine learning techniques can be used to protect against these threats [10].

Several methods have been proposed for malicious URL detection. In [39], the analysis of phishing and benign URLs was evaluated. A URL is blacklisted according to its malicious behavior. For example, a red flag keyword such as ebayisapi is generated for eBay sites for phishing attacks. In [40], data from Twitter spam messages and emails was considered, and features collected from areas such as network traffic, webpage content, lexicon, DNS, and link popularity. ML has been considered with lexical analysis, but the accuracy in detecting malware and spam URLs was poor. ML has been applied for the detection of 2 million malicious URLs [41]. The results showed that 9% of the malicious attributes were missed because only descriptive features were used.

In this project, ML is employed using the WEKA tool. This tool provides a set of visualization techniques and methodologies for data analysis and predictive modeling [11]. It has numerous ML classifiers which can be used to detect malicious URLs. The ISCX-URL-2016 dataset is used for evaluation purposes. The ML classifiers considered in this project are RF, KNN, Decision Tree, BayesNet, and Simple Logistic.

1.3 Report Outline

The structure of this report is as follows.

Chapter 1 introduced malicious URLs and the types of malicious URLs. The project motivation and related work were also presented.

Chapter 2 presents the classification of malicious URLs and blacklisting methods. The types of URL attacks and ML methods for the prevention of malicious URL attacks are also discussed.

Chapter 3 introduces ML with a brief explanation of the classifiers used in this work. A discussion of the WEKA tool is given as it is employed for detecting malicious URLs.

Chapter 4 describes the method used to detect malicious URLs as well as the test environment. The hardware and software configurations and the performance metrics are presented along with a discussion of the results.

Chapter 5 presents the conclusion and suggestions for future work.

Chapter 2 URL Classification and Malicious Attacks

A URL is known as a specific unique resource on the Internet [12]. URLs are associated with resources such as HTML pages, CSS documents, and images. There are a few exceptions where resources either do not exist or have been moved from the servers [12].

2.1 URL Classification

A URL identifies and locates a web resource. The type of protocol, source domain, top, second, and third-level domains, primary domains, and pathways are the components that comprise a URL. The complexity of a URL is determined by the resource being referenced, as well as how and where it is located. URLs are used to gain access to the worldwide web. Figure 2.1 shows the most important components of a URL [6].

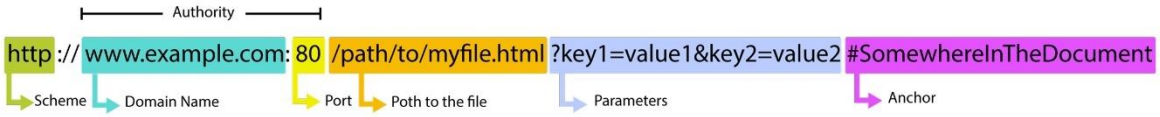

Figure 2.1: The URL components [6].

2.1.1 Scheme

Scheme is the first component of a URL. It specifies the protocol that the browser utilizes for resource requests. A protocol is a set of instructions for data exchange or transfer over the internet [5]. Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS) are the most common protocols for webpages. HTTP is an unsecured version and HTTPS is a secured version. Browsers can also employ other schemes, for example mailto: to launch a mail client [6][7].

2.1.2 Authority

The authority component is followed by the pattern :// [5]. When the domain, e.g. www.example.com and port, e.g. 80 exist, the authority divides them with a colon (:). The domain identifies the server being accessed. This is usually a domain name but can also be an IP address.

The port is the logical gate that allows access to the server resources. The server grants access to its resources using conventional HTTP ports (80 for HTTP and 443 for HTTPS) [6][7].

2.1.3 Path

A path locates physical files on the server through its location as shown in Figure 1.1, e.g. /path/to/myfile.html [5][6].

2.1.4 Parameters

Parameters are extra values in a URL as shown in Figure 1.1, e.g. ?key1=value1&key2=value2. The symbol & is used to separate key and value pairs. The server can utilize parameters to do further tasks. Every server is unique in terms of the rules to handle parameters [5].

2.1.5 Anchor

The anchor is the last component of the URL as shown in Figure 1.1, e.g. #SomewhereInTheDocument. It is a link to a different section of the document. An anchor acts as a bookmark within the resource, instructing the browser to display the content at the bookmarked location [5].

2.2 Spam URLs

Spam URLs can spread through a variety of channels including emails, texts, and social media platforms. Social media is an easy and common channel for spammers and fraudsters. For a successful attack, personal information is often required, and it is easier to collect such information using these channels. This could make it more likely for a user to click on an unknown URL [17][18].

2.2.1 URL Shortening

URL shortening facilities, such as Bitly, Google URL shortener, Is Good, and TinyURL, are popular spam masking methods [18]. For link sharing, an attacker may create many short versions of a long URL. Spam attackers use URL shortening to hide the true landing page of a malicious URL. Shortening a URL is a common approach, however social media platforms rarely detect and block them. Figure 2.2 gives an example of URL shortening.

Figure 2.2: An example of URL shortening [18].

2.3 Malware URLs

A malware URL is a link that can take a user to a false webpage or website. The goal of creating malicious webpages is to carry out an attack agenda, which can be a political agenda, or to steal personal or organizational data [19]. Actions such as simply clicking on a malicious URL, trying to open an attached file, or trying to engage with an advertisement can have significant effects. Opening a malicious URL may download the payload to the machine. The payload contains malicious code which can harm the computer and compromise the data.

2.4 Phishing URLs

Phishing is a type of social engineering attack that seeks to trick people into giving up personal information. Attackers focus on user personal details such as bank information, corporate data, login credentials, and anything valuable. Due to a lack of security awareness, organizations can have vulnerabilities. Attackers can find vulnerable people to infiltrate organizations using phishing attacks. One successful phishing attack on an employee can put an entire corporation in jeopardy. A solution to this problem is to effectively train users to identify malicious webpages.

2.5 Blacklisting and Heuristic Techniques

In general, there are two approaches for classifying URLs, blacklisting and heuristic techniques. These methods rely on database lookup to allow or restrict good or bad URLs, respectively. A large database of blacklisted URLs is maintained which is acquired from trustworthy sources. As a result, when a new URL is added to the list, the utility software checks the database to see if it is in the list. If the URL matches one in the list, the user will be notified of a potential threat, otherwise

it will be regarded as non-malicious or benign. These traditional methods take significant time, and it is hard to keep track of the URLs, especially the ones which have been shortened [13][14].

2.6 Machine Learning Techniques

Machine Learning (ML) techniques learn URL patterns using information gathered in a variety of ways. Feature extraction methods can be static or dynamic. Static features are typically collected from graphical images of webpages, URL strings, and scripting languages such as HTML and JavaScript [15]. Dynamic feature extraction is done by monitoring the dynamic behavior of the system for anomalous activity. This is accomplished by looking for unusual or abnormal behavior in the system logs and sequence calls. Because the systems are vulnerable to attacks, dynamic feature extraction methods are difficult to generalize and implement [15][16]. ML techniques can be used to solve this problem.

2.7 Proposed Framework

Figure 2.1 gives the proposed framework for training and testing the ML models. The first step is to preprocess the ISCX-URL-2016 dataset. The WEKA tool is employed for preprocessing and building the ML models.

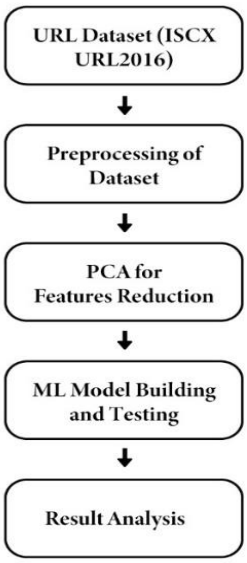

Figure 2.3: The proposed framework.

2.8 Dataset

The ISCX-URL-2016 dataset contains four different types of malicious URLs, namely spam, malware, phishing, and benign. About 12,000 spam URLs were collected from WEBSPAM–UK2007, about 10,000 phishing URLs from the OpenPhish repository, 35,300 benign URLs from Alexa.com, and over 11,500 malware URLs from the DNS-BH malware site list [25].

2.9 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is used to reduce the dimensionality of features in a dataset. It is commonly used to convert a large collection of variables into a smaller dataset. ML models are more efficient at exploring and visualizing smaller datasets with extraneous features removed [29]. In this project, the WEKA tool is first used to standardize the dataset features. Then the correlation matrix is obtained to determine the relationship between the features. Eigen decomposition is then used to obtain the eigenvectors and eigenvalues. The eigenvalues are the variances of the components, whereas the eigenvectors are the principal components. They are then sorted in descending order so the eigenvector with the highest eigenvalue is the first principal component of the dataset [29]. The less important components with smaller eigenvalues are eliminated. In this project, the top 10 and 25 features were chosen based on their eigenvalues for evaluation purposes [38].

2.10 Feature Selection Using PCA

Table 2.1 lists the 50 most important features selected using PCA. The top 10 and 25 features will be used to train the classifiers.

Number	Feature Name	Eigenvalue
1	ArgLen	21.67
2	SymbolCount_URL	9.216
3	DomainUrlRatio	7.906
4	NumberRate_DirectoryName	4.917
5	Directory_LetterCount	4.604
6	Domain_LongestWord	3.753
7	Ldl_Filename	3.459
8	Host_DigitCount	2.589
9	Ldl_Domain	2.521
10	URL_Type_obf_TypeMalware	1.817
11	Path_LongestWordLength	1.585
12	Filename_LetterCount	1.454
13	File_Name_DigitCount	1.362
14	URL_SensitiveWord	1.184
15	Delimeter_Domain	1.104
16	Executable	1.024
17	Executable_Delimeter	0.998
18	PortEighty	0.995
19	URL_SensitiveWord	0.934
20	Arguments_Longest	0.887
21	URL_Type_obf_Type	0.802
22	Avgpathtokenlen	0.739
23	Entropy	0.651
24	Dld_GetArg	0.633
25	Entropy_Domain	0.553
26	LongestWordLength	0.477
27	NumberRate_DirectoryName	0.422
28	Entropy_Afterpath	0.351
29	This.fileExtLen	0.309
30	ArgPathRatio	0.284
31	LargPathRatio	0.273
32	Entropy_URL	0.243
33	URLQueries_variable	0.226
34	Sub-Directory_LongestWord	0.199
35	PathurlRatio	0.190
36	LetterCount	0.164
37	FileNameLen	0.148
38	SpcharUrl	0.128

39	Ldld_Url	0.115
40	Charcompace	0.106
41	Query_DigitCount	0.092
42	Entropy_Filename	0.086
43	DigitCount	0.081
44	NumberRate_Domain	0.071
45	dld_getArg_Entropy	0.066
46	Query_DigitCount	0.061
47	SymbolCount	0.055
48	Longdomaintokenlen	0.053
49	Longdomain	0.048
50	NumberRate	0.045

Table 2.1: The 50 most important features selected using PCA.

Chapter 3 Machine Learning

Machine Learning (ML) is an area of Artificial Intelligence (AI). ML is a data analysis technique that automatically forms analytical models. These models can learn from data, recognize patterns, and make decisions with minimal human intervention [22][23]. The most important task in ML is feature selection. As ML algorithms are developed based on the results of training data, they are non-interactive so previous observations are used to make predictions. Accurate prediction can be a challenging task [24]. In this work, six ML classifiers are employed for malicious URL detection [25]. ML classifiers can be divided into two categories, namely supervised learning and unsupervised learning [26][33], as described below.

3.1 Supervised Learning

Supervised ML is used with labeled datasets. This data is used to train the model and predict the outcome. The outcome is usually a class or value. Supervised learning can solve a variety of complex problems, for example identifying and classifying viruses or spam emails in an inbox. Random Forest (RF), Logistic Regression, Neural Networks, Linear Regression, Naive Bayes, Support Vector Machine (SVM), are examples of supervised ML classifiers.

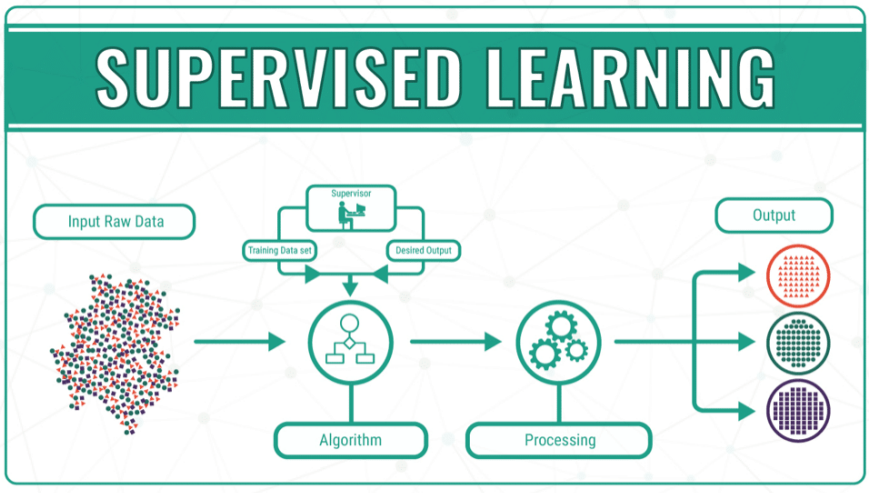

Figure 3.1: Supervised machine learning model [26].

3.2 Unsupervised Learning

Unsupervised ML algorithms are used with unlabeled datasets. Hidden patterns can be detected by these algorithms with no human interference. Due to their ability to identify differences and resemblances in data, they are commonly used in data analysis, product selling strategies, pattern recognition, and customer segmentation. Unsupervised learning is also used for feature extraction via dimensionality reduction. Unsupervised ML algorithms include K-means clustering and probabilistic clustering [28].

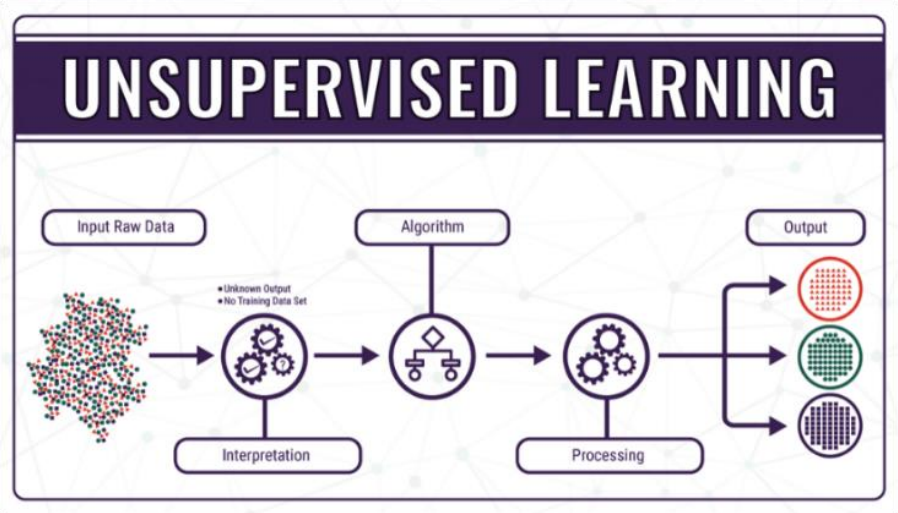

Figure 3.2: Unsupervised machine learning model [28].

3.3 Machine Learning Classifiers

In this project, five ML classifiers are employed for data classification, namely Random Forest (RF), Decision Tree, Simple Logistic, K-Nearest Neighbors (KNN), and Bayes Network (BayesNet). These classifiers are widely used in practice for supervised learning purposes. In addition, these classifiers have been shown to perform well for such classification problems.

3.3.1 Random Forest

Random Forest (RF) is a supervised learning classifier that has been extensively employed for classification and regression problems. RF constructs decision trees from samples collected and averages the results to improve the prediction accuracy. The data in continuous variables in regression and categorical variables in classification are managed by the RF. RF has been shown to have fewer classification errors than other algorithms with imbalanced datasets [30].

3.3.2 K-Nearest Neighbors (KNN)

KNN can be used to solve classification and regression problems. KNN calculates the distances of all points near the unknown data and chooses the shortest distances. Thus, it is also known as a distance-based algorithm. Training KNN classifiers can be slow with missing values. KNN can also be computationally expensive in terms of both time and storage if the dataset is very large. This is generally not the case with other supervised learning models [31].

3.3.3 Bayesian Network (BayesNet)

A Bayesian network classifier considers a collective probability model to solve complex problems. The network is composed of nodes and their causal relation which represent edges and random variables, respectively. The goal is to provide information from random variables related to the nodes and the statistical probabilities related to the edges. Bayesian networks are very good at modeling probabilistic relations and predicting the probabilities that possible known causes are contributing factors.

3.3.4 Simple Logistic

The simple logistic classifier is based on linear logistic regression. It is commonly used for binary classification due to its simplicity. Simple logistic performs well when the data is associated

linearly, but performs badly when the relationship is complex and nonlinear. It also struggles with datasets having missing values [33].

3.3.5 Decision Tree

Decision tree classifiers are employed for classification and regression. They can predict the target value of a variable from the data features using simple decision rules. Small variations in the data can result in unstable decision trees, but this can be mitigated by using ensembled decision trees [34][35].

3.4 Data Splitting

The choice of data splitting for training and testing is an important decision. The most common strategies are *k*-fold cross-validation and percentage split. Percentage split is a simple way to split data into training and test sets with 80:20 and 70:30 ratios often used. In this work, *k*-fold cross-validation is used with $k = 5$ and 10. It has been shown to produce less biased results than percentage split [35]. A cross-validation model is trained using *k* partitions with one partition used as the test set. The cross-validation accuracy is the average of the results for the *k* different test partitions [37]. Figure 3.3 illustrates the partitions for *k*-fold cross-validation with $k = 5$.

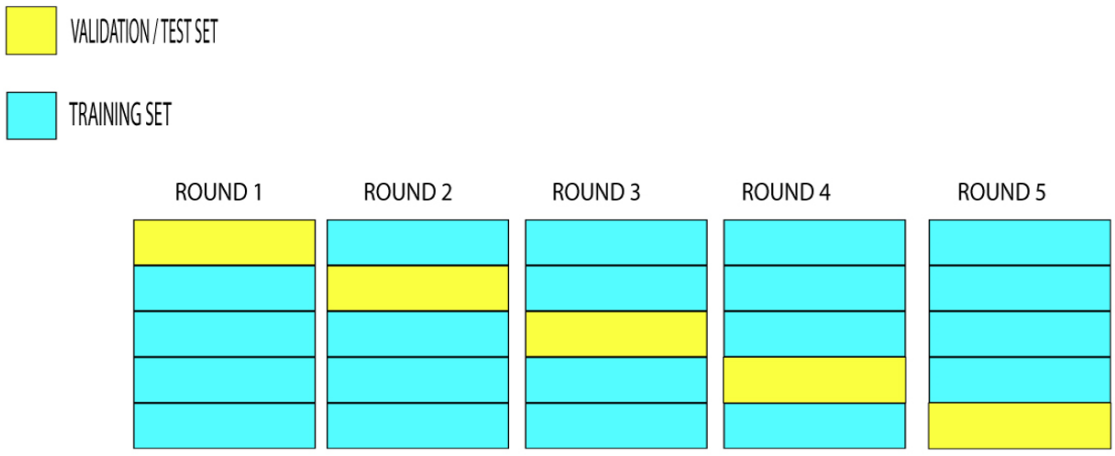

Figure 3.3: Illustration of *k*-fold cross-validation with $k = 5$.

3.5 The WEKA Machine Learning Tool

WEKA is a free and open-source ML tool developed on the Java platform. It is accessible via a Graphical User Interface (GUI), standard terminal applications, or a Java Application Programming Interface (API). It was developed at the University of Waikato in New Zealand and has since been widely used in academic, scientific, and industrial applications. It includes supervised, semi-supervised, and unsupervised ML classifiers like Random Forest, Linear Regression, KNN, BayesNet, and Decision Tree. These classifiers can be tuned by changing their parameters (known as hyperparameters), making this tool extremely powerful. Tuning can improve classifier accuracy but this is often done experimentally as it is highly dependent on the ML problem. The WEKA tool GUI is shown in Figure 3.4.

Figure 3.4: The WEKA tool GUI.

WEKA Explorer is used to enter the dataset. The dataset is chosen from the Preprocess panel. Various formats including .csv and .arff are supported. A snapshot of the WEKA explorer preprocess panel is shown in Figure 3.5. It provides dataset information such as the number of features and classes. Figure 3.6 shows dataset visualization in WEKA.

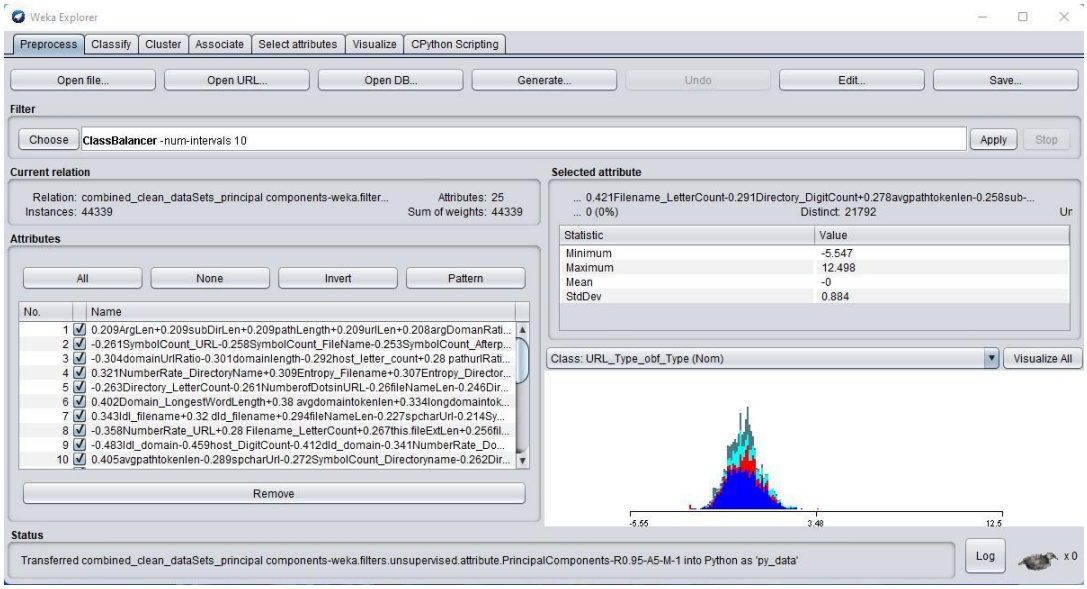

Figure 3.5: The WEKA Explorer preprocess panel.

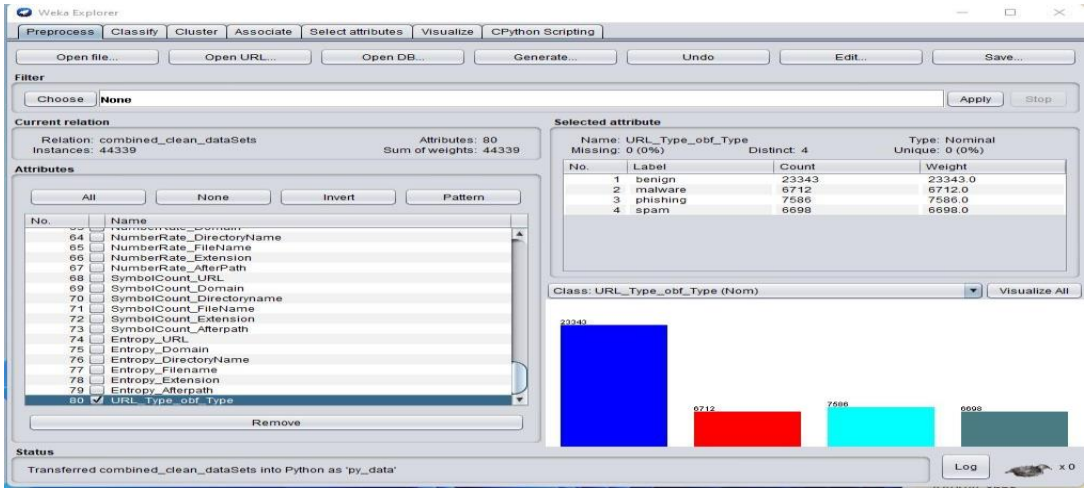

Figure 3.6: The dataset classes visualized in WEKA.

Chapter 4 Performance Evaluation

In this chapter, the performance results are presented and discussed. Five ML classifiers with default parameters are used namely RF, Decision Tree, Simple Logistic, KNN (K = 1), and BayesNet. The results were obtained using a personal computer with the hardware and software specifications given in Table 4.1. The ISCX-URL-2016 dataset of malicious URLs is employed to evaluate the ML models. There are four classes of URLs including malicious and benign. Figure 4.1 shows the number of instances in the classes. After processing, the number of benign URLs is 23343, malware URLs is 6712, phishing URLs is 7586, and spam URLs is 6698.

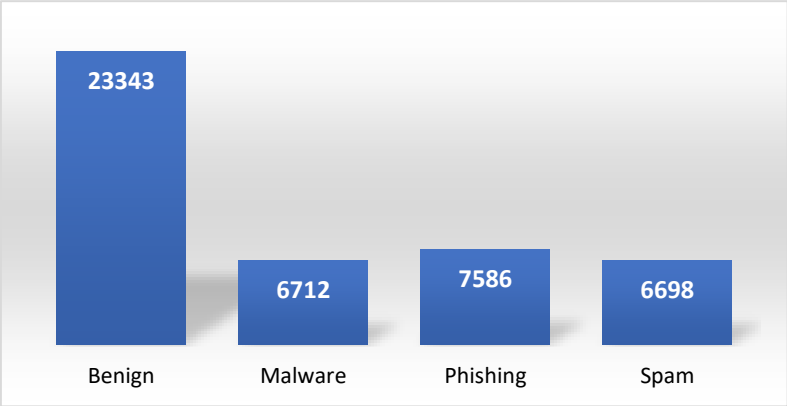

Figure 4.1: The number of instances in the four classes.

Item	Specification
Model	Acer Aspire E5-575G
Operating System	Windows 10 Professional
System Type	64-bit Operating System, x64-based Processor
Processor Type	Intel (R) Core (TM) i5-7200 CPU
Installed Memory (RAM)	16 GB
Processor Speed	2.50 GHz
Number of Cores	2
Number of Threads	4
Machine Learning Tool	WEKA Version 3.9.4

Table 4.1: The hardware and software specifications.

4.1 Evaluation Metrics

The performance metrics used are as follows.

Precision is the ratio of true positive to the sum of false positive and true positive

$$\frac{tp}{tp + fp}$$

where true positive (tp) is the number of malicious URLs correctly classified and false positive (fp) is the number of URLs incorrectly classified.

Recall is the ratio of true positive to the sum of false negative and true positive

$$\frac{tp}{tp + fn}$$

where false negative (fn) is the number of incorrectly classified URLs.

Accuracy is the number of correct classifications of either malicious or benign URLs out of all URLs in the dataset

$$\frac{tn + tp}{tn + tp + fn + fp}$$

where true negative (tn) is the number of correct classifications of benign as benign.

F-Measure is the harmonic mean of recall and precision

$$\frac{2tp}{2tp + fp + fn}$$

Execution Time is the time required to train and test the classification model.

The accuracy, precision, recall, and F-measure are all expressed as percentages in this chapter.

4.2 Performance of the Classifiers with 10-Fold Cross-validation Using 79 Features

In this section, the dataset is trained and tested with all 79 features using 10-fold cross-validation. Table 4.2 gives the test results for the ML classifiers. This shows that RF has the highest accuracy, precision, recall, and F-measure at 98.7, 99.1, 99.9, 99.5, followed by KNN at 98.3, 98.3, 98.3, 98.3, Decision Tree at 97.8, 97.9, 97.9, 97.9, BayesNet at 91.4, 91.8, 91.4, 91.3, and Simple Logistic at 89.7, 89.5, 89.7, 89.4. In terms of execution time, KNN had the lowest at 0.06 s followed by BayesNet at 2.92 s, Decision Tree at 9.04 s, Random Forest at 24.24 s, and Simple Logistic at 126.2 s.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	Execution time (s)
RF	98.7	99.1	99.9	99.5	24.24
Decision Tree	97.8	97.9	97.9	97.9	9.04
KNN	98.3	98.3	98.3	98.3	0.06
BayesNet	91.4	91.8	91.4	91.3	2.92
Simple Logistic	89.7	89.5	89.7	89.4	126.2

Table 4.2: Performance of the ML classifiers with 10-fold cross-validation using 79 features.

4.3 Performance of the Classifiers with 10-Fold Cross-validation Using 25 Features

In this section, the dataset is trained and tested with 25 features using 10-fold cross-validation. Table 4.3 gives the test results for the ML classifiers. This shows that KNN has the highest accuracy, precision, recall, and F-measure at 98.3, 98.0, 98.0, 98.0, followed by RF at 98.1, 98.0, 98.0, 98.0, Decision Tree at 96.4, 96.4, 96.5, 96.4, BayesNet at 87.8, 88.5, 87.1, 86.9, and Simple Logistic at 83.4, 82.8, 83.4, 82.4. In terms of execution time, KNN had the lowest at 0.02 s followed by BayesNet at 2.68 s, Decision Tree at 5.46 s, Simple Logistic at 30.77 s, and RF at 38.52 s.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	Execution time (s)
RF	98.1	98.0	98.0	98.0	38.52
Decision Tree	96.4	96.4	96.5	96.4	5.46
KNN	98.3	98.0	98.0	98.0	0.02
BayesNet	87.8	88.5	87.1	86.9	2.68
Simple Logistic	83.4	82.8	83.4	82.4	30.77

Table 4.3: Performance of the ML classifiers with 10-fold cross-validation using 25 features.

4.4 Performance of the Classifiers with 10-Fold Cross-validation Using 10 Features

In this section, the dataset is trained and tested with 10 features using 10-fold cross-validation. Table 4.4 gives the test results for the ML classifiers. This shows that KNN has the highest accuracy, precision, recall, and F-measure at 97.6, 97.6, 97.6, 97.6, followed by RF at 97.3, 97.3, 97.3, 97.3, Decision Tree at 94.7, 94.7, 97.8, 94.7, BayesNet at 75.9, 77.1, 75.9, 75.6, and Simple Logistic at 56.5, 54.6, 56.6, 50.0. In terms of execution time, KNN had the lowest at 0.01 s followed by BayesNet at 0.36 s, Decision Tree at 1.63 s, Simple Logistic at 6.15 s, and RF at 18.05 s.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	Execution time (s)
RF	97.3	97.3	97.3	97.3	18.05
Decision Tree	94.7	94.7	97.8	94.7	1.63
KNN	97.6	97.6	97.6	97.6	0.01
BayesNet	75.9	77.1	75.9	75.6	0.36
Simple Logistic	56.5	54.6	56.6	50.0	6.15

Table 4.4: Performance of the ML classifiers with 10-fold cross-validation using 10 features.

4.5 Performance of the Classifiers with 5-Fold Cross-validation Using 79 Features

In this section, the dataset is trained and tested with all 79 features using 5-fold of cross-validation. Table 4.5 gives the test results for the ML classifiers. This shows that RF has the highest accuracy, precision, recall, and F-measure at 98.6, 98.6, 98.6, 98.6, followed by KNN at 98.1, 98.1, 98.2, 98.1, Decision Tree at 97.4, 97.4, 97.5, 97.4, BayesNet at 91.2, 91.6, 91.2, 91.2, and Simple Logistic at 89.7, 89.5, 89.7, 89.4. In terms of execution time, KNN had the lowest at 0.02 s followed by BayesNet at 4.41 s, Decision Tree at 19.59 s, Random Forest at 43.73 s, and Simple Logistic at 185.3 s.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	Execution time (s)
RF	98.6	98.6	98.6	98.6	43.73
Decision Tree	97.4	97.4	97.5	97.4	19.59
KNN	98.1	98.1	98.2	98.1	0.02
BayesNet	91.2	91.6	91.2	91.2	4.41
Simple Logistic	89.7	89.5	89.7	89.4	185.3

Table 4.5: Performance of the ML classifiers with 5-fold cross-validation using 79 features.

4.6 Performance of the Classifiers with 5-Fold Cross-validation Using 25 Features

In this section, the dataset is trained and tested with 25 features using 5-fold cross-validation. Table 4.6 gives the test results for the ML classifiers. This shows that RF has the highest accuracy, precision, recall, and F-measure at 97.9, 98.4, 99.9, 99.2, followed by KNN at 97.8, 98.8, 99.9, 99.4, Decision Tree at 95.9, 97.9, 99.4, 98.7, BayesNet at 86.9, 88.9, 96.0, 92.3, Simple Logistic at 83.4, 82.8, 96.3, 82.5. In terms of execution time, KNN had the lowest at 0.01 s followed by BayesNet at 1.88 s, Decision Tree at 5.54 s, Simple Logistic at 46.82 s, and RF at 40.47 s.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	Execution time (s)
RF	97.9	98.4	99.9	99.2	40.47
Decision Tree	95.9	97.9	99.4	98.7	5.54
KNN	97.8	98.8	99.9	99.4	0.01
BayesNet	86.9	88.9	96.0	92.3	1.88
Simple Logistic	83.4	88.2	96.3	82.5	46.82

Table 4.6: Performance of the ML classifiers with 5-fold cross-validation using 25 features.

4.7 Performance of the Classifiers with 5-Fold Cross-validation Using 10 Features

In this section, the dataset is trained and tested with 10 features using 5-fold cross-validation. Table 4.7 gives the test results for the ML classifiers. This shows that RF has the highest accuracy, precision, recall, and F-measure at 97.1, 97.2, 97.2, 97.1, followed by KNN at 96.8, 96.9, 96.9, 96.8, Decision Tree at 94.1, 94.0, 94.1, 94.0, BayesNet at 75.4, 74.5, 76.6, 75.5, and Simple Logistic at 56.7, 54.7, 56.7, 50.2. In terms of execution time, KNN had the lowest at 0.01 s followed by BayesNet at 0.71 s, Decision Tree at 3.43 s, Simple Logistic at 12.05 s, and RF at 33.45 s.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	Execution time (s)
RF	97.1	97.2	97.2	97.1	33.45
Decision Tree	94.1	94.0	94.1	94.0	3.43
KNN	96.8	96.9	96.9	96.8	0.01
BayesNet	75.4	76.6	75.4	75.51	0.71
Simple Logistic	56.7	54.7	56.7	50.2	12.05

Table 4.7: Performance of the ML classifiers with 5-fold cross-validation using 10 features.

4.8 Discussion

RF and KNN performed better than the other classifiers in terms of accuracy, precision, recall, and F-measure with both 5-fold and 10-fold cross-validation. However, in most cases KNN was the fastest classifier in terms of execution time, whereas RF and Simple Logistic were the slowest classifiers.

In 10-fold cross-validation with 79 features, RF has the highest accuracy at 98.7% and execution time at 24.24 s followed by KNN having accuracy at 98.3%, whereas in terms of execution time, KNN was only 0.06 s (Table 4.2). Simple Logistic was the slowest classifier with execution time of 126.19 s and accuracy of 89.7% (Table 4.2). Decision Tree performed better than Simple Logistic and BayesNet with accuracy of 97.8%, but the execution time was higher at 9.04 s (Table 4.2).

In 5-fold cross-validation with 79 features, RF has the highest accuracy at 98.6% and execution time at 43.73 s, followed by KNN with accuracy at 98.1%, whereas in terms of execution time, KNN was only 0.02 s (Table 4.5). Simple Logistic was the slowest classifier with execution time of 185.3 s and accuracy of 89.7% (Table 4.5). Decision Tree accuracy is better than Simple Logistic at 97.4%, but the execution time is higher at 19.59 s, followed by BayesNet with accuracy at 91.2% and execution time at 4.41 s (Table 4.5). The results discussed above are illustrated in Figure 4.2 which shows that 10-fold cross-validation provides better accuracy than 5-fold cross-validation for all five classifiers

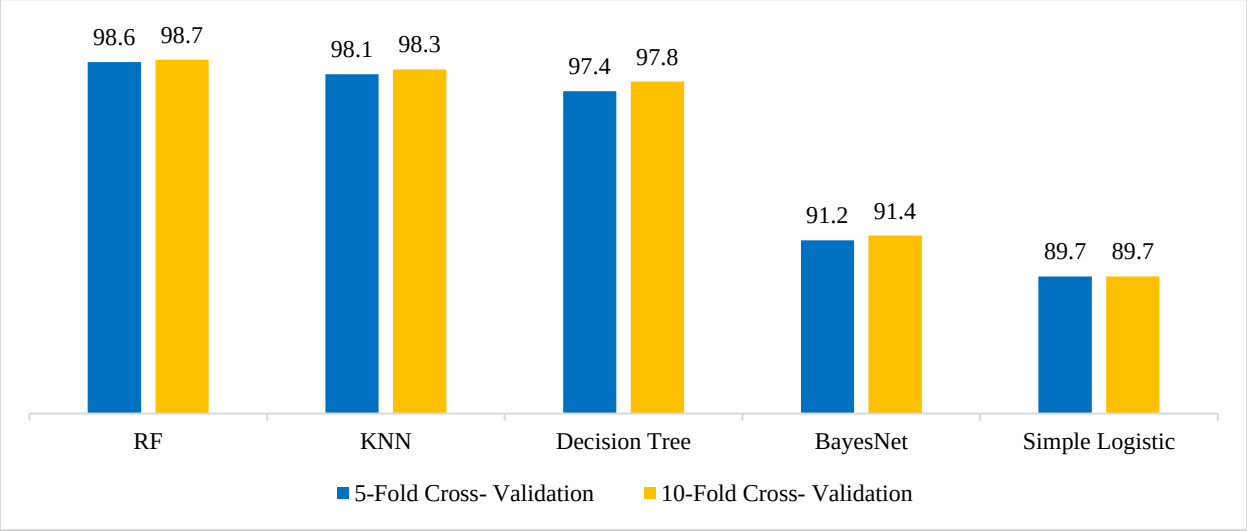

Figure 4.2: 10-fold and 5-fold cross-validation accuracy with 79 features.

In 10-fold cross-validation with 25 features, KNN has the highest accuracy at 98.3%, followed by RF at 98.1% (Table 4.3), whereas in terms of execution time, KNN was only 0.02 s versus 38.52 s. Simple Logistic was the second slowest classifier with execution time of 30.77 s and accuracy of 83.4% (Table 4.3). Decision Tree accuracy is better than Simple Logistic at 96.4%, but the execution time was higher at 5.46 s. BayesNet accuracy is at 87.8% but the execution time is 2.68 s (Table 4.3).

In 5-fold cross-validation with 25 features RF has the highest accuracy at 97.9% followed by KNN at 97.8%, whereas in terms of execution time, KNN outperformed RF at 0.01 s versus 40.47 s (Table 4.6). Simple Logistic was the second slowest classifier with execution time of 46.82 s and accuracy of 83.4% (Table 4.6). The accuracy of Decision Tree is 95.9% but the execution time is 5.54 s, followed by BayesNet, whose accuracy is 86.9% and execution time is 1.88 s (Table 4.6). The results discussed above are illustrated in Figure 4.3 which shows that 10-fold cross-validation provides better accuracy than 5-fold cross-validation for all five classifiers.

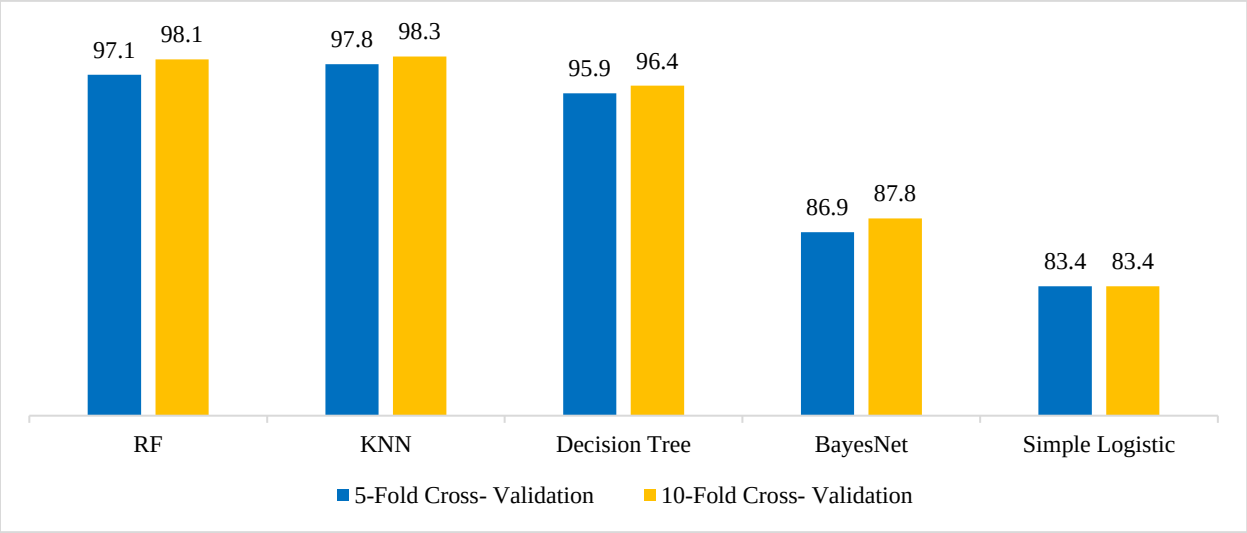

Figure 4.3: 10-fold and 5-fold cross-validation accuracy with 25 features.

In 10-fold cross-validation with 10 features, KNN has the highest accuracy at 97.6%, followed by RF at 97.3% (Table 4.4), whereas in terms of execution time KNN was better at 0.01 s versus 18.05 s. Simple Logistic has the lowest accuracy at 56.5% and execution time at 6.15 s. (Table 4.4). Decision Tree accuracy is better than BayesNet at 94.7%, but the execution time was higher at 1.63 s, followed by BayesNet with accuracy at 75.9% which outperformed Decision Tree with execution time at 0.36 s (Table 4.4).

In 5-fold cross-validation with 10 features, RF has the highest accuracy at 97.1% followed by KNN at 96.8%, whereas in terms of execution time, KNN is better at 0.01 s versus 33.45 s (Table 4.7). Simple Logistic was the second slowest classifier with execution time of 12.05 s and accuracy of 56.7% (Table 4.7). Decision Tree accuracy is better than Simple Logistic at 94.1% but the execution time is higher at 3.43 s. BayesNet has accuracy of 75.4%, while the execution time of 0.71 s is lower than that of RF, Decision Tree, and Simple Logistic (Table 4.7). The results discussed above are illustrated in Figure 4.4 which shows that 10-fold cross-validation provides better accuracy than 5-fold cross-validation for all five classifiers.

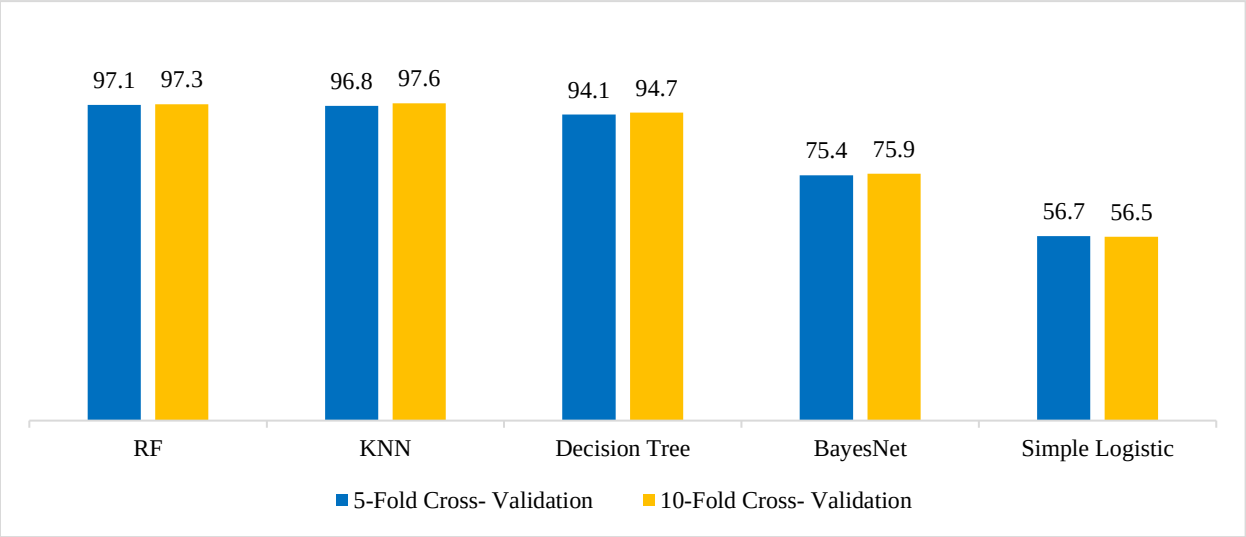

Figure 4.4: 10-fold and 5-fold cross-validation accuracy with 10 features.

Chapter 5 Conclusion and Future Work

This project considered ML using the Waikato Environment for Knowledge Analysis (WEKA) tool for malicious URL detection. The ISCX-URL-2016 dataset was used for evaluation. It contains four classes of URLs, namely spam, malware, phishing, and benign. The dimensionality of the dataset was reduced using PCA. Five supervised ML classifiers were considered, namely RF, Decision Tree, KNN, BayesNet, and Simple Logistic. Accuracy, precision, recall, F-measure, and execution time were used as performance metrics. 5-fold and 10-fold cross-validation were used with the 79 original features as well as 10 and 25 features selected using PCA.

Among the results, RF using 79 features and 10-fold cross-validation provides the highest accuracy at 98.7% with an execution time of 24.24 s. However, KNN using 25 features and 10-fold cross-validation had the second highest accuracy at 98.3% and an execution time of only 0.02 s. In general, KNN provides the best combination of accuracy and execution time. It is also evident from Table 4.4 that 10-fold cross-validation using 10 features provides the best balance between high accuracy and low execution time. 5-fold cross-validation using 10 features had the lowest accuracy in general for the classifiers. Overall, 10-fold cross-validation provided better performance than 5-fold cross-validation.

For future work, other datasets can be used to evaluate model performance. Deep learning techniques can also be utilized for analysis. Instead of using *k*-fold cross-validation, splitting methods can be employed to obtain training and testing sets. Unsupervised ML can also be considered to find patterns and similarities between different URL types.

Bibliography

[1] C. Jones, 50 Web Security Stats You Should Know in 2022, 2002, https://expertinsights.com/insights/50-web-security-stats-you-should-know/.

[2] M. S. I. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, Detecting Malicious URLs Using Lexical Analysis, Network and System Security, Springer, Berlin, pp. 467-482, 2016.

[3] R. Patgiri, H. Katari, R. Kumar, and D. Sharma, Empirical Study on Malicious URL Detection Using Machine Learning, International Conference on Distributed Computing and Internet Technology, pp. 380-388, Bhubaneswar, India, 2019.

[4] A. S. Raja, R. Vinodini, and A. Kavitha, Lexical Features Based Malicious URL Detection Using Machine Learning Techniques, International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems, pp. 163-166, Chennai, India, 2021.

[5] I. Bouarfa, Malicious URL Detection with Machine Learning, 2019, https://medium.com/@ismaelbouarfa/malicious-url-detection-with-machine-learning-d57890443dec.

[6] C. D. Mills, M. Fuji, and D. Belt, What is a URL, 2021, https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL.

[7] R. Petke, Registration Procedure for URL Scheme Names, UUNET Technologies, 1999, https://datatracker.ietf.org/doc/html/rfc2717.

[8] J. Acharya, A. Chuadhary, A. Chhabria, and S. Jangale, Detecting Malware, Malicious URLs and Virus Using Machine Learning and Signature Matching, International Conference for Emerging Technology, Belagavi, India, 2021.

[9] O. V. Lee, H. Ahmad, F. A. Mohd, M. Raffei, A. Farihan, E. P. Nincarean, K. Shahreen, and T. Sutikno, A Malicious URLs Detection System Using Optimization and Machine Learning Classifiers, Indonesian Journal of Electrical Engineering and Computer Science, vol. 17, no. 3, pp. 1210-1214, 2020.

[10] M. Yeo, Y. Koo, Y. Yoon, T. Hwang, J. Ryu, and J. Song, Flow-based Malware Detection Using Convolutional Neural Network, International Conference on Information Networks, pp. 910-913, Chiang Mai, Thailand, 2018.

[11] R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, and A. Seewald, WEKA Tool, 2013, https://statweb.stanford.edu/~lpekelis/13_datafest_cart/WekaManual-3-7-8.pdf.

[12] R. Ramakrishnan, URL Feature Engineering and Classification, 2021, https://medium.com/nerd-for-tech/url-feature-engineering-and-classification-66c0512fb34d.

[13] R. S. Rao, S. T. Ali, and P. Shield, A Desktop Application to Detect Phishing Webpages Through Heuristic Approach, International Conference on Communication Networks, pp. 147-156, Bangalore, India, 2015.

[14] D. Sahoo, C. Liu, and S. Hoi, Malicious URL Detection Using Machine Learning, 2019, https://www.scribd.com/document/513730585/1-MaliciourURLDetectionMachineLearning.

[15] B. Eshete, A. V. Fiorita, and K. Weldemariam, Holistic Analysis and Detection of Malicious Webpages, International Conference on Security and Privacy in Communication Networks, pp 149-166, Padua, Italy, 2012.

[16] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, Learning to Detect Malicious Web Sites from Suspicious URLs, International Conference on Knowledge Discovery and Data Mining, pp. 1245-1254, Paris, France, 2009.

[17] N. Gupta, A. Aggarwal, and P. K. Guru, Bit.ly/malicious: Deep Dive into Short URL Based e-crime Detection, Symposium on Electronic Crime Research, pp. 14-24, Birmingham, AB, USA, 2014.

[18] S. Chandran, How Spammer Conduct Mass URL Spam Attack, 2018, https://www.datavisor.com/blog/attack-technique-how-attackers-use-link-shorteners-to-spread-url-spam/.

[19] Savvy Security, What is a Malicious URL and How You Can Avoid Them, 2021, https://cheapsslsecurity.com/blog/what-is-a-malicious-url.

[20] RAPID Solution, What is a Phishing Attack, 2018, https://www.rapid7.com/fundamentals/phishing-attacks/.

[21] Imperva Cyber Security Leader, Phishing Attack, 2021, https://www.imperva.com/learn/application-security/phishing-attack-scam/.

[22] E. Alpaydin, Introduction to Machine Learning, MIT Press, Cambridge, MA, USA, 2020.

[23] S. Angra, and S. Ahuja, Machine Learning and its Applications, International Conference on Big Data Analytics and Computational Intelligence, pp. 57-60, Chirala, Andhra Pradesh, India, 2017.

[24] R. E. Schapire, The Boosting Approach to Machine Learning: An Overview in Non-linear Estimation and Classification, pp. 149-171, Springer, New York, NY, USA, 2003.

[25] University of New Brunswick, Canadian Institute of Cyber Security, Malicious URL Dataset, 2016, https://www.unb.ca/cic/datasets/url-2016.html.

[26] C. M. Bishop, Pattern Recognition and Machine Learning, 2006, https://link.springer.com/book/9780387310732.

[27] IBM Cloud Education, Machine Learning, 2020, https://www.ibm.com/cloud/learn/machine-learning.

[28] S. Ahmed, Y.Lee, S. Hyun, and I. Koo, Unsupervised Machine Learning-based Detection of Covert Data Integrity Assault in Smart Grid Networks Utilizing Isolation Forest, IEEE Transactions on Information Forensic and Security, vol. 14 no. 10, pp. 2765-2777, 2019.

[29] Z. Jaadi, A Step-by-Step Explanation of Principal Component Analysis (PCA), 2022, https://builtin.com/data-science/step-step-explanation-principal-component-analysis

[30] J. K. Jaiswal and R. S. Kannu, Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression, World Congress on Computing and Communication Technologies, pp. 65-68, Tiruchirappalli, India, 2017.

[31] K. Taunk, S. De, S. Verma, and A. Swetapadma, A Brief Review of Nearest Neighbor Algorithm for Learning and Classification, International Conference on Intelligent Computing and Control Systems, pp. 1255-1260, Madurai, India, 2019.

[32] E. Charniak, Bayesian Networks without Tears, AI Magazine, vol. 12, no. 4, pp. 50–63, Dec. 1991.

[33] S. A. Nimeh, D. Nappa, X. Wang, and S. Nair, A Comparison of Machine Learning Techniques for Phishing Detection, Anti Phishing Working Groups Annual eCrime Researchers Summit, pp. 60–69, Pittsburgh, PA, USA, 2007.

[34] A. Sharma, Machine Learning 101: Decision Tree Algorithm for Classification, 2021, https://www.analyticsvidhya.com/blog/2021/02/machine-learning-101-decision-tree-algorithm-for-classification/.

[35] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, New York, NY, USA, 2009.

[36] P. Gupta, Cross-validation in Machine Learning, 2017, https://medium.com/towards-data-science/cross-validation-in-machine-learning-72924a69872f.

[37] S. Gupta, Decision Trees Towards Data Science, 2017, https://medium.com/towards-data-science/decision-trees-in-machine-learning-641b9c4e8052.

[38] M. Rekha, Eigen Decomposition and PCA, 2019, https://blog.clairvoyantsoft.com/eigen-decomposition-and-pca-c50f4ca15501.

[39] K. McGrath and M. Gupta, Behind Phishing: An Examination of Phisher Modi Operandi, Usenix Workshop on Large-Scale Exploits and Emergent Threats, San Francisco, CA, USA, 2008.

[40] M. Justin, K. Saul Lawrence, Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs, International Conference on Knowledge Discovery and Data Mining, pp. 1245-1254, Paris, France, 2009.

[41] M. Justin, S. Savage, and K. S. Lawrence, Identifying Suspicious URLs: An Application of Large-scale Online Learning, International Conference on Machine Learning, pp. 681-688, Montreal, QC, Canada, 2009.

[42] R. Islam and J. Isphany, Detecting Malicious COVID-19 URLs Using Machine Learning Techniques, International Conference on Pervasive Computing and Communications, Virtual, Germany, 2021.