Evaluation of Machine Learning Classifiers for Phishing Detection

by

Rabail Kazi
B.Eng., Mehran University of Engineering and Technology, 2011

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

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering

Evaluation of Machine Learning Classifiers for Phishing Detection

by

Rabail Kazi
B.Eng., Mehran University of Engineering and Technology, 2011

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Samer Moein, Departmental Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Samer Moein, Departmental Member
(Department of Electrical and Computer Engineering)

ABSTRACT

One of the common techniques used by attackers to break security and steal private and confidential information is phishing. An effective way to defend against phishing is to use an add-on filter. However, it is vital for the phishing detection system to be accurate. The phishing detection system used in this project is a website filter based on the Simple Logistic heuristic which is a machine learning algorithm. Weka is a tool used for implementing machine learning algorithms. In this report, several classifiers present inside Weka are tested against a fixed data set. The aim is to examine machine learning classifiers for detection of phishing. Experimental results are presented which demonstrate that Random Forest outperforms all other classifiers with an accuracy of 93%. The accuracy is further improved for Random Forest by using the Auto-WEKA classifier. This classifier is able to detect up to 99% of phishing websites, with a False Positive Rate (FPR) of only 1%. Thus, the accuracy of the phishing detection system can be improved by using the Random Forest classifier and Auto-WEKA.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My supervisor, Dr. T. Aaron Gulliver for his guidance, patience and support. He is always open and honest in communicating with his students and I never would have completed my masters degree without his supervision.

My Parents, who taught me how to speak my first words and how to be a better person. This report would not have been possible without their love and support, and I cannot be thankful enough for all their sacrifices.

My Siblings, for always being there to support and motivate me. Their advice always helped me to gain motivation and to use my efforts in the best way possible.

University of Victoria, for an extremely supportive environment which helped me to focus and dedicate my efforts towards this degree.

DEDICATION

I dedicate this report to my parents.

Chapter 1

Introduction

In the past few years, phishing has become a persistent threat to online users. Findings from the Anti-Phishing Working Group [1] indicate that the number of phishing websites observed in the first quarter of 2016 increased 250% from the last quarter of 2015, as shown in Figure 1.1. The number of phishing attacks observed per month rose from 48,114 in October 2015 to 123,555 in March 2016. There were 289,371 unique reports received in the first quarter of 2016, compared to 136,347 just 12 months earlier [1].

In 1996, the word Phishing was first used when a group of hackers stole America Online (AOL) accounts by tricking AOL users into giving away their passwords [2]. Phishing is one of the most profitable crimes these days. Trillions of dollars are lost every year because of users entering private information into fake websites. Phishers lure people into giving away private information by acting as a recognized company or organization [3]. In almost all phishing attacks, phishers lure people to a fake website. To host a fake website, phishers register a new domain, use a compromised machine, or use free web space [4]. Hence, it is vital to filter out these fake websites to eliminate phishing attacks.

Two distinct approaches have been used so far to prevent users from visiting phishing websites. The first approach is the blacklist method and is the most common approach used by modern web browsers as it incurs no false positives. In the blacklist method, phishing websites are detected by comparing the URL of a website a user visits with a database of verified phishing websites [5]. However, it is difficult to build an effective blacklist due to the substantial increase in the number of phishing websites. Furthermore, it is not very efficient in the window of vulnerability, e.g. the time period between detection of a threat and the protective steps taken against

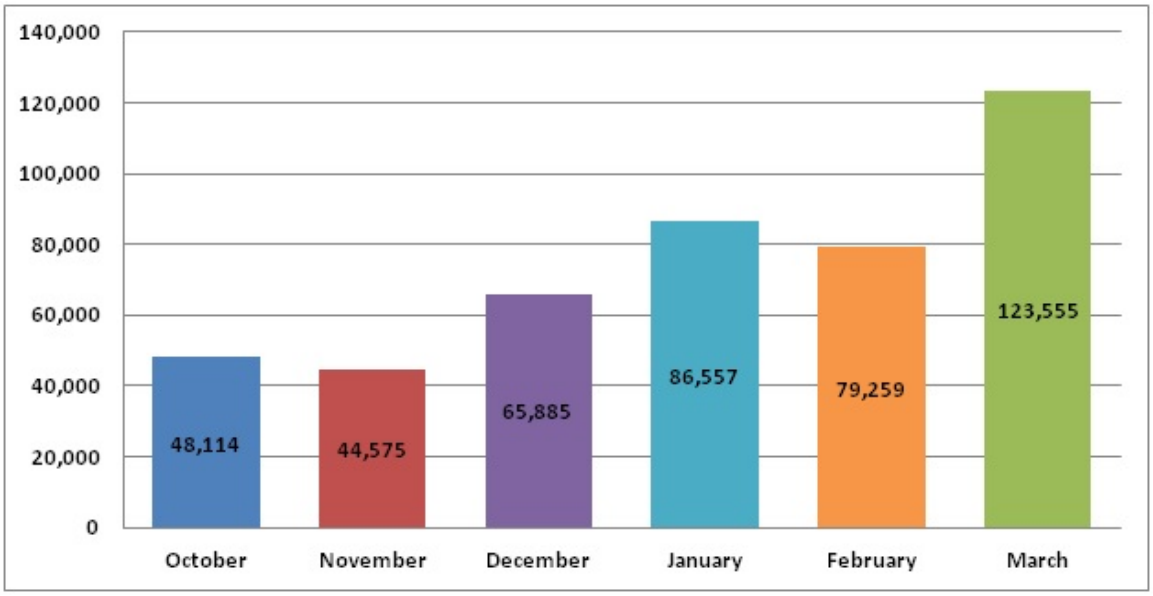


Figure 1.1: The number of unique phishing websites detected from October 2015 to March 2016 [1].

it. The second approach is heuristic-based. In this approach, different features of a website are collected and the heuristic determines if a website seems to be a phishing website. Different from the blacklist method, the heuristic-based method can identify new phishing websites. Although there are numerous solutions proposed for the detection and prevention of phishing attacks using the heuristic approach, most have poor detection accuracy or a large number of false positives [5].

1.1 Motivation

Despite educating users and implementing efficient phishing detection systems, users are still extremely susceptible to phishing attacks. In [6], an exact replica of a bank website was designed to trap employees and obtain their credentials. Around 120 employees were targeted out of which 52 submitted their credentials. These results indicate that phishing is very dangerous because users can easily be victimized.

The heuristic approach is an effective means of limiting users from visiting phishing websites. This approach can support users by informing them that they are about to visit a phishing website. The phishing detection system used in this project is based on the Simple Logistic heuristic which is a machine learning algorithm [7]. Unfortunately, the detection accuracy of this system is poor, so users may not trust

warnings from the system even if it correctly identifies a phishing website. The aim of this project is to find a machine learning classifier which can improve the detection accuracy of the phishing detection system.

1.2 Related Work

In order to address the phishing issue, a lot of research has been carried out. Diverse approaches have been utilized for phishing detection, which includes whitelists [8], blacklists [9], and heuristics [10]. In the whitelist approach, only URLs present in a list of legitimate URLs are qualified to be safe and all other URLs are believed to be phishing websites. The disadvantage with this approach is that it is impossible to create a list of all legitimate websites. Blacklist is the most common approach used by current web browsers. Phishing websites are detected by comparing the URL of a website with a database of verified phishing websites [11]. However, it is difficult to create an effective blacklist because of the substantial increase in the number of phishing websites. Another approach is the heuristic-based method. In this approach, different features of a website are collected and the heuristic determines if a website seems to be a phishing website. Unfortunately, the majority of these phishing detection systems suffer from high levels of false positives.

Several heuristic approaches have been used to identify phishing websites [9], [11], [12]. However, they have poor detection accuracy. Jo et al. [7], [13] developed an interactive website filter which detects phishing websites based on machine learning. The key feature of this system is that it considers the dissimilarity between the true identity of a website and its observed identity. A website with high dissimilarity will be considered a phishing website.

In [13], a data set of 2084 URLs was used for training and testing. This data set is composed of 1684 phishing websites and 400 legitimate websites. The results showed that the True Positive Rate (TPR) was 98.5% (1659/1684) and the False Positive Rate (FPR) was 0.5% (2/400) using the Simple Logistic classifier. This TPR is high because the data set consists of around 80% phishing websites and 20% legitimate websites. However, in real life, the number of legitimate websites is usually more than the number of phishing websites. Therefore, approximately 50% legitimate and 50% phishing websites is better, since the number of phishing websites is also increasing significantly.

In [14], the heuristic-based approach presented in [7], [13] was integrated into a

web browser via a Firefox add-on. The phishing filter in [14] is also based on machine learning techniques as in [7], [13]. The Random Forest Classifier was implemented and up to 94.3% detection accuracy was achieved. A data set of 1761 URLs was used for training and testing. The data set in [14] was composed of 829 phishing websites and 932 legitimate websites. The results showed that the True Positive Rate (TPR) was 94.3% and the False Positive Rate (FPR) was 5.8%. However, the performance evaluation is not that meaningful as the data set and the number of detected phishing and legitimate websites is unknown.

In [13], [14], it is claimed that the proposed algorithms have high detection accuracy. In this project, the machine classifiers in Weka are evaluated to increase the phishing detection accuracy and decrease the number of false positives. The phishing detection accuracy is measured for the Simple Logistic, Random Forest, AdaBoost, MultiLayer Perceptron and Rotation Forest classifiers, present in Weka. The data set consists of 1761 websites which were collected from various sources, 829 websites are phishing and the remaining 932 websites are legitimate. The performance evaluation of these classifiers is evaluated and compared with the results in [13]. These results demonstrate that Random Forest outperforms all other classifiers with an accuracy of 93%. Accuracy of Random Forest is further increased with Auto-WEKA. The Auto-WEKA classifier is able to detect up to 99% of phishing websites with a False Positive Rate (FPR) of only 1%. Thus, the accuracy of the phishing detection system can be improved by using the Random Forest classifier and Auto-WEKA.

1.3 Report Outline

This report is structured as follows.

Chapter 1 provided a brief introduction to phishing, phishing detection techniques and how phishing takes place. The motivation of the project along with the related work was discussed.

Chapter 2 presents the design of the phishing detection system used in this project. The main features are discussed along with how machine learning is employed.

Chapter 3 introduces machine learning. The machine learning software WEKA used in this project along with different machine learning classifiers are described. Auto-WEKA is also explained briefly.

Chapter 4 presents the methodology and the experiments employed. The first part provides the hardware and software configuration and the evaluation metrics. The second part compares the detection accuracy of the classifiers. The phishing detection accuracy is measured for the Simple Logistic, Random Forest, AdaBoost, MultiLayer Perceptron and Rotation Forest classifiers present in Weka. Auto-WEKA is used to improve the classifier performance.

Chapter 5 concludes the report and provides suggestions for future work.

Chapter 2

The Phishing Detection System

The phishing detection system used in this project is an interactive website filter which is based on heuristics [7]. The main feature of this system is that it considers the disparity between the websites observed and their true identities. Observed identities are features of a website which are easy to be spoofed, e.g. frequent terms. True identities are those features of a website which are hard to spoof, e.g. the host domain.

For a given website, the detection system finds the observed and true identities from its WHOIS record, URL, and content features. Then, it measures the dissimilarity between these identities using text similarity. A website with significant dissimilarity is believed to be a phishing website, and the user is notified that it maybe a phishing website. The user then decides whether to browse the website or not. The reason the detection system is not entirely automated is that the heuristics are not able to attain perfect phishing detection accuracy.

2.1 Main Components of the System

The phishing detection system employs three main features, URL, content, and WHOIS for a website. These features are used to extract information about a website, and the results are passed to the machine learning classifiers. A block diagram of the phishing detection system is shown in Figure 2.1.

The classifiers are supervised machine learning algorithms. The training is based on features, in this case whether a website is phishing or not. The classifiers are trained with known phishing and legitimate websites and then provide binary deci-

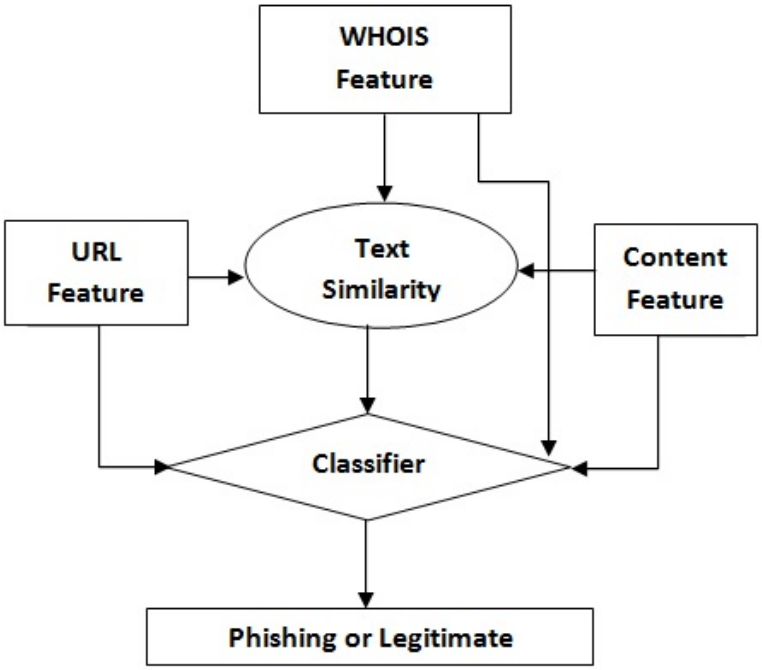


Figure 2.1: Block Diagram of the phishing detection system [13].

sions for websites. Chapter 3 provides a brief overview of machine learning classifiers as this is one of the major components of this project. Subsections 2.1.1 and 2.1.2 discuss the main features required for detecting phishing websites.

2.1.1 URL, Content and WHOIS Features

URL obfuscation is an important part of phishing attacks. A URL obfuscation technique fools the user into trusting a website. The phishing detection system classifies URL obfuscation techniques into six types as shown in Table 2.1.

Content features are considered as observed identities. In [7], four features were gathered from the content of websites. Table 2.2 shows the four features used in the phishing detection system.

Phishing websites are likely to have poor WHOIS records, e.g. fundamental registrant records. A simple WHOIS search for a website may not even be successful. For a website, the phishing detection system sends a WHOIS query for its domain and extracts the following features [7].

- Registrant - Who is responsible for the particular domain?

Type I	Whether IP address or port number is present, e.g. `http://200.68.203.165/psh/pshcm1.html`
Type II	Whether URL contains domains except the host name, e.g. `http://muetinfotelecom.com/www.paypal.com/IBlognt.html`
Type III	Whether multiple domain names are embedded in the host name, e.g. `http://youonlineaccount.rescuenational.co.uk.poolho a.net/`
Type IV	The presence of obfuscated characters in the URL, e.g. `http://www.soundsterio.com/ServicE\%20W\%20give\_file s/images/`
Type V	Whether the website is using HTTPS protocol
Type VI	Whether the website is sent via a chain of redirection

Table 2.1: The six types of URL features used in the phishing detection system [7].

Type I	Common Name candidates in website identity features
Type II	Frequency of terms
Type III	Collecting resource domains
Type IV	Malicious behaviour features

Table 2.2: The four types of content features used in the phishing detection system [7].

- Dates - What are the dates of updates, registration, and expiration?
- Name server - How many domain name servers are present for this website?

2.1.2 Dissimilarity Feature

Following extraction of the WHOIS, URL, and content features, the phishing detection system uses the approximate string matching algorithm to compute the text similarity [7]. Approximate string matching is a technique used to find strings whose patterns match closely but not exactly [14]. The text similarities are found for the website observed and true identities. Jo et al. [7], confirmed that a legitimate website will have a greater text similarity than a phishing website. The measures obtained by the string matching technique are then passed to the classifiers for predicting phishing websites.

2.2 Features used in the System

This section presents the 10 features used in the system for phishing detection [7]. The system uses machine learning classifiers, (explained in Chapter 3), string matching techniques for finding text similarity, and the WHOIS server API. The system initially fetches the features of the URL of a website and then the text similarity is computed. Table 2.3 lists the features which are employed in the system.

Feature	Description
Protocol	Whether the URL uses a secure connection, e.g. Https or Http
Frequent Terms vs. Registrant	Text similarity between the most frequent terms in content and the name of the domain name registrant via WHOIS
No. of Domain Name Candidates in Hostname and Content	Text similarity of domain registrant that appears both in content as well as the host domain
No. of Domain Name Candidates in Hostname	Text similarity of domain registrant that appears in the host domain
Numeric Hostname	Whether the host domain name is an IP address
Title vs. Registrant	Text similarity between the registrant and the title from WHOIS
Title vs. Domain Name Candidates in Hostname	Text similarity between the Domain Name and title in the host domain
Frequent Terms vs. Host Domain	Text similarity between the repeated terms in the host domain and content
Copyright vs. Registrant	Text similarity between registrant and copyright holder from WHOIS
Domain Name Candidates in Anchors vs. Registrant	Text similarity between Domain Name candidates from anchors and registrant from WHOIS

Table 2.3: The 10 features and their descriptions used in the phishing detection system [7].

Chapter 3

Machine Learning

Machine learning is widely utilized in the area of information security. There are two kinds of machine learning, supervised and unsupervised. Both supervised and unsupervised machine learning approaches have been used for identifying phishing websites, e.g. [5], [6], [7], [11]. In this project, supervised learning is utilized, e.g. pre-existing phishing websites and legitimate websites are used for training. A phishing data set is used to assess the performance of the models. Following training and testing, these models are used to categorize new websites. The goal is to distinguish between phishing and legitimate websites using the 10 features.

3.1 Machine Learning Process

Machine learning is a continuous process. As time goes by it gathers additional knowledge and adapts to make better predictions. This enables a system to automatically evaluate data and make decisions on what information is the most significant [14]. The stages of the machine learning process are described below.

- **Prepare Data** In this step, data is collected from various websites, e.g. PhishTank, Alexa [15], [16]. For this project, 1761 websites were collected. The phishing detection system takes this data in the form of URLs. The detection system fetches the 10 features described in Section 2.2. Next, the text similarity is obtained via approximate string matching.

- **Machine Learning Classifiers** There are numerous classifiers available in Weka [17]. In this project, five classifiers are selected from support vector machine, logistic regression, multilayer perceptron, and decision tree approaches.

All of the classifiers selected for this project have been used previously for detection of phishing [5], [7], [18]. Section 3.3 provides a brief overview of each classifier used in this project. After selecting the classifiers, the next step is to build and then simulate the model.

3.1.1 Types of Machine Learning Tasks

There are two kinds of machine learning tasks, supervised and unsupervised. For supervised learning, a training data set is forwarded to the classifier. The expected output and inputs are defined for this data set [14]. The classifier learns from the inputs and then estimates the result of future inputs. This approach is like a conventional schooling system where the student initially learns and afterward appears for an exam. In unsupervised learning, no training data is given to the classifier. The classifier determines the necessary parameters required for accurate results.

3.2 The Weka Machine Learning Tool

Weka [17] is a Java based machine learning program. It is open source software which consists of a variety of machine learning algorithms, e.g. logistic regression, support vector machines, decision trees and multilayer perceptrons. In this project, five types of classifiers, e.g. Random Forest, Simple Logistic, AdaBoost, MultiLayer Perceptron and Rotation Forest are evaluated using Weka. For a fair-comparison, all of these classifiers were used with the default options. The reason these classifiers were chosen is because they are popular and represent diverse classification categories.

3.3 Overview of the Machine Learning Classifiers

This section briefly explains the machine learning classifiers used in this project.

Simple Logistic

This classifier is typically used for building models based on linear logistic regression. It is the most extensively utilized statistical model for binary data prediction, e.g. 0 or 1. Being a generalized linear model, it usually utilizes the function called *logit* [19]. Weka uses LogitBoost for logistic models. LogitBoost is a boosting algorithm and the number of iterations to be performed in LogitBoost is usually cross validated [13].

The Simple Logistic regression classifier performs well when the relationship in the data is approximately linear, but performs poorly if complex nonlinear relationships are present among the variables.

Random Forest

This is an ensemble learning algorithm containing regression as well as classification. In Random Forest, the prediction is attained by means of decision trees. This classifier combines several tree predictors, and every tree depends on the values of a random vector sampled independently [18]. Random forests are able to handle large numbers of variables for a given data set [18]. In addition, they can approximate missing data well. The main disadvantage of random forests is that they lack reproducibility because the procedure of building the forest is random.

AdaBoost

Within the group of classifier ensemble models, AdaBoost is one of the best [20]. In AdaBoost, a higher weight is assigned to heuristics that are capable of labeling a website correctly (e.g. Phishing), and assigns a lower weight to a heuristic which labels it inaccurately [21]. The name of AdaBoost in Weka is AdaBoost M1. AdaBoost is intended to be used as a supervised learning algorithm. The main disadvantage of AdaBoost is the overfitting problem, which occurs when a model is very complicated, e.g. too many parameters relative to the number of observations, which can create random errors.

Multilayer Perceptron

Multilayer Perceptron networks are widely used nonlinear models consisting of a number of neuron layers. A Multilayer Perceptron is one the most well-known neural network models utilizing a back propagation algorithm. Every neuron contains a function that maps an input data set to an output data set [22].

Rotation Forest

This classifier is an ensemble method based on decision trees. It trains a large number of decision trees autonomously by means of a diverse set of extracted characteristics for each tree [20]. Rotation Forest aims to build precise and distinct classifiers.

Decision trees are trained via Random Forest from the given data set. This classifier draws upon the idea of Random Forest [20].

3.3.1 Auto-WEKA

The performance of machine learning classifiers depends significantly on the parameter settings. This represents a challenge for machine learning that given a data set, choosing a machine learning classifier and setting its parameters to obtain good performance [23]. Auto-WEKA is an automated approach which considers a range of feature selection techniques and all classification approaches present inside Weka. It has a more than 768 dimensional parameter space which includes all algorithms [23].

Auto-WEKA provides optimized classification methods and feature selectors inside Weka [23]. Auto-WEKA draws on the complete range of classification algorithms in Weka. This makes it easy for non-experts to construct efficient classifiers for specific applications. A wide-ranging empirical comparison of 21 well-recognized data sets demonstrated that Auto-WEKA repeatedly outperforms benchmark classifiers [23], particularly on huge data sets. Auto-WEKA by default chooses from 39 Weka classifiers. Out of these classifiers, 27 are considered as base classifiers e.g. used independently. The remaining ten classifiers are meta-classifiers, e.g. Auto-WEKA takes one base classifier along with its parameters as inputs.

Chapter 4

Performance Evaluation

In this chapter, the detection accuracy of different classifiers in Weka is evaluated. Machine learning classifiers are employed, supervised learning is performed, and phishing websites are distinguished from legitimate websites. A single data set is used for training and testing. Furthermore, different metrics for performance evaluation along with the hardware and software configuration are described. The accuracy measures of different classifiers is evaluated and Random Forest was found to be the best. Finally, Auto-WEKA is used to further optimize the Random Forest classifier for the phishing detection system.

4.1 Data Set

The number of URLs in the phishing data set and the distribution is shown in Table 4.1. The phishing data set contains 829 phishing and 932 legitimate websites. The phishing websites have been gathered from Phishtank [15], which is a well-recognized organization used for providing phishing samples. For legitimate URLs, Alexa [16] is used. Founded by Amazon, Alexa is a well-known source of high-traffic legitimate websites.

All ten features present in the phishing detection system are employed in the data set. For evaluation, all of the classifiers were trained and tested using 10 fold cross-

Total No. of Websites	Legitimate Websites	Phishing Websites
1761	932	829

Table 4.1: Data set distribution for phishing detection.

validation. Ten fold cross-validation gives high-quality estimates of the inaccuracy of a classifier [24]. The data set is divided into ten different parts, and nine parts out of ten are utilized to train the classifier. Then, the information gained from the training phase is used to confirm the tenth part. This is completed ten times and at the end of the testing and training phase, all of the parts are utilized as both testing and training data. This cross validation method guarantees that the training data will be different from the test data [24].

4.2 Hardware and Software Configuration

All of the tests were performed on a laptop with the software and hardware parameters shown in Table 4.2.

Manufacturer	Samsung Electronics
System Type	64-bit Operating System
Operating System	Windows 7 Home Premium
Processor Name	Intel(R)Core(TM)i3
Processor Speed	2.30GHz
Installed Memory (RAM)	6.00 GB
Total Number of Cores	4
Machine Learning Tool	Weka version 3.8.0

Table 4.2: Hardware and software paramters of the laptop used to carry out the experiments.

4.3 Evaluation Metrics

For any phishing detection system, one of the main requirements is that it should attain high detection accuracy. User safety will be compromised if the detection system tags phishing websites as legitimate. Moreover, users will complain if the detection system tags legitimate websites as phishing websites. In order to evaluate the detection accuracy for the phishing detection system, the evaluation metrics are as follows.

- True Positive (tp) - The number of phishing websites identified as phishing.
- True Negative (tn) - The number of legitimate websites identified as legitimate.

- False Positive (fp) - The number of legitimate websites misidentified as phishing.
- False Negative (fn) - The number of phishing websites misidentified as legitimate.

  A summary of the evaluation metrics is shown in Table 4.3.

	actual phishing websites	actual legitimate websites
predict phishing websites	tp	fp
predict legitimate websites	fn	tn

Table 4.3: The evaluation metrics, tp, tn, fp and fn, used for measuring phishing detection accuracy.

- **Precision (p)** measures the rate of phishing websites which are identified correctly as the websites detected as phishing. To be precise, it measures the degree to which the blocked websites are in fact phishing.

$$p = \frac{tp}{(tp + fp)} \tag{4.1}$$

- **Recall (r)** measures the rate of phishing websites which the phishing detection system identifies correctly as phishing websites.

$$r = \frac{tp}{(tp + fn)} \tag{4.2}$$

- **F-Measure ($f1 - measure$)** is the weighted harmonic mean of precision and recall. This project uses the $f1 - measure$ as an index for testing accuracy.

$$f1 - measure = \frac{2 \cdot p \cdot r}{(p + r)} \tag{4.3}$$

- **Error Rate (ER)** can be calculated by dividing the number of incorrectly identified websites by the number of all websites in the data set. The average ER is a reasonable metric to indicate the detection accuracy.

$$ER = \frac{(fp + fn)}{(tp + tn + fp + fn)} \tag{4.4}$$

- **True Positive Rate (TPR)** is the number of phishing websites detected as phishing divided by the number of phishing websites in the data set. It is also called the hit ratio.

$$TPR = \frac{\sum\ tp}{\sum\ \text{phishing websites in data set}} \tag{4.5}$$

- **False Positive Rate (FPR)** is the number of legitimate websites that were wrongly detected as phishing websites divided by the total number of legitimate websites in a data set.

$$FPR = \frac{\sum\ fp}{\sum\ \text{legitimate websites in a data set}} \tag{4.6}$$

According to [5], $f1-measure$ and ER are the common metrics for the evaluation of detection accuracy. The higher the $f1-measure$ and the lower the ER, the better the phishing detection accuracy. Hence, in this project, we use $f1-measure$ and ER to detect phishing detection accuracy. Moreover, we use the TPR and FPR metrics to compare the evaluation results of this project with the results presented in [13].

4.4 The Accuracy of Different Classifiers

In order to evaluate the classfier accuracy, the ten features present in the phishing detection system have been employed in Weka as shown in Section 2.2. The phishing detection accuracy has been measured for the classifiers, Simple Logistic, Random Forest, AdaBoost, MultiLayer Perceptron and Rotation Forest. The simulation was carried out with Weka, Figure A.1 shows the simulation for the five classifiers. The phishing detection accuracy measures are shown in Figure 4.1, Table 4.4, and Table 4.5.

The detection accuracy of all five classifiers is evaluated by measuring ER and $f1-measure$. The TPR and FPR evaluation results are also compared with the original phishing detection approach (Simple Logistic) used in [13]. Initially, the F-Measure ($f1-measure$), Precision (p) and Recall (r) average rates of all five classifiers were measured in Weka for the data set mentioned in Section 4.1. The 10 fold cross-validation was executed and the rates calculated as shown in Table 4.4. The highest $f1-measure$ is 0.930 with Random Forest, followed by Rotation Forest (0.917),

	p	r	$f1 - measure$
Simple Logistic	0.851	0.851	0.851
Random Forest	0.930	0.930	0.930
AdaBoost M1	0.873	0.872	0.872
Multilayer Perceptron	0.858	0.857	0.858
Rotation Forest	0.917	0.917	0.917

Table 4.4: Precision (p), Recall (r) and F-Measure ($f1 - measure$) for phishing detection.

AdaBoost M1 (0.872), Multilayer Perceptron (0.858), and finally Simple Logistic (0.851).

The ER is calculated using (4.4) as shown in Figure 4.1. The lowest error rate is 0.070 with Random Forest, followed by Rotation Forest (0.083), AdaBoost M1 (0.128), Multilayer Perceptron (0.142) and finally Simple Logistic (0.149).

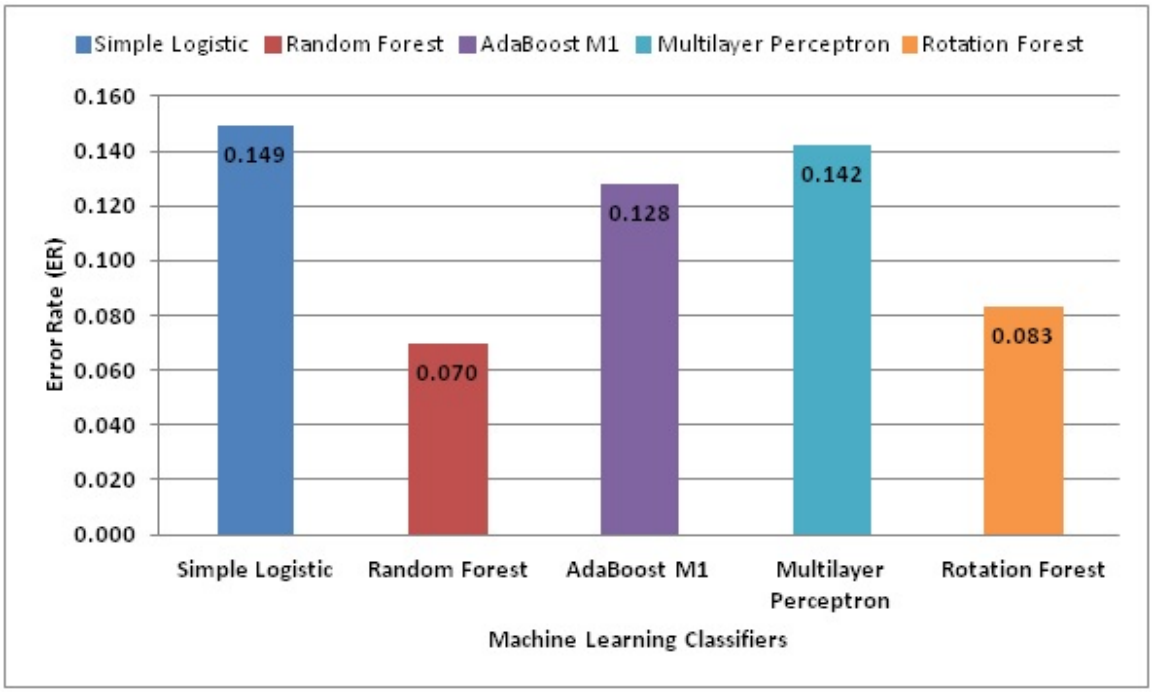


Figure 4.1: Error Rate (ER) for all classifiers measured after 10 fold cross-validation.

Using (4.5) and (4.6), the TPR and FPR were calculated and the results are shown in Table 4.5. The highest TPR was 93% (780/829) with Random Forest, and the lowest FPR was 6.9% (49/932) with Random Forest. In [13], the performance evaluation showed that the TPR was 98.5% (1659/1684) and the FPR was 0.5%

	TPR %	FPR %
Simple Logistic	85.1 (698/829)	15 (131/932)
Random Forest	93 (780/829)	6.9 (49/932)
AdaBoost M1	87.2 (739/829)	12.6 (90/932)
Multilayer Perceptron	85.7 (717/829)	14.2 (112/932)
Rotation Forest	91.7 (756/829)	8.4 (73/932)

Table 4.5: True Positive Rate (TPR) and False Positive Rate (FPR) results for the phishing detection accuracy measures.

(2/400) using Simple Logistic. Given the fact that this project uses 829 phishing websites and 932 legitimate websites, the websites are divided close to 50%. On the other hand, in [13], the data set has around 80% phishing websites and 20% legitimate websites. Furthermore, it can be seen that for the data set in this project Simple Logistic is the least accurate classifier, with a TPR of only 85.1% (698/829) and an FPR of 15% (131/932). Using this in real time would be annoying for users. Hence, according to these comparisons, four out of five classifiers, namely Random Forest, Rotation Forest, AdaBoost M1 and Multilayer Perceptron, outperform the Simple Logistic classifier. Tree classifiers have better detection accuracy compared to the remaining four classifiers. Random Forest [18], a tree classifier, had the highest accuracy amongst all other classifiers. Many classifiers have good accuracy but tree classifiers outperformed all other classifiers and this was shown when the data set is divided equally between phishing and legitimate websites. Thus, Random Forest is the best classifier for the phishing detection system.

4.5 Simulation of Auto-WEKA for the Optimization of Random Forest

In Section 4.4, Random Forest was found to be the best classifier for the phishing detection system. Since finding the optimal parameters for a classifier is a difficult process, Auto-WEKA is used in an attempt to further improve the detection accuracy of the phishing detection system using Random Forest as its base classifier. Auto-WEKA is an automated approach, so it determines the best classifier and uses the best feature selection method for any given data set [23]. Figure A.8 shows the simulation results for Auto-WEKA, and as expected it used Random Forest as its base classifier.

	$TPR\,\%$	$FPR\,\%$	ER	p	r	$f1-measure$
Auto-WEKA	99 (825/829)	1 (4/932)	0.010	0.990	0.990	0.990

Table 4.6: True Positive Rate (TPR), False Positive Rate (FPR), Error Rate (ER), Precision (p), Recall (r) and F-Measure ($f1-measure$) for phishing detection using Auto-WEKA.

As in the previous section, 10 fold cross-validation was executed on Weka using Auto-WEKA and the evaluation metrics were calculated and are shown in Table 4.6. The results of Auto-WEKA are very impressive. Using Auto-WEKA, the $f1-measure$ increased from 0.930 to 0.990. The ER was also decreased from 0.070 to merely 0.010. Furthermore, the TPR is increased to 99% while the FPR is reduced to only 1%. This shows that for the phishing detection system, Auto-WEKA can be used with Random Forest.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this project the accuracy of the phishing detection system developed in [7] was evaluated. For a given website, the detection system finds the observed and true identities from its WHOIS record, URL, and content features. Then, it measures the dissimilarity between these identities using text similarity. A website with significant dissimilarity is believed to be a phishing website, and users are notified that it is a phishing website. The user then decides whether to browse the website or not. The system was evaluated with five classifiers on a fixed data set. The classifiers included Simple Logistic, Random Forest, AdaBoost, MultiLayer Perceptron and Rotation Forest. A total of 1761 URLs were collected for the phishing data set. The phishing data set consisted of 829 phishing samples and 932 legitimate URL samples. Ten fold cross-validation was executed and evaluation metrics were calculated. In order to evaluate the phishing detection accuracy, the main evaluation metrics were F-Measure ($f1-measure$), and Error Rate (ER). The performance evaluation showed that Random forest outperformed all other classifiers with the highest $f1-measure$ of 0.930 and the lowest ER of 0.070.

The detection accuracy was further tested with Auto-WEKA. Auto-WEKA is a classification approach which chooses the best classifier for a given system and sets its parameter to provide the best performance [23]. As expected, Auto-WEKA chose Random Forest as its base classifier and provided the highest F-Measure (0.990), and a negligible Error Rate of only 0.010.

The True Positive Rate (TPR) and False Positive Rate (FPR) were also com-

pared with the original phishing detection heuristic (Simple Logistic). In [13], the performance evaluation showed that the TPR was 98.5% (1659/1684) and the FPR was 0.5% (2/400). Given the fact that this project uses 829 phishing websites and 932 legitimate websites, the websites are divided close to 50%. On the other hand in [13], the data set had around 80% phishing websites and 20% legitimate websites. Furthermore, it can be seen that for the data set used in this project Simple Logistic was the least accurate classifier, with a TPR of 85.1% (698/829) and an FPR of 15% (131/932). Using this in real time would be annoying for users. Users will stop taking this detection system seriously and will be more likely to assume a True Positive (tp) is a False Positive (fp). According to the comparisons made, four out of five classifiers, namely Random Forest, Rotation Forest, AdaBoost M1 and Multilayer Perceptron, outperform the Simple Logistic classifier. Thus, for larger data sets and where the phishing websites and legitimate websites are close to be equal, Random Forest is the best out of the other four classifiers.

When Auto-WEKA was implemented to further improve the performance of Random Forest, the TPR increased to 99% while the FPR was reduced to only 1%. This shows that for the phishing detection system, Random Forest and Auto-WEKA can be used for real-world scenarios where there are a significant number of phishing websites.

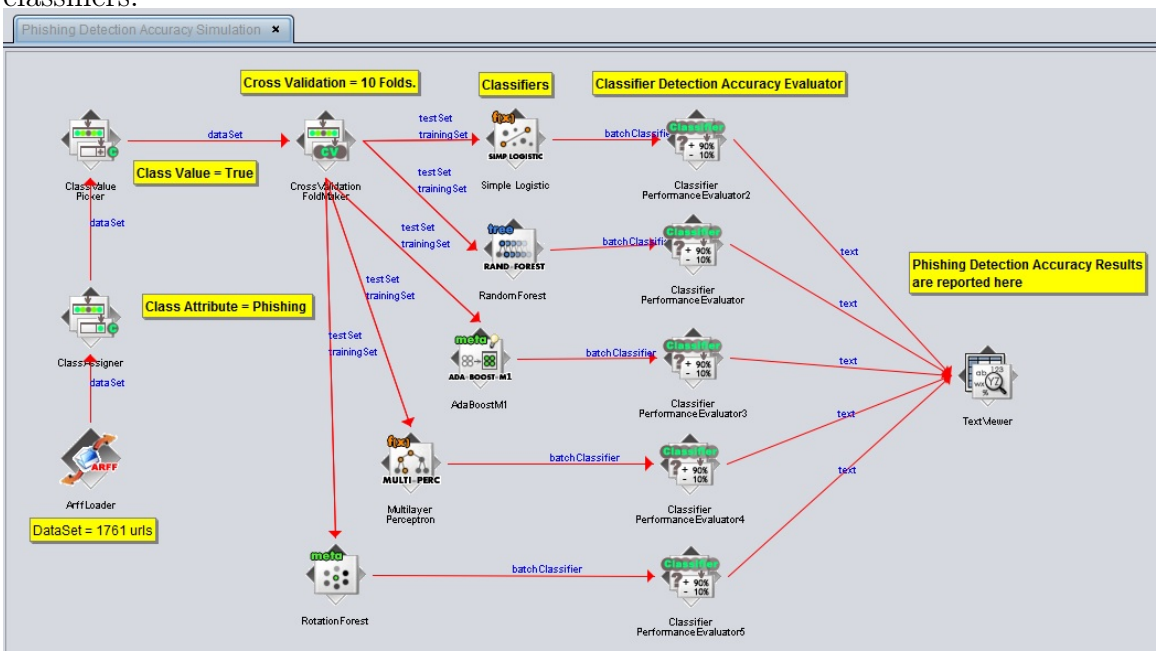
5.2 Future Work

The evaluation results in Chapter 4 motivate future work to include additional websites in the data set. Apart from additional websites, more features can be added to the phishing detection system. For instance, websites can also be included from Virus Total [25] as its knowledge base contains recent phishing websites. This will improve the prediction and decrease the misidentification rate of the classifiers. Furthermore, developing an automated phishing detection system can also be considered.

Appendix A

Simulation Models and Results

Figure A.1: Simulation model with Weka for the phishing detection system using five classifiers.



In Figure A.1, the simulation model created with Weka's Knowledge Flow visualizer is shown. In order to process and analyze the data, all of the components were first selected via the tool bar and then placed on the layout canvas. Weka supports only .Arff files as input hence the data set was created in .Arff format. The data set was loaded into the Arff Loader. Next the Class Assigner was added which permits users to choose the class attribute. In this project, the Class Attribute was Phishing and the Class value was selected as True. For the evaluation, the three main com-

ponents are Cross Validation Fold Maker, Classifiers, and the Classifier Performance Evaluator. From the toolbar, Cross Validation Fold Maker was selected and was set to 10. Then, the five classifiers used in this project were added and connected individually with the Cross Validation Fold Maker twice by initially choosing the training set and then the test set. For the performance evaluation, the Classifier Performance Evaluator was selected and connected with the individual classifiers. For the visualization of the results for each classifier, the Text Viewer component was chosen and then the simulation model was loaded and executed. The results are shown in Figures A.2, A.3, A.4, A.5, and A.6.

Figure A.2: Evaluation results for the Simple Logistic classifier with Weka.

```
=== Evaluation result ===

Scheme: SimpleLogistic
Options: -I 0 -M 500 -H 50 -W 0.0
Relation: Phishing.Detection-weka.filters.unsupervised.instance.Randomize-S42


Correctly Classified Instances        1498               85.0653 %
Incorrectly Classified Instances       263               14.9347 %
Kappa statistic                          0.7003
Mean absolute error                      0.2173
Root mean squared error                  0.3249
Relative absolute error                 43.6059 %
Root relative squared error             65.1013 %
Coverage of cases (0.95 level)          99.3754 %
Mean rel. region size (0.95 level)      85.2073 %
Total Number of Instances             1761

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.842    0.142    0.841      0.842    0.841      0.700    0.927     0.899     true
               0.858    0.158    0.859      0.858    0.859      0.700    0.927     0.938     false
Weighted Avg.  0.851    0.150    0.851      0.851    0.851      0.700    0.927     0.920

=== Confusion Matrix ===

   a   b   <-- classified as
 698 131 |   a = true
 132 800 |   b = false
```

The evaluation results show various evaluation matrices which are presented in Weka by default. However, in this project for detecting phishing accuracy, the matrices used from the results shown are True Positive Rate (TPR), False Positive Rate (FPR), Precision (p), and Recall (r). Moreover, the confusion matrix shows the number of websites detected as True Postive (tp), False Positive (fp), True Negative (tn) and False Negative (fn). All of the evaluation results are discussed in detail in Section 4.4.

Figure A.3: Evaluation results for the Random Forest classifier with Weka.

```
=== Evaluation result ===

Scheme: RandomForest
Options: -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation: Phishing.Detection-weka.filters.unsupervised.instance.Randomize-S42


Correctly Classified Instances        1637               92.9585 %
Incorrectly Classified Instances       124                7.0415 %
Kappa statistic                          0.8589
Mean absolute error                      0.1162
Root mean squared error                  0.2363
Relative absolute error                 23.3227 %
Root relative squared error             47.3345 %
Coverage of cases (0.95 level)          98.9211 %
Mean rel. region size (0.95 level)      67.8308 %
Total Number of Instances             1761

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.941    0.080    0.912      0.941    0.926      0.859    0.975     0.970     true
               0.920    0.059    0.946      0.920    0.933      0.859    0.975     0.975     false
Weighted Avg.  0.930    0.069    0.930      0.930    0.930      0.859    0.975     0.973

=== Confusion Matrix ===

   a   b   <-- classified as
 780  49 |   a = true
  75 857 |   b = false
```

Similar to Figure A.1, the simulation model for the Auto-WEKA classifier was created with Weka's Knowledge Flow for optimizing the Random Forest classifier and is shown in Figure A.7. The results for the phishing detection accuracy can be seen below in Figure A.8. Again, the matrices used from the results shown are True Positive Rate (TPR), False Positive Rate (FPR), Precision (p) and Recall (r). Moreover, the confusion matrix shows the number of websites detected as True Postive (tp), False Positive (fp), True Negative (tn), and False Negative (fn). The evaluation results for Auto-WEKA are discussed in Section 4.5.

Figure A.4: Evaluation results for the AdoBoostM1 classifier with Weka.

```
=== Evaluation result ===

Scheme: AdaBoostM1
Options: -P 100 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump
Relation: Phishing.Detection-weka.filters.unsupervised.instance.Randomize-S42


Correctly Classified Instances        1535               87.1664 %
Incorrectly Classified Instances       226               12.8336 %
Kappa statistic                          0.7432
Mean absolute error                      0.1859
Root mean squared error                  0.3045
Relative absolute error                 37.3069 %
Root relative squared error             61.0093 %
Total Number of Instances             1761

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.891    0.146    0.845      0.891    0.867      0.744    0.940     0.921     true
                 0.854    0.109    0.898      0.854    0.876      0.744    0.940     0.946     false
Weighted Avg.    0.872    0.126    0.873      0.872    0.872      0.744    0.940     0.934

=== Confusion Matrix ===

   a   b   <-- classified as
 739  90 |   a = true
 136 796 |   b = false
```

Figure A.5: Evaluation results for the Multilayer Perceptron classifier with Weka.

```
=== Evaluation result ===

Scheme: MultilayerPerceptron
Options: -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: Phishing.Detection-weka.filters.unsupervised.instance.Randomize-S42


Correctly Classified Instances        1510               85.7467 %
Incorrectly Classified Instances       251               14.2533 %
Kappa statistic                          0.7145
Mean absolute error                      0.1925
Root mean squared error                  0.3286
Relative absolute error                 38.6396 %
Root relative squared error             65.8288 %
Total Number of Instances             1761

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.865    0.149    0.838      0.865    0.851      0.715    0.922     0.891     true
                 0.851    0.135    0.876      0.851    0.863      0.715    0.922     0.933     false
Weighted Avg.    0.857    0.142    0.858      0.857    0.858      0.715    0.922     0.913

=== Confusion Matrix ===

   a   b   <-- classified as
 717 112 |   a = true
 139 793 |   b = false
```

Figure A.6: Evaluation results for the Rotation Forest classifier with Weka.

```
=== Evaluation result ===

Scheme: RotationForest
Options: -G 3 -H 3 -P 50 -F "weka.filters.unsupervised.attribute.PrincipalComponents -R 1.0 -A 5 -M -
Relation: Phishing.Detection-weka.filters.unsupervised.instance.Randomize-S42


Correctly Classified Instances        1614               91.6525 %
Incorrectly Classified Instances       147                8.3475 %
Kappa statistic                          0.8325
Mean absolute error                      0.1402
Root mean squared error                  0.2527
Relative absolute error                 28.1404 %
Root relative squared error             50.6188 %
Total Number of Instances             1761

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   MCC      ROC Area  PRC Area  Class
                 0.912    0.079     0.911      0.912    0.911      0.832    0.965     0.952     true
                 0.921    0.088     0.922      0.921    0.921      0.832    0.965     0.968     false
Weighted Avg.    0.917    0.084     0.917      0.917    0.917      0.832    0.965     0.961

=== Confusion Matrix ===

   a   b   <-- classified as
 756  73 |   a = true
  74 858 |   b = false
```

Figure A.7: Simulation of the Auto-WEKA classifier with Weka for optimizing Random Forest.

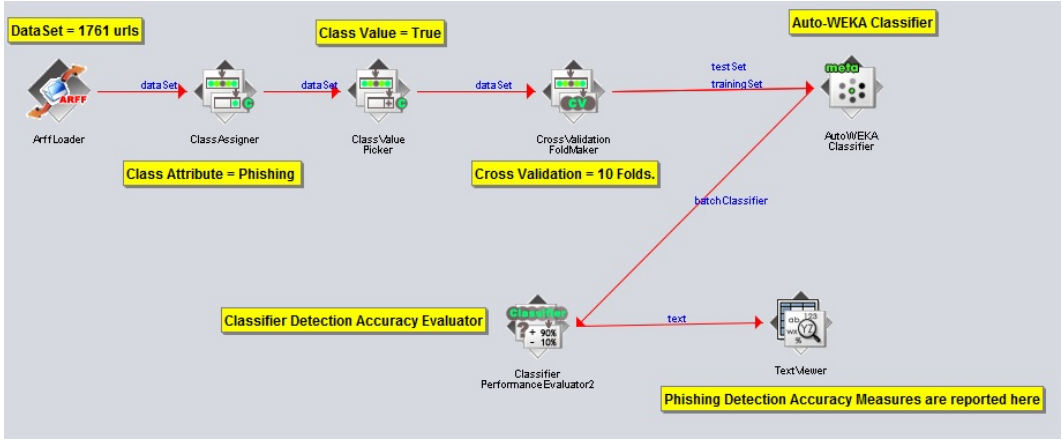


Figure A.8: Evaluation results for the Auto-WEKA classifier with Weka.

```
=== Run information ===

Scheme:       weka.classifiers.meta.AutoWEKAClassifier -seed 123 -timeLimit 15 -memLimit 1024
Relation:     Phishing.Detection-weka.filters.unsupervised.instance.Randomize-S42
Instances:    1761
Attributes:   11
              protocol
              FrequentTermsvsRegistrant
              NOofDomainNamecandidatesinHostnameandContent
              NOofDomainNamecandidatesinHostname
              NumericHostname
              TitlevsRegistrant
              TitlevsDomainNamecandidatesinHostname
              FrequentTermsvsHostDomain
              CopyrightvsRegistrant
              DomainNamecandidatesinAnchorsvsRegistrant
              phising
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

best classifier: weka.classifiers.trees.RandomForest
arguments: [-I, 14, -K, 1, -depth, 19]
attribute search: null
attribute search arguments: []
attribute evaluation: null
attribute evaluation arguments: []
estimated error: 0.3975014196479236


Correctly Classified Instances        1743               98.9779 %
Incorrectly Classified Instances        18                1.0221 %
Kappa statistic                          0.9795
Mean absolute error                      0.0469
Root mean squared error                  0.1139
Relative absolute error                  9.4044 %
Root relative squared error             22.8227 %
Total Number of Instances             1761

=== Confusion Matrix ===

   a   b   <-- classified as
 825   4 |   a = true
  14 918 |   b = false

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.995    0.015    0.983      0.995    0.989      0.980    0.999     0.999     true
                 0.985    0.005    0.996      0.985    0.990      0.980    0.999     0.999     false
Weighted Avg.    0.990    0.010    0.990      0.990    0.990      0.980    0.999     0.999
```

Bibliography

[1] Anti Phishing Working Group, "APWG phishing activity trends report - 1Q 2016", Online: `http://docs.apwg.org/reports/apwg_trends_report_q1_2016.pdf`, May 2016.

[2] B. B. Gupta, A. Tewari, A. K. Jain and D. P. Agrawal, "Fighting against phishing attacks: State of the art and future challenges", Journal of Neural Computing and Applications, Vol. 27, pp. 1–26, Jan. 2016

[3] J. S. Downs, M. B. Holbrook and L. F. Cranor, "Decision strategies and susceptibility to phishing", in Proceedings of the Symposium on Usable Privacy and Security, pp. 79–90, Jul. 2006.

[4] J. Hong, "The state of phishing attacks", Communications of the ACM, Vol. 55, pp. 74–81, Jan. 2012.

[5] M. Daisuke, H. Hazeyama and Y. Kadobayashi, "An evaluation of machine learning-based methods for detection of phishing sites", in Proceedings of the International Conference on Neural Information Processing, pp. 539–546, Nov. 2008.

[6] M. Aburrous, M. A. Hossain, K. Dahal and F. Thabtah, "Predicting phishing websites using classification mining techniques with experimental case studies", in Proceedings of IEEE International Conference on Information Technology: New Generations, pp. 176–181, Apr. 2010.

[7] I. Jo, E. Jung and H. Y. Yeom, "Interactive website filter for safe web browsing", Journal of Information Science and Engineering, Vol. 29, pp. 115–31, Jan. 2013.

[8] Y. Cao, W. Han and Y. Le, "Anti-phishing based on automated individual white-list", in Proceeding of the ACM Workshop on Digital Identity Management, Oct. 2008.

[9] P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks", in Proceedings of the International Conference on Computer Communications, pp. 1–5, Apr. 2010.

[10] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh and J. C. Mitchell, "Client-side defense against web-based identity theft", in Proceedings of Network and Distributed System Security Symposium, Feb. 2004.

[11] Y. Zhang, J. Hong and L. Cranor, "CANTINA: A content-based approach to detecting phishing web sites", in Proceedings of the International World Wide Web Conference, pp. 639–648, Feb. 2007.

[12] G. Xiang and J. Hong, "A hybrid phish detection approach by identity discovery and keywords retrieval", in Proceedings of the International World Wide Web Conference, pp. 571–580, Apr. 2009.

[13] I. Jo, E. Jung and Y. H. Yeom, "You're not who you claim to be: Website identity check for phishing detection", in Proceedings of the International Conference on Computer Communications and Networks, pp. 1–6, Aug. 2010.

[14] R. Joshi, "Interactive phishing filter", Master's Project, Online: `http://scholarworks.sjsu.edu/etd_projects/430`, Sep. 2015.

[15] LL. OpenDNS, "PhishTank: An anti-phishing site", Online: `https://www.phishtank.com/`, May 2016.

[16] Alexa Internet Inc., "Alexa the web information company", Online: `http://www.alexa.com/topsites`, May 2016.

[17] Weka, Online: `http://www.cs.waikato.ac.nz/ml/weka/`, Aug. 2016.

[18] A. A. Akinyelu and A.O. Adewumi, "Classification of phishing email using random forest machine learning technique", Journal of Applied Mathematics, Vol. 2014, Apr. 2014.

[19] A. Saeed, D. Nappa, X. Wang and S. Nair, "A comparison of machine learning techniques for phishing detection", in Proceedings of the Anti-Phishing Working Groups eCrime Researchers Summit, pp. 60–69, Oct. 2007.

[20] L. I. Kuncheva and J. J. Rodrguez, "An experimental study on rotation forest ensembles", in Proceedings of Multiple Classifier Systems, LNCS, Vol. 4472, pp. 459–468, May 2007.

[21] M. Daisuke, H. Hazeyama and Kadobayashi, "A proposal of the AdaBoost-based detection of phishing sites", in Proceedings of the Joint Workshop on Information Security, Jan. 2007.

[22] L. V. Santhana and M. S. Vijaya. "Efficient prediction of phishing websites using supervised learning algorithms", Procedia Engineering, pp. 798–805, Dec. 2012

[23] C. Thornton, F. Hutter, H. H. Hoos and K. Leyton-Brown, "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms", in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 847–855, Aug. 2013.

[24] D. Michie, D. J. Spiegelhalter and C. C. Taylor, "Machine learning, neural and statistical classification", Ellis Horwood, Feb. 1994.

[25] Virus Total, "VirusTotal - Free online virus, malware and URL scanner", Online: `https://www.virustotal.com/en/`, Jan. 2012.