FINGERPRINTING TOR PROTOCOL NETWORK TRAFFIC

BY

MAJDI SAEED MOHAMMED BIN SALMAN

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

November 2014

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **MAJDI SAEED MOHAMMED BIN SALMAN** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE.**

Thesis Committee

Dr. Sami Zhioua
(Advisor)

Md. Rafiul Hassan

Dr. Md Rafiul Hassan
(Member)

Dr. Farag Azzedin
(Member)

عادل

Dr. Adel F. Ahmed
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

13/1/15

Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedication

To my Parents, my wife and my sons Saeed & Huda
for their endless love,support and encouragement

ACKNOWLEDGMENTS

In the name of Allah, most Kind, most Merciful. Thanks are to Allah who gave me strength, patience and ability to accomplish this research. I would like to thank my family who supported me during my study. I am particularly grateful to my wife who supported me during this work. I would like to thank King Fahd University for Petroleum and Minerals and Hadramout Establishment for Human Development for giving me the opportunity to complete my study in KFUPM university.My special sincere thanks go to sheikh Abdullah Ahmed Bugshan for his unlimited support.

No words can express my thanks to my thesis advisor, Dr. Sami Zhioua, for his guidance, support and advice.

I would like also to express my sincere gratitude to Dr. Md Rafiul Hassan for his support and advice.

Thanks are due to my thesis committee members Dr. Farag Azzedin for his cooperation, valuable comments, and support. .

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **v**

LIST OF TABLES **ix**

LIST OF FIGURES **xi**

LIST OF ABBREVIATIONS **xvi**

ABSTRACT (ENGLISH) **xvii**

ABSTRACT (ARABIC) **xix**

CHAPTER 1 INTRODUCTION **1**

CHAPTER 2 ANONYMITY SYSTEMS **6**

2.1 Anonymous Proxy Servers 7

2.2 Java Anonymous Proxy (JAP) 8

2.3 Tor protocol 9

2.3.1 Circuits and Onion Encryption 14

2.3.2 Tor Cell 17

CHAPTER 3 WEBSITE FINGERPRINTING **20**

3.1 Website Fingerprinting Defenses 21

3.2 Website Fingerprinting Attacks 22

CHAPTER 4 TRAFFIC CAPTURING AND DATA PROCESSING **27**
4.1 Data collection . . . 27
4.1.1 Environment setup . . . 28
4.1.2 Tools . . . 29
4.2 Data Processing . . . 36
4.2.1 Filtering Tor packet data . . . 37
4.2.2 Identifying Tor packet data for each website . . . 38
4.2.3 Extracting Tor packet sizes . . . 39
4.2.4 Extracting Inter packet time . . . 40

CHAPTER 5 FEATURE EXTRACTION AND CLASSIFIERS **42**
5.1 Feature extraction methods . . . 43
5.1.1 Edit distance . . . 43
5.1.2 Wavelet Packet Decomposition . . . 46
5.2 Classifiers . . . 53
5.2.1 Weka Calssification Tool . . . 53
5.2.2 Support vector machine . . . 55
5.2.3 Multilayer perceptron neural network . . . 60
5.2.4 Naive bayesian . . . 62
5.3 MEASURING PERFORMANCE . . . 63

CHAPTER 6 EXPERIMENTAL ANALYSIS **70**
6.1 Datasets . . . 70
6.2 Feature extraction method and evaluation . . . 71
6.2.1 Edit Distance - levenshtein distance- . . . 71
6.2.2 Wavelet Packet Decomposition . . . 81

CHAPTER 7 CONCLUSION **88**
7.1 Thesis Contributions . . . 88
7.2 Limitations of the Current Work . . . 90

7.3 Future Work 91

REFERENCES **93**

APPENDIX A: Visited Websites List **101**

APPENDIX B : Performance of SVM classifier and WPD method **103**

APPENDIX C : Performance of SVM classifier and WPD method **128**

VITAE **133**

LIST OF TABLES

4.1 Environment setup and Machine specification for website traffic generation and capturing 28

6.1 websites recognition accuracy of Edit distance method with different classifiers on Cai dataset 72

6.2 websites recognition accuracy of Edit distance method with different classifiers on the Chrome Browser 73

6.3 websites recognition accuracy of Edit distance method with different classifiers on Firefox Browser 74

6.4 websites recognition accuracy of Edit distance method with different classifiers on IE Browser 75

6.5 websites recognition accuracy of Edit distance method with different classifiers on Opera Browser 76

6.6 websites recognition accuracy of Edit distance method with different classifiers on Safari Browser 77

6.7 websites recognition accuracy of Edit distance method with different classifiers on Tor Browser 78

6.8 Performance metrics for Edit distance method + MultilayerPerceptron 80

6.9 WPD families Results 81

6.10 Cross validation performance metrics for Wavelet Bior1.5 method on our dataset 84

6.11 Cross validation performance metrics for Wavelet Bior1.5 method on our dataset 84
6.12 Cross validation performance metrics for Wavelet Bior1.5 method on our dataset 85
6.13 Cross validation performance metrics for Wavelet Bior1.5 method on our dataset 85
6.14 Performance metrics for Wavelet Bior1.5 + MultilayerPerceptron . 86

LIST OF FIGURES

2.1 proxy server connection 7
2.2 Mix network connection 8
2.3 Jap network Structure 9
2.4 IP packet structure 11
2.5 How Tor is working Step 1: Tor documentation source 12
2.6 How Tor is working Step2: Tor documentation source 13
2.7 How Tor is working Step3: Tor documentation source 14
2.8 Message encryption layers and routing path 16
2.9 Message with symmetric keys through Tor path 16
2.10 The format of Tor Cell 17
2.11 Relay Cell Payload Format 18
2.12 Circuit Creation Workflow 19

4.1 FireFox Browser configuration set up to use Tor 29
4.2 IE- Browser configuration set up to use Tor 29
4.3 Chickenfoot automation plugin 30
4.4 Chickenfoot script editor 31
4.5 Chickenfoot Java Script file 32
4.6 Chickenfoot output 33
4.7 Selenium WebDriver 34
4.8 Packet capture from the network using Windump 35
4.9 Sample of packet data captured by windump 36
4.10 IP's of Tor relays in local configuration file of Vidalia bundle . . . 38

4.11 Timing details for website page load 39
4.12 Timing details for Tor Captured packets 39
4.13 Sequence of Tor packet size for the website 1 for 5 visits(samples) 40
4.14 Sequence of IPT's for the website 1 for 5 visits(samples) 41

5.1 Pseudocode of DamerauLevenshtein distance 45
5.2 wavelet packet decomposition with 3 levels 48
5.3 Signal decomposition using WPD 49
5.4 Bior1.5 Wavelet and scaling functions 51
5.5 Bior1.5 Coefficients 51
5.6 Plot of packet Sizes for website1 sample1 of Firefox browser . . . 52
5.7 Level 4 WPD bior1.5 coefficients for website1 sample1 of Firefox browser 53
5.8 Attribute class values in Weka ARFF data file 55
5.9 The support vectors and the margin of the classifier. 56
5.10 Sigmoid activation function 62
5.11 The feed-forward network architecture and perceptron neuron model 62
5.12 Sample of classification accuracy for edit distance method on Chrome dataset 65
5.13 TP rate and Recall metrics for chrome dataset classification using edit distance method 66
5.14 FP rate for chrome dataset classification using edit distance method 67
5.15 Classification Precision for edit distance method on chrome dataset 68

6.1 Graphical presentation of website fingerprinting using Edit distance method on Cai and our datasets 72
6.2 Graphical presentation of website fingerprinting using Edit Distance method for Chrome Browser. 73
6.3 Graphical presentation of website fingerprinting using Edit Distance method for FireFox Browser. 74

6.4 Graphical presentation of website fingerprinting using Edit Distance method for IE Browser. 75
6.5 Graphical presentation of website fingerprinting using Edit Distance method for Opera Browser. 76
6.6 Graphical presentation of website fingerprinting using Edit Distance method for Safari Browser. 77
6.7 Graphical presentation of website fingerprinting using Edit Distance method for Tor Browser. 78
6.8 Performance metrics for ED+ SVM classification algorithm 79
6.9 Performance metrics for ED + MultilayerPerceptron classifier . . 80
6.10 Sample of chrome trace with the low and high-pass coefficients of Haar wavelet 82
6.11 Sample of chrome trace with the low and high-pass coefficients of Sym2 wavelet 82
6.12 Sample of chrome trace with the low and high-pass coefficients of Bior1.5 wavelet 83
6.13 Accuracy rate for the website fingerprinting using different WPD families 83
6.14 Performance metrics for WPD+ MultilayerPerceptron classifier . 86
6.15 SVM Vs NEURAL NETWORK(Time in seconds) 87

B.1 Accuracy of WPD first level for chrome browser 103
B.2 Confusion matrix of WPD first level for chrome browser 104
B.3 Accuracy of WPD second level for chrome browser 104
B.4 Confusion matrix of WPD second level for chrome browser 105
B.5 Accuracy of WPD third level for chrome browser 105
B.6 Confusion matrix of WPD third level for chrome browser 106
B.7 Accuracy of WPD fourth level for chrome browser 106
B.8 Confusion matrix of WPD fourth level for chrome browser 107
B.9 Accuracy of WPD first level for Firefox browser 107

B.10 Confusion matrix of WPD first level for Firefox browser 108
B.11 Accuracy of WPD second level for Firefox browser 108
B.12 Confusion matrix of WPD second level for Firefox browser 109
B.13 Accuracy of WPD third level for Firefox browser 109
B.14 Confusion matrix of WPD third level for Firefox browser 110
B.15 Accuracy of WPD fourth level for Firefox browser 110
B.16 Confusion matrix of WPD fourth level for Firefox browser 111
B.17 Accuracy of WPD first level for safari browser 111
B.18 Confusion matrix of WPD first level for safari browser 112
B.19 Accuracy of WPD second level for safari browser 112
B.20 Confusion matrix of WPD second level for safari browser 113
B.21 Accuracy of WPD third level for safari browser 113
B.22 Confusion matrix of WPD third level for safari browser 114
B.23 Accuracy of WPD fourth level for safari browser 114
B.24 Confusion matrix of WPD fourth level for safari browser 115
B.25 Accuracy of WPD first level for IE browser 115
B.26 Confusion matrix of WPD first level for IE browser 116
B.27 Accuracy of WPD second level for IE browser 116
B.28 Confusion matrix of WPD second level for IE browser 117
B.29 Accuracy of WPD third level for IE browser 117
B.30 Confusion matrix of WPD third level for IE browser 118
B.31 Accuracy of WPD fourth level for IE browser 118
B.32 Confusion matrix of WPD fourth level for IE browser 119
B.33 Accuracy of WPD first level for opera browser 119
B.34 Confusion matrix of WPD first level for opera browser 120
B.35 Accuracy of WPD second level for opera browser 120
B.36 Confusion matrix of WPD second level for opera browser 121
B.37 Accuracy of WPD third level for opera browser 121
B.38 Confusion matrix of WPD third level for opera browser 122
B.39 Accuracy of WPD fourth level for opera browser 122

B.40 Confusion matrix of WPD fourth level for opera browser 123
B.41 Accuracy of WPD first level for tor browser 123
B.42 Confusion matrix of WPD first level for tor browser 124
B.43 Accuracy of WPD second level for tor browser 124
B.44 Confusion matrix of WPD second level for tor browser 125
B.45 Accuracy of WPD third level for tor browser 125
B.46 Confusion matrix of WPD third level for tor browser 126
B.47 Accuracy of WPD fourth level for tor browser 126
B.48 Confusion matrix of WPD fourth level for tor browser 127
C.1 Accuracy of WPD for chrome browser 128
C.2 Confusion matrix of WPD for chrome browser 128
C.3 Accuracy of WPD for Firefox browser 129
C.4 Confusion matrix of WPD for Firefox browser 129
C.5 Accuracy of WPD for IE browser 130
C.6 Confusion matrix of WPD for IE browser 130
C.7 Accuracy of WPD for Opera browser 131
C.8 Confusion matrix of WPD for Opera browser 131
C.9 Accuracy of WPD for Tor browser 132
C.10 Confusion matrix of WPD for Tor browser 132

LIST OF ABBREVIATIONS

WF	:	Website Fingerprinting
WPD	:	Wavelet Packet Decomposition
DLD	:	Damerau-Levenshtein Distance
OSAD	:	Optimal String Alignment Distance
CV	:	Cross Validation
SVM	:	Support Vector Machine
MLP	:	Multi-Layer Perceptron
TPR	:	True Positive Rate
FPR	:	False Positive Rate
TNR	:	True Negative Rate
FNR	:	False Negative Rate

THESIS ABSTRACT

NAME:	Majdi Saeed Mohammed Bin Salman
TITLE OF STUDY:	Fingerprinting Tor Protocol Network Traffic
MAJOR FIELD:	INFORMATION AND COMPUTER SCIENCE
DATE OF DEGREE:	November 2014

Website fingerprinting (WF) is an attack on anonymity systems that affects browsing privacy. Many research works in the literature have used website fingerprinting to attack internet users with anonymity systems, in particular, Tor protocol. However, most of the fingerprinting attacks on Tor protocol have been studied based on dataset generated by a single web browser namely Firefox (Tor Browser version). In This thesis, a dataset of Tor protocol traffic was collected using the six popular web browsers, namely Firefox, Chrome, Internet Explorer, Opera, Safari and Tor Browser. Two feature extraction methods based on edit distance (ED) and wavelet packet decomposition (WPD) for WF on Tor protocol were investigated. ED algorithm was used to compute the similarity between the Tor traffic instances which is then used to train the classifiers. Different classifiers are applied for our extracted features and the accuracy computed by each classi-

fier was compared. A new approach based on WPD was applied to our generated dataset for feature extraction. The WPD was used to extract the approximation and detail coefficients for Tor packet sizes sequence for each website. An empirical analysis of applying these features for website fingerprinting using a freely available datasets (Cai dataset) and our datasets has been carried out. To the best of our knowledge, this is the first work in the literature that uses all popular web browsers to collect the datasets and uses wavelet packet decomposition method for extracting the features of Tor traffic packet sequences for the website. The empirical analysis showed promising results which are comparable to similar work in the literature. This confirms our initial intuitions that WPD method is suitable for use with website fingerprinting focusing on packet size, packet order and sequence. Our work also shows the different results of website fingerprinting with respect to the major web browsers.

ملخص الرسالة

الاسم الكامل: مجدي سعيد محمد بن سلمان

عنوان الرسالة: التعرف على حركة بيانات الإرسال لنظام التور

التخصص: علوم الحاسب الآلي

تاريخ الدرجة العلمية: نوفمبر 2014

آلية التعرف على هوية المواقع الالكترونية هو هجوم إلكتروني على أنظمة إخفاء الهوية التي تؤثر على خصوصية التصفح لمستخدمي الانترنت . عدد من الدراسات والبحوث أستخدمت آلية التعرف على هوية المواقع الالكترونية لمهاجمة مستخدمي الانترنت الذين يستخدمون أنظمة إخفاء الهوية وبالتحديد نظام التور. معظم تلك الدراسات والبحوث أعتمدت في دراساتها على على قاعدة بيانات تم إنشائها بإستخدام متصفح إنترنت وحيد وهو متصفح تور فايرفوكس (إصدار خاص بنظام التور).

في هذه الدراسة تم إنشاء قاعدة بيانات لحركة بيانات الإرسال لعدد من مواقع الانترنت عبر نظام التور بإستخدام عدد من متصفحات المواقع الأكثر شيوعاً في الإستخدام وهي متصفح الفايرفوكس, متصفح الجوجل كروم, متصفح الانترنت إكسبلورور, متصفح الاوبرا, متصفح السفاري وأخير المتصفح الخاص بنظام التور. كما تم إستخدام وتطبيق طريقتين مختلفتين لإستخراج الميزات والسمات من بيانات حركة الإرسال للمواقع وهما طريقة خوارزمية مسافة ليفنشتاين وطريقة تحليل المويجات . تم إستخدام خوارزمية مسافة ليفنشتاين لحساب التشابه بين مجموعتين من بيانات حركة الارسال لنفس الموقع أو موقعين مختلفين وتدريب المصنفات للقيام بالتمييز ومعرفة مدى التشابه في بيانات الحركة للموقعين . كما تم تطبيق عدد من المصنفات على السمات والميزات المستخرجة بطريقة خوارزمية مسافة ليفنشتاين وتم مقارنة نتائج دقة التمييز للمصنفات المستخدمة في الدراسة. كما تم أيضاً تطبيق طريقة جديدة في هذه الدراسة وهي طريقة تحليل المويجات لاستخراج الميزات والسمات من بيانات حركة الارسال للمواقع من قاعدة البيانات المنشئة. طريقة تحليل المويجات تعتمد على إستخراج معاملات التقريب ومعاملات التفاصيل من متسلسلة تضم أحجام مختلفة لحزمة من البيانات لكل موقع إنترنت في قاعدة البيانات.

كما تم في هذا العمل عرض النتائج التجريبية الحاصلة من تطبيق الميزات السابقة بإستخدام قاعدة البيانات التي تم تطويرها في هذا العمل وكذلك بإستخدام قاعدة بيانات أخرى متوفرة (قاعدة بيانات تم إستخدامها في دراسة سابقة).

وتعد هذه المرة الأولى - وذلك على حد علمنا - التي تتم فيها إنشاء قاعدة بيانات لدراسة آلية التعرف على هوية المواقع الالكترونية بإستخدام عدد من متصفحات المواقع الأكثر شيوعاً وكذلك تطبيق طريقة تحليل المويجات لإستخراج السمات من بيانات حركة الارسال لمواقع الانترنت التي تمت زيارتها.

التجارب التحليلية لهذه الدراسة أظهرت نتائج واعدة يمكن مقارنتها بنتائج الدراسات السابقة , وهذا يؤكد حدسنا الاول بأن طريقة تحليل المويجات يمكن إستخدامها في التعرف على هوية المواقع الالكترونية بالاعتماد على إستخراج السمات من حجم وترتيب حزمة بيانات الارسال للمواقع التي تم زيارتها. كما أن هذه الدراسة أظهرت إختلاف نتائج التعرف على هوية المواقع الالكترونية عند إستخدام متصفحات مختلفة للمواقع.

CHAPTER 1

INTRODUCTION

Since last two decades, the internet has had a significant impact society and businesses. In both fields, sharing information for the society or business purpose has become the main factor for the internet users and service providers. Therefore, a lot of attention has been given to achieve the privacy objective for the internet users to hide their identities from the eavesdropping adversary.

Information privacy can be defined as the right to be free from surveillance. The main concern of the internet users is to share their personal information with the third party without letting the unwanted observers to access their information.

Many systems have been developed to achieve the privacy objective by encrypting the client data precisely the client identity and his website destination address and content. One of these systems is a Tor anonymity system. The Tor anonymity system is low-latency anonymity system [1, 2, 3] that is being used by 500,000 users everyday [2]. The Tor Anonymity Network consists of 4000 relays [4] from which circuit of three relays is built to route the client data to destination [2, 5].

In other words, the communication between the user client and web server for requesting specific website/web-service through the Tor network is routed through a number of volunteer relays using multiple layers of encryption [2].

Most practical attacks against the Tor system are based on traffic analysis. Attackers have ability to reveal some information about the client and the visited website identity by observing and analyzing the packet traffic and extracting patterns that primarily consist of specific features of the traffic packet such as packet time, packet size, order and direction of the packets, etc.

website traffic fingerprinting is " an attack where the adversary attempts to recognize the encrypted traffic patterns of specific web pages" [2]. The attackers collect the traffic packets for the target websites (websites visited by the client through the Tor system) and extract some features to be used in the classification process. Training and testing the extracted features will be used to identify and classify the website class (assigning the traffic to a certain website) by calculating the similarities between the features of the trained traffic packets and the new traffic packets using some classification techniques such as Support Vector Machine (SVM)[2] and Multinomial Naive-Bayes (MNB)[6].

Recently, few researches proposed website fingerprinting techniques on Tor anonymity system [6, 7, 8]. These studies have shown that applying website fingerprinting on Tor system is more challenging than other anonymity systems for three main factors[2]:

1. The size of data unit sent through the tor network is fixed (512 bytes)

2. More than one circuit of three relays are used to transfer the information from the Tor client, which affect the performance in terms of network latency, network bandwidth and transfer congestion.

3. Data transferred through Tor traffic are affected with unnecessary additional data which are resulted from the activities performed by Tor such as network construction, network testing and controlling network.

Despite the challenges of applying website fingerprinting on the Tor anonymity system, the studies have shown the possibility of obtaining a high accuracy rate over 80% based on simplified feature extraction that aim to remove features (e.g SENDMEs control cell [2], ACKs [4, 5]) that provide no useful information and reduce the accuracy of the classification.

This research investigates a new websites fingerprinting technique based on wavelet packet decomposition method using websites packet traces visited by different web browsers. Most of existing works have reported their results based on packet sizes, packet order, packet sequence and directions as a features to train the classifiers. Recently edit distance methods have been more involved in website fingerprinting [2, 5]. To the best of our knowledge, all the previous work have been implemented their approaches using dataset collected from Tor browser. In this work, we have built our own dataset for websites packet traces using six different browsers (Firefox, Chrome, Internet explorer, Safari, Opera and Tor). We have introduced the preprocessing steps to build up our dataset starting from capturing, parsing, analyzing, filtering the traffic of the websites. Besides, we have developed automation scripts to automate websites visiting process using different web browsers. We validate our generated dataset for website fingerprinting by using two different approaches. We firstly implemented Edit distance method for website fingerprinting on our generated dataset. We used Levenshtein distance to calculate the distances between the traces of certain sample of websites traces and we used different classifiers to evaluate the classification method. Then we introduced a new method based on wavelet packet decomposition. The wavelet packet decomposition method was applied to the packet sequences of the websites traces to extract the pattern for the classification. The wavelet transform method has been widely used in pattern recognition, especially in image processing and signal processing, face classification and audio classification. To the best of our knowledge, wavelet method was not used for website fingerprinting in the previous works.

The main contributions of this work are as follows:

1. Building a new dataset for website fingerprinting using six different web browsers. All previous website fingerprinting techniques evaluate their approach to the data collected by Tor browser.

2. Feature extraction and selection. We investigate two different feature extraction methods using edit distance and wavelet packet decomposition. We applied the two feature extraction method on our generated dataset.

3. Evaluating the feature extraction method using different classifiers (support vector machine, MultilayerPerceptron and the Naive Bayes).

This thesis is organized as follows: Chapter 2 provides a background of anonymity systems. It surveys the structure of the most anonymity systems and how they work. Tor anonymity protocol structure and work mechanism has been extensively presented in this chapter since it is the main subject of this research. Chapter 3 surveys the website fingerprinting techniques at attack type and level, preprocessing phase, the used features, classifiers, the obtained accuracy, and the datasets used by researchers. The different phases of building the dataset (traffic capturing, packets filtering and data preprocessing) using different web browsers are presented in chapter 4. Chapter 5 discusses the different methods for extracted the features of the packet traces of the websites and the used classifiers. The results of our experiments are discussed in chapter 6. finally, conclusions are presented in chapter 7.

CHAPTER 2

ANONYMITY SYSTEMS

The communication networks use addresses to route the traffic, which are visible to anyone observing the network. These addresses (such as IP addresses, or Ethernet MACs) are the unique identifiers of the users that appear in their communication. Linking theses address to the users will compromise their privacy. Therefore, anonymizing the communication is necessary to protect the privacy of users against traffic analysis and attackers and prevent them to obtain the sensitive information and Identities of users .

Anonymous communication systems play a vital role in protecting privacy of people from network surveillance and traffic analysis. They provide the ability for the users to hide their network identity and prevent the observers to know the actual source or destination of messages. Most of Anonymous systems work based on transmitting the traffic via one or more proxies and encrypting the traffic. These systems are classified to low-latency and high-latency anonymous systems.

In this chapter we present a background for the main low- latency Anonymous systems that are designed for interactive applications such as web browsing. The presented low- latency Anonymous systems share the deterministic routing feature which means the path of nodes or proxies for sending the traffic is known in advance.

2.1 Anonymous Proxy Servers

Proxy servers are the systems or applications that work as a mediator between the client and other servers. One type of proxy server is the forward proxy that used to retrieve the requests to the user and connect the user with the target. On the other word, the target is communicating with the proxy server as the owner of the request and the real user will be unknown to the target. Figure 2.1 shows the proxy server connection.The client connects to the proxy server through the client's ISP for requesting some resources or services such as web page. The proxy server evaluates the request and contacts the web server to retrieve the requested web page.

Figure 2.1: proxy server connection

2.2 Java Anonymous Proxy (JAP)

Java Anonymous Proxy (JAP) is a client application developed for AN.ON project, which is also known as (JonDo). JAP and AN.ON is developed by The University of Dresden in 2001 [9].

The anonymity part is taking place in JAP proxy when the user connects to the web server using Mix networks. Mix networks are a chain of proxy servers which are used to deliver the messages from different users to different destinations. A chain of proxy servers is known as mixes. The messages or requests from the users to the destinations will be mixed using Mix Cascades approach to achieve the anonymity and observability. In such case the data traffic will be unrecognizable to whom it belongs to [9, 10].

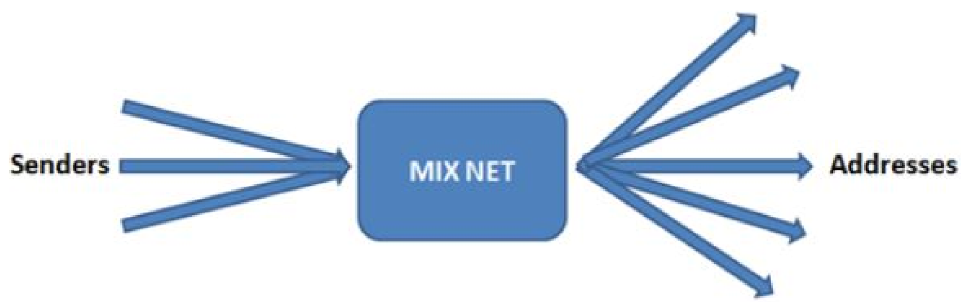


Figure 2.2: Mix network connection

As shown in figure 2.3, the JAP client application is installed in the client computer (referred as User A, User B and User C) and combined with web browser for the anonymity objective. Using the info service, a list of the available cascades (mixes) will be retrieved to the JAP client to choose the desired mix and connects the user in this mix (cascade). It's the responsibility of the JAP client to check the traffic load, online users, availability cascades etc.

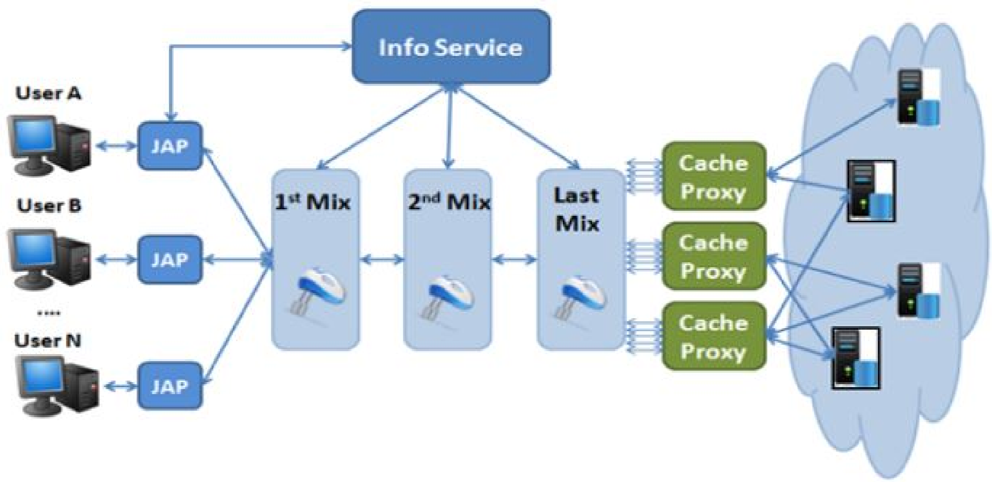


Figure 2.3: Jap network Structure

In the JAP application, the data traffic is encrypted based on the mixes that are used. Symmetric encryption (RSA 1024+ bit key length) is used for key encryption (between the mixes) and asymmetric encryption (AES 128 bit key length) for the data traffic for better efficiency [9]. The encryption process has taken place within the mixes, which mean that when the data traffic leaves the cascade, it will be unencrypted and requires the users to provide another encryption method for such level.

2.3 Tor protocol

Tor protocol is an anonymous communication systems that can be used as a virtual tunnel for the Internet user to be communicated privately and securely. Tor protocol has been designed and developed by Onion Routing technology project to resist the traffic analysis and network observation. It is a free software and an open network that has been implemented based on an onion routing to achieve the online anonymity for the client [4, 11]. Tor protocol is the second generation onion router system with many advantages compared to the previous generation.

Besides the anonymity browsing services that are provided by Tor, and using the instant messaging application without leaking the contents and objects of the conversation, Tor application can be used to break the blocked websites by Internet service provider and do not leak any identifiable information to that Internet service provider.

Traffic analysis is a common surveillance technique on the Internet that can be used reveal the address and location information on communication sides. Traffic data packet consists of two main parts, the first part is data payload which is the content that will be transmitted to the recipient. The data payload part is encrypted commonly. The second one is the packet header that keeps information regarding to the IP addresses of the source and destination. This part of the traffic packet is unencrypted which means that the traffic analysis can access to the IP address information and retrieve more information about the users. Figure2.4 shows the structure of the IP packet.

As we see from the header part of the IP packet, it discloses reasonable information such as the source and destination IPs, size, timing, and so on. With the expose of source and destination information, the eavesdropper can easily get sensitive information about the Identity and the location of the Internet user.

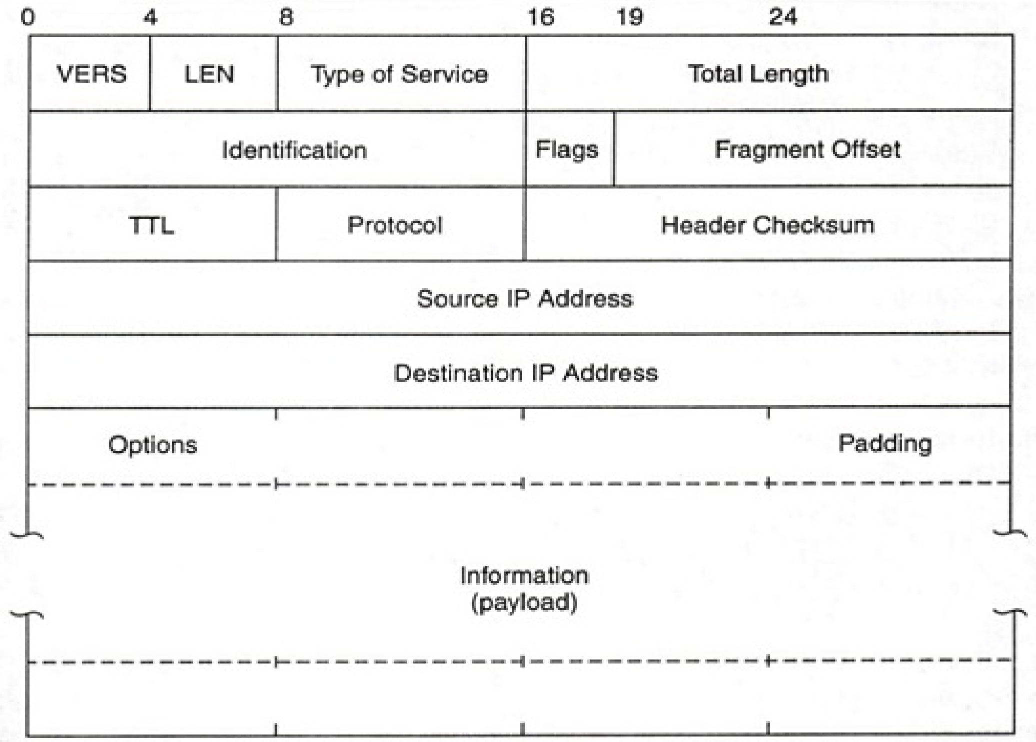


Figure 2.4: IP packet structure

Currently, Tor network is considered the most widely used low latency anonymity networks with approximately 500,000 users daily and it has been consisted of more than 4000 nodes known as Onion Routers distributed across the world [12].

By default, Tor circuits consist of three nodes, the guard node which is the only node that recognize the client's identity, the middleman that is responsible for exchanging the encrypted cell between the guard node and the exit node, and the exit node which is the only node that recognize the destination identity.

Tor client contacts Tor network, which creates a random path to the destination server. As the Tor client wants to communicate with the destination server with a hidden identity, he will send the message to the destination through different Tor machine node on the Tor network. Therefore, the first step is for the Tor client to contact a directory server to obtain a list of Tor nodes. The directory server is a Tor node that provides the list of Tor nodes to the Tor client to choose a list of nodes - usually three nodes- to send the message to the destination server through them. Figure 2.5 shows the first step of communication to obtain a list of Tor nodes from a directory server.

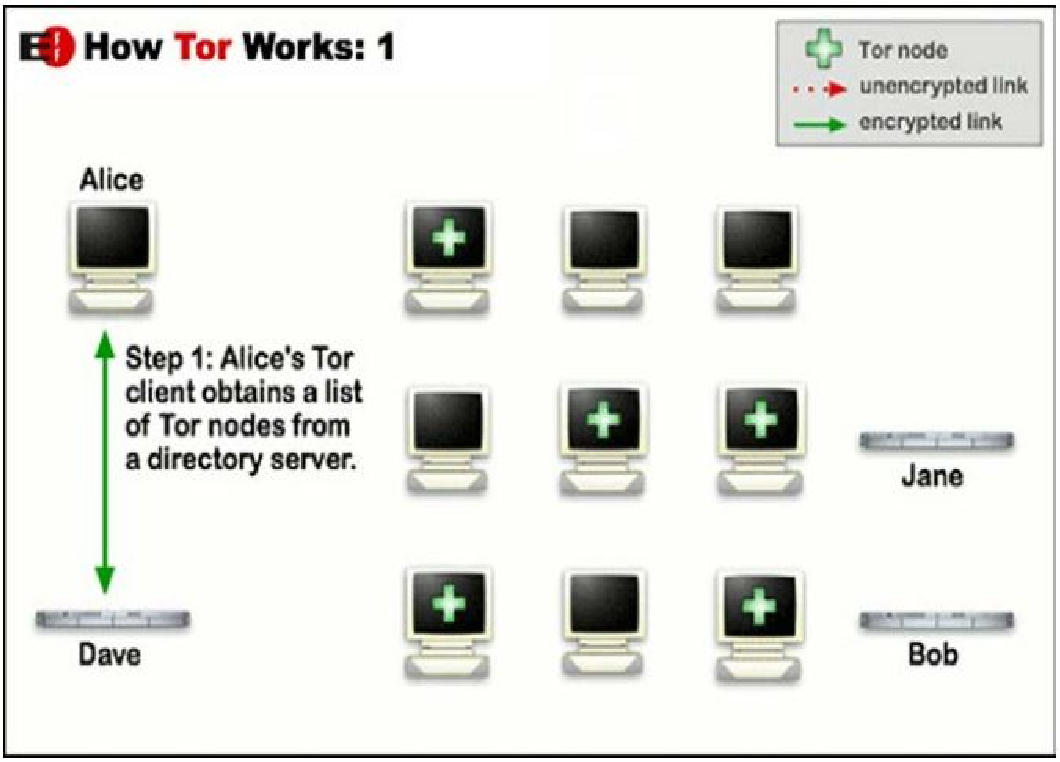


Figure 2.5: How Tor is working Step 1: Tor documentation source

When the path of the selected Tor network is established, the Tor client can send the message to the first node (guard node) which will forward the next node in the path (the middleman) until it reaches the exit node, the last node in the selected path before the destination server. The exit node is responsible of delivering the message destination server. The destination server will deal with the exit node as the origin of the message request which means that the identity of the original client will be hidden from the destination server. Figure 2.6 shows the chosen path from the Tor client to the destination server.

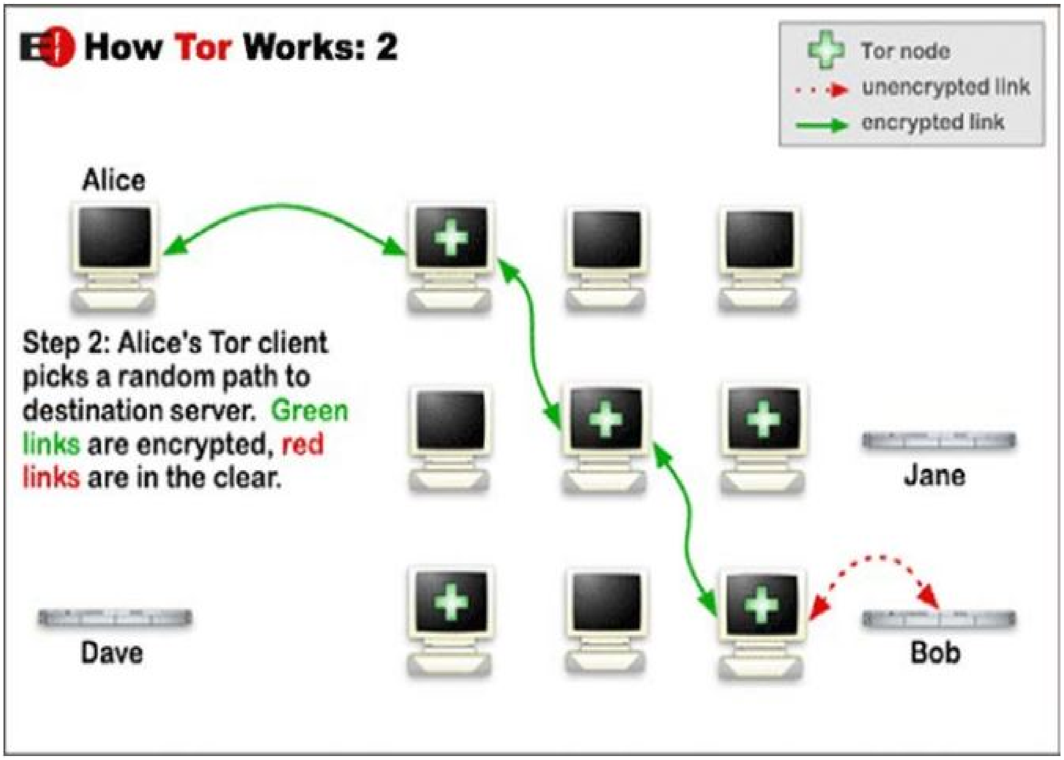


Figure 2.6: How Tor is working Step2: Tor documentation source

The selected Tor network can be changed at any time due the inactivity of one of the tor network nodes or the time of tor network activity reached for a certain time limit (as per Tor project documentation, the time limit for the selected path to be active must be exceed 10 minutes). Figure 2.7 shows the changing of the Tor network path.

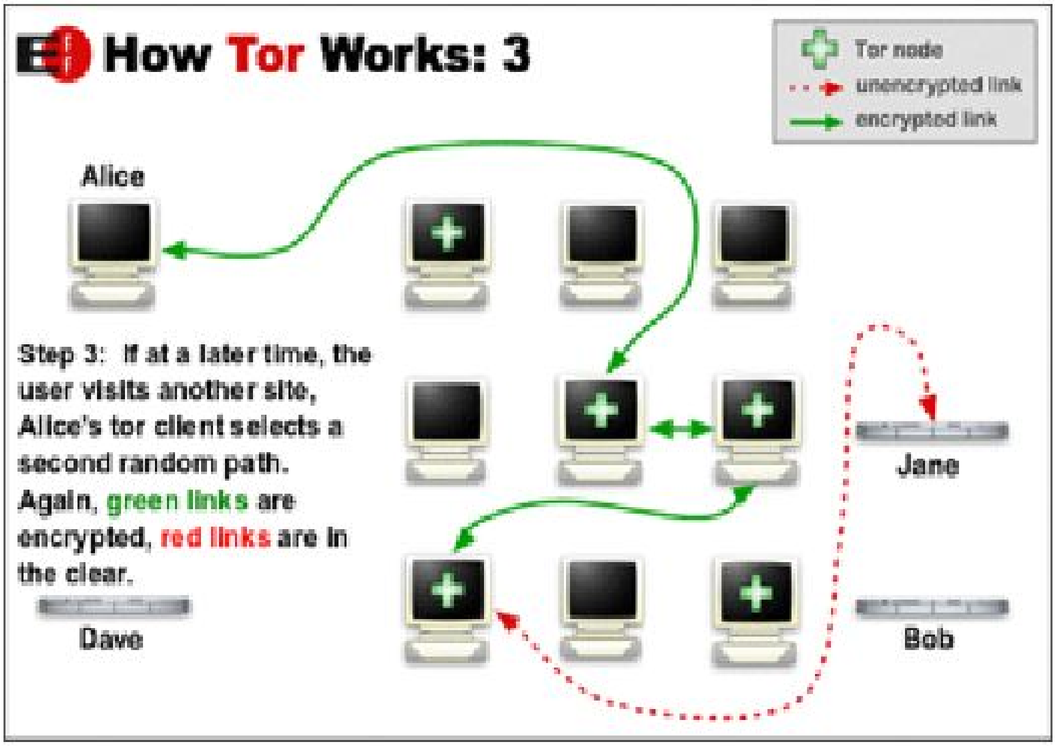


Figure 2.7: How Tor is working Step3: Tor documentation source

Using the Tor protocol, the data traffic or stream is divided into fixed sized cells, the main aspect of encryption and traffic anonymity on Tor protocol, encrypted with identifying key for the Tor circuit nodes.

2.3.1 Circuits and Onion Encryption

An onion proxy (OP) is used to handle the Tor circuit creation and encryption transparently. Tor protocol uses layered encryption, which mean the encryption process is taking place between the onion routers to ensure that each onion router knows only the adjacent node in the Tor circuit. Since the traffic between each two onion nodes of the Tor circuit is completely different in the encryption, it will be difficult for the attacker and traffic analysis to compare the traffic between two adjacent nodes in the Tor circuit. However, the traffic between the exit node and destination, where the client is connecting to, is not encrypted, which means that the encryption process is taking place within the routers (nodes) of the Tor network.

The encryption process of the traffic between the Tor circuit nodes is done by negotiating a symmetric key with each node and encrypting the messages with the negotiated key in every node. Figure 2.8 shows the process of message encryption. Message is encrypted using different encryption layers and then sent through the circuit. A plain text message is first encrypted with the public key of the third relay, then another encryption layer is added with the public key of the middle relay, and finally an encryption layer with the public key of the entry relay. That message will be processed as follows when it is transferred: Like someone peeling or removing the outer cover of the onion, each onion router removes the encryption layer using its public key and forward the message next router. This process is repeated until the message arriving to the last node in the Tor circuit. This process of removing the encryption layers, using the public key of each Tor node, preventing the intermediary nodes from getting address information (source and destination of the massage) and content of the message.

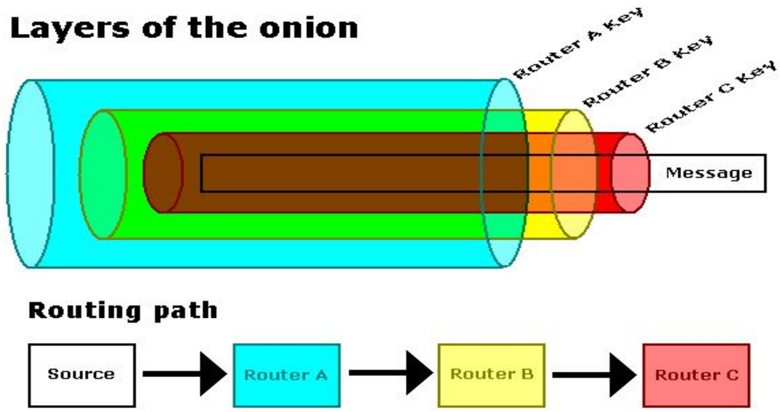


Figure 2.8: Message encryption layers and routing path

Let's assume that Alice (Tor client) uses the Tor protocol to communicate with Bob (destination) through a Tor network circuit consisting of of three tor routers (R1, R2, R3). Symmetric keys (K1, K2, and K3) will be negotiated through the nodes circuit. When the message M is sent, M will be encrypted with K3K2K1 respectively. As the message passes through the circuit, the first node in the circuit decrypts the message with its symmetric key. The R1 will decrypt M using K1, R2 will decrypt M using K2 and so on. Figure 2.9 shows the decryption process of the message through the Tor path.

$$Alice \xrightarrow{[M]_{K_{3,2,1}}} R_1 \xrightarrow{[M]_{K_{3,2}}} R_2 \xrightarrow{[M]_{K_3}} R_3 \xrightarrow{M} Bob$$

Figure 2.9: Message with symmetric keys through Tor path

2.3.2 Tor Cell

In Tor protocol, the Tor routers are communicating by using Tor cells. Tor cell is 512-bytes long formatted as shown in figure 2.10:

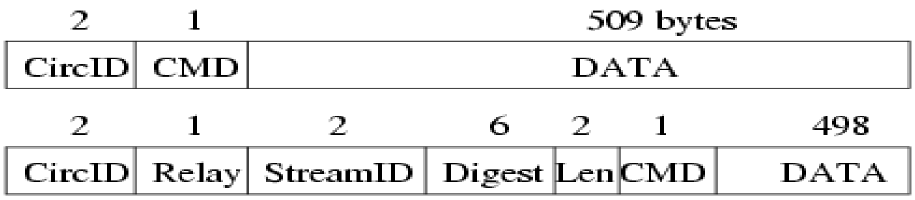


Figure 2.10: The format of Tor Cell

There are two types of Tor cells [13]:

1. Control Cell: the value of the command filed of the control cell is:

    (a) CELL PADDING: used for keepalive and optionally used for link padding, although not used currently. [13].

    (b) CELL CREATE: used to initiate a connection between two Tor processes. This command mainly used to create the first hop between the onion proxy and the first onion router in the circuit. It is also used to extend the Tor circuit by one hop through the communication between the Tor onion nodes.

    (c) CREATED: used for the confirmation of the Create cell command.

    (d) DESTROY: used for releasing a Tor circuit.

2. Relay cell: The command field (Command) of a relay cell defines the purpose of the relay cell. BEGIN, END, and CONNECTED relay commands are

used for setting up and demolishing TCP streams on the Tor circuit. In addition, the DATA relay command is used to send data through TCP stream. For constructing a new circuit EXTEND and EXTENDED relay cells are used. The recognized field relay cell header is used to tell the onion router whether the cell is fully decrypted by setting its value to zero. If the cell is fully decrypted, then the digest will be the four bytes of the running digest of all of the bytes destined for or originated from this hop in the circuit. [13] The StreamID field is used by the onion proxy and the exit router to differentiate between various streams on the Tor circuit. Finally, the Length field of the relay cell indicates to the number of bytes of the data field that contain the real data. Figure 2.11 shows relay cell format.

Relay command	'Recognized'	StreamID	Digest	Length	Data
1 byte	2 bytes	2 bytes	4 bytes	2 bytes	498 bytes

Figure 2.11: Relay Cell Payload Format

Figure 2.12 below shows the circuit creation Workflow. The diagram shows the steps and the commands of creation Tor circuit that consists of three router nodes R1, R2 and R3. Alice is the client Tor who is running the onion proxy. K1, K2 and K3 are the symmetric keys assigned during the Tor circuits creation

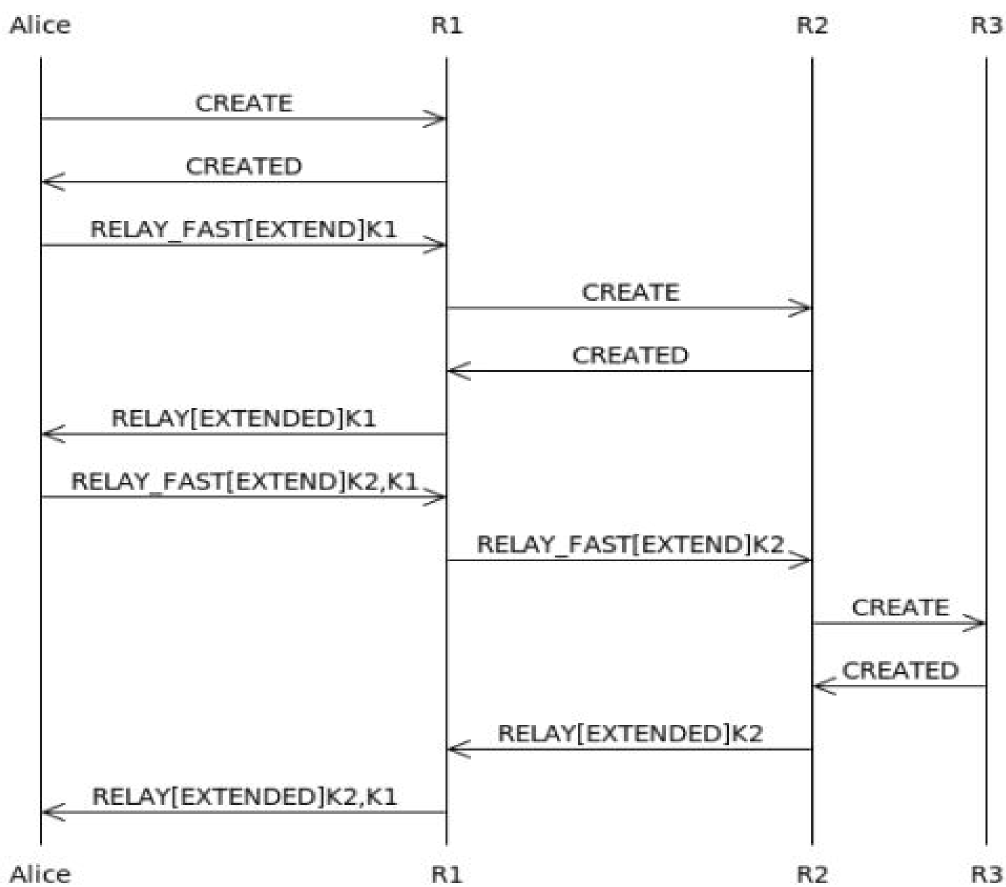


Figure 2.12: Circuit Creation Workflow

CHAPTER 3

WEBSITE FINGERPRINTING

Website fingerprinting [14] is a variant of passive traffic analysis that can be carried out by a local eavesdropper or by any entity observing Tor client traffic. In traffic analysis attack, the adversary analyses the traffic to extract patterns that can reveal the identity of the website accessed by the client. Patterns are constructed from certain features in the traffic such as the size of transferred data, the timing, the order of packets, etc.

Website fingerprinting was first used to analyze encrypted HTTP traffic [14, 15, 16, 17]. Most of these attacks were based on tracking the size the objects fetched by the main web page. With the migration to HTTP/1.1 which makes use of persistent connections and pipelining, it is no longer possible to easily distinguish between single objects fetching. Only a few works focused on implementing website fingerprinting on anonymity systems [2, 5, 6, 7, 8, 18]. It turned out that website fingerprinting is much challenging when applied on anonymity systems in particular Tor. The reason is that Tor protocol performs some structural modifications in the traffic: restructuring the traffic into fixed size cells, merging small packets together, multiplexing TCP streams, etc. However, despite these challenges, recent works showed that the accuracy of website fingerprinting could be as high as 87% [5] and 91% [2] when applied on Tor.

In this chapter, a survey of the previous works on website fingerprinting is presented. We classified the previous works to defenses and attacks approaches.

3.1 Website Fingerprinting Defenses

Fu, et al. [19] present a defense approach based on inserting dummy packets with randomized intervals between the packets. They claimed that using non-constant intervals between the introduced dummy packets will reduce the success rate of the traffic analysis. Padding packets schemes were proposed in different works to defeat the traffic analysis [16, 17, 20]. They proposed different techniques for padding packets, such as pad-to-MTU, exponential padding, random padding, etc. Dyer, et al., proved that most of the padding schemes were ineffective against their evaluated attacks [21]. An alternative approach to packet padding approach was proposed by Wright et al. [22]. They introduced a traffic morphing scheme to transform the distribution of source packet size using splitting or padding techniques for the packets to imitate the distribution of the target website. A new scheme of traffic morphing, based on padding and fragmenting the packet sizes to n-grams, was proposed by [23]. Dyer, et al., also showed that the defense approach based traffic morphing was ineffective [21].

Dyer, et al., proposed new defense approach known as Buffered Fixed-Length Obfuscator (BuFLO) for hiding the loading total time and bandwidth. The approach works by transmitting fixed-length packets at fixed interval for fixed time. Dyer, et al claimed that their new defense BuFLO approach doesnt leak the packet timing information which helps in reducing the best attack recognition rate.

Another approach called HTTPOS defense has been proposed in [24]. The proposed approach allows the client to hide the actual packet length by inserting request objects using maximum TCP segmentation, size to reduce and hide packet sizes [24].

The first successful attack on the Tor protocol was proposed by Panchenko et al. [7]. As a result of this successful attack, new defense was developed by Tor developers to resist the new successful attack. The new defense approach works by enabling the HTTP pipelining which allows the Multiple Simultaneous Requests taken into account that the pipeline size the order of the request are randomized.

3.2 Website Fingerprinting Attacks

Several researches studied attacks on anonymity systems from different angles. In this section, we present the previously published works on website fingerprinting attacks and we address the systems or protocols that each proposed attack target. Features and parameters used for each attack will be addressed, including the datasets to test and evaluate the attack.

Fingerprinting attacks have been first proposed against the encrypted web traffic system such as SSH, VPN, IPsec or WPA. [6, 14, 15, 16, 17, 23, 25]. The proposed attacks focused mainly on packet length.

Liberatore and Levine[15] proposed an approach for website fingerprinting based on length of two directions packet (incoming and outgoing packets). They used packet size frequencies for a traced traffic to classify the observed instance of packet size of the tested website. Naive Bayes classifier has used in this work for classification purpose. They got good results based on the packet size frequency histogram. They applied their approach under the simple encryption system (SSH tunnel) not under the Tor system.

Herrmann et al. [6] used text mining techniques for website fingerprinting [2] and obtained better classification results than the previous work of Liberatore and Levine. Both [6, 15] used the packets size, frequency histogram and discarded the other two main elements in traffic tracing which are ordering and timing. They tested their experiment on 775 websites [6]. Multinomial Naive Bayes classifier (MNB) was used with consideration that the packet length frequency is used as an exponent to the relevant probability value. However, their recognition rate of the Tor system was very low (only 3%) compared with the recognition rate obtained by the simple encryption system. Low recognition accuracy under Tor anonymity system shows that website fingerprinting on Tor system is more challenging than website fingerprinting on simple encrypting systems.

Shi et al [8] has presented a new approach for website fingerprinting on the Tor. In his study, the number of incoming or outgoing packets of top 20 websites in Japan in certain time was traced and then the time for a certain number of packets were represented as a vector [8]. The study has reported identification rate of 50%.

Panchenko et al. [7] presented an approach for website fingerprinting on two anonymity system (Tor and JonDonym) and used extra features rather than the packet size and direction features that have used in previous works. Some of the new features that the study have used are: sizes of all packets in one direction during interval time, packet size marker, packet number marker, the percentage of incoming bytes, etc.. [7]. Support Vector Machine classifier has used under Weka environment[26] to classify the website fingerprinting On anonymity networks (Tor and JonDonym). The study used the same data set of websites that have used in Herrmann et al. [6] in the closed-world experiment part. The study succeeded to increase the recognition rate under Tor from 3% to 55%.

Aggarwal, et al. [27] proposed complete analysis of security and privacy of the modern browsers. They showed that each of modern browser has each own design structure and mechanism to support the privacy in the browsing. The study has classified the attacks that private browsing tries to avoid to local attacks and web attacks. Complete analysis of the implementation of the most four popular browsers from the point of view of security of the browsing has conducted in the study. As per Aggarwal, et al. [27], browser extensions and plugins has a negative effect to achieve the goals of private browsing.

One of the most recent works for website fingerprinting on Tor is Cai et al.[5].

In this study, traffic traces are represented as a sequence of positive packet lengths for outgoing packets and negative packet lengths for incoming packets. The distance between the traffic traces is computed by optimal string alignment distance algorithm[5]. Support Vector Machine has used with a distance based kernel. They used Damerau-Levenshtein[28, 29] edit distance to calculate the distance between the traces and normalizing this distance with respect to the lengths of the shortest trace between the two traces. The study used 800 websites according to Alexa for classification and evaluation purpose. The filtered websites were visited under Tor system In closed world model. The study has reported accuracy recognition rate of 87% for the visited websites.

The most recent contribution was by Tao Wang et al [2]. The study claims that Collecting Data on Tor should be more accurate since there are a number of factors that may have a negative impact on the obtained results. As per the study, the factors that should be taken into account are circuit construction, timing and website localization[2]. Using Tor controller, different circuits and modification of the top sites list are the procedures that the study used them to avoid the impact in data collection because of the previous factors. The study also proposed new data processing approach based on Tor cell sequences instead of TCP/IP packet sequences. The study has reported accuracy of 91% as compared to 87% of accuracy obtained previous works in the closed-world experiments [2]. relatively high accuracy rates, most of the existing works depend on packet size, packet length frequency and packet ordering as main features for classification.

It is obvious that using packet sizes without the timing and order information to attack the Tor traffic is more challenging than the other encrypted systems due to strong padding packet mechanism of tor system that provides less information [5, 6, 7, 8, 18].

We strongly think that using a feature extraction method that depend on packet sizes and packet order that represents the time sequence of the packet size will have an impact on the accuracy of the website fingerprinting. We strongly think that using wavelet packet decomposition as feature extraction of the packet sizes sequence as the wavelet method has proven to be used successfully different fields of pattern recognition [30, 31, 32, 33, 34, 35, 36, 37].

Besides that, all existing works carried out their experiment on websites using a single web browser to evaluate the accuracy of the techniques. We developed our own dataset using the six common browsers to evaluate our proposed method and investigate the impact of the website fingerprinting on different web browsers.

CHAPTER 4

TRAFFIC CAPTURING AND DATA PROCESSING

In this research, we conducted our experiments based on two data sets of packet sizes for the websites visited through the Tor protocol. The first dataset shared by Cai [5] which consists in traffic data for 100 websites and 40 samples. The second dataset was generated by us using different tools for visit automation and packet capturing. This chapter presents the design and implementation of our data collection process. It presents the environment setup and processes of collecting and processing the data using special tools for each process.

4.1 Data collection

For the purpose of this research, we have built and developed a dataset for websites fingerprinting. The developed dataset contains packet traces for 20 websites and 15 samples for each website visited by six different browsers. Each website trace contains a sequence of packet sizes of website visit session under the Tor protocol. The following subsections describe the environment setup and tools that are used for collecting- capturing- the raw packet data for the visited websites through the Tor protocol.

4.1.1 Environment setup

A client Machine was set up and utilized for the purpose of website traffic generation, capturing and processing. Table 4.1 shows the specifications of the machines and systems that were used to collect and process the data (website traffic). We started collecting and capturing the website traffic using the Ubuntu operating system and Firefox 3.6 version. We used the Chickenfoot automation tools for automating the websites visiting. Since our main goal of this research is to investigate the website fingerprinting in different browsers, we set up a second machine with windows 8 operating system and we used Selenium WebDriver tools with some code modification to automate websites visiting on all six browsers.

	Dataset 1	Dataset 2
CPU	Intel Core i5	Intel Core2 Duo
RAM	6 GB	4 GB
Operating System	Microsoft Windows 8	Ubuntu 10.04.4
Web Brwoser	FireFox,Chrome,IE,Safari,Opera and Tor	FireFox
Tor Vidalia	Vidalia 0.2.21	Vidalia 0.2.21
packet analyzer	Windump	Tcpdump
web automation/scripting	Selenium	Chickenfoot

Table 4.1: Environment setup and Machine specification for website traffic generation and capturing

4.1.2 Tools

Several tools were used in data collection phase to generate and develop the data set for website fingerprinting. We used web browser automation tools to automate a browser to visit a list of websites. At the same time, a tool for capturing and analyzing the website traffic was used to capture the raw packets and display them in human-readable format.

4.1.2.1 Tor Vidalia Bundle

In order to connect to Tor network and configure the browsers to use the Tor protocol, we used Vidalia Bundles for Windows which comes with Tor and Vidalia (a cross-platform graphical controller for the Tor) [38]. The called browsers were configured to use Tor protocol by setting up the SOCKS5 proxy server to localhost or 127.0.0.1 with port 9050.

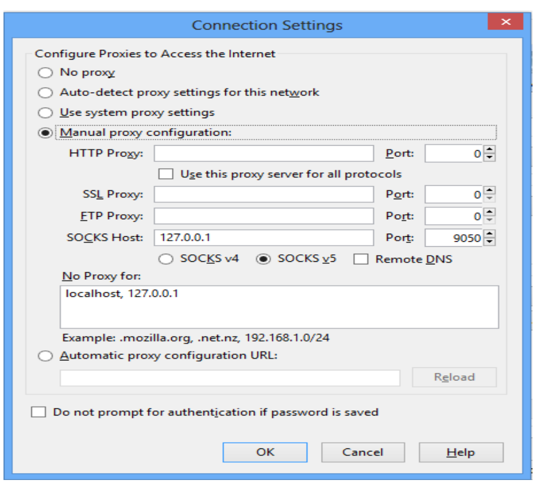


Figure 4.1: FireFox Browser configuration set up to use Tor

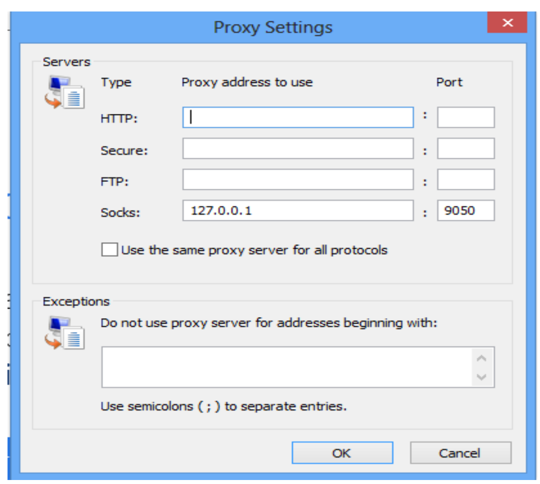


Figure 4.2: IE- Browser configuration set up to use Tor

4.1.2.2 Chickenfoot

Chickenfoot [39] is a Firefox extension and add-on that provides a programming environment in the browsers sidebar which enable users to write scripts to automate and customize the web browsing. The scripts are written with predefined Chickenfoot functions to perform specific web tasks.

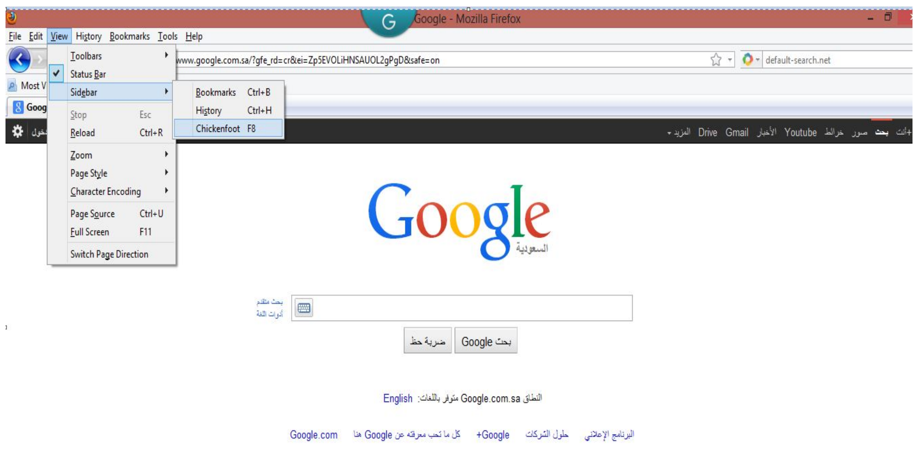


Figure 4.3: Chickenfoot automation plugin

Chickenfoot supports the Firefox browser 3.6. It's available as a sidebar option in the Firefox view menu. Chickenfoot plugs contains a JavaScript editor that allows to enter the command in JavaScript. The lower portion of the Chickenfoot sidebar presents an interface with four tabs: Output, Patterns, Actions, and Triggers. Here's a brief description of the tabs cite miller2010rewriting:

- **Output:** shows the results or output of the running script.
- **Patterns:** shows the search patterns for locating common elements in a page.
- **Actions:** Contains all user actions within the browser page.

- **Triggers:** contains the scripts that will be automatically run when a certain Web page is visited.

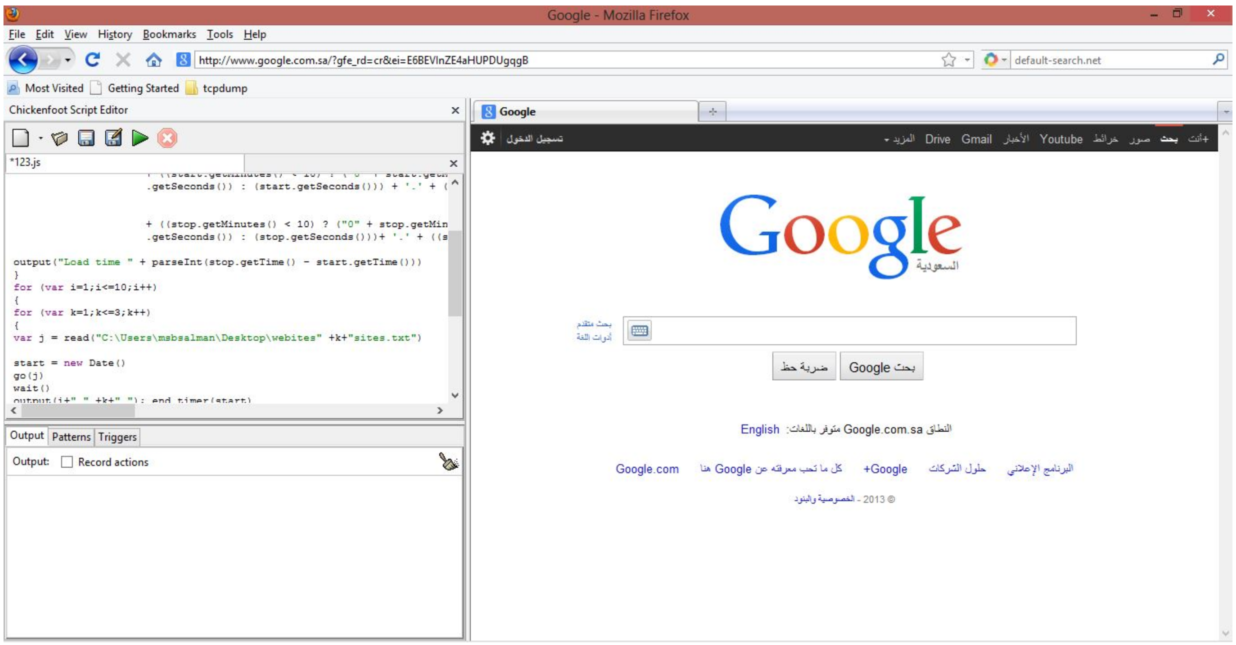


Figure 4.4: Chickenfoot script editor

For our automation purpose, we create a JavaScript file to perform the websites visiting automation using predefined functions of Chickenfoot such as go (), wait () and output (). We put 5 seconds interval time between every website visit to avoid packets correlation for website packets.

123.js

```
include("fileio.js")
function end_timer(start){
    stop = new Date()

output("start time " + start.getHours() + ':'

                        + ((start.getMinutes() < 10) ? ("0" + start.getMinutes()) : (start.getMinutes())) + ':' + ((start.getSeconds() < 10) ? ("0" + start
                        .getSeconds()) : (start.getSeconds())) + '.' + ((start.getMilliseconds() < 10) ? ("0" + start.getMilliseconds()) : (start.getMilliseconds()

                        + ((stop.getMinutes() < 10) ? ("0" + stop.getMinutes()) : (stop.getMinutes())) + ':' + ((stop.getSeconds() < 10) ? ("0" + stop
                        .getSeconds()) : (stop.getSeconds()))+ '.' + ((stop.getMilliseconds() < 10) ? ("0" + stop.getMilliseconds()) : (stop.getMilliseconds())));

output("Load time " + parseInt(stop.getTime() - start.getTime()))
}
for (var i=1;i<=20;i++)
{
for (var k=1;k<=20;k++)
{
var j = read("/home/majdi/Desktop/" +k+"sites.txt")

start = new Date()
go(j)
wait(5)
output(i+" " +k+" "); end_timer(start)

}
}
```

Figure 4.5: Chickenfoot Java Script file

The output file of the websites visit automation will include the website number, visit number, start time, end time and website loading time interval. Start time and Stop time for the website visits will be in milliseconds.

```
9 1
start time 2013-12-13 14:18:07.901 stop time 2013-12-13 14:18:27.418 Load time 19517
9 2
start time 2013-12-13 14:18:27.431 stop time 2013-12-13 14:18:56.483 Load time 29052
9 3
start time 2013-12-13 14:18:56.486 stop time 2013-12-13 14:20:22.547 Load time 86061
9 4
start time 2013-12-13 14:20:22.548 stop time 2013-12-13 14:21:21.969 Load time 59421
9 5
start time 2013-12-13 14:21:21.972 stop time 2013-12-13 14:22:16.499 Load time 54527
9 6
start time 2013-12-13 14:22:16.507 stop time 2013-12-13 14:22:58.710 Load time 42203
9 7
start time 2013-12-13 14:22:58.716 stop time 2013-12-13 14:24:17.319 Load time 78603
9 8
start time 2013-12-13 14:24:17.325 stop time 2013-12-13 14:24:48.921 Load time 31596
9 9
start time 2013-12-13 14:24:48.927 stop time 2013-12-13 14:26:31.655 Load time 102728
9 10
start time 2013-12-13 14:26:31.661 stop time 2013-12-13 14:27:02.446 Load time 30785
9 11
start time 2013-12-13 14:27:02.456 stop time 2013-12-13 14:27:56.774 Load time 54318
9 12
start time 2013-12-13 14:27:56.811 stop time 2013-12-13 14:28:37.891 Load time 41080
9 13
start time 2013-12-13 14:28:37.904 stop time 2013-12-13 14:29:08.267 Load time 30363
9 14
start time 2013-12-13 14:29:08.298 stop time 2013-12-13 14:29:29.958 Load time 21660
9 15
start time 2013-12-13 14:29:29.965 stop time 2013-12-13 14:29:42.541 Load time 12576
9 16
start time 2013-12-13 14:29:42.545 stop time 2013-12-13 14:31:08.663 Load time 86118
```

Figure 4.6: Chickenfoot output

4.1.2.3 Selenium WebDriver

Selenium WebDriver is one of the most used tools for browser automation. It is a portable, open source software available for Windows that allows users and developers alike to use and develop a functional process to drive and automate the browser. Selenium has been developed using JavaScript that any browser that support JavaScript [40] .

Selenium WebDriver has been used in this work to automate the process of visiting a list of websites by different browsers. We developed Java code with Selenium WebDriver using browser webdriver to make direct calls to the browser and automate the visiting of a list of websites. Figure 4.7 shows our customized codes to call the WebDrivers of the used web browsers.

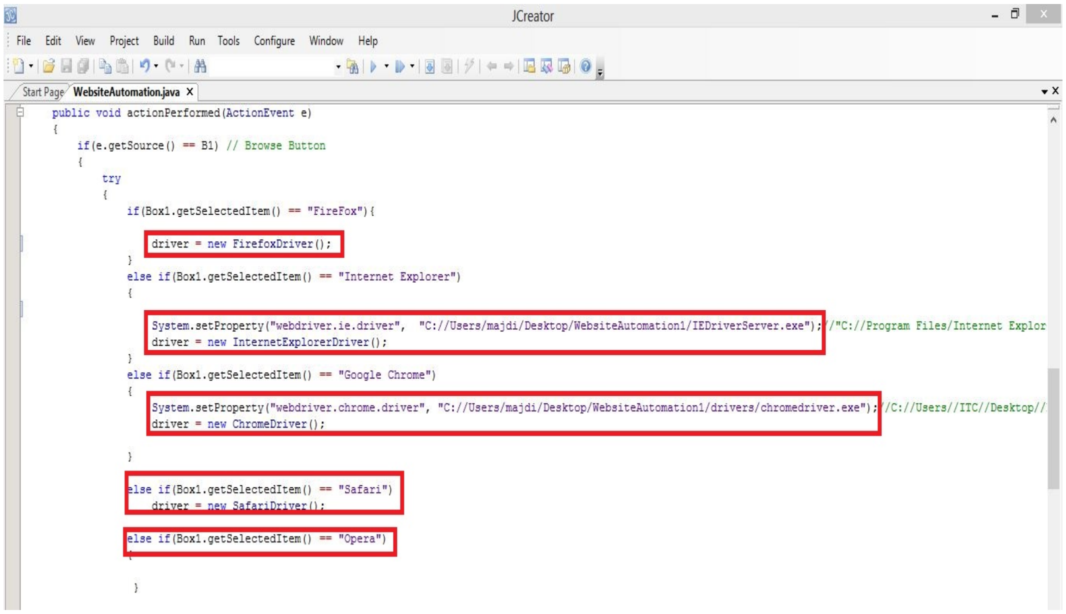


Figure 4.7: Selenium WebDriver

For every website visit, when a website is fully loaded, a detailed timing information is provided, including the start time and end time and total load time - in microseconds - for website loading. We put a 5 second gap between two website visits to avoid the overlap in capturing the raw packet data for each website. The browser cache property was disabled and all cached website page contents were deleted after the website is fully loaded to guarantee that the next visit to the same website will load all page contents from the server and provide the actual time load for the visited website.

4.1.2.4 WinDump

The WinDump tool (windows version of tcpdump) is used in the windows environment to capture and view the data packets for the network traffic. WinDump is an executable file available on http://www.winpcap.org/windump/. It runs on Command Prompt window and provides options to display and specify the network interfaces, filter the captured network packets,etc . . .

Figure 4.8 shows the windump command options and parameters that are used in this work:

1. -i2 : For specifying which Ethernet interfaces that the network traffic will be captured from it.

2. -nn : To avoid converting the address and port numbers to names.

3. -tttt : To view a timestamp in default format for each packet.

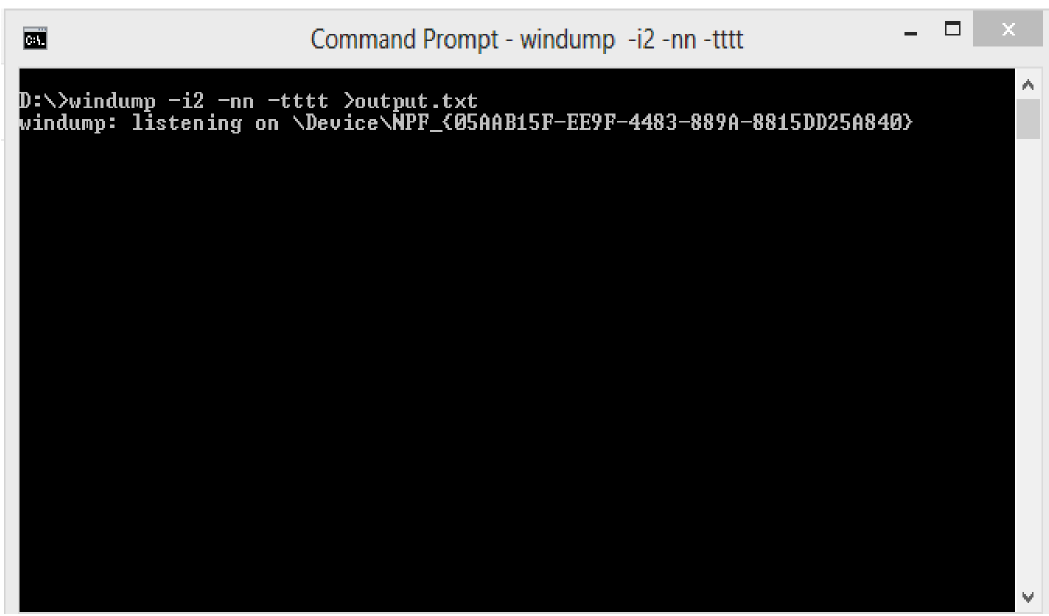


Figure 4.8: Packet capture from the network using Windump

The WinDump program can be run with the w or >options which cause the packet data to be saved in data files such as .txt or .pcap data format.

Figure 4.9 shows sample of the data packets captured using windump tool.

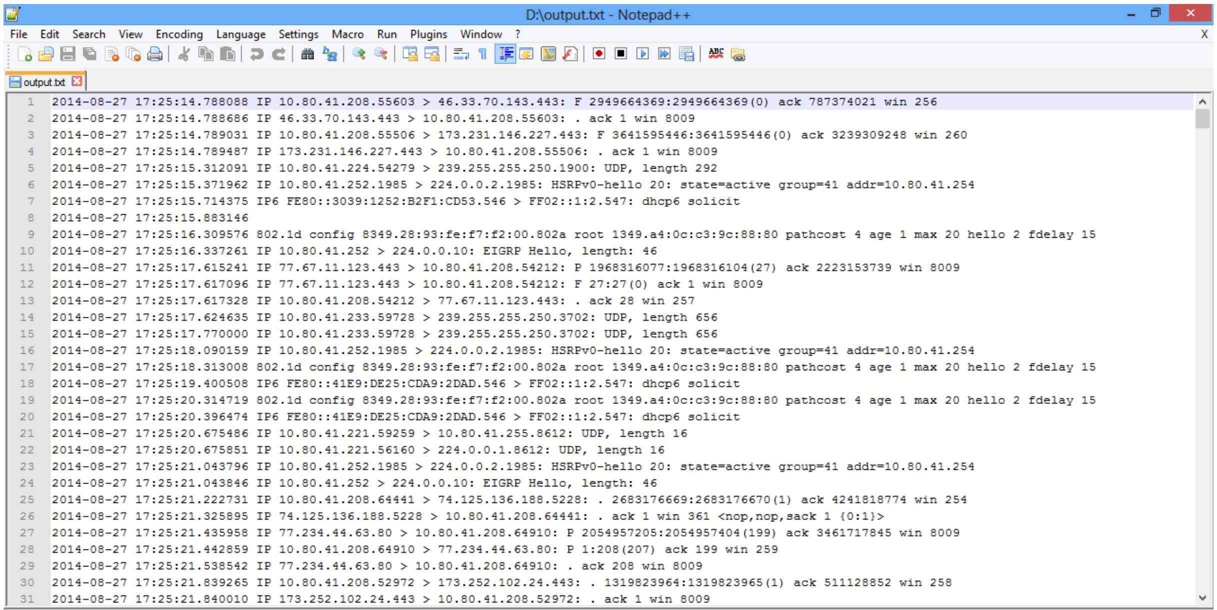


Figure 4.9: Sample of packet data captured by windump

4.2 Data Processing

During the data collection phase, WinDump was used to capture the network traffic on the determined Ethernet interface. The process of websites visiting with selenium webdriver tool generates significant network traffic, which is captured using WinDump. As a result of the WinDump capturing process for the network traffic, a raw packet data will be stored data file format.

Twenty websites are selected to be visited automatically using our developed tool 20 times. As a result of the data processing phase, we end up with 400 files (20 website x 20 sample). We exclude undesirable website traces, usually the website traces that contain no packets or have too few numbers of packets compared to the other traces for the same website, and we select the top 15 samples out of 20 samples for every website.

Next subsections show how we filtered the raw packet data for building our dataset.

4.2.1 Filtering Tor packet data

In this stage, we filter the captured packets based on whether the packets are Tor packet or non-Tor packet. To achieve this goal, we have implemented a filter to check whether the source or destination IPs in the data packet match any IP in the Tor relays IPs list. We keep updating for all Tor relays IPs by two different ways:

1. We extract the all relays nodes IPs from the local configuration file of the Tor Vidalia bundle. These IPs are all IPs used to construct the Tor path network. The relays configuration file provides other information about the date and time that the relays were in use and the ports that the relays accessed through. Figure 4.10 shows the local configuration file with the name of cached-microdesc-consensus that contains ip's of Tor relay node. In this file, each line starts with r character has an ip for Tor relay.

2. We obtain the latest Tor exit node IPs list from https://collector.torproject.org/archive/exit-lists/. And all the latest Tor relies IPs list from https://collector.torproject.org/relay-descriptors/microdescs/.

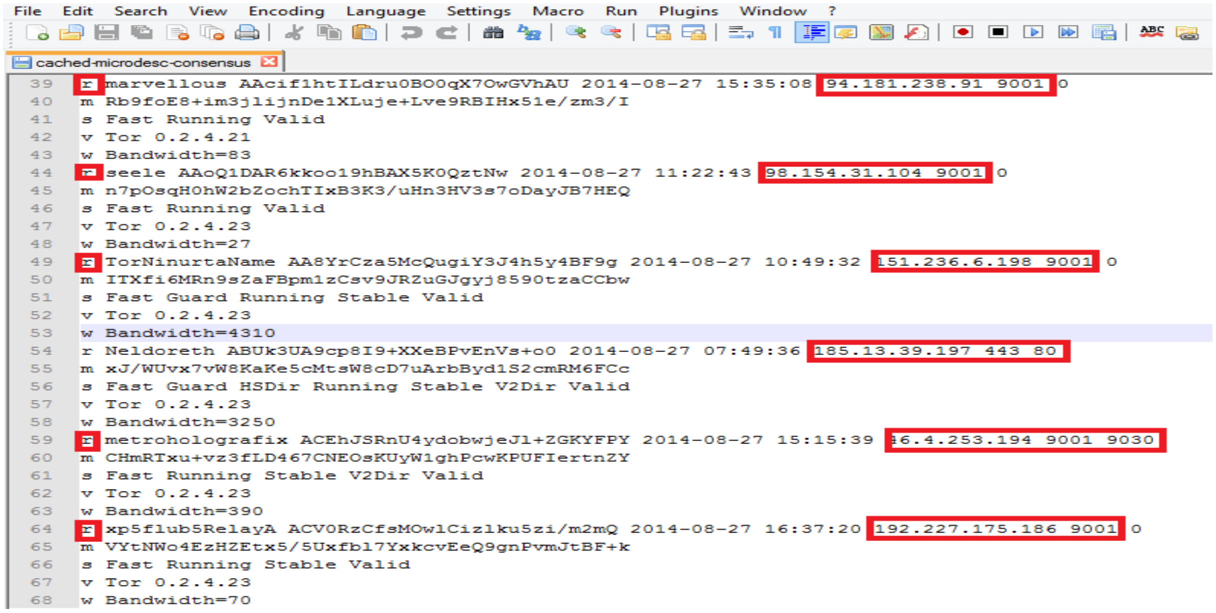
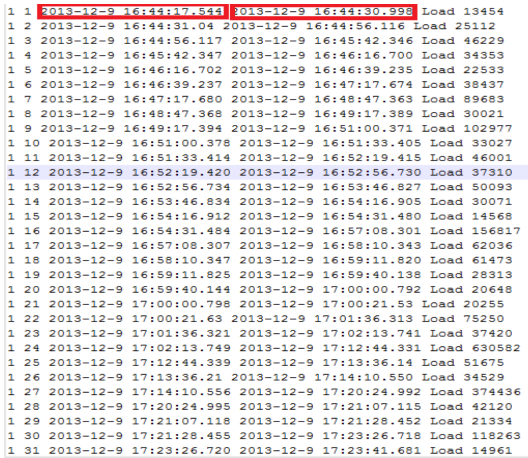


Figure 4.10: IP's of Tor relays in local configuration file of Vidalia bundle

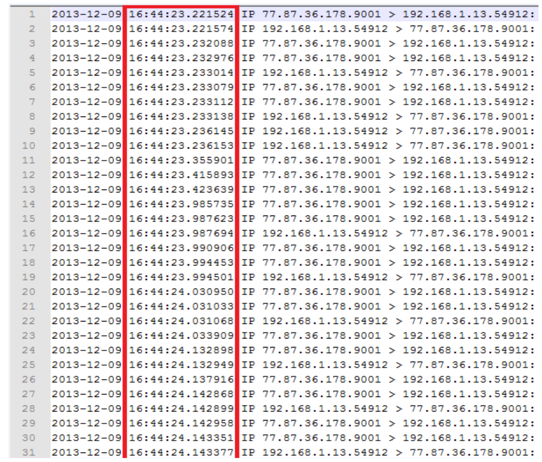
4.2.2 Identifying Tor packet data for each website

In this stage, the start and stop times for loading website page were used to compare with the time stamp of the packets captured by WinDump. If the time stamp of the packet being captured is located within the time interval for website page loading, the packet will be exported to a file that contains all packets for a specific website and visit. The timestamps for both website loading and packets capturing were in microseconds to avoid missing some packets that related to certain websites. Figure 4.11 shows the start time and stop time for loading certain website that compared with the time stamp of the captured packet presented in figure 4.12.



```
1 1 2013-12-9 16:44:17.544 2013-12-9 16:44:30.998 Load 13454
1 2 2013-12-9 16:44:31.04 2013-12-9 16:44:56.116 Load 25112
1 3 2013-12-9 16:44:56.117 2013-12-9 16:45:42.346 Load 46229
1 4 2013-12-9 16:45:42.347 2013-12-9 16:46:16.700 Load 34353
1 5 2013-12-9 16:46:16.702 2013-12-9 16:46:39.235 Load 22533
1 6 2013-12-9 16:46:39.237 2013-12-9 16:47:17.674 Load 38437
1 7 2013-12-9 16:47:17.680 2013-12-9 16:48:47.363 Load 89683
1 8 2013-12-9 16:48:47.368 2013-12-9 16:49:17.389 Load 30021
1 9 2013-12-9 16:49:17.394 2013-12-9 16:51:00.371 Load 102977
1 10 2013-12-9 16:51:00.378 2013-12-9 16:51:33.405 Load 33027
1 11 2013-12-9 16:51:33.414 2013-12-9 16:52:19.415 Load 46001
1 12 2013-12-9 16:52:19.420 2013-12-9 16:52:56.730 Load 37310
1 13 2013-12-9 16:52:56.734 2013-12-9 16:53:46.827 Load 50093
1 14 2013-12-9 16:53:46.834 2013-12-9 16:54:16.905 Load 30071
1 15 2013-12-9 16:54:16.912 2013-12-9 16:54:31.480 Load 14568
1 16 2013-12-9 16:54:31.484 2013-12-9 16:57:08.301 Load 156817
1 17 2013-12-9 16:57:08.307 2013-12-9 16:58:10.343 Load 62036
1 18 2013-12-9 16:58:10.347 2013-12-9 16:59:11.820 Load 61473
1 19 2013-12-9 16:59:11.825 2013-12-9 16:59:40.138 Load 28313
1 20 2013-12-9 16:59:40.144 2013-12-9 17:00:00.792 Load 20648
1 21 2013-12-9 17:00:00.798 2013-12-9 17:00:21.53 Load 20255
1 22 2013-12-9 17:00:21.63 2013-12-9 17:01:36.313 Load 75250
1 23 2013-12-9 17:01:36.321 2013-12-9 17:02:13.741 Load 37420
1 24 2013-12-9 17:02:13.749 2013-12-9 17:12:44.331 Load 630582
1 25 2013-12-9 17:12:44.339 2013-12-9 17:13:36.14 Load 51675
1 26 2013-12-9 17:13:36.21 2013-12-9 17:14:10.550 Load 34529
1 27 2013-12-9 17:14:10.556 2013-12-9 17:20:24.992 Load 374436
1 28 2013-12-9 17:20:24.995 2013-12-9 17:21:07.115 Load 42120
1 29 2013-12-9 17:21:07.118 2013-12-9 17:21:28.452 Load 21334
1 30 2013-12-9 17:21:28.455 2013-12-9 17:23:26.718 Load 118263
1 31 2013-12-9 17:23:26.720 2013-12-9 17:23:41.681 Load 14961
```

Figure 4.11: Timing details for website page load



```
2013-12-09 16:44:23.221524 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.221574 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:23.232088 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.232976 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.233014 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:23.233079 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.233112 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.233138 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:23.236145 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:23.236153 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:23.355901 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.415893 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.423639 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.985735 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.987623 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.987694 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:23.990906 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.994453 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:23.994501 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:24.030950 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:24.031033 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:24.031068 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:24.033909 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:24.132898 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:24.132949 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:24.137916 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:24.142868 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:24.142899 IP 192.168.1.13.54912 > 77.87.36.178.9001:
2013-12-09 16:44:24.142958 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:24.143351 IP 77.87.36.178.9001 > 192.168.1.13.54912:
2013-12-09 16:44:24.143377 IP 192.168.1.13.54912 > 77.87.36.178.9001:
```

Figure 4.12: Timing details for Tor Captured packets

4.2.3 Extracting Tor packet sizes

After we identified the Tor packet data for each website and visit, we extract the Tor packet size of the raw packet data and assign the positive or negative sign based on whether the packet is incoming or outgoing. Packet sizes of value zero were excluded since they are usually acknowledgment packets that provide no information. Figure 4.13 presents sample of packet sizes for website 1 for five visits that extracted.

Website1 Sample 1	Website1 Sample 2	Website1 Sample 3	Website1 Sample 4	Website1 Sample 5
565	565	565	565	565
565	565	565	-565	565
-565	-565	-565	-1360	-565
565	565	565	-282	565
565	565	565	565	565
-565	-565	-565	-565	-565
-565	-565	-565	565	-565
565	565	565	565	565
565	565	565	-565	565
-565	-565	-565	-565	-565
565	565	565	-565	-565
-565	-565	-565	565	565
565	565	565	565	565
-565	565	565	-565	-565
565	-565	-565	-1360	565
-565	-565	-565	-335	-1360
-1360	565	565	-1360	-88
-1360	1077	1077	-282	-1360
-1360	-565	-565	-1360	-88
-191	-565	-565	-1360	-1360
565	565	565	-1360	-52
1130	565	565	-687	565

Figure 4.13: Sequence of Tor packet size for the website 1 for 5 visits(samples)

4.2.4 Extracting Inter packet time

The time between two packets arriving at a host is known as inter-packet arriving time (IPT). We have extracted the inter packet time of the arriving packets from the Tor packet data for each website and visit. We used the IPT's for every two arriving packets in the Tor packet data sequence as a features for the Tor traffic of the website. We investigated the ability of using the IPT as a feature for the Tor traffic for websites classification but unfortunately we have obtained poor accuracy. This is due to the timing issue of the Tor circuit which including the time to construct and choose the Tor circuit beside the time to select another random.

Website 1 Sample 1	Website 1 Sample 2	Website 1 Sample 3	Website 1 Sample 4	Website 1 Sample 5
19.976	698.278	0.025	27.08	23.877
420.974	257.003	16.631	0.008	242.951
900.772	2512.66	200.064	215.932	0.087
130.321	204.236	55	0.079	303.82
2512.376	125.234	3.103	72.15	73.694
237.499	206.787	0.095	8.288	58.069
0.083	166.711	163.848	175.996	0.051
173.232	198.334	174.354	102.089	0.079
681.293	35.698	0.049	90.225	37.811
0.989	236.019	4.55	0.107	0.065
130.976	3499.792	104.036	115.646	3.448
82.832	2.436	3.513	117.662	191.535
2.859	130.87	0.053	0.062	50.44
133.323	3067.07	11.451	1.523	199.638
2691.018	1.024	0.119	263.134	288.715
2.028	130.416	4.818	0.081	2.228
208.881	1333.628	0.809	15.163	374.974
3171.279	197.508	122.76	1.55	533.489
2.025	117.126	183.051	133.416	25.358
135.687	4.489	3.473	92.246	35.208
3.158	129.318	0.048	45.553	0.06

Figure 4.14: Sequence of IPT's for the website 1 for 5 visits(samples)

CHAPTER 5

FEATURE EXTRACTION AND CLASSIFIERS

Over the last few years, several researchers addressed the problem of website fingerprinting recognition. Most of the available feature extraction methods were based on the packet information features such as the packet size, packet direction etc.. Some researchers used integrated feature extraction methods such packet size with editing distance [2, 5]. Even though the published results in these papers are very good, some issues are still remaining like using different browsers for collecting data and extract the features and investigating other classifiers for the classification.

One of the goals of this research is to use wavelet feature extraction method for website fingerprinting recognition with acceptable recognition rates. We have investigated different types of wavelet functions using the packet size sequence as a feature of website traces. One of the investigated methods (wavelet packet decomposition) was used in [41] with a neural network classifier. In [41], wavelet method was used to extract several hidden features of the time-frequency information of network traffic.

In this chapter, several types of features are extracted for website fingerprinting. Two main features extraction methods were used in this research work. We implemented the edit distance feature extraction method of Cai work [5] on our new developed dataset. We also investigated a new feature extraction method (wavelet method for signal processing) using two different datasets. Size and direction of the Tor packet for the website are the primary data input in both Cai dataset and our developed dataset. Both feature extraction methods depend on extracting features from the sequence of the packet size. The results of the new wavelet extraction method show that our extraction method is a suitable for the website fingerprinting and it opens new direction to use signal processing techniques for website fingerprinting.

5.1 Feature extraction methods

5.1.1 Edit distance

The similarity and dissimilarity between two sequences can be calculated based on the number of operations needed to transform one sequence into another. Given a set of packet size sequences, the distance between pairs of them helps in finding the similarity between the two sequences and derive structural relationship amongst them.

A traffic instance under Tor protocol is represented as a sequence of packet sizes with positive or negative sign indicating to outgoing and incoming packets respectively. In order classify the traffic traces and get acceptable success rate of website fingerprinting, distance-based metrics have been used in some previous work. Distance-based metrics compute the sequence of the packets size as they train the classifiers for website fingerprinting

5.1.1.1 Optimal String Alignment Distance(OSAD)

Optimal string alignment distance (OSAD) was used in previous work to calculate the distance between two traffic traces by identifying the minimal number of operations (insertions, deletions, substitutions and transpositions) required to transfer one sequence to another. It takes into account that the transpositions operations can be held only on the adjacent elements of the string.

The cost of each operation can be assigned differently without affecting the validity of the optimal string alignment distance algorithm [2]. For example the distance between the strings xyz and zx will be 3 operations (delete y, delete z, insert z before x) instead of 2 operations (delete y, transpose z and x) since the restriction of transposition operation for the non-adjacent elements. The cost assigned for the insertions, deletions, substitutions operations should be higher than the cost of the transposition operation to keep distance symmetry [2]

5.1.1.2 Damerau-Levenshtein Distance (DLD)

The distance between two sequences of fixed length where minimum transform operations are applied to transform one sequence into other is referred as Damerau-Levenshtein Distance [42].

Damerau-Levenshtein Distance describes the number of insertions, deletions, substitutions and transpositions operations required to transform a sequence of characters to another [2]. Damerau-Levenshtein Distance differs from Optimal String Alignment Distance in the restriction of the transportation operation. Damerau-Levenshtein distance remove the restriction on transpositions operation. Figure 5.1 shows the pseudo code of Dameraul Levenshtein distance.

```
int DamerauLevenshteinDistance(char S[1..M], char T[1..N])
 {
    declare int d[0..M, 0..N]
    declare int i, j, cost
    for i from 0 to M
      d[i,0] := i //the distance of any first string to an empty second string
    for j from 0 to N
      d[0, j] := j //the distance of any second string to an empty first string
    for i from 1 to M
    {
        for j from 1 to N{
        if S[i] = T[j] then cost := 0
                   else cost := 1
          d[i, j] := minimum(d[i-1, j] + 1, //a deletion
                             d[i, j-1] + 1, //an insertion
                             d[i-1, j-1] + 1) //a substitution

         if(i > 1 and j > 1 and S[i] = T[j-1] and S[i-1] = M[j]) then
                d[i, j] := minimum(
                                   d[i, j],
                                   d[i-2, j-2] + cost) // transposition

      }
    }
    return d[M, N]
 }
```

Figure 5.1: Pseudocode of DamerauLevenshtein distance

5.1.2 Wavelet Packet Decomposition

The wavelet transform has been extensively used in signal processing analysis. It has proven to be very efficient in many engineering fields problems [31, 33, 35, 37, 43]. The frequency content of the signal can be represented by window frames using Fourier transform which ignores the time localization information of the signal. For achieving the time localization information, the window frame should be varied so that it will be in a wider range of low frequencies and a slight range of higher frequencies. Thus the Wavelet transform has provided a flexibility to represent the time and the frequencies of the signal. It provides detailed information for low level frequencies and detailed information about the time at high frequencies which makes it suitable for the analysis of inconsistence data patterns over different time intervals.

Wavelet packet decomposition (WPD) is an extension and simplification of wavelets which use the entire decomposition for the signal including the low-pass and high-pass to create the complete representation of the signals. It is obtained by applying a recursion of a decimation process to reduce the sample rate of the signals or the size of the data sample. WPD provides a level by level transformation of a signal from the time domain into the frequency domain [43] such that the high frequencies of a given signal could be resolved within a small time frame while the low frequencies nee be resolved by a large time frame.

At each level of WPD, two sets of coefficients are generated, the approximation coefficients and detail coefficients. The approximation coefficients are produced by convolving the input signal (packet sequence sizes) with the low-pass filter and down-sampling by a factor of two. The detail coefficients are similarly produced by convolving the input signal (packet sequence sizes) with high-pass filter. For the new level, unlike the wavelet decomposition, both approximation coefficients and detail coefficients will be convolved with the low-pass filter and high-pass filter and down-sampling to generate a new level of decomposition. The ability to iterate both low-pass and high-pass filter in wavelet decomposition method will lead generating more than one basis function rather than one basis function -and two basic functions in the last level- in the case of the wavelet transform. Iterating the low-pass and high-pass filters will generate the complete tree basis where the top level of the WPD tree is the time representation of the signal and the bottom level of a fully decomposed tree is the frequency representation of the signal. For the n-level decomposition, the original signal S is split as illustrated in Fig.5.1. The original signal S is decomposed to the first layer A1 and D1 signals. The similar decomposition process can be applied to the first layer A1 and D1 signal to obtain the second layer that consists of AA2, DA2, AD2 and DD2. In each layer, every approximation and detail signals will be decomposed to new approximation and details.

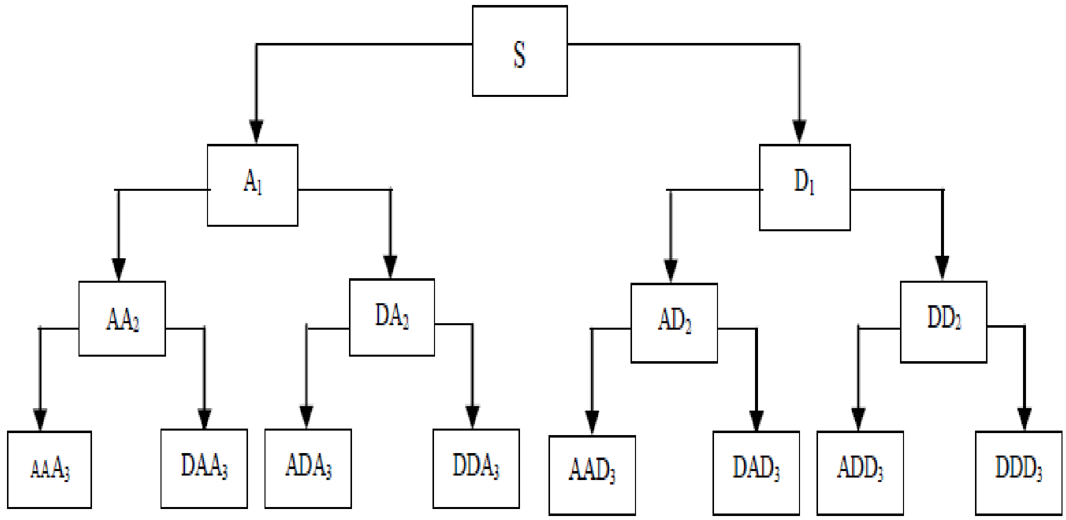


Figure 5.2: wavelet packet decomposition with 3 levels

To extract the features from the website Tor packet sequences, we use the wpdec function for For one dimensional data to do wavelet decomposition of our data. The main signal for the decomposition is represented by the pure Tor packet sequences for each website visit. Thus, for 20 websites and 15 visits for each website we will have 300 input data vectors (signals). The length of data vectors is varied since the number of Tor packet for every website visit are varied.

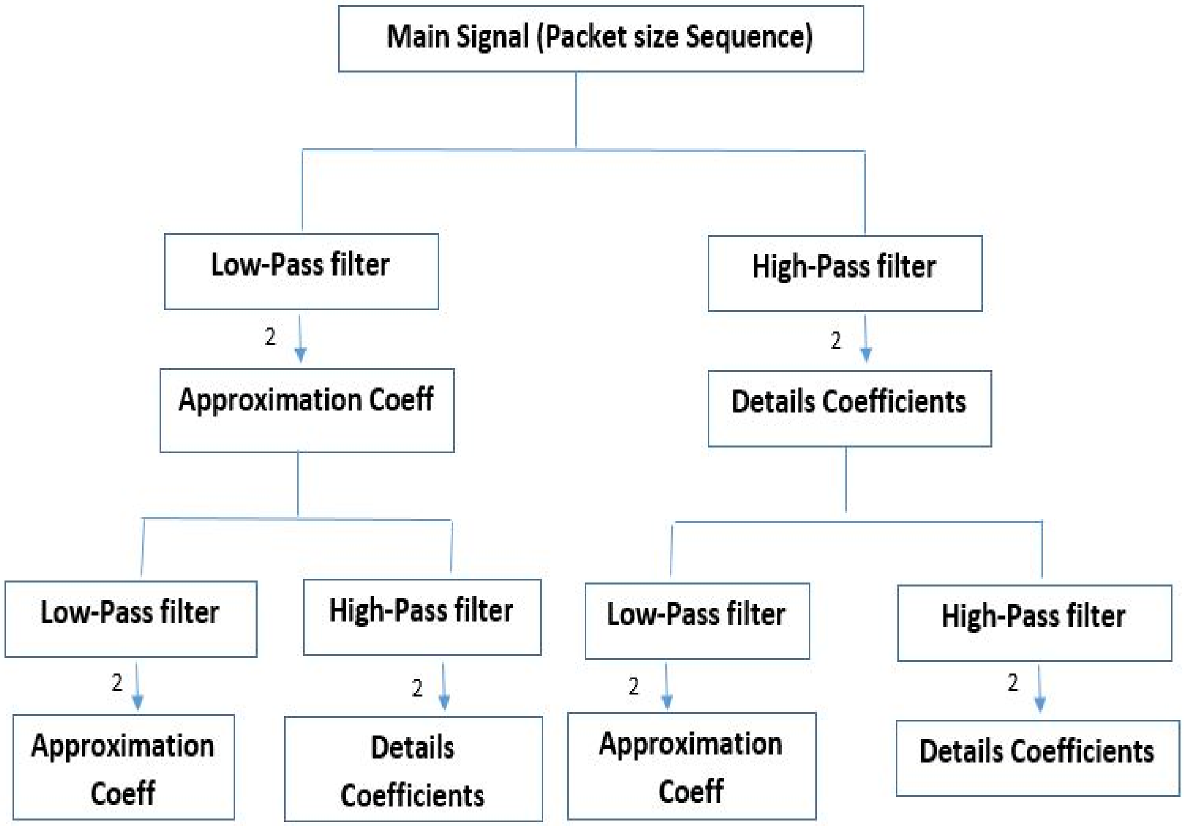


Figure 5.3: Signal decomposition using WPD

Since our input will be the sequences of packet sizes of the websites traces, we applied the 1D wavelet decomposition to extract the low-pass and the high-pass coefficients of our original signals, which represent the packet size sequence of the traces, and we investigated different wavelet families to check which of the wavelet families will provide more informative coefficients to be used for the traces classification.

One-D wavelet decomposition of our signals implemented using wpdec ( ) function with the following parameters:

$$T = Wpdec(Data, "Level", "Wavelet") \tag{5.1}$$

Where

T is the wavelet decomposition tree

Data is an array storing the Tor packet sizes instant sample

Level is the level of decomposition (by default it's 0)

Wavelet is the family of wavelet (e.g. Bior1.5)

Figure 5.4 shows the wavelet and scaling functions(low-pass and high-pass coefficients) of the Bior1.5 wavelet family. We have mainly used the Bior1.5 wavelet family since it generated the coefficients that provide us the best accuracy results comparing to the other wavelet family outputs. Meanwhile, we applied the decomposition up to the fourth level as we figured out that the results of classification is improved when the next level of the decomposition is taking place. The detailed results for the all four levels are presented in details in the next chapter.

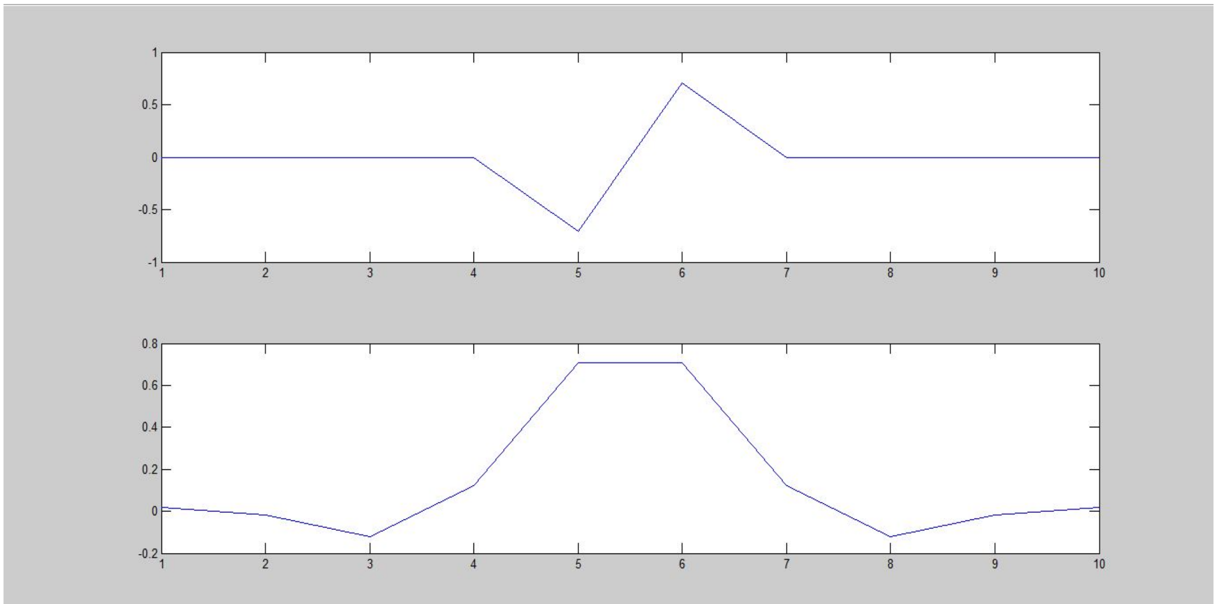


Figure 5.4: Bior1.5 Wavelet and scaling functions

Figure 5.5 shows the values of high-pass low-pass filters or coefficients of bior1.5 wavelet family that are used for convolving the input signal (Packet sizes sequence) with the wavelet family (Bior1.5) for decomposition.

Decomposition high-pass filter	Decomposition low-pass filter
0	0.016572815
0	-0.016572815
0	-0.121533978
0	0.121533978
-0.707106781	0.707106781
0.707106781	0.707106781
0	0.121533978
0	-0.121533978
0	-0.016572815
0	0.016572815

Figure 5.5: Bior1.5 Coefficients

Then, we extract the a coefficients list from the decomposed wavelet tree using the following function

$$C = Wpcoef(T, "Level") \tag{5.2}$$

The function returns the coefficients associated with certain nodes in certain level. We examine the returned coefficients for different nodes at each level to train our classifier and we found that the coefficients of the first node at each level provide the best accuracy results. Figure 5.6 shows a plot graph for website packet sizes sample whereas the Figure 5.7 shows a plot of extracted coefficients of the same website packet sizes using WPD in in the 4^{th} level.

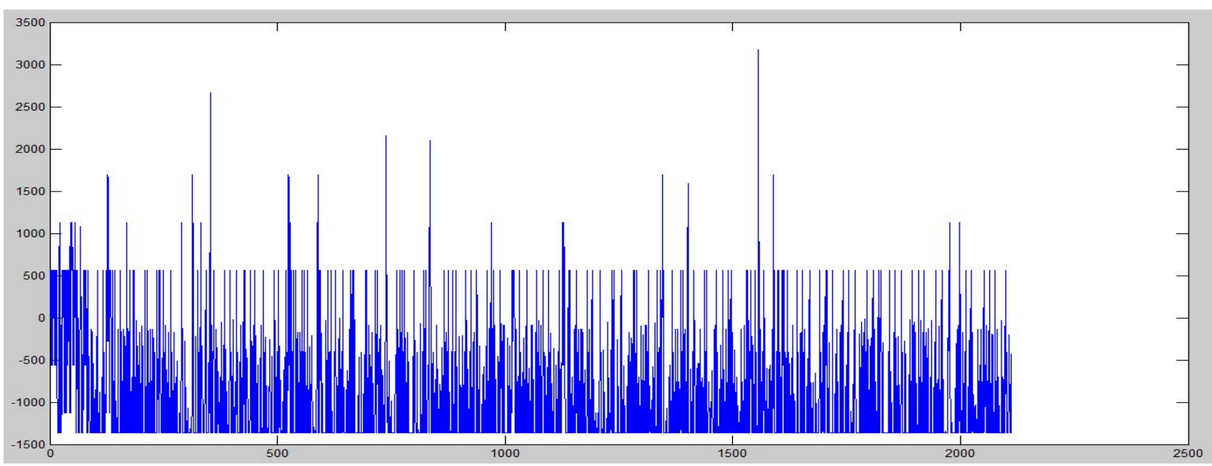


Figure 5.6: Plot of packet Sizes for website1 sample1 of Firefox browser

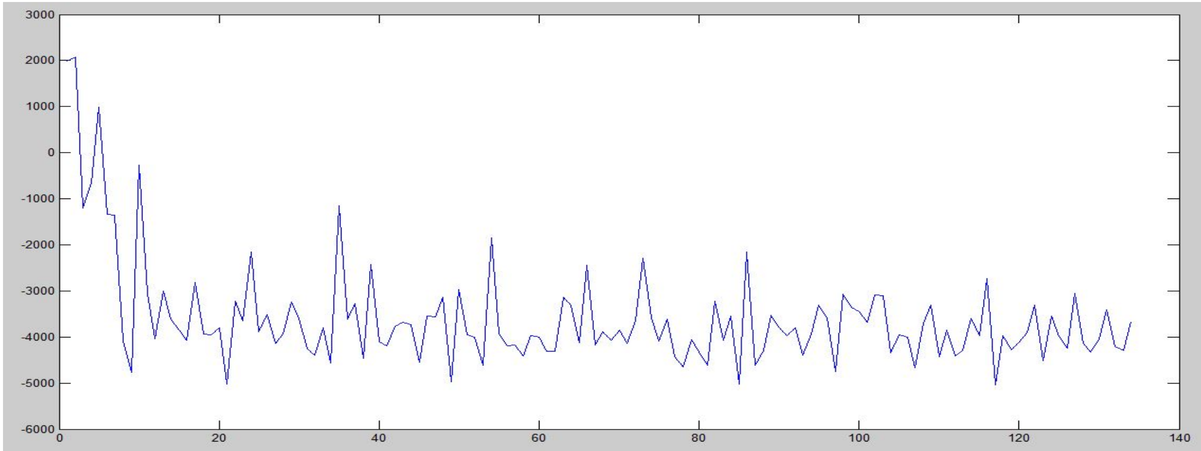


Figure 5.7: Level 4 WPD bior1.5 coefficients for website1 sample1 of Firefox browser

5.2 Classifiers

5.2.1 Weka Calssification Tool

WEKA (Waikato Environment for Knowledge Analysis) is the product of the University of Waikato (New Zealand) [44]. It is a comprehensive collection of machine learning algorithms that performs data mining tasks such as data preprocessing, classification, clustering and regression. It is written in the Java language and provides a GUI to facilitate the interacting with data files and producing visual results. Weka also allows users to perform different classification algorithms on their data sets. Weka provides the ability for the users apply the algorithms directly to a dataset or call them from outsource Java code. In addition, it is possible to develop new machine learning schemes.

The main GUI for Weka presents as four application interface with different functions:

1. Explorer: preprocessing, classifying, clustering, selecting attributes, etc..

2. Experimenter: Setup, Run and analyze the machine learning algorithms.

3. Knowledge Flow: Visual design for KDD process.

4. Simple CLI: a simple command interface.

In this thesis, we only use the Explorer interface which is satisfying our needs to perform the classification process on our dataset.

WEKA can load ARFF files Attribute Relation File Format. ARFF has two sections:

1. The Header section defines the relation (dataset) name, attribute name, and type (the class).

2. The Data section lists the data instances.

Since our preprocessing or feature extraction phase in this thesis work is done in Matlab program. The output of the feature extracted from Matlab is a CSV data file format. We convert the data format form CSV to ARFF using the following steps:

1. Load the CSV file that contains the features data into Weka using preprocess >>>open file.

2. Save the CSV file to ARFF format.

3. Change the last attribute field in the ARRF file to include the classes to be classified based on them. See figure 5.8.

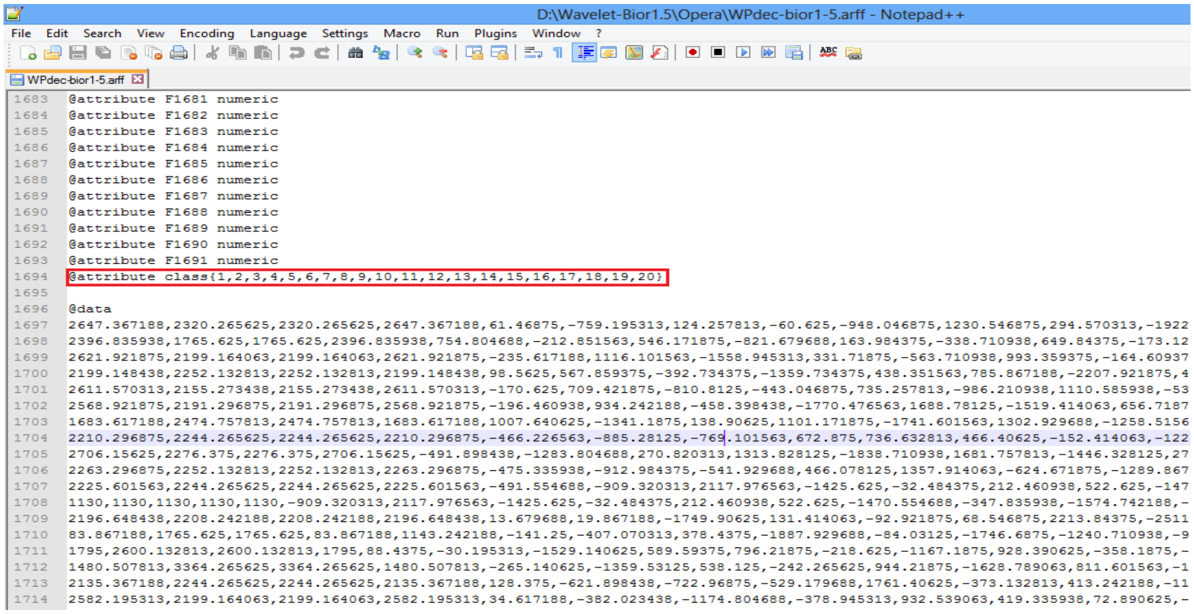


Figure 5.8: Attribute class values in Weka ARFF data file

5.2.2 Support vector machine

Support vector machines (SVMs) is a supervised machine learning that used for analyzing the data samples and recognizing the patterns among the data. Given a set of training data labeled with the category that belonged to, a model for assigning a new data into the corresponding category will be built using SVM algorithms. The support vector machine builds a hyperplane that separate the two categories of the data samples. In SVM model, the data samples are represented as points in space such that the data samples of the same category are separated from the data samples of the other category by a hyperplane as wide as possible. The new data samples are predicted to be belonged to one of the categories based on which side of the gap they fall on.

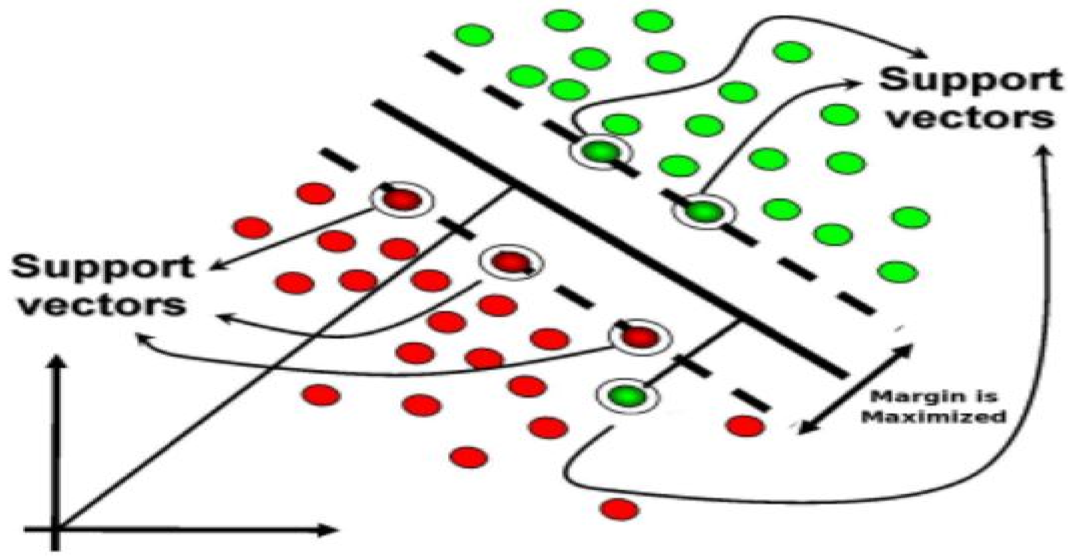


Figure 5.9: The support vectors and the margin of the classifier.

Given a set of training data

$$D = (X_i, y_i)|X_i \in \Re^p, y_i \in \{-1, 1\} \tag{5.3}$$

The linear discriminant function is given by

$$f(x) = w^T + b \tag{5.4}$$

where the value of **w** denotes to the parameter vector and **b** represents the bias. The Hyperplane decision surface function of support vector machine gaps the feature space to two half- spaces.

$$f(x) = 0 = w^T + b \tag{5.5}$$

Giving a training dataset, we say that the training dataset is linearly separable if :

$$w^T x_i + b \geq 0 for y_i = +1 \tag{5.6}$$

$$w^T x_i + b \leq 0 for y_i = -1 \tag{5.7}$$

The margin width M is given by:

$$\frac{M = 2}{\sqrt{w.w}} \tag{5.8}$$

Maximized the hyperplane margin and minimizing the classification error is solved as a convex quadratic programming problem. This gives the Lagrangian:

$$L_D = \sum_i a_i \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i x_j \tag{5.9}$$

Where

L_D is maximized with respect to a

a_i is Lagrange multiplier

y_i is the vector class

x_i is the train vector

To classify vector x:

$$f(x) = \sum_{i=1}^{N} a_i y_i s_i x + b \tag{5.10}$$

Where

N is the number of support vectors

s_i is support vector with $class_i$

b can be calculated with :

$$b = \frac{1}{y_i} - \sum_{j=1}^{N} a_j y_j x_j x_i \tag{5.11}$$

Where a <C

Different kernel functions can be used for SVM classification. The kernel functions map the input vector to a higher dimensional space where a better hyperplane with minimal classification error can be obtained. The kernel functions are defined as

$$b = \frac{1}{y_i} - \sum_{j=1}^{N} a_j y_j x_j x_i \tag{5.12}$$

The most common functions used by SVM are

Linear Kernel

$$k(x, y) = x^T y + c \tag{5.13}$$

Polynomial Kernel

$$k(x, y) = \left(\alpha x^T y + c\right)^d \tag{5.14}$$

where

alpha is the slope

c is a constant term

d polynomial degree

Gaussian Kernel (example of radial basis function kernel)

$$k(x, y) = \exp\left(-\frac{\parallel x - y \parallel^2}{2\sigma^2}\right) \tag{5.15}$$

A sequential minimal optimization (SMO) is a Support Vector Machine algorithm developed by John Platt which is made for training a support vector classifier [45]. It implements the sequential minimal optimization algorithm for training a support vector classifier, using polynomial or Gaussian kernels. SMO solves the SVM QP (Quadratic Programming), since SVM only classified binary problems, by decomposing the overall problem into a series of the smallest possible QP sub-problems. This implementation generally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. Multi-class problems are solved using pairwise classification [45]. In this work, SMO is evaluated with the following parameters: c = 1.0 and 1024; epsilon = 1.0E-12; kernel = PolyKernel; num-Folds = -1; randomSeed = 1. (Default parameters).

5.2.3 Multilayer perceptron neural network

Multilayer perceptron neural network (MLP) is an extension of the single layer perceptron network where the hidden layers will be used beside the input and output layers for data training. MLP is considered as a feed forward flow network where input data passing through different layers that consist of neurons. MLPs neural network model able to learn and predict complicated patterns in data through finding a set of input weights of the neurons that maximize the fit to the training data. In feed-forward networks, the output of a neuron has no more effect on its Input and it is only forward to be input for the next layer. Unlike the feed-forward networks, the output of neurons in recurrent networks, or called back-propagation network, are given as their input based on the error difference of the neuron's output. In this thesis a feed forward neural network with back-propagation algorithm was used to classify instances. The aim of using a back-propagation algorithm is to minimize the output squared error function through adjusting the input weights of the neurons. The artificial neuron receives one or more input and sums them to produce an output. The sums of each node are weighted and activated using a proper activation function (either threshold or sigmoid function) passed through a nonlinear function known as an activation or transfer function.

Mathematically, the input to the neuron consists of three main elements: Input signals, weights and bias. First, a linear combination of the input signals (x_i) and the weights (w_i) is formed, then the bias (b) will be added.

$$u = b + \sum w_{ji} \tag{5.16}$$

The linear combination in Equation 5.16 with an activation function g will provide the output of the neuron

$$y(x; b, w) = g\left(b + \sum_{i=1}^{n} w_i x_i\right) \tag{5.17}$$

The activation functions or transfer functions are usually used to transform the weighted sum of the inputs to generate the neuron output value. The purpose of the activation function is to introduce non-linearity to the neural network. The choice of activation function will have an impact on the performance of the training algorithm. Nonlinear activation function activation function is widely used in neural networks. In a backpropagation neural network, the activation function should be differentiable (non-linear) to be bound in a specific limited range.

For the backpropagation neural network learning, the most common activation functions used is the sigmoid. The sigmoid activation function is a bounded differentiable real function that is defined for all real input values and has a positive derivative at each point [46].

The expression of the sigmoid function is given by:

$$y = \frac{1}{1 + e^{-x}} \tag{5.18}$$

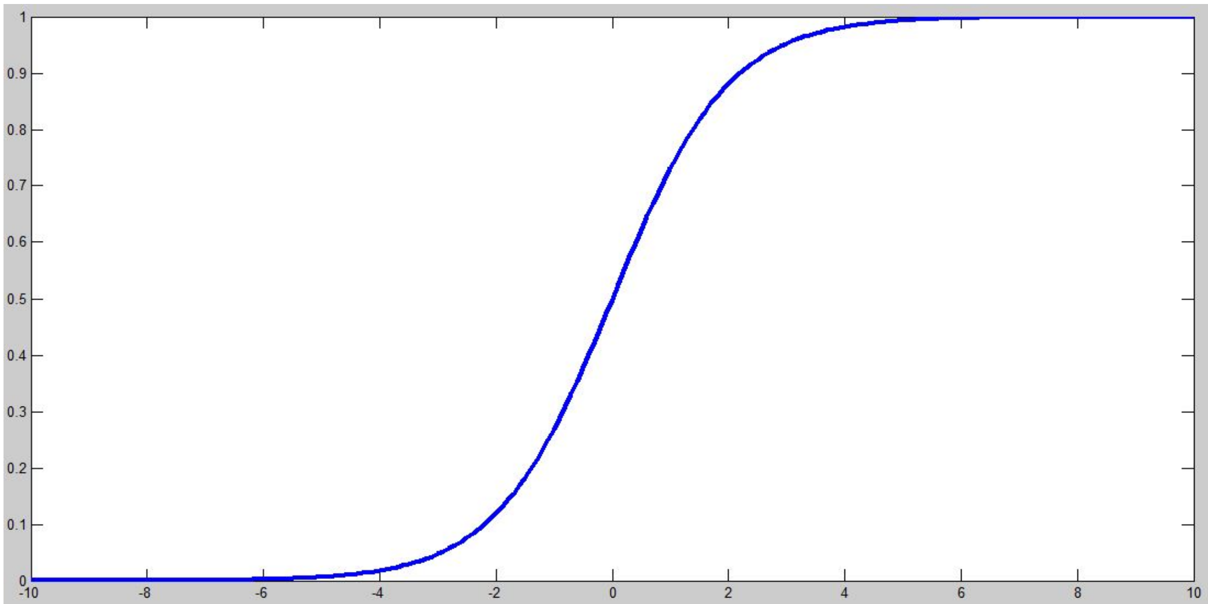


Figure 5.10: Sigmoid activation function

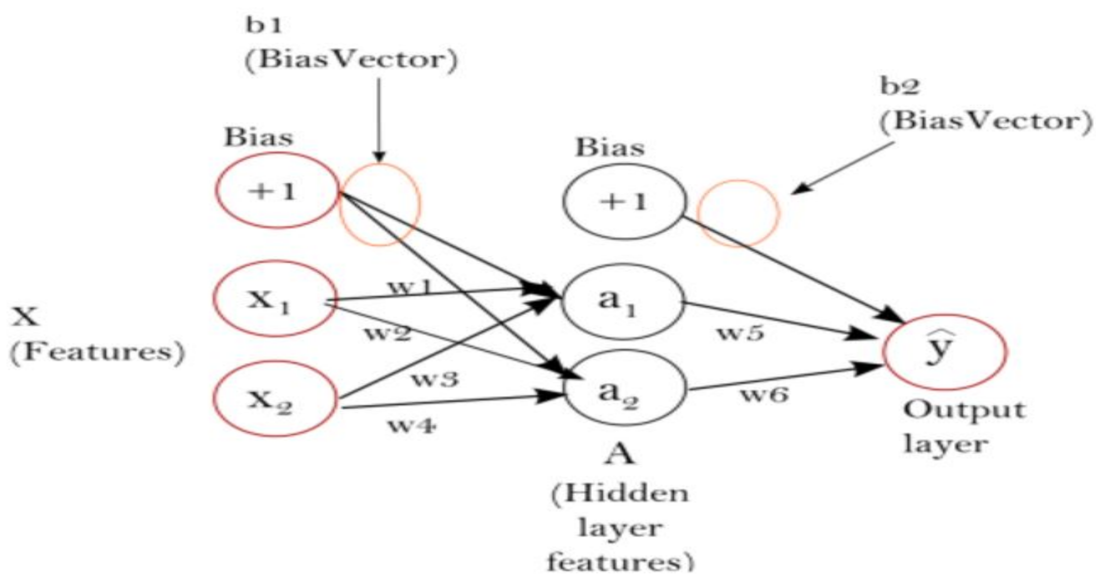


Figure 5.11: The feed-forward network architecture and perceptron neuron model

5.2.4 Naive bayesian

The Naive Bayesian classifier technique is a probabilistic classifier based on Bayesian theorem. It is one of the most efficient classification algorithms. It supports the independence theorem on the attributes of the classes. It provides a good results in case of high feature dimensional. Naive Bayes classifier needs a small amount of training data to determine the parameters (means and variances) necessary for classification. [47]

Let say that we have the following classes [w1,w2,...,wc], And the Feature vector is x = [x1, x2,, xd] Then:

The Naive Bayes assumption:

$$P(x_1, ..., x_d|w_j) = \prod_i P(x_i|w_j) \tag{5.19}$$

And the Naive Bayes classifier :

$$w_{NB} = argmax_{wj} P(w_j) \prod_i P(x_i|w_j) \tag{5.20}$$

5.3 MEASURING PERFORMANCE

The proposed approach can be validated using different classifiers. In this work experiments, K-Fold cross validation scheme is chosen with 10 folds. In The K-Fold cross validation model, training data sets is divided into k subsets where one of these subsets is used for testing while others are used for training. Then, another different set is chosen for testing each time and the average error rate is computed among all data sets. The advantage of this model is that each data instance is used in testing exactly once, and in training k-1 times.

In machine learning, different classifiers can be used on various datasets to provide a best results since no single classifier can provide best results on all problems. The usefulness of the classifier on various datasets can be determined by evaluating the performance of the classifier on the datasets.

A confusion matrix clarifies the accuracy of the classification problem. Given N classes which indicate to the websites in this work - a confusion matrix is a M x N matrix, where Xi,j indicates the number of tuples from data samples that were assign to class X_{ij} but where the correct class is X_i.

A confusion matrix shows the best classification results whenever the values outside the diagonal of the matrix are mostly have a zero value which means that each sample is correctly classified to its corresponding class.

A confusion matrix provides information about real and predicted classifications produced by the classifier. Number of performance metrics can be derived from the confusion matrix as follows:

1. Accuracy(AC) :

   The accuracy (AC) is the percentage of the total number of predictions that were correctly classified. AC shows the correctness of the model using the ratio of total number of correct classifications to overall number of the classification for a single class.

   There are some terms that are commonly used to describe the accuracy metric. These terms are true positive (TP), true negative (TN), false negative (FN), and false positive (FP).

Accuracy of the classification model is defined - based on the above terms - as:

$$Accuracy = \frac{(TN + TP)}{(TN + TP + FN + FP)} \tag{5.21}$$

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         246               82      %
Incorrectly Classified Instances        54               18      %
Kappa statistic                          0.8105
Mean absolute error                      0.0905
Root mean squared error                  0.2095
Relative absolute error                 95.1982 %
Root relative squared error             96.0776 %
Coverage of cases (0.95 level)          99      %
Mean rel. region size (0.95 level)      80      %
Total Number of Instances              300
```

Figure 5.12: Sample of classification accuracy for edit distance method on Chrome dataset

2. Recall Or TP Rate

   Recall or TP rate is the percentage of positive cases that were correctly classified. Recall or TP rate is a measure of a classification model be able to select instances of a certain class from a data set [48]. Recall or TP rate is defined as :

$$Recall = TPR = \frac{TP}{(TP + FN)} \tag{5.22}$$

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.933	0.000	1.000	0.933	0.966	0.964	0.954	0.938	1
	0.867	0.000	1.000	0.867	0.929	0.928	0.900	0.875	2
	0.733	0.007	0.846	0.733	0.786	0.777	0.941	0.746	3
	0.800	0.018	0.706	0.800	0.750	0.738	0.927	0.642	4
	0.933	0.004	0.933	0.933	0.933	0.930	0.967	0.877	5
	0.867	0.014	0.765	0.867	0.813	0.804	0.936	0.742	6
	0.867	0.004	0.929	0.867	0.897	0.892	0.925	0.823	7
	0.733	0.004	0.917	0.733	0.815	0.812	0.934	0.775	8
	0.933	0.004	0.933	0.933	0.933	0.930	0.995	0.902	9
	0.667	0.004	0.909	0.667	0.769	0.769	0.940	0.676	10
	0.933	0.046	0.519	0.933	0.667	0.676	0.972	0.537	11
	0.933	0.000	1.000	0.933	0.966	0.964	0.992	0.952	12
	0.733	0.011	0.786	0.733	0.759	0.747	0.978	0.669	13
	0.800	0.039	0.522	0.800	0.632	0.624	0.894	0.454	14
	0.800	0.004	0.923	0.800	0.857	0.853	0.957	0.791	15
	0.867	0.000	1.000	0.867	0.929	0.928	0.959	0.927	16
	0.733	0.011	0.786	0.733	0.759	0.747	0.900	0.652	17
	0.667	0.011	0.769	0.667	0.714	0.702	0.984	0.660	18
	0.800	0.000	1.000	0.800	0.889	0.890	0.860	0.818	19
	0.800	0.014	0.750	0.800	0.774	0.762	0.970	0.652	20
Weighted Avg.	0.820	0.009	0.850	0.820	0.827	0.822	0.944	0.755	

Figure 5.13: TP rate and Recall metrics for chrome dataset classification using edit distance method

3. FP Rate The false positive rate (FP) is the percentage of negative samples that were incorrectly classified as positive. It's also known as false alarm rate. FP rate is calculated using the following equation:

$$FPR = \frac{FP}{(TN + FP)} \tag{5.23}$$

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.933	0.000	1.000	0.933	0.966	0.964	0.954	0.938	1
	0.867	0.000	1.000	0.867	0.929	0.928	0.900	0.875	2
	0.733	0.007	0.846	0.733	0.786	0.777	0.941	0.746	3
	0.800	0.018	0.706	0.800	0.750	0.738	0.927	0.642	4
	0.933	0.004	0.933	0.933	0.933	0.930	0.967	0.877	5
	0.867	0.014	0.765	0.867	0.813	0.804	0.936	0.742	6
	0.867	0.004	0.929	0.867	0.897	0.892	0.925	0.823	7
	0.733	0.004	0.917	0.733	0.815	0.812	0.934	0.775	8
	0.933	0.004	0.933	0.933	0.933	0.930	0.995	0.902	9
	0.667	0.004	0.909	0.667	0.769	0.769	0.940	0.676	10
	0.933	0.046	0.519	0.933	0.667	0.676	0.972	0.537	11
	0.933	0.000	1.000	0.933	0.966	0.964	0.992	0.952	12
	0.733	0.011	0.786	0.733	0.759	0.747	0.978	0.669	13
	0.800	0.039	0.522	0.800	0.632	0.624	0.894	0.454	14
	0.800	0.004	0.923	0.800	0.857	0.853	0.957	0.791	15
	0.867	0.000	1.000	0.867	0.929	0.928	0.959	0.927	16
	0.733	0.011	0.786	0.733	0.759	0.747	0.900	0.652	17
	0.667	0.011	0.769	0.667	0.714	0.702	0.984	0.660	18
	0.800	0.000	1.000	0.800	0.889	0.890	0.860	0.818	19
	0.800	0.014	0.750	0.800	0.774	0.762	0.970	0.652	20
Weighted Avg.	0.820	0.009	0.850	0.820	0.827	0.822	0.944	0.755	

Figure 5.14: FP rate for chrome dataset classification using edit distance method

4. Precision (P) is the Probability that positive cases are correctly predicted. It is also known as positive predictive value and calculated using the following equation:

$$Precision = \frac{TP}{(TP + FP)} \tag{5.24}$$

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.933	0.000	1.000	0.933	0.966	0.964	0.954	0.938	1
	0.867	0.000	1.000	0.867	0.929	0.928	0.900	0.875	2
	0.733	0.007	0.846	0.733	0.786	0.777	0.941	0.746	3
	0.800	0.018	0.706	0.800	0.750	0.738	0.927	0.642	4
	0.933	0.004	0.933	0.933	0.933	0.930	0.967	0.877	5
	0.867	0.014	0.765	0.867	0.813	0.804	0.936	0.742	6
	0.867	0.004	0.929	0.867	0.897	0.892	0.925	0.823	7
	0.733	0.004	0.917	0.733	0.815	0.812	0.934	0.775	8
	0.933	0.004	0.933	0.933	0.933	0.930	0.995	0.902	9
	0.667	0.004	0.909	0.667	0.769	0.769	0.940	0.676	10
	0.933	0.046	0.519	0.933	0.667	0.676	0.972	0.537	11
	0.933	0.000	1.000	0.933	0.966	0.964	0.992	0.952	12
	0.733	0.011	0.786	0.733	0.759	0.747	0.978	0.669	13
	0.800	0.039	0.522	0.800	0.632	0.624	0.894	0.454	14
	0.800	0.004	0.923	0.800	0.857	0.853	0.957	0.791	15
	0.867	0.000	1.000	0.867	0.929	0.928	0.959	0.927	16
	0.733	0.011	0.786	0.733	0.759	0.747	0.900	0.652	17
	0.667	0.011	0.769	0.667	0.714	0.702	0.984	0.660	18
	0.800	0.000	1.000	0.800	0.889	0.890	0.860	0.818	19
	0.800	0.014	0.750	0.800	0.774	0.762	0.970	0.652	20
Weighted Avg.	0.820	0.009	0.850	0.820	0.827	0.822	0.944	0.755	

Figure 5.15: Classification Precision for edit distance method on chrome dataset

5. ROC area

   In addition to the above metrics, Area under Receiver Operating Characteristics (ROC) is also considered. In order to decide which classifier is better than the other, the ROC performance is reduced to a scalar value that represents the expected performance. A ROC area is a plot of the false positive rate on the X axis against the true positive rate on the Y axis. It shows a trade-off between the true positives and false positives predications. The point (0, 1) in ROC graph shows that the classifier has classified all positive and negative cases correctly. The point (0, 0) shows that the classifier predicts all cases as negative unlike the point (1, 1) that shows that the classifier predicts all cases as positive.

CHAPTER 6

EXPERIMENTAL ANALYSIS

Extensive experiments were conducted to evaluate the edit distance method and wavelet method in our generated dataset and Cai dataset. This chapter presents and analysis the experiments results of the different approaches.

6.1 Datasets

Two different datasets are used in our experiments (Cai dataset and our dataset). Cai dataset consists of packet traces for (100) websites with (40) samples for each website [5]. Cai dataset was generated using one web browser (FireFox Tor version). In our experimentations,(20 websites x 15 visits) samples were used to regenerate the results of Cai method for classification and investigate other classifiers on it.

We developed a second database for this research. The developed dataset Consists of packet traces for (20) websites with (15) samples for each website. We generated the dataset using six different web browsers (Firefox, Chrome, Safari, IE, Opera and Tor browser).

6.2 Feature extraction method and evaluation

We conducted several experiments using cai datasets and our datasets. To standardize the two datasets, we used the same sample sizes (20 websites x 15 visits) in our experiments.

6.2.1 Edit Distance - levenshtein distance-

We investigated the edit distance levenshtein distance- feature extraction method on the two different datasets (Cai dataset and our dataset) with uniform sample sizes (20 websites x 15 visits) using three different classifiers (SMO, NaiveBayes and Naivbayesmultinomai) in weka toolbox. We used two different costs for SMO classifier (1 and 1024) as these costs were used in the SMO classifier in previous works.

Table 6.1 shows the accuracy rates of the edit distance method using CAI dataset and our dataset with one sample size (20 websites x 15 visits). It shows that the Edit Distance -levenshtein distance- with a Naive Bayes classifier has the highest recognition rate of (92%) on CAI dataset whereas the SMO classifier with the cost parameter 1, the default cost of SMO, provide the highest recognition rate of (86%) on our dataset specifically on Firefox browser.

Tor FireFox Browser							
Classifier/Browser	FireFox	IE	Chrome	Opera	Safari	Tor	Cai Data
SMO-c1	81.00%	86.00%	82.00%	75.00%	72.00%	79.00%	86.00%
SMO-c1024	80.00%	84.00%	80.00%	75.00%	67.00%	72.00%	83.00%
Naive Bayes	79.00%	86.00%	78.00%	61.00%	70.00%	76.00%	92.00%
Naivbayesmultinomai	53.00%	62.00%	45.00%	44.00%	44.00%	44.00%	58.00%

Table 6.1: websites recognition accuracy of Edit distance method with different classifiers on Cai dataset

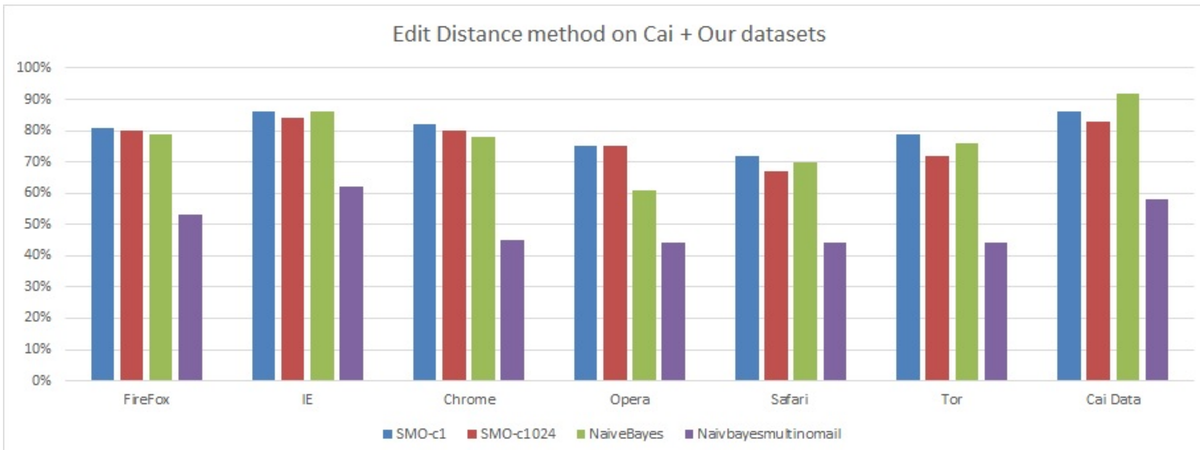


Figure 6.1: Graphical presentation of website fingerprinting using Edit distance method on Cai and our datasets

We conducted the same edit distance Levenshtein distance- feature extraction method on our generated dataset for every web browser using three sample sizes (10 websites x 10 visits), (20 websites x 15 visits) and (20 websites x 20 visits) and three classifiers as we did in the previous experiment.

Tables 6.2 - 6.7 show the accuracy rates for edit distance feature extraction method on the Chrome, Firebox, IE, Opera, Safari and Tor browsers datasets respectively. They show that SVM classifier with cost 1 provide the best recognition rate of (82%) (81%) (86%) (75%) (72%) (79%) using (20 websites x 15 visits). Nave Bayes classifier provides best results for the safari and tor browsers with sample size (10 websites x 10 visits).

Chrome Browser				
Sampe Size/Classifier	SMO-c1	SMO-c1024	NaiveBayes	Naivbayesmultinomail
10 x 10	79.00%	68.00%	81.00%	69.00%
20 x 20	74.00%	70.00%	64.00%	45.00%
20 x 15	82.00%	80.00%	78.00%	45.00%

Table 6.2: websites recognition accuracy of Edit distance method with different classifiers on the Chrome Browser

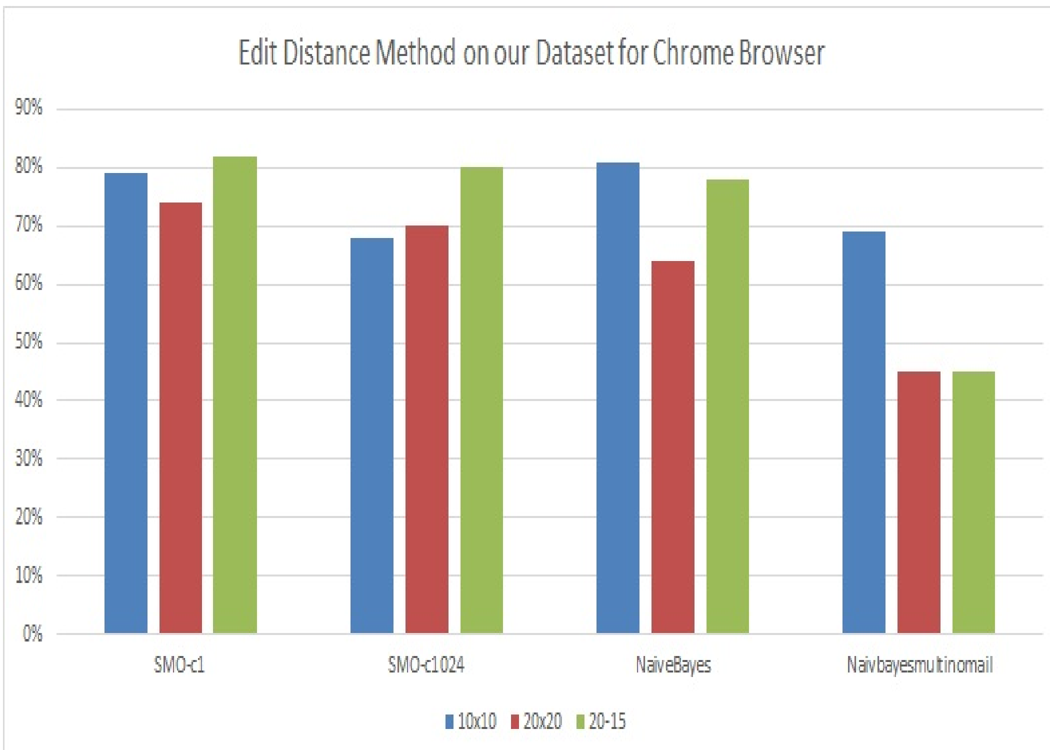


Figure 6.2: Graphical presentation of website fingerprinting using Edit Distance method for Chrome Browser.

FireFox browser Browser				
Sampe Size/Classifier	SMO-c1	SMO-c1024	NaiveBayes	Naivbayesmultinomail
10 x 10	78.00%	75.00%	77.00%	72.00%
20 x 20	75.00%	73.00%	68.00%	51.00%
20 x 15	81.00%	80.00%	79.00%	53.00%

Table 6.3: websites recognition accuracy of Edit distance method with different classifiers on Firefox Browser

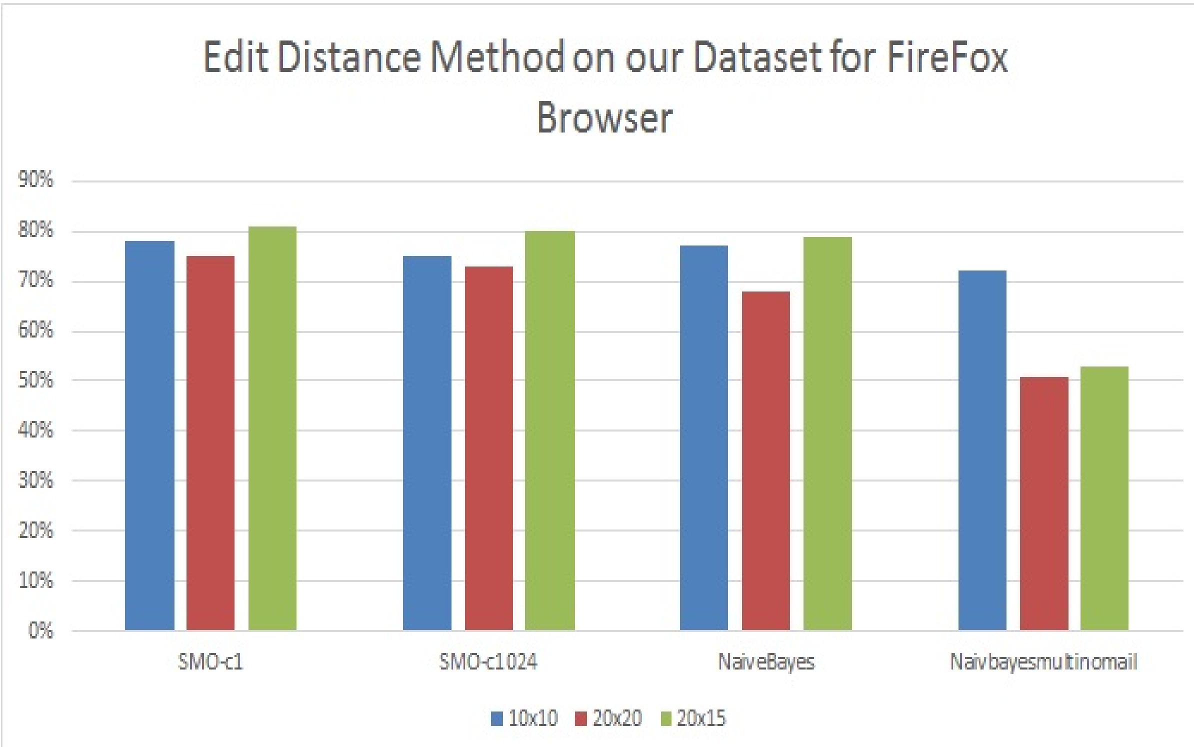


Figure 6.3: Graphical presentation of website fingerprinting using Edit Distance method for FireFox Browser.

IE Browser				
Sample Size/Classifier	SMO-c1	SMO-c1024	NaiveBayes	Naivbayesmultinomail
10 x 10	86.00%	85.00%	86.00%	74.00%
20 x 20	79.00%	76.00%	73.00%	55.00%
20 x 15	86.00%	84.00%	86.00%	62.00%

Table 6.4: websites recognition accuracy of Edit distance method with different classifiers on IE Browser

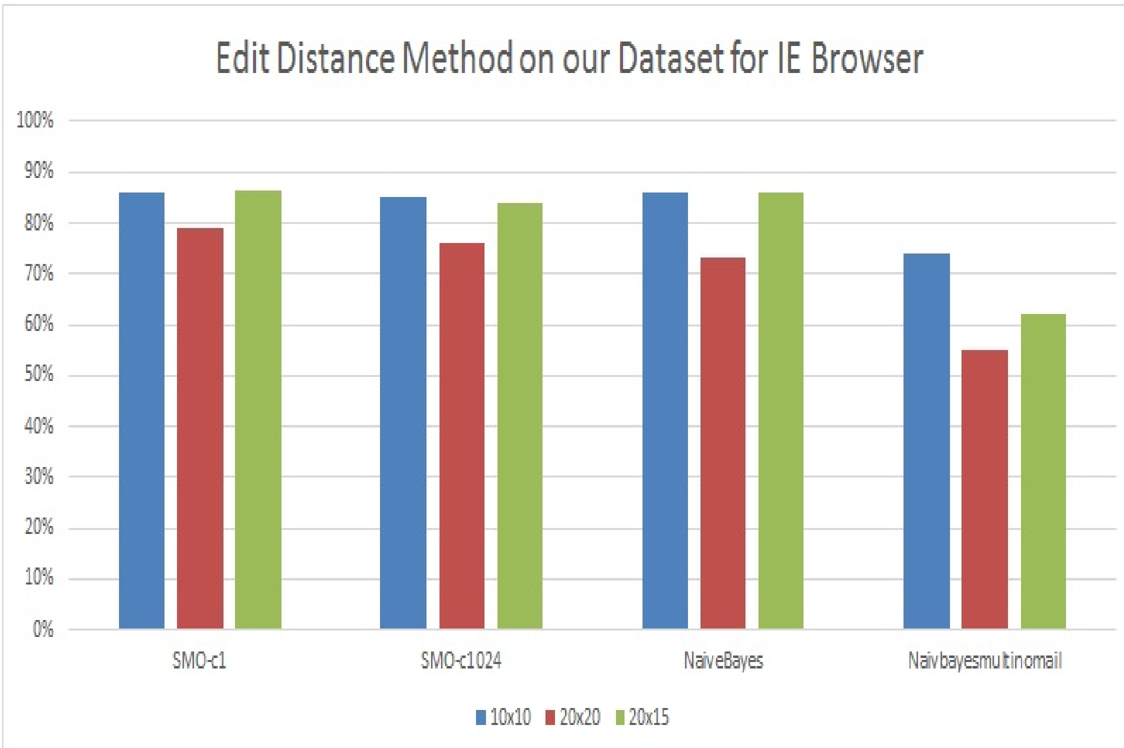


Figure 6.4: Graphical presentation of website fingerprinting using Edit Distance method for IE Browser.

Opera Browser				
Sampe Size/Classifier	SMO-c1	SMO-c1024	NaiveBayes	Naivbayesmultinomail
10 x 10	75.00%	64.00%	67.00%	60.00%
20 x 20	72.00%	69.00%	50.00%	21.00%
20 x 15	75.00%	75.00%	61.00%	44.00%

Table 6.5: websites recognition accuracy of Edit distance method with different classifiers on Opera Browser

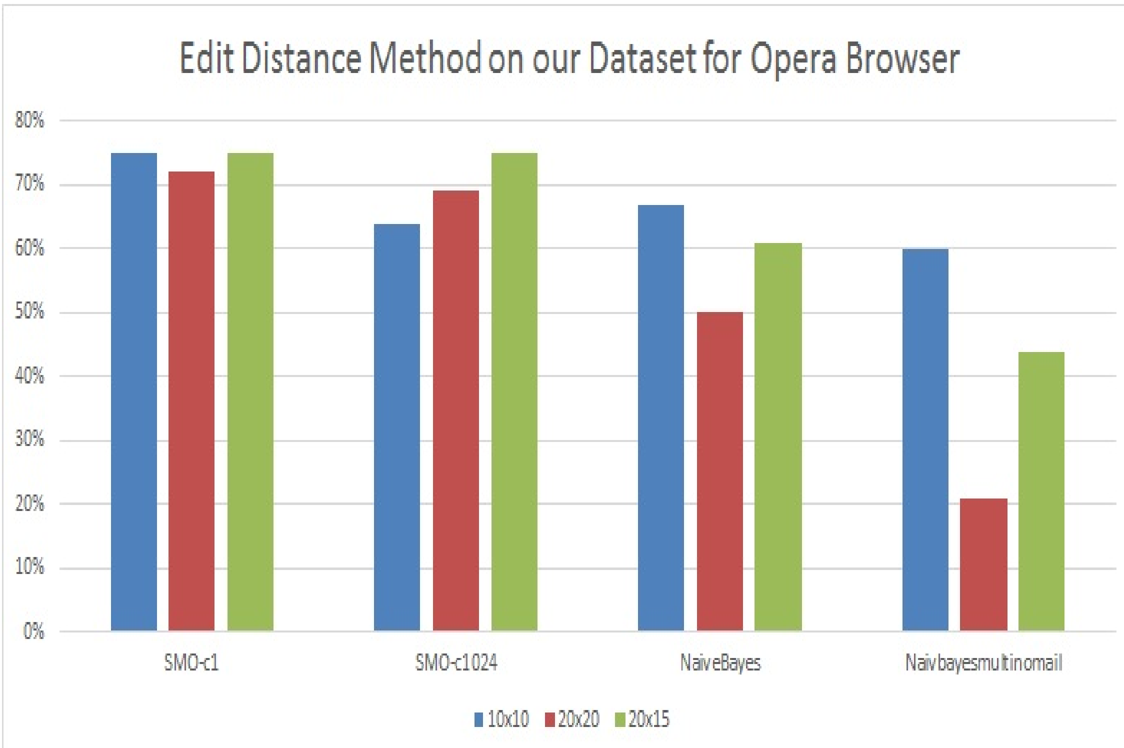


Figure 6.5: Graphical presentation of website fingerprinting using Edit Distance method for Opera Browser.

Safari Browser				
Sampe Size/Classifier	SMO-c1	SMO-c1024	NaiveBayes	Naivbayesmultinomail
10 x 10	81.00%	73.00%	86.00%	80.00%
20 x 20	67.00%	63.00%	63.00%	44.00%
20 x 15	72.00%	67.00%	70.00%	44.00%

Table 6.6: websites recognition accuracy of Edit distance method with different classifiers on Safari Browser

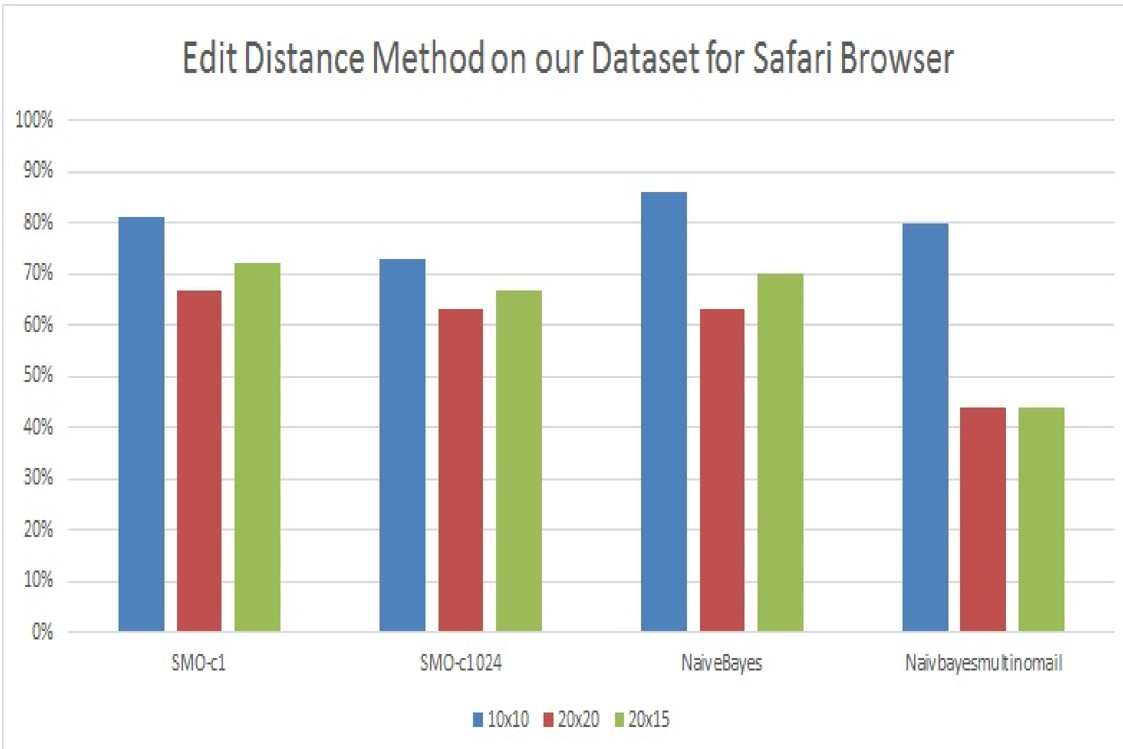


Figure 6.6: Graphical presentation of website fingerprinting using Edit Distance method for Safari Browser.

Tor Browser				
Sampe Size/Classifier	SMO-c1	SMO-c1024	NaiveBayes	Naivbayesmultinomail
10 x 10	79.00%	74.00%	87.00%	74.00%
20 x 20	79.00%	72.00%	76.00%	44.00%
20 x 15	79.00%	72.00%	76.00%	44.00%

Table 6.7: websites recognition accuracy of Edit distance method with different classifiers on Tor Browser

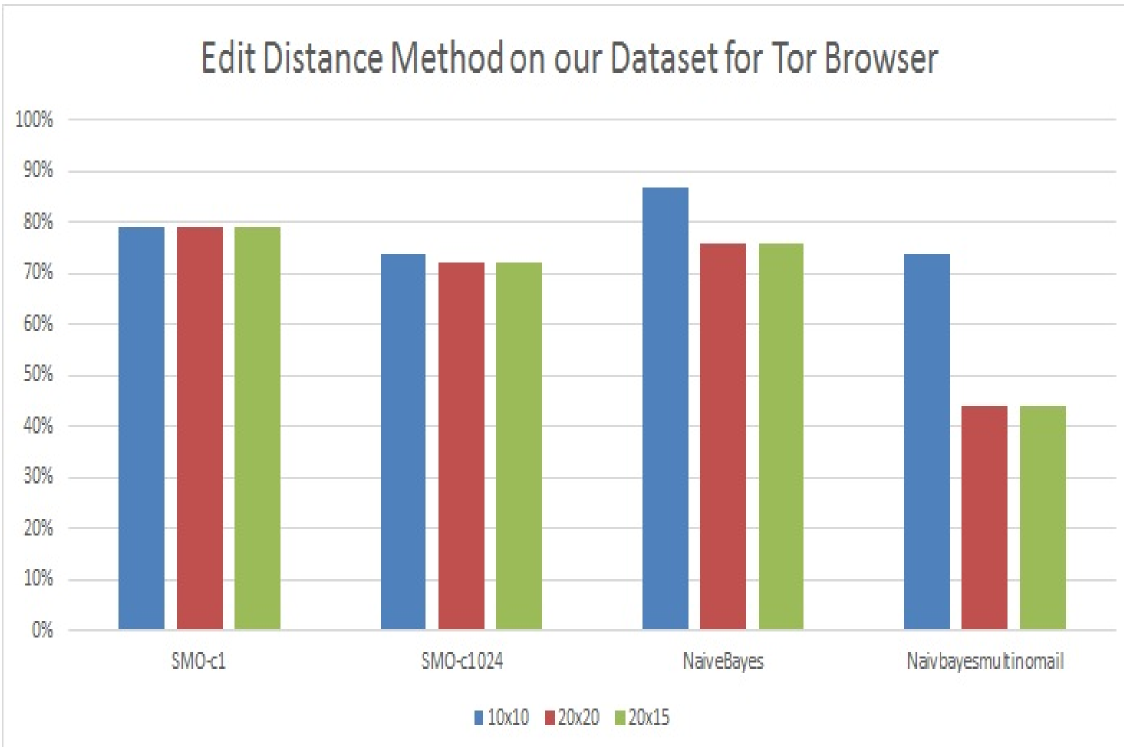


Figure 6.7: Graphical presentation of website fingerprinting using Edit Distance method for Tor Browser.

Using SVM classifier with Edit distance method, the best results for the accuracy rates that we have obtained in terms of the Accuracy rate, TP Rate, FP Rate, Precision, Recall and ROC metrics were for the IE browser, Chrome browser and Firefox browser datasets respectively. The highest Accuracy rate achieved is 86.33% for IE browser and the lowest rate obtained is 72% for the Safari browser. Table 6.2.1 shows the experimental results in terms of TP Rate, FP Rate, Precision, Recall and ROC for the six selected web browsers.

Results: Edit distance with SVM Classifier						
Browser	FireFox	Chrome	Safari	Opera	IE	Tor
Accuracy Rate 10 Folds	81.33 %	82.00%	72.00 %	75.33 %	86.33 %	78.67%
TP Rate	0.813	0.820	0.72	0.753	0.863	0.787
FP Rate	0.01	0.009	0.015	0.013	0.007	0.011
Precision	0.828	0.850	0.766	0.766	0.882	0.781
Recall	0.813	0.820	0.72	0.753	0.863	0.787
ROC	0.919	0.944	0.931	0.941	0.958	0.951

captionCross validation perfomance metrics for Edit distance method on our dataset

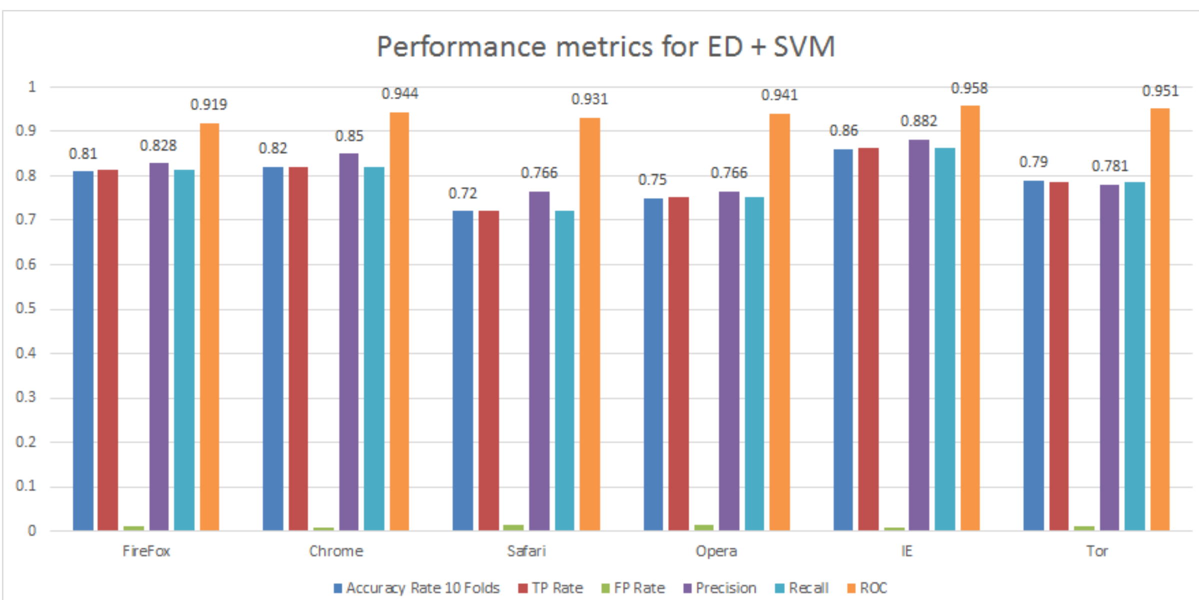


Figure 6.8: Performance metrics for ED+ SVM classification algorithm

Then we applied another classifier called MultilayerPerceptron with our features extracted by the edit distance method for the all data browsers. Table 6.8 shows the results of the MultilayerPerceptron classifier on features extracted by edit distance method.

Results: Edit distance with MultilayerPerceptron Classifier						
Browser	FireFox	Chrome	Safari	Opera	IE	Tor
Accuracy Rate 10 Folds	81.333 %	82.667%	70.667%	77.000%	87.667%	76.333%
TP Rate	0.813	0.827	0.707	0.770	0.877	0.763
FP Rate	0.01	0.009	0.015	0.012	0.006	0.012
Precision	0.824	0.844	0.725	0.779	0.888	0.749
Recall	0.813	0.827	0.707	0.770	0.877	0.763
ROC	0.954	0.972	0.941	0.962	0.975	0.957

Table 6.8: Performance metrics for Edit distance method + MultilayerPerceptron

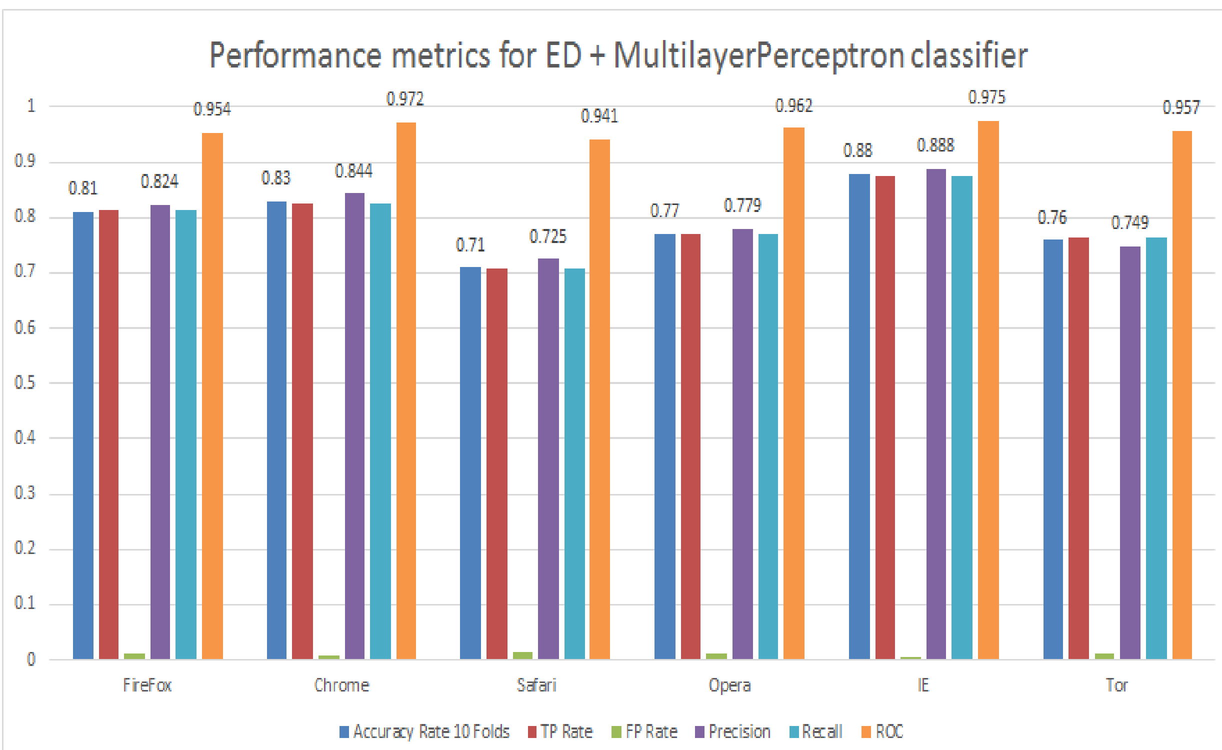


Figure 6.9: Performance metrics for ED + MultilayerPerceptron classifier

6.2.2 Wavelet Packet Decomposition

In our experiments, we investigated wavelet packet decomposition as feature extraction method for the website packets traces. We conducted our experiments on our datasets and Cai datasets. Since there are certain numbers of mother wavelets, we have investigated the main three types of the mother wavelet (Haar, Sym2, and Bior1.5) [49, 50, 51, 52, 53]. Table 6.9 shows the results of WPD for the fourth level of decomposition using the three mentioned wavelet families (Haar, Sym2, and Bior1.5). As a result, we find that the Bior1.5 provides the best accuracy rate compared to other investigated wavelet packet decomposition families which means that our main signals that represented the websites traces is quite identical or similar to the signal shape of the mother wavelet Bior1.5. on the other word, the low-pass and high-pass coefficients can be integrated with our main signals and produce new readable signals that can be easy detect the high and low frequency of our main signals-website packet size traces- . Figure 6.13 shows the accuracy rate for different wavelet families using WPD in our generated dataset and Cai dataset.

WPD fourth level results for different wavelet families							
Browser	FireFox	Chrome	Safari	Opera	IE	Tor	Cai Data
Bior1.5	79.666 %	70.333%	59.333%	49.666%	78.667%	68.000%	79.333%
Haar	78.667 %	67.333%	56%	45%	75.666%	64.666%	79.667
SYM2	78.333 %	69.333%	58%	42.333%	76.333%	61%	79.667%

Table 6.9: WPD families Results

Figures 6.10,6.11 and 6.12 show the plot of the main signal (the packet sizes sequence) with the low and high coefficients (filters) of the used wavelet families.

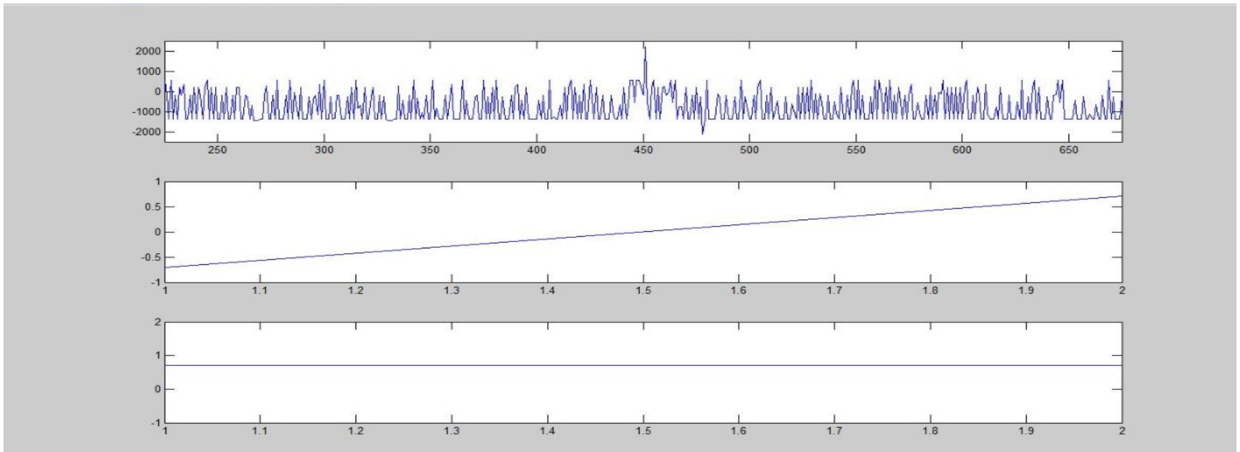


Figure 6.10: Sample of chrome trace with the low and high-pass coefficients of Haar wavelet

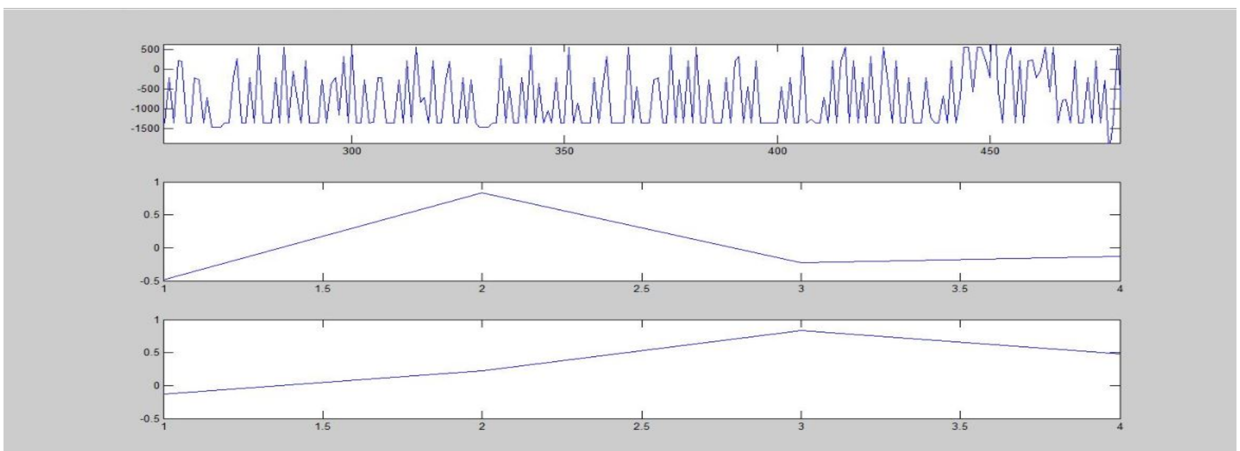


Figure 6.11: Sample of chrome trace with the low and high-pass coefficients of Sym2 wavelet

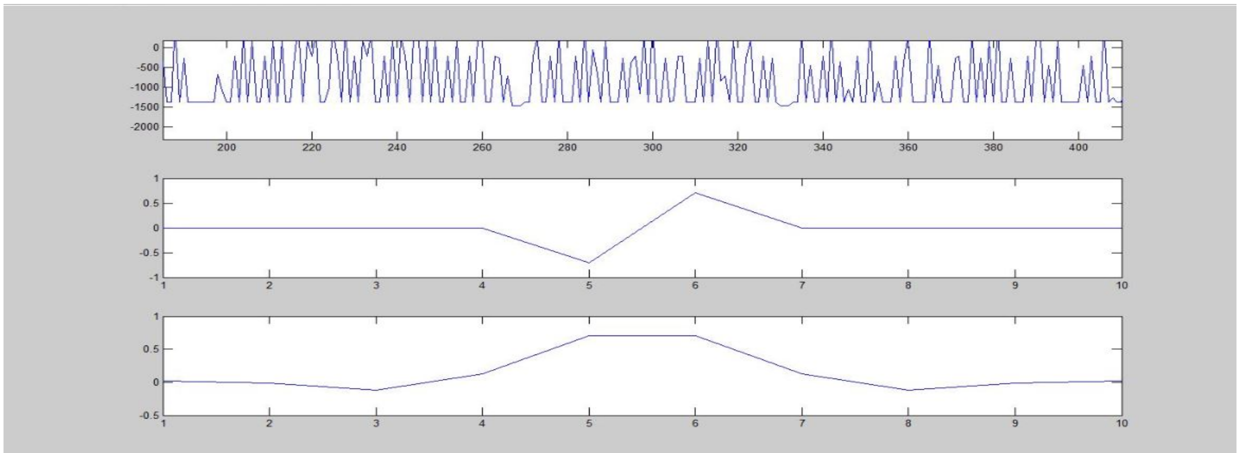


Figure 6.12: Sample of chrome trace with the low and high-pass coefficients of Bior1.5 wavelet

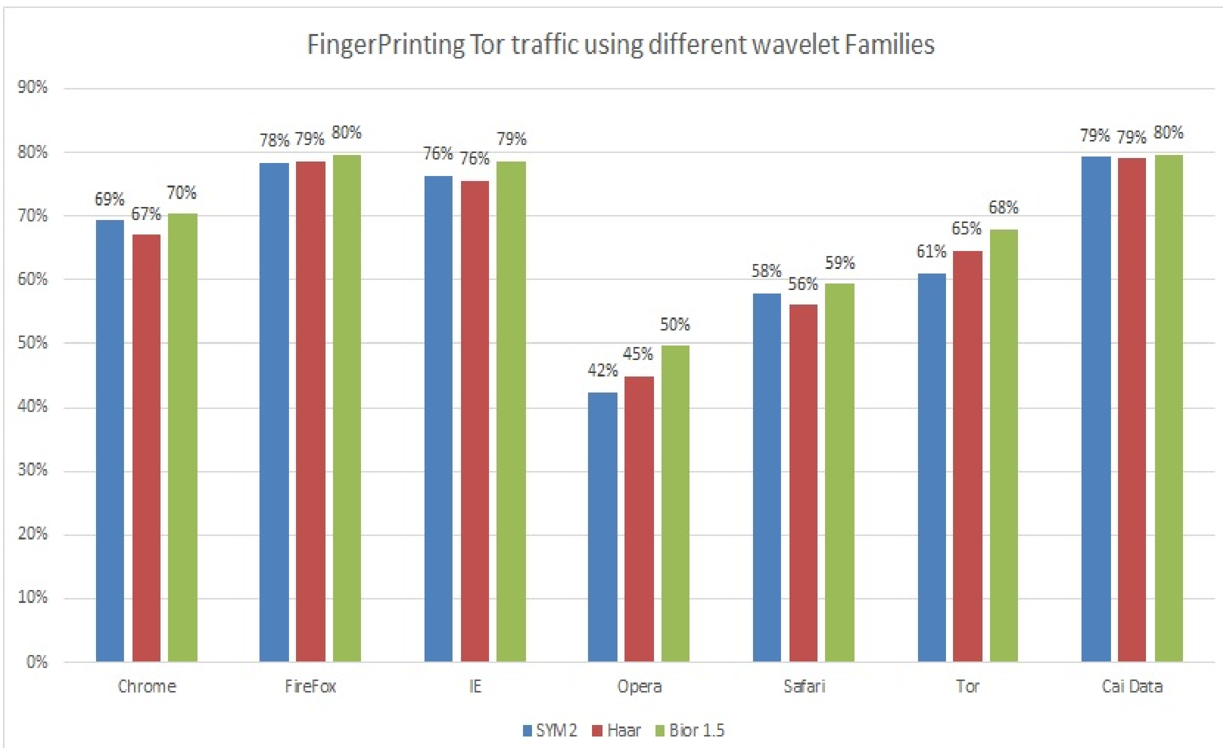


Figure 6.13: Accuracy rate for the website fingerprinting using different WPD families

Then extensive experiments have been conducted to investigate the best accuracy rate for bior1.5 wavelet packet decomposition. We used our generated dataset with (20 websites x 15 visits) samples for six browsers beside to Cai dataset. The website classes labeled by numbers due to the data file format requirement in Weka toolbox (See appendix for more details for website classs name). We have used four levels of wavelet packet decomposition to investigate how the accuracy improved at each level and to determine some metrics of the improvements. The results for every WPD level on our dataset and Cai dataset using SVM classifier(with SMO classification algorithm) are shown in the following part.

1. **Wavelet Packet Decomposition Level 1 results :**

Results: Wavelet Packet Decomposition Bior1.5 on Level1							
Browser	FireFox	Chrome	Safari	Opera	IE	Tor	Cai Data
Accuracy	69.333%	46.00%	47.667%	30.333%	57.333%	38.333%	67.667%
TP Rate	0.693	0.460	0.447	0.303	0.573	0.383	0.667
FP Rate	0.016	0.028	0.028	0.037	0.22	0.032	0.017
Precision	0.767	0.867	0.570	0.484	0.641	0.575	0.735
Recall	0.693	0.584	0.477	0.303	0.573	0.383	0.667
ROC	0.949	0.873	0.883	0.818	0.937	0.857	0.956

Table 6.10: Cross validation performance metrics for Wavelet Bior1.5 method on our dataset

2. **Wavelet Packet Decomposition Level 2 results :**

Results: Wavelet Packet Decomposition Bior1.5 on Level2							
Browser	FireFox	Chrome	Safari	Opera	IE	Tor	Cai Data
Accuracy	69.333 %	59.667%	53.000 %	39.333%	67.000%	46.667%	72.000%
TP Rate	0.693	0.597	0.530	0.393	0.670	0.467	0.720
FP Rate	0.016	0.021	0.025	0.032	0.017	0.028	0.015
Precision	0.767	0.684	0.609	0.481	0.704	0.580	0.760
Recall	0.693	0.597	0.530	0.393	0.670	0.467	0.720
ROC	0.949	900	0.905	0.838	0.947	0.887	0.962

Table 6.11: Cross validation performance metrics for Wavelet Bior1.5 method on our dataset

3. **Wavelet Packet Decomposition Level 3 results :**

Results: Wavelet Packet Decomposition Bior1.5 on Level 3							
Browser	FireFox	Chrome	Safari	Opera	IE	Tor	Cai Data
Accuracy	74%	68.00%	59.667 %	46.667%	74.667%	61.667%	74.333%
TP Rate	0.740	0.680	0.597	0.467	0.747	0.617	0.743
FP Rate	0.014	0.017	0.021	0.028	0.013	0.020	0.14
Precision	0.798	0.722	0.663	0.473	0.737	0.672	0.762
Recall	0.740	0.680	0.597	0.467	0.747	0.617	743
ROC	0.951	0.920	0.912	0.859	0.951	0.911	0.964

Table 6.12: Cross validation performance metrics for Wavelet Bior1.5 method on our dataset

4. **Wavelet Packet Decomposition Level 4 results :**

Results: Wavelet Packet Decomposition Bior1.5 on Level 4							
Browser	FireFox	Chrome	Safari	Opera	IE	Tor	Cai Data
Accuracy	79.666%	70.333%	59.667%	49.666%	78.667%	68.000%	79.333%
TP Rate	0.797	0.703	0.597	0.497	0.787	0.680	0.793
FP Rate	0.011	0.016	0.021	0.026	0.011	0.017	0.011
Precision	0.836	0.732	0.663	0.539	0.799	0.692	0.811
Recall	0.797	0.703	0.597	0.497	0.787	0.680	0.793
ROC	0.958	0.933	0.912	0.882	0.954	0.931	0.959

Table 6.13: Cross validation performance metrics for Wavelet Bior1.5 method on our dataset

Then we have used the MultilayerPerceptron classifier on the Wavelet data features for all selected browsers and Cai dataset. Using the MultilayerPerceptron classifier, we have applied the wavelet packet decomposition features of the datasets for the 4^{th} level of the WPD of all browser data. The table 6.14 shows the results of applying the MultilayerPerceptron classifier for the level 4 of the WPD of the datasets.

Results: Wavelet Packet Decomposition Bior1.5 on Level 4+ MultilayerPerceptron classifier							
Browser	Firefox	Chrome	Safari	Opera	IE	Tor	Cai Data
Accuracy	78.333%	75.000%	59.667 %	45.000%	82.000%	72.667%	78.000%
TP Rate	0.783	0.750	0.597	0.450	0.820	0.727	0.780
FP Rate	0.011	0.013	0.021	0.029	0.009	0.014	0.012
Precision	0.795	0.754	0.663	0.459	0.830	0.723	0.788
Recall	0.783	0.750	0.597	0.450	0.820	0.723	0.780
ROC	0.961	0.960	0.912	0.873	0.965	0.951	0.954

Table 6.14: Performance metrics for Wavelet Bior1.5 + MultilayerPerceptron

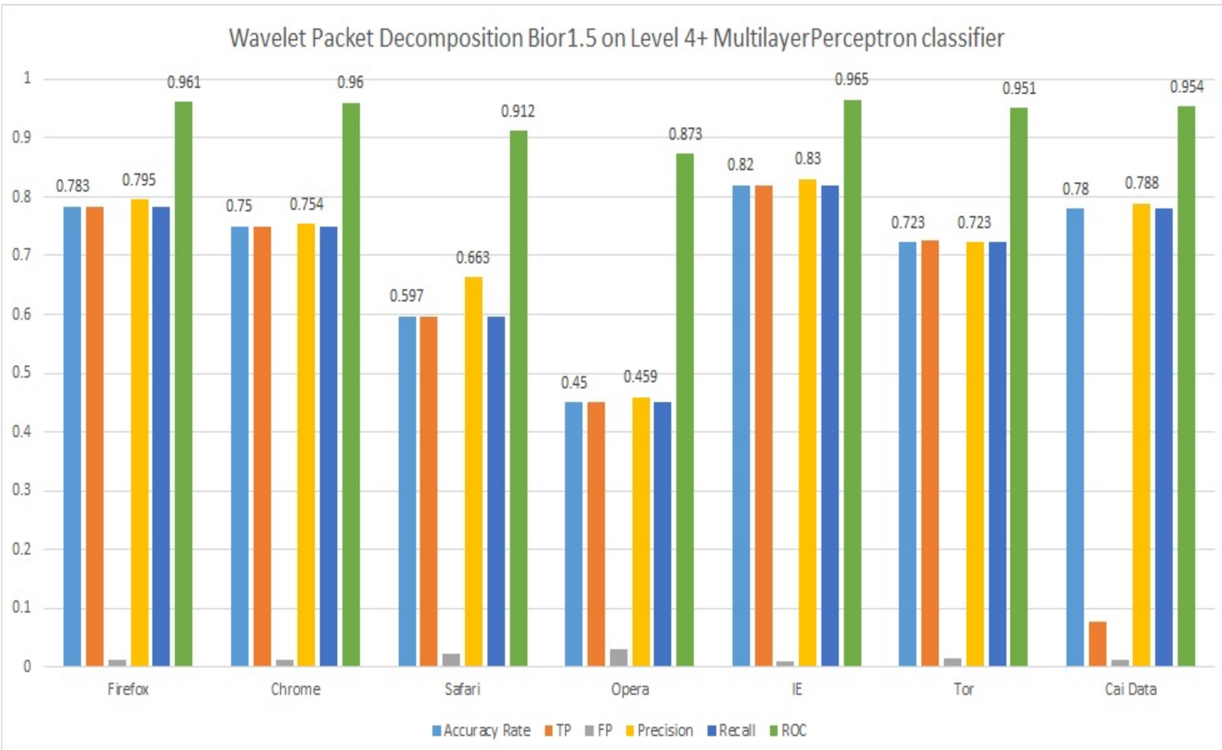


Figure 6.14: Performance metrics for WPD+ MultilayerPerceptron classifier

Although MLP classifier increases the classification accuracy rate for Internet explorer, Chrome, Safari and Tor datasets against the SVM classifier, the time required to build a model using MLP classifier is significantly higher than the time required to build the SVM classifier. Fig. 6.15 shows the time required to build the model using the two investigated classifiers for the six web browsers datasets.

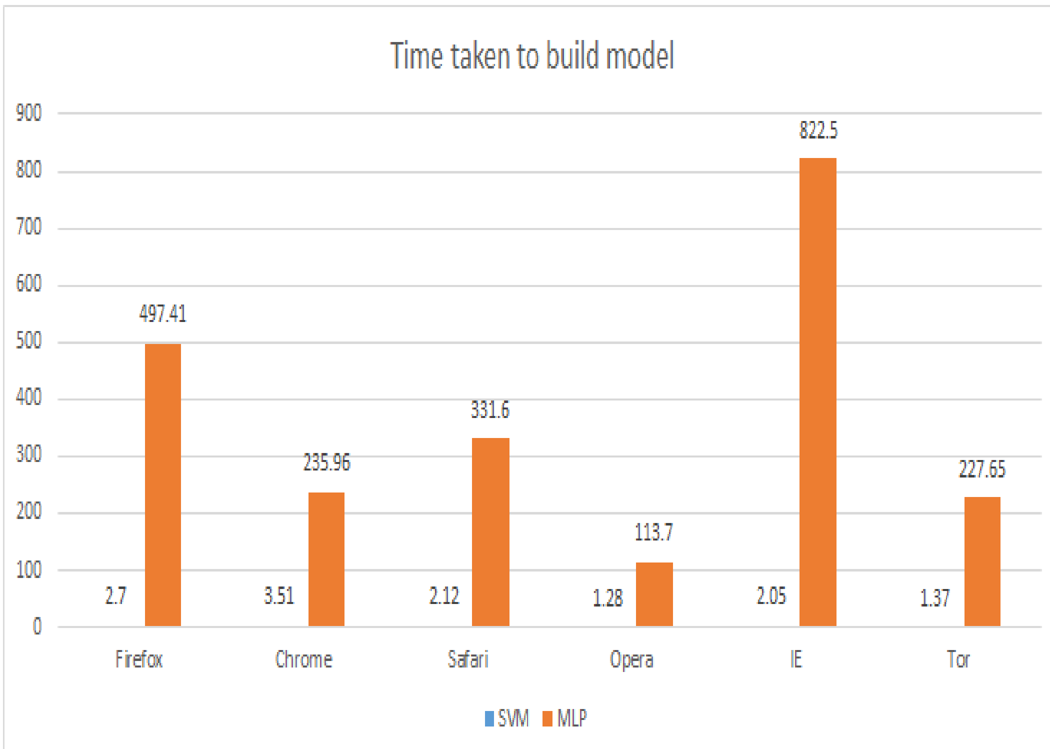


Figure 6.15: SVM Vs NEURAL NETWORK(Time in seconds)

CHAPTER 7

CONCLUSION

7.1 Thesis Contributions

In this work we examined different approaches for website fingerprinting on Tor protocol. We collected a dataset of website traffic using six web browsers (Firefox, Chrome, Internet Explorer, Opera, Safari and Tor) with 20 websites and 15 samples. We presented the stages and phases of designing our dataset starting by describing the environment set up for data collection and the tools that were used to collect the data. We also described the second phase and stage for designing the dataset which is the data processing phase. It describes in details how we filtered, identified and extracted the data from the raw packet data to build our final format dataset to be used in our experiment (feature extraction).

To validate our generated dataset for website fingerprinting, we implemented edit distance feature extraction method which was used in recent research works for website fingerprinting. We achieved accuracy results of 86% on dataset generated by IE browser using SMO algorithm for classification. Accuracy rate of 88% was achieved on dataset generated by IE browser using MultilayerPerceptron classifier. We used dataset of 20 websites and 15 samples in our experiments.

Then we introduced a new feature extraction method based on wavelet transform technique. We used wavelet packet decomposition method to extract feature vectors from the Tor packet size sequence for the websites. Since there are a number of wavelet families that can be applied to perform the transformation and decomposition, we investigated three main wavelet families on our generated dataset to determine which of them can be used further based on the obtained accuracy results. Based on the wavelet families experimental results, we choose to use Bior1.5 wavelet for the WPD experiments in certain levels. We investigated different levels of WPD to show how the accuracy improved through the levels of WPD. For the 4th level of WPD for out Tor packet size sequence, we achieved an accuracy rate of 82% on IE dataset using MultilayerPerceptron classifier and 80% on Firefox dataset using SMO classifier. We used other performance metrics (TP rate, FP rate, Precision, Recall and ROC) to evaluate and validate our approach on our generated dataset.

Although the obtained results of applying the new feature extraction method (WPD) are not quite perfect, they give a good indication that wavelet technique is suitable for website fingerprinting if we improve the preprocessing phase of the data collection.

7.2 Limitations of the Current Work

There were several limitations to our work summarized as follows:

Firstly, we have built our approach based on closed-world dataset not on open-world data set. In closed-world dataset classification, the set of classes (websites, identities) is known in advance. This type of classification is known as supervised classification. Open-world classification deals with scenarios in which the set of classes is not known in advance. In Open-world classification problem, the classifier should be able to first detect the pattern among all websites traces (unsupervised learning since no label for websites traces are available during building the model) and second classify the corresponding websites.

Secondly, our proposed approach is not suitable if the extracted packet size sequence has a lot of noise (sizes of packet that are not belong to Tor traffic). Using wavelet packet decomposition as a feature extraction requires the data signal (packet size sequence) to has a clear pattern in order to extract the features. This implies that the preprocessing phase is the main factor in the WPD approach to be successful.

Thirdly, our approach is working based on extracting coefficients features from the packet size sequences of the traces. It's hard to classify website traces using WPD for combined features of the website traces (packet size, inter-arrival packet time, etc...) since the WPD extracted coefficients rely on the frequencies of packet size not on combined elements.

Finally, packet size values of the website traces are very important for applying WPD to extract the coefficients for the packet size traces. Since the size of data unit sent through the tor network is fixed (512 bytes), the website trace will almost contain the packet size of (512, 1024, 2048) values with plus-minus sign () indicating whether the packet is incoming or outgoing. These values will be repeated frequently, which in turn represent the trace packet sizes sequence in the format that allows WDP to extract more informative coefficients for classification. That means, our approach of WPD is not suitable if the packet sequence contains more variable values.

7.3 Future Work

Our research work can be extended to four main directions in the future:

- The aim of applying our proposed method in closed-world is to evaluate the ability of the classification method (feature extraction method + classifier used) to distinguish between the visited websites and to be used as a basis for comparison with the other proposed works in the field. For the realistic classification results, open world dataset can be used to evaluate the Wavelet packet decomposition approach for website fingerprinting.
- Traffic flow features are important to the classification problem, and different features may result in different classification results. Therefore, combining

statistical feature of the traffic flow with the packet size of traces will have a positive in the classification results.

- Since we obtained the best results of the websites classification using WPD method with back-propagation (BP) neural network classifier in our work, BP neural network can be optimized using specific optimization algorithms such as Particle swarm optimization (PSO) in order to increase the classification performance.

- Applying our proposed method to fingerprint a traffic of web services. This should have a high value in the anonymity systems literature since it will accurately indicate the impact of using web services on the anonymity of users (how much anonymity is lost when using web services).

REFERENCES

[1] E. Nygren, R. K. Sitaraman, and J. Sun, “The akamai network: a platform for high-performance internet applications,” *ACM SIGOPS Operating Systems Review*, vol. 44, no. 3, pp. 2–19, 2010.

[2] T. Wang and I. Goldberg, “Improved website fingerprinting on tor,” in *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. ACM, 2013, pp. 201–212.

[3] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh, “Stegotorus: a camouflage proxy for the tor anonymity system,” in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 109–120.

[4] P. C. González, “Fingerprinting tor,” *Information Management & Computer Security*, vol. 21, no. 2, pp. 73–90, 2013.

[5] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, “Touching from a distance: Website fingerprinting attacks and defenses,” in *Proceedings of the 2012 ACM*

conference on Computer and communications security. ACM, 2012, pp. 605–616.

[6] D. Herrmann, R. Wendolsky, and H. Federrath, “Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier,” in *Proceedings of the 2009 ACM workshop on Cloud computing security.* ACM, 2009, pp. 31–42.

[7] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, “Website fingerprinting in onion routing based anonymization networks,” in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society.* ACM, 2011, pp. 103–114.

[8] Y. Shi and K. Matsuura, “Fingerprinting attack on the tor anonymity system,” in *Information and Communications Security.* Springer, 2009, pp. 425–438.

[9] H. Federrath, “Japanonymity & privacy,” *URL http://anon. inf. tu-dresden. de/index en. html. Accessed*, vol. 10, 2007.

[10] M. Rennhard and B. Plattner, “Practical anonymity for the masses with morphmix,” in *Financial Cryptography.* Springer, 2004, pp. 233–250.

[11] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second-generation onion router,” DTIC Document, Tech. Rep., 2004.

[12] T. M. Portal, “Users. retrieved 7, december 2013.”

[13] S. DeFabbia-Kane, “Analyzing the effectiveness of passive correlation attacks on the tor anonymity network,” Ph.D. dissertation, Wesleyan University, 2011.

[14] A. Hintz, “Fingerprinting websites using traffic analysis,” in *Privacy Enhancing Technologies*. Springer, 2003, pp. 171–178.

[15] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine, “Privacy vulnerabilities in encrypted http streams,” in *Privacy Enhancing Technologies*. Springer, 2006, pp. 1–11.

[16] M. Liberatore and B. N. Levine, “Inferring the source of encrypted http connections,” in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 255–263.

[17] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, “Statistical identification of encrypted web browsing traffic,” in *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*. IEEE, 2002, pp. 19–30.

[18] T. Wang and I. Goldberg, “Comparing website fingerprinting attacks and defenses,” Technical Report 2013-30, CACR, 2013. http://cacr. uwaterloo. ca/techreports/2013/cacr2013-30. pdf, Tech. Rep.

[19] X. Fu, B. Graham, R. Bettati, and W. Zhao, “On countermeasures to traffic analysis attacks,” in *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society*. IEEE, 2003, pp. 188–195.

[20] H. Cheng and R. Avnur, "Traffic analysis of ssl encrypted web browsing," *URL citeseer. ist. psu. edu/656522. html*, 1998.

[21] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail," in *Security and Privacy (SP), 2012 IEEE Symposium on.* IEEE, 2012, pp. 332–346.

[22] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic morphing: An efficient defense against statistical traffic analysis." in *NDSS*, 2009.

[23] L. Lu, E.-C. Chang, and M. C. Chan, "Website fingerprinting and identification using ordered feature sequences," in *Computer Security–ESORICS 2010.* Springer, 2010, pp. 199–214.

[24] X. Luo, P. Zhou, E. W. Chan, W. Lee, R. K. Chang, and R. Perdisci, "Httpos: Sealing information leaks with browser-side obfuscation of encrypted flows." in *NDSS*, 2011.

[25] X. Gong, N. Kiyavash, and N. Borisov, "Fingerprinting websites using remote traffic analysis," in *Proceedings of the 17th ACM conference on Computer and communications security.* ACM, 2010, pp. 684–686.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[27] G. Aggarwal, E. Bursztein, C. Jackson, and D. Boneh, "An analysis of private browsing modes in modern browsers." in *USENIX Security Symposium*, 2010, pp. 79–94.

[28] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[29] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," in *Soviet physics doklady*, vol. 10, 1966, p. 707.

[30] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 11, pp. 1186–1191, 1993.

[31] P. Srinivasan and L. H. Jamieson, "High-quality audio compression using an adaptive wavelet packet decomposition and psychoacoustic modeling," *Signal Processing, IEEE Transactions on*, vol. 46, no. 4, pp. 1085–1093, 1998.

[32] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *Multimedia, IEEE Transactions on*, vol. 1, no. 3, pp. 264–277, 1999.

[33] G. G. Yen and K.-C. Lin, "Wavelet packet feature extraction for vibration monitoring," *Industrial Electronics, IEEE Transactions on*, vol. 47, no. 3, pp. 650–667, 2000.

[34] E. Rydgren, E. Thomas, F. Amiel, F. Rossant, and A. Amara, "Iris features extraction using wavelet packets," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, vol. 2. IEEE, 2004, pp. 861–864.

[35] L. Eren and M. J. Devaney, "Bearing damage detection via wavelet packet decomposition of the stator current," *Instrumentation and Measurement, IEEE Transactions on*, vol. 53, no. 2, pp. 431–436, 2004.

[36] H. Ocak, K. A. Loparo, and F. M. Discenzo, "Online tracking of bearing wear using wavelet packet decomposition and probabilistic modeling: A method for bearing prognostics," *Journal of sound and vibration*, vol. 302, no. 4, pp. 951–961, 2007.

[37] S. Ekici, S. Yildirim, and M. Poyraz, "Energy and entropy-based feature extraction for locating fault on transmission lines by using neural network and wavelet packet decomposition," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2937–2944, 2008.

[38] M. Edman and J. Hipple, "Vidalia: towards a usable tor gui," in *Proceedings of the 3rd symposium on Usable privacy and security*. ACM, 2007, pp. 163–164.

[39] R. Miller, M. Bolin, L. Chilton, G. Little, M. Webber, and C.-H. Yu, "Rewriting the web with chickenfoot," *No Code Required: Giving Users Tools to Transform the Web*, pp. 39–62, 2010.

[40] D. Burns, *Selenium 2 Testing Tools: Beginner's Guide.* Packt Publishing Ltd, 2012.

[41] J. Tan, X.-s. Chen, M. Du, and K. Zhu, "A novel internet traffic identification approach using wavelet packet decomposition and neural network," *Journal of Central South University*, vol. 19, pp. 2218–2230, 2012.

[42] D. Gusfield, *Algorithms on strings, trees and sequences: computer science and computational biology.* Cambridge University Press, 1997.

[43] M. Gokhale, D. K. Khanduja *et al.*, "Time domain signal analysis using wavelet packet decomposition approach," *Int'l J. of Communications, Network and System Sciences*, vol. 3, no. 03, p. 321, 2010.

[44] I. H. Witten, E. Frank, L. E. Trigg, M. A. Hall, G. Holmes, and S. J. Cunningham, "Weka: Practical machine learning tools and techniques with java implementations," 1999.

[45] J. Platt *et al.*, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[46] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *From Natural to Artificial Neural Computation.* Springer, 1995, pp. 195–201.

[47] H. Zhang and J. Su, "Naive bayesian classifiers for ranking," in *Machine Learning: ECML 2004.* Springer, 2004, pp. 501–512.

[48] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[49] J. S. Kaleka and R. Sharma, “Comparative performance analysis of haar, symlets and bior wavelets on image compression using discrete wavelet transform,” *International Journal of Computers & Distributed Systems*, vol. 1, no. 2, pp. 11–16, 2012.

[50] T. Gandhi, B. K. Panigrahi, and S. Anand, “A comparative study of wavelet families for eeg signal classification,” *Neurocomputing*, vol. 74, no. 17, pp. 3051–3057, 2011.

[51] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, “An optimal wavelet function based on wavelet denoising for multifunction myoelectric control,” in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on*, vol. 2. Ieee, 2009, pp. 1098–1101.

[52] E. Avci, “Comparison of wavelet families for texture classification by using wavelet packet entropy adaptive network based fuzzy inference system,” *Applied Soft Computing*, vol. 8, no. 1, pp. 225–231, 2008.

[53] ——, “Performance comparison of wavelet families for analog modulation classification using expert discrete wavelet neural network system,” *Expert Systems with Applications*, vol. 33, no. 1, pp. 23–35, 2007.

APPENDIX A : Visited Websites List

1. http://www.google.com
2. http://www.facebook.com
3. http://www.youtube.com
4. http://www.yahoo.com
5. http://www.baidu.com
6. http://www.en.wikipedia.org
7. http://www.ebay.com
8. http://www.live.com
9. http://www.taobao.com
10. http://www.linkedin.com
11. http://www.sina.com.cn
12. http://www.twitter.com
13. http://www.amazon.com
14. http://www.hao123.com
15. http://www.google.co.in
16. http://www.blogspot.com

17. http://www.weibo.com

18. http://www.tmall.com

19. http://www.wordpress.com

20. http://www.ask.com

APPENDIX B : Detailed Performance Metrics For SVM classifier and WPD method

1. Chrome browser

   (a) WPD 1^{st} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.400	0.025	0.462	0.400	0.429	0.402	0.876	0.366	1
	0.667	0.011	0.769	0.667	0.714	0.702	0.901	0.623	2
	0.267	0.004	0.800	0.267	0.400	0.448	0.936	0.664	3
	0.067	0.004	0.500	0.067	0.118	0.169	0.538	0.099	4
	0.933	0.105	0.318	0.933	0.475	0.510	0.934	0.351	5
	0.733	0.025	0.611	0.733	0.667	0.650	0.874	0.515	6
	0.400	0.000	1.000	0.400	0.571	0.623	0.948	0.694	7
	0.600	0.004	0.900	0.600	0.720	0.724	0.997	0.899	8
	0.467	0.000	1.000	0.467	0.636	0.674	0.870	0.759	9
	0.667	0.228	0.133	0.667	0.222	0.221	0.762	0.117	10
	0.533	0.004	0.889	0.533	0.667	0.677	0.997	0.906	11
	0.800	0.014	0.750	0.800	0.774	0.762	0.954	0.670	12
	0.000	0.000	0.000	0.000	0.000	0.000	0.903	0.690	13
	0.333	0.032	0.357	0.333	0.345	0.312	0.777	0.194	14
	0.467	0.032	0.438	0.467	0.452	0.422	0.876	0.308	15
	0.467	0.028	0.467	0.467	0.467	0.439	0.929	0.372	16
	0.200	0.000	1.000	0.200	0.333	0.438	0.700	0.309	17
	0.000	0.014	0.000	0.000	0.000	-0.027	0.882	0.237	18
	0.667	0.007	0.833	0.667	0.741	0.734	0.935	0.677	19
	0.533	0.035	0.444	0.533	0.485	0.457	0.870	0.340	20
Weighted Avg.	0.460	0.028	0.584	0.460	0.461	0.467	0.873	0.489	

Figure B.1: Accuracy of WPD first level for chrome browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
6	0	0	0	3	0	0	0	0	3	0	0	0	0	2	0	0	0	0	1	a = 1
0	10	0	0	0	0	0	0	0	3	0	1	0	0	0	1	0	0	0	0	b = 2
3	0	4	0	0	0	0	0	0	4	0	0	0	0	1	0	0	1	0	2	c = 3
1	1	0	1	2	0	0	0	0	7	0	1	0	0	0	1	0	0	0	1	d = 4
0	0	0	0	14	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	0	11	0	0	0	4	0	0	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	2	0	6	0	0	4	0	1	0	0	0	0	0	0	0	2	g = 7
0	0	0	0	1	0	0	9	0	3	0	0	0	0	0	2	0	0	0	0	h = 8
1	0	0	0	0	0	0	0	7	6	0	0	0	0	0	1	0	0	0	0	i = 9
0	1	0	0	0	0	0	0	0	10	0	0	0	0	0	2	0	0	0	2	j = 10
0	0	0	0	0	1	0	0	0	0	8	0	0	2	0	1	0	3	0	0	k = 11
0	0	0	0	0	0	0	0	0	3	0	12	0	0	0	0	0	0	0	0	l = 12
0	0	1	0	3	1	0	0	0	3	0	0	0	6	0	0	0	0	0	1	m = 13
0	0	0	0	5	1	0	0	0	4	0	0	0	5	0	0	0	0	0	0	n = 14
2	0	0	0	1	1	0	0	0	1	0	0	0	1	7	0	0	0	2	0	o = 15
0	0	0	0	1	0	0	1	0	6	0	0	0	0	0	7	0	0	0	0	p = 16
0	0	0	0	3	2	0	0	0	3	0	0	0	0	4	0	3	0	0	0	q = 17
0	0	0	1	9	0	0	0	0	2	1	1	0	0	0	0	0	0	0	1	r = 18
0	0	0	0	0	1	0	0	0	2	0	0	0	0	2	0	0	0	10	0	s = 19
0	1	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	8	t = 20

Figure B.2: Confusion matrix of WPD first level for chrome browser

(b) WPD 2^{nd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.533	0.021	0.571	0.533	0.552	0.529	0.927	0.479	1
	0.800	0.007	0.857	0.800	0.828	0.819	0.898	0.723	2
	0.667	0.007	0.833	0.667	0.741	0.734	0.975	0.776	3
	0.067	0.007	0.333	0.067	0.111	0.131	0.607	0.206	4
	0.933	0.077	0.389	0.933	0.549	0.574	0.961	0.461	5
	0.800	0.018	0.706	0.800	0.750	0.738	0.944	0.667	6
	0.733	0.004	0.917	0.733	0.815	0.812	0.962	0.758	7
	0.867	0.007	0.867	0.867	0.867	0.860	0.996	0.888	8
	0.733	0.000	1.000	0.733	0.846	0.850	0.906	0.807	9
	0.600	0.133	0.191	0.600	0.290	0.280	0.807	0.159	10
	0.933	0.004	0.933	0.933	0.933	0.930	0.998	0.934	11
	0.867	0.007	0.867	0.867	0.867	0.860	0.961	0.789	12
	0.467	0.000	1.000	0.467	0.636	0.674	0.942	0.761	13
	0.400	0.018	0.545	0.400	0.462	0.444	0.834	0.302	14
	0.600	0.042	0.429	0.600	0.500	0.477	0.910	0.322	15
	0.533	0.021	0.571	0.533	0.552	0.529	0.933	0.478	16
	0.200	0.014	0.429	0.200	0.273	0.268	0.733	0.190	17
	0.133	0.000	1.000	0.133	0.235	0.357	0.884	0.372	18
	0.667	0.007	0.833	0.667	0.741	0.734	0.947	0.684	19
	0.400	0.032	0.400	0.400	0.400	0.368	0.878	0.307	20
Weighted Avg.	0.597	0.021	0.684	0.597	0.597	0.598	0.900	0.553	

Figure B.3: Accuracy of WPD second level for chrome browser

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t   <-- classified as
  8  0  0  0  2  0  0  0  0  1  0  0  0  0  2  0  0  0  0  2 |  a = 1
  0 12  0  1  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0 |  b = 2
  0  0 10  0  1  0  0  0  0  1  0  0  0  0  0  0  2  0  0  1 |  c = 3
  1  1  1  1  2  1  0  0  0  5  0  0  0  0  0  0  0  0  0  3 |  d = 4
  0  0  0  0 14  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0 |  e = 5
  0  0  0  0  0 12  0  0  0  3  0  0  0  0  0  0  0  0  0  0 |  f = 6
  2  0  0  0  0  0 11  0  0  1  0  0  0  0  0  0  0  0  0  1 |  g = 7
  0  0  0  0  0  0  0 13  0  0  0  0  0  0  0  2  0  0  0  0 |  h = 8
  0  0  0  0  0  0  1  0 11  2  0  0  0  0  0  1  0  0  0  0 |  i = 9
  0  1  0  0  0  1  0  1  0  9  0  2  0  0  0  1  0  0  0  0 |  j = 10
  0  0  0  0  0  0  0  0  0  0 14  0  0  1  0  0  0  0  0  0 |  k = 11
  0  0  0  0  0  0  0  0  0  2  0 13  0  0  0  0  0  0  0  0 |  l = 12
  0  0  1  0  2  0  0  0  0  2  0  0  7  1  0  0  0  0  0  2 |  m = 13
  0  0  0  0  3  1  0  0  0  4  0  0  0  6  1  0  0  0  0  0 |  n = 14
  1  0  0  0  2  0  0  0  0  0  0  0  0  0  9  0  1  0  2  0 |  o = 15
  0  0  0  0  0  0  0  1  0  6  0  0  0  0  0  8  0  0  0  0 |  p = 16
  0  0  0  0  3  2  0  0  0  2  0  0  0  0  5  0  3  0  0  0 |  q = 17
  0  0  0  1  6  0  0  0  0  1  1  0  0  3  0  0  1  2  0  0 |  r = 18
  0  0  0  0  1  0  0  0  0  2  0  0  0  0  2  0  0  0 10  0 |  s = 19
  2  0  0  0  0  0  0  0  0  5  0  0  0  0  2  0  0  0  0  6 |  t = 20
```

Figure B.4: Confusion matrix of WPD second level for chrome browser

(c) WPD 3^{rd} level

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.667    0.035    0.500      0.667    0.571      0.552    0.932     0.450     1
                 0.733    0.011    0.786      0.733    0.759      0.747    0.916     0.698     2
                 0.733    0.007    0.846      0.733    0.786      0.777    0.981     0.731     3
                 0.200    0.011    0.500      0.200    0.286      0.295    0.718     0.274     4
                 0.933    0.035    0.583      0.933    0.718      0.722    0.940     0.598     5
                 0.867    0.035    0.565      0.867    0.684      0.681    0.934     0.546     6
                 0.800    0.004    0.923      0.800    0.857      0.853    0.964     0.874     7
                 0.867    0.004    0.929      0.867    0.897      0.892    0.995     0.879     8
                 0.800    0.004    0.923      0.800    0.857      0.853    0.867     0.752     9
                 0.733    0.081    0.324      0.733    0.449      0.449    0.860     0.285     10
                 0.933    0.007    0.875      0.933    0.903      0.898    0.996     0.875     11
                 0.867    0.011    0.813      0.867    0.839      0.830    0.975     0.746     12
                 0.733    0.000    1.000      0.733    0.846      0.850    0.985     0.868     13
                 0.333    0.011    0.625      0.333    0.435      0.437    0.847     0.321     14
                 0.533    0.028    0.500      0.533    0.516      0.490    0.910     0.376     15
                 0.667    0.018    0.667      0.667    0.667      0.649    0.976     0.597     16
                 0.400    0.021    0.500      0.400    0.444      0.421    0.797     0.377     17
                 0.467    0.000    1.000      0.467    0.636      0.674    0.915     0.625     18
                 0.800    0.004    0.923      0.800    0.857      0.853    0.979     0.805     19
                 0.533    0.014    0.667      0.533    0.593      0.578    0.902     0.475     20
Weighted Avg.    0.680    0.017    0.722      0.680    0.680      0.675    0.920     0.608
```

Figure B.5: Accuracy of WPD third level for chrome browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
8	0	0	0	2	0	0	0	0	1	0	0	0	0	2	0	0	0	0	2	a = 1
0	12	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	b = 2
0	0	10	0	1	0	0	0	0	1	0	0	0	0	0	0	2	0	0	1	c = 3
1	1	1	1	2	1	0	0	0	5	0	0	0	0	0	0	0	0	0	3	d = 4
0	0	0	0	14	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	0	12	0	0	0	3	0	0	0	0	0	0	0	0	0	0	f = 6
2	0	0	0	0	0	11	0	0	1	0	0	0	0	0	0	0	0	0	1	g = 7
0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	2	0	0	0	0	h = 8
0	0	0	0	0	0	1	0	11	2	0	0	0	0	0	1	0	0	0	0	i = 9
0	1	0	0	0	1	0	1	0	9	0	2	0	0	0	1	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	0	14	0	0	1	0	0	0	0	0	0	k = 11
0	0	0	0	0	0	0	0	0	2	0	13	0	0	0	0	0	0	0	0	l = 12
0	0	1	0	2	0	0	0	0	2	0	0	7	1	0	0	0	0	0	2	m = 13
0	0	0	0	3	1	0	0	0	4	0	0	0	6	1	0	0	0	0	0	n = 14
1	0	0	0	2	0	0	0	0	0	0	0	0	0	9	0	1	0	2	0	o = 15
0	0	0	0	0	0	0	1	0	6	0	0	0	0	0	8	0	0	0	0	p = 16
0	0	0	0	3	2	0	0	0	2	0	0	0	0	5	0	3	0	0	0	q = 17
0	0	0	1	6	0	0	0	0	1	1	0	0	3	0	0	1	2	0	0	r = 18
0	0	0	0	1	0	0	0	0	2	0	0	0	0	2	0	0	0	10	0	s = 19
2	0	0	0	0	0	0	0	0	5	0	0	0	0	2	0	0	0	0	6	t = 20

Figure B.6: Confusion matrix of WPD third level for chrome browser

(d) WPD 4^{th} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.600	0.021	0.600	0.600	0.600	0.579	0.920	0.433	1
	0.733	0.018	0.688	0.733	0.710	0.694	0.924	0.604	2
	0.733	0.014	0.733	0.733	0.733	0.719	0.971	0.604	3
	0.267	0.014	0.500	0.267	0.348	0.342	0.784	0.318	4
	0.867	0.018	0.722	0.867	0.788	0.779	0.977	0.794	5
	0.800	0.025	0.632	0.800	0.706	0.694	0.970	0.622	6
	0.867	0.004	0.929	0.867	0.897	0.892	0.971	0.830	7
	0.867	0.011	0.813	0.867	0.839	0.830	0.993	0.816	8
	0.800	0.007	0.857	0.800	0.828	0.819	0.886	0.754	9
	0.667	0.070	0.333	0.667	0.444	0.433	0.889	0.269	10
	0.933	0.004	0.933	0.933	0.933	0.930	0.998	0.921	11
	0.800	0.014	0.750	0.800	0.774	0.762	0.986	0.696	12
	0.733	0.000	1.000	0.733	0.846	0.850	0.985	0.865	13
	0.467	0.028	0.467	0.467	0.467	0.439	0.895	0.371	14
	0.733	0.021	0.647	0.733	0.688	0.671	0.942	0.537	15
	0.800	0.007	0.857	0.800	0.828	0.819	0.982	0.758	16
	0.600	0.025	0.563	0.600	0.581	0.558	0.817	0.418	17
	0.400	0.004	0.857	0.400	0.545	0.572	0.897	0.506	18
	0.800	0.000	1.000	0.800	0.889	0.890	0.962	0.861	19
	0.600	0.011	0.750	0.600	0.667	0.656	0.912	0.576	20
Weighted Avg.	0.703	0.016	0.732	0.703	0.705	0.697	0.933	0.628	

Figure B.7: Accuracy of WPD fourth level for chrome browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
9	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	0	0	2	a = 1
0	11	0	0	0	0	0	0	0	2	0	1	0	0	0	1	0	0	0	0	b = 2
0	0	11	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	c = 3
1	1	1	4	1	0	0	0	0	3	0	0	0	1	0	0	3	0	0	0	d = 4
0	0	0	0	13	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	0	12	0	1	0	2	0	0	0	0	0	0	0	0	0	0	f = 6
0	0	1	0	0	0	13	0	0	0	1	0	0	0	0	0	0	0	0	0	g = 7
0	0	0	0	0	0	0	13	0	2	0	0	0	0	0	0	0	0	0	0	h = 8
0	0	0	1	0	0	0	0	12	1	0	0	0	0	0	0	0	0	0	1	i = 9
0	1	0	0	0	1	0	1	0	10	0	1	0	1	0	0	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	0	14	0	0	1	0	0	0	0	0	0	k = 11
0	1	0	0	0	0	0	0	0	2	0	12	0	0	0	0	0	0	0	0	l = 12
0	0	1	1	0	0	0	0	1	0	0	0	11	1	0	0	0	0	0	0	m = 13
1	0	1	0	2	3	0	0	0	0	0	0	0	7	1	0	0	0	0	0	n = 14
2	0	0	0	0	1	0	0	0	0	0	0	0	0	11	0	1	0	0	0	o = 15
0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	12	0	0	0	0	p = 16
0	0	0	1	0	0	0	0	0	2	0	0	0	1	2	0	9	0	0	0	q = 17
0	1	0	0	2	1	0	0	0	1	0	0	0	3	0	0	1	6	0	0	r = 18
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	12	0	s = 19
2	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	9	t = 20

Figure B.8: Confusion matrix of WPD fourth level for chrome browser

2. Firefox browser

   (a) WPD 1^{st} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.467	0.028	0.467	0.467	0.467	0.439	0.900	0.325	1
	0.933	0.000	1.000	0.933	0.966	0.964	0.999	0.975	2
	0.733	0.004	0.917	0.733	0.815	0.812	0.997	0.909	3
	0.067	0.004	0.500	0.067	0.118	0.169	0.821	0.370	4
	1.000	0.028	0.652	1.000	0.789	0.796	0.989	0.714	5
	0.933	0.018	0.737	0.933	0.824	0.819	0.987	0.709	6
	0.800	0.000	1.000	0.800	0.889	0.890	0.948	0.937	7
	0.800	0.018	0.706	0.800	0.750	0.738	0.972	0.724	8
	0.800	0.000	1.000	0.800	0.889	0.890	0.997	0.953	9
	0.867	0.126	0.265	0.867	0.406	0.436	0.901	0.242	10
	0.867	0.004	0.929	0.867	0.897	0.892	0.997	0.916	11
	0.867	0.018	0.722	0.867	0.788	0.779	0.930	0.653	12
	0.333	0.004	0.833	0.333	0.476	0.513	0.924	0.731	13
	0.600	0.011	0.750	0.600	0.667	0.656	0.947	0.625	14
	0.600	0.025	0.563	0.600	0.581	0.558	0.964	0.480	15
	0.933	0.025	0.667	0.933	0.778	0.776	0.986	0.659	16
	0.600	0.007	0.818	0.600	0.692	0.688	0.882	0.657	17
	0.267	0.000	1.000	0.267	0.421	0.507	0.989	0.850	18
	0.800	0.004	0.923	0.800	0.857	0.853	0.988	0.822	19
	0.600	0.004	0.900	0.600	0.720	0.724	0.858	0.633	20
Weighted Avg.	0.693	0.016	0.767	0.693	0.689	0.695	0.949	0.694	

Figure B.9: Accuracy of WPD first level for Firefox browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
7	0	0	0	1	0	0	0	0	2	0	0	0	0	4	1	0	0	0	0	a = 1
0	14	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	11	0	0	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	c = 3
0	0	1	1	0	0	0	1	0	10	0	1	0	1	0	0	0	0	0	0	d = 4
0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	f = 6
1	0	0	0	0	0	12	0	0	0	0	0	0	0	1	1	0	0	0	0	g = 7
1	0	0	0	0	0	0	12	0	0	0	0	0	0	0	2	0	0	0	0	h = 8
0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	3	0	0	0	0	i = 9
0	0	0	0	0	0	0	2	0	13	0	0	0	0	0	0	0	0	0	0	j = 10
1	0	0	1	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	k = 11
0	0	0	0	0	0	0	0	0	1	0	13	0	0	1	0	0	0	0	0	l = 12
0	0	0	0	0	0	0	0	0	9	1	0	5	0	0	0	0	0	0	0	m = 13
0	0	0	0	0	0	0	0	0	6	0	0	0	9	0	0	0	0	0	0	n = 14
5	0	0	0	0	0	0	0	0	1	0	0	0	0	9	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	14	0	0	0	0	p = 16
0	0	0	0	1	1	0	0	0	2	0	1	0	0	1	0	9	0	0	0	q = 17
0	0	0	0	5	0	0	0	0	2	0	2	0	2	0	0	0	4	0	0	r = 18
0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	12	0	s = 19
0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	2	0	1	9	t = 20

Figure B.10: Confusion matrix of WPD first level for Firefox browser

(b) WPD 2^{nd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.467	0.028	0.467	0.467	0.467	0.439	0.900	0.325	1
	0.933	0.000	1.000	0.933	0.966	0.964	0.999	0.975	2
	0.733	0.004	0.917	0.733	0.815	0.812	0.997	0.909	3
	0.067	0.004	0.500	0.067	0.118	0.169	0.821	0.370	4
	1.000	0.028	0.652	1.000	0.789	0.796	0.989	0.714	5
	0.933	0.018	0.737	0.933	0.824	0.819	0.987	0.709	6
	0.800	0.000	1.000	0.800	0.889	0.890	0.948	0.937	7
	0.800	0.018	0.706	0.800	0.750	0.738	0.972	0.724	8
	0.800	0.000	1.000	0.800	0.889	0.890	0.997	0.953	9
	0.867	0.126	0.265	0.867	0.406	0.436	0.901	0.242	10
	0.867	0.004	0.929	0.867	0.897	0.892	0.997	0.916	11
	0.867	0.018	0.722	0.867	0.788	0.779	0.930	0.653	12
	0.333	0.004	0.833	0.333	0.476	0.513	0.924	0.731	13
	0.600	0.011	0.750	0.600	0.667	0.656	0.947	0.625	14
	0.600	0.025	0.563	0.600	0.581	0.558	0.964	0.480	15
	0.933	0.025	0.667	0.933	0.778	0.776	0.986	0.659	16
	0.600	0.007	0.818	0.600	0.692	0.688	0.882	0.657	17
	0.267	0.000	1.000	0.267	0.421	0.507	0.989	0.850	18
	0.800	0.004	0.923	0.800	0.857	0.853	0.988	0.822	19
	0.600	0.004	0.900	0.600	0.720	0.724	0.858	0.633	20
Weighted Avg.	0.693	0.016	0.767	0.693	0.689	0.695	0.949	0.694	

Figure B.11: Accuracy of WPD second level for Firefox browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
7	0	0	0	1	0	0	0	0	2	0	0	0	0	4	1	0	0	0	0	a = 1
0	14	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	11	0	0	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	c = 3
0	0	1	1	0	0	0	1	0	10	0	1	0	1	0	0	0	0	0	0	d = 4
0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	f = 6
1	0	0	0	0	0	12	0	0	0	0	0	0	0	1	1	0	0	0	0	g = 7
1	0	0	0	0	0	0	12	0	0	0	0	0	0	0	2	0	0	0	0	h = 8
0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	3	0	0	0	0	i = 9
0	0	0	0	0	0	0	2	0	13	0	0	0	0	0	0	0	0	0	0	j = 10
1	0	0	1	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	k = 11
0	0	0	0	0	0	0	0	0	1	0	13	0	0	1	0	0	0	0	0	l = 12
0	0	0	0	0	0	0	0	0	9	1	0	5	0	0	0	0	0	0	0	m = 13
0	0	0	0	0	0	0	0	0	6	0	0	0	9	0	0	0	0	0	0	n = 14
5	0	0	0	0	0	0	0	0	1	0	0	0	0	9	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	14	0	0	0	0	p = 16
0	0	0	0	1	1	0	0	0	2	0	1	0	0	1	0	9	0	0	0	q = 17
0	0	0	0	5	0	0	0	0	2	0	2	0	2	0	0	0	4	0	0	r = 18
0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	12	0	s = 19
0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	2	0	1	9	t = 20

Figure B.12: Confusion matrix of WPD second level for Firefox browser

(c) WPD 3^{rd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.467	0.021	0.538	0.467	0.500	0.477	0.900	0.347	1
	1.000	0.014	0.789	1.000	0.882	0.882	0.995	0.833	2
	0.667	0.004	0.909	0.667	0.769	0.769	0.994	0.905	3
	0.200	0.004	0.750	0.200	0.316	0.373	0.879	0.490	4
	1.000	0.021	0.714	1.000	0.833	0.836	0.989	0.714	5
	0.867	0.011	0.813	0.867	0.839	0.830	0.990	0.770	6
	0.933	0.000	1.000	0.933	0.966	0.964	0.948	0.937	7
	0.800	0.018	0.706	0.800	0.750	0.738	0.976	0.768	8
	0.867	0.000	1.000	0.867	0.929	0.928	0.998	0.965	9
	0.800	0.095	0.308	0.800	0.444	0.457	0.900	0.293	10
	0.933	0.000	1.000	0.933	0.966	0.964	0.999	0.981	11
	0.867	0.004	0.929	0.867	0.897	0.892	0.937	0.851	12
	0.533	0.000	1.000	0.533	0.696	0.721	0.931	0.850	13
	0.667	0.014	0.714	0.667	0.690	0.674	0.947	0.570	14
	0.733	0.021	0.647	0.733	0.688	0.671	0.968	0.566	15
	0.933	0.018	0.737	0.933	0.824	0.819	0.992	0.764	16
	0.733	0.021	0.647	0.733	0.688	0.671	0.870	0.526	17
	0.333	0.000	1.000	0.333	0.500	0.567	0.979	0.842	18
	0.800	0.004	0.923	0.800	0.857	0.853	0.977	0.791	19
	0.667	0.007	0.833	0.667	0.741	0.734	0.854	0.679	20
Weighted Avg.	0.740	0.014	0.798	0.740	0.739	0.741	0.951	0.722	

Figure B.13: Accuracy of WPD third level for Firefox browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
7	1	0	0	0	0	0	0	0	1	0	0	0	0	5	0	0	0	0	1	a = 1
0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	10	1	0	1	0	0	0	0	0	0	0	1	0	0	2	0	0	0	c = 3
0	1	1	3	0	0	0	0	0	8	0	1	0	0	0	0	1	0	0	0	d = 4
0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = 5
1	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	f = 6
0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	1	0	0	0	0	g = 7
1	0	0	0	0	0	0	12	0	0	0	0	0	0	0	2	0	0	0	0	h = 8
0	0	0	0	0	2	0	0	13	0	0	0	0	0	0	0	0	0	0	0	i = 9
0	0	0	0	0	0	0	3	0	12	0	0	0	0	0	0	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	1	0	0	0	0	k = 11
0	0	0	0	0	0	0	0	0	1	0	13	0	0	1	0	0	0	0	0	l = 12
0	0	0	0	0	0	0	0	0	7	0	0	8	0	0	0	0	0	0	0	m = 13
0	0	0	0	0	0	0	0	0	5	0	0	0	10	0	0	0	0	0	0	n = 14
3	0	0	0	0	0	0	0	0	1	0	0	0	0	11	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	14	0	0	0	0	p = 16
0	2	0	0	1	0	0	0	0	1	0	0	0	0	0	0	11	0	0	0	q = 17
0	0	0	0	5	0	0	0	0	0	0	0	0	3	0	0	2	5	0	0	r = 18
0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	12	0	s = 19
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	10	t = 20

Figure B.14: Confusion matrix of WPD third level for Firefox browser

(d) WPD 4^{th} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.400	0.021	0.500	0.400	0.444	0.421	0.919	0.331	1
	1.000	0.014	0.789	1.000	0.882	0.882	0.989	0.753	2
	0.733	0.000	1.000	0.733	0.846	0.850	0.998	0.959	3
	0.400	0.007	0.750	0.400	0.522	0.532	0.880	0.433	4
	0.933	0.028	0.636	0.933	0.757	0.757	0.984	0.645	5
	0.933	0.007	0.875	0.933	0.903	0.898	0.995	0.848	6
	0.933	0.000	1.000	0.933	0.966	0.964	0.958	0.938	7
	0.867	0.018	0.722	0.867	0.788	0.779	0.955	0.663	8
	1.000	0.004	0.938	1.000	0.968	0.967	1.000	1.000	9
	0.667	0.042	0.455	0.667	0.541	0.522	0.880	0.343	10
	0.933	0.004	0.933	0.933	0.933	0.930	0.994	0.899	11
	0.867	0.007	0.867	0.867	0.867	0.860	0.935	0.777	12
	0.867	0.000	1.000	0.867	0.929	0.928	0.938	0.925	13
	0.733	0.011	0.786	0.733	0.759	0.747	0.922	0.619	14
	0.800	0.028	0.600	0.800	0.686	0.674	0.969	0.540	15
	0.933	0.004	0.933	0.933	0.933	0.930	0.998	0.973	16
	0.733	0.028	0.579	0.733	0.647	0.631	0.868	0.453	17
	0.533	0.000	1.000	0.533	0.696	0.721	0.977	0.839	18
	0.800	0.007	0.857	0.800	0.828	0.819	0.939	0.713	19
	0.600	0.000	1.000	0.600	0.750	0.767	0.871	0.707	20
Weighted Avg.	0.783	0.011	0.811	0.783	0.782	0.779	0.948	0.718	

Figure B.15: Accuracy of WPD fourth level for Firefox browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
6	1	0	0	0	0	0	0	1	0	0	0	0	0	6	0	1	0	0	0	a = 1
0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	11	1	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	c = 3
0	1	0	6	0	0	0	0	0	4	0	1	0	2	0	0	1	0	0	0	d = 4
0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	e = 5
1	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	0	0	14	0	0	0	1	0	0	0	0	0	0	0	0	0	g = 7
1	0	0	0	0	0	0	13	0	0	0	0	0	0	0	1	0	0	0	0	h = 8
0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	i = 9
0	0	0	0	1	0	0	3	0	10	0	0	0	1	0	0	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	0	14	0	0	0	1	0	0	0	0	0	k = 11
0	1	0	0	0	0	0	0	0	0	0	13	0	0	0	0	1	0	0	0	l = 12
0	0	0	0	0	0	0	0	0	2	0	0	13	0	0	0	0	0	0	0	m = 13
0	0	0	0	0	0	0	0	0	4	0	0	0	11	0	0	0	0	0	0	n = 14
2	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	1	0	0	0	o = 15
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	14	0	0	0	0	p = 16
0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	11	0	0	0	q = 17
0	0	0	0	4	0	0	0	0	0	0	1	0	0	0	0	2	8	0	0	r = 18
0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	12	0	s = 19
2	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	9	t = 20

Figure B.16: Confusion matrix of WPD fourth level for Firefox browser

3. Safari browser

(a) WPD 1^{st} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.400	0.014	0.600	0.400	0.480	0.469	0.954	0.453	1
	0.800	0.091	0.316	0.800	0.453	0.464	0.891	0.302	2
	0.267	0.007	0.667	0.267	0.381	0.404	0.824	0.343	3
	0.467	0.011	0.700	0.467	0.560	0.554	0.878	0.552	4
	0.667	0.154	0.185	0.667	0.290	0.291	0.779	0.169	5
	0.800	0.063	0.400	0.800	0.533	0.535	0.953	0.392	6
	0.467	0.011	0.700	0.467	0.560	0.554	0.857	0.392	7
	0.667	0.014	0.714	0.667	0.690	0.674	0.958	0.623	8
	0.067	0.000	1.000	0.067	0.125	0.252	0.803	0.607	9
	0.800	0.070	0.375	0.800	0.511	0.515	0.951	0.359	10
	0.333	0.000	1.000	0.333	0.500	0.567	0.849	0.719	11
	0.733	0.035	0.524	0.733	0.611	0.596	0.905	0.436	12
	0.000	0.007	0.000	0.000	0.000	-0.019	0.775	0.214	13
	0.533	0.011	0.727	0.533	0.615	0.606	0.981	0.622	14
	0.667	0.018	0.667	0.667	0.667	0.649	0.980	0.611	15
	0.667	0.007	0.833	0.667	0.741	0.734	0.967	0.676	16
	0.267	0.004	0.800	0.267	0.400	0.448	0.891	0.538	17
	0.000	0.000	0.000	0.000	0.000	0.000	0.700	0.171	18
	0.467	0.025	0.500	0.467	0.483	0.457	0.849	0.340	19
	0.467	0.011	0.700	0.467	0.560	0.554	0.916	0.522	20
Weighted Avg.	0.477	0.028	0.570	0.477	0.458	0.465	0.883	0.452	

Figure B.17: Accuracy of WPD first level for safari browser

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t   <-- classified as
  6  1  0  0  0  0  0  0  0  3  0  0  0  0  3  0  0  0  2  0 |  a = 1
  0 12  0  0  0  1  0  0  0  1  0  1  0  0  0  0  0  0  0  0 |  b = 2
  0  6  4  0  2  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  c = 3
  0  2  0  7  0  0  0  0  0  1  0  0  0  0  0  1  0  0  3  1 |  d = 4
  0  3  0  0 10  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1 |  e = 5
  0  0  0  0  2 12  0  0  0  0  0  0  0  0  0  0  0  0  1  0 |  f = 6
  0  1  0  0  2  2  7  0  0  1  0  2  0  0  0  0  0  0  0  0 |  g = 7
  0  0  0  1  4  0  0 10  0  0  0  0  0  0  0  0  0  0  0  0 |  h = 8
  0  2  0  0  2  4  1  0  1  1  0  2  1  0  0  0  0  0  0  1 |  i = 9
  0  0  0  0  0  0  0  0  0 12  0  3  0  0  0  0  0  0  0  0 |  j = 10
  0  1  1  2  2  0  0  0  0  1  5  1  1  1  0  0  0  0  0  0 |  k = 11
  0  3  0  0  0  0  0  0  0  0  0 11  0  0  0  1  0  0  0  0 |  l = 12
  0  0  0  0  8  2  0  0  0  4  0  0  0  0  1  0  0  0  0  0 |  m = 13
  0  0  0  0  6  0  0  0  0  1  0  0  0  8  0  0  0  0  0  0 |  n = 14
  2  1  0  0  0  0  2  0  0  0  0  0  0  0 10  0  0  0  0  0 |  o = 15
  0  0  0  0  1  1  0  1  0  0  0  0  0  2  0 10  0  0  0  0 |  p = 16
  0  1  0  0  7  0  0  1  0  2  0  0  0  0  0  0  4  0  0  0 |  q = 17
  1  4  1  0  3  3  0  0  0  2  0  0  0  0  1  0  0  0  0  0 |  r = 18
  1  1  0  0  1  1  0  0  0  3  0  1  0  0  0  0  0  0  7  0 |  s = 19
  0  0  0  0  4  0  0  2  0  0  0  0  0  0  0  0  1  0  1  7 |  t = 20
```

Figure B.18: Confusion matrix of WPD first level for safari browser

(b) WPD 2^{nd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.667	0.039	0.476	0.667	0.556	0.536	0.965	0.474	1
	0.800	0.063	0.400	0.800	0.533	0.535	0.939	0.382	2
	0.333	0.032	0.357	0.333	0.345	0.312	0.856	0.274	3
	0.400	0.014	0.600	0.400	0.480	0.469	0.901	0.431	4
	0.667	0.060	0.370	0.667	0.476	0.462	0.879	0.342	5
	0.600	0.060	0.346	0.600	0.439	0.419	0.883	0.291	6
	0.333	0.004	0.833	0.333	0.476	0.513	0.896	0.518	7
	0.800	0.014	0.750	0.800	0.774	0.762	0.970	0.704	8
	0.467	0.004	0.875	0.467	0.609	0.627	0.941	0.668	9
	0.733	0.060	0.393	0.733	0.512	0.505	0.951	0.367	10
	0.467	0.011	0.700	0.467	0.560	0.554	0.915	0.617	11
	0.800	0.046	0.480	0.800	0.600	0.595	0.917	0.435	12
	0.200	0.011	0.500	0.200	0.286	0.295	0.731	0.243	13
	0.733	0.014	0.733	0.733	0.733	0.719	0.977	0.680	14
	0.533	0.007	0.800	0.533	0.640	0.639	0.979	0.677	15
	0.533	0.018	0.615	0.533	0.571	0.552	0.913	0.495	16
	0.600	0.011	0.750	0.600	0.667	0.656	0.909	0.617	17
	0.067	0.000	1.000	0.067	0.125	0.252	0.826	0.415	18
	0.400	0.021	0.500	0.400	0.444	0.421	0.849	0.337	19
	0.467	0.011	0.700	0.467	0.560	0.554	0.907	0.495	20
Weighted Avg.	0.530	0.025	0.609	0.530	0.519	0.519	0.905	0.473	

Figure B.19: Accuracy of WPD second level for safari browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
10	0	0	0	0	1	0	0	0	2	0	0	0	0	1	0	0	0	1	0	a = 1
0	12	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	b = 2
0	3	5	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = 3
0	2	0	6	2	0	0	0	0	0	0	2	0	0	0	1	0	0	2	0	d = 4
0	2	0	0	10	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	e = 5
0	2	1	0	1	9	0	0	0	1	0	0	0	0	0	1	0	0	0	0	f = 6
3	1	1	0	0	1	5	0	0	2	0	2	0	0	0	0	0	0	0	0	g = 7
0	0	0	1	0	0	0	12	0	0	0	0	0	0	0	1	0	0	0	1	h = 8
0	1	1	0	1	2	0	0	7	1	0	0	1	0	0	0	0	0	0	1	i = 9
0	0	0	0	0	0	0	0	0	11	0	4	0	0	0	0	0	0	0	0	j = 10
0	0	1	1	0	1	0	0	0	0	7	0	2	1	1	1	0	0	0	0	k = 11
0	2	0	0	0	0	0	0	0	0	0	12	0	0	0	1	0	0	0	0	l = 12
0	0	1	0	3	0	0	0	0	4	1	0	3	1	0	0	1	0	1	0	m = 13
0	0	1	0	2	0	0	0	0	1	0	0	0	11	0	0	0	0	0	0	n = 14
4	1	0	0	0	0	1	0	0	0	0	1	0	0	8	0	0	0	0	0	o = 15
0	0	0	1	3	1	0	1	0	0	0	0	0	1	0	8	0	0	0	0	p = 16
2	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	9	0	0	1	q = 17
1	0	3	0	2	1	0	0	1	2	2	0	0	0	0	0	1	1	1	0	r = 18
1	2	0	0	1	0	0	0	0	4	0	1	0	0	0	0	0	0	6	0	s = 19
0	1	0	1	1	0	0	2	0	0	0	0	0	1	0	0	1	0	1	7	t = 20

Figure B.20: Confusion matrix of WPD second level for safari browser

(c) WPD 3^{rd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.800	0.035	0.545	0.800	0.649	0.640	0.977	0.579	1
	0.733	0.025	0.611	0.733	0.667	0.650	0.943	0.605	2
	0.467	0.025	0.500	0.467	0.483	0.457	0.920	0.399	3
	0.467	0.011	0.700	0.467	0.560	0.554	0.872	0.429	4
	0.800	0.063	0.400	0.800	0.533	0.535	0.898	0.405	5
	0.867	0.053	0.464	0.867	0.605	0.610	0.955	0.428	6
	0.467	0.000	1.000	0.467	0.636	0.674	0.874	0.590	7
	0.733	0.025	0.611	0.733	0.667	0.650	0.941	0.562	8
	0.600	0.000	1.000	0.600	0.750	0.767	0.964	0.782	9
	0.733	0.063	0.379	0.733	0.500	0.494	0.941	0.373	10
	0.533	0.011	0.727	0.533	0.615	0.606	0.856	0.653	11
	0.733	0.039	0.500	0.733	0.595	0.581	0.910	0.418	12
	0.267	0.011	0.571	0.267	0.364	0.370	0.729	0.231	13
	0.867	0.011	0.813	0.867	0.839	0.830	0.985	0.749	14
	0.667	0.007	0.833	0.667	0.741	0.734	0.977	0.702	15
	0.400	0.025	0.462	0.400	0.429	0.402	0.893	0.325	16
	0.600	0.004	0.900	0.600	0.720	0.724	0.908	0.703	17
	0.200	0.004	0.750	0.200	0.316	0.373	0.893	0.575	18
	0.600	0.007	0.818	0.600	0.692	0.688	0.874	0.582	19
	0.400	0.011	0.667	0.400	0.500	0.498	0.935	0.461	20
Weighted Avg.	0.597	0.021	0.663	0.597	0.593	0.592	0.912	0.527	

Figure B.21: Accuracy of WPD third level for safari browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
12	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	a = 1
0	11	0	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	b = 2
0	0	7	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = 3
0	0	0	7	2	1	0	1	0	2	0	0	0	0	0	1	0	0	1	0	d = 4
0	1	0	0	12	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	e = 5
0	1	0	0	1	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = 6
1	0	1	0	0	1	7	0	0	1	0	3	0	0	0	1	0	0	0	0	g = 7
0	0	0	1	1	0	0	11	0	0	0	0	0	0	0	1	0	0	0	1	h = 8
0	0	0	0	1	2	0	1	9	0	0	0	1	0	0	1	0	0	0	0	i = 9
0	0	0	0	0	0	0	0	0	11	0	4	0	0	0	0	0	0	0	0	j = 10
0	0	0	0	1	1	0	0	0	1	8	0	1	0	1	1	0	1	0	0	k = 11
0	3	0	0	0	0	0	0	0	0	0	11	0	0	0	1	0	0	0	0	l = 12
1	0	1	0	0	0	0	0	0	6	1	0	4	0	1	1	0	0	0	0	m = 13
0	0	0	0	2	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	n = 14
4	0	0	0	0	0	0	0	0	1	0	0	0	0	10	0	0	0	0	0	o = 15
0	0	0	1	3	0	0	2	0	0	0	0	0	2	0	6	0	0	0	1	p = 16
0	0	0	1	3	0	0	1	0	1	0	0	0	0	0	0	9	0	0	0	q = 17
2	0	4	0	1	0	0	0	0	1	2	0	1	1	0	0	0	3	0	0	r = 18
2	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	9	0	s = 19
0	1	1	0	3	0	0	2	0	0	0	0	0	0	0	0	1	0	1	6	t = 20

Figure B.22: Confusion matrix of WPD third level for safari browser

(d) WPD 4^{th} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.867	0.028	0.619	0.867	0.722	0.716	0.972	0.700	1
	0.733	0.039	0.500	0.733	0.595	0.581	0.949	0.474	2
	0.600	0.014	0.692	0.600	0.643	0.627	0.951	0.522	3
	0.400	0.004	0.857	0.400	0.545	0.572	0.931	0.505	4
	0.667	0.077	0.313	0.667	0.426	0.416	0.840	0.298	5
	0.800	0.056	0.429	0.800	0.558	0.557	0.892	0.351	6
	0.533	0.000	1.000	0.533	0.696	0.721	0.905	0.627	7
	0.533	0.018	0.615	0.533	0.571	0.552	0.940	0.460	8
	0.733	0.004	0.917	0.733	0.815	0.812	0.934	0.751	9
	0.667	0.021	0.625	0.667	0.645	0.626	0.973	0.551	10
	0.400	0.014	0.600	0.400	0.480	0.469	0.847	0.466	11
	0.733	0.060	0.393	0.733	0.512	0.505	0.910	0.338	12
	0.200	0.018	0.375	0.200	0.261	0.247	0.798	0.243	13
	0.733	0.018	0.688	0.733	0.710	0.694	0.974	0.558	14
	0.533	0.000	1.000	0.533	0.696	0.721	0.979	0.851	15
	0.533	0.014	0.667	0.533	0.593	0.578	0.870	0.455	16
	0.667	0.025	0.588	0.667	0.625	0.605	0.914	0.516	17
	0.400	0.007	0.750	0.400	0.522	0.532	0.949	0.534	18
	0.600	0.011	0.750	0.600	0.667	0.656	0.905	0.591	19
	0.533	0.004	0.889	0.533	0.667	0.677	0.953	0.688	20
Weighted Avg.	0.593	0.021	0.663	0.593	0.597	0.593	0.919	0.524	

Figure B.23: Accuracy of WPD fourth level for safari browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
13	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	a = 1
0	11	0	0	0	1	0	0	0	0	0	3	0	0	0	0	0	0	0	0	b = 2
0	0	9	0	0	5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	c = 3
0	1	0	6	1	3	0	0	0	1	0	0	0	0	0	2	0	0	1	0	d = 4
0	1	0	0	10	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	0	12	0	1	0	0	0	2	0	0	0	0	0	0	0	0	f = 6
1	0	0	0	1	1	8	0	0	1	0	3	0	0	0	0	0	0	0	0	g = 7
0	2	0	0	4	0	0	8	0	0	0	1	0	0	0	0	0	0	0	0	h = 8
0	0	0	0	2	0	0	1	11	0	1	0	0	0	0	0	0	0	0	0	i = 9
0	0	0	0	0	0	0	0	0	10	0	5	0	0	0	0	0	0	0	0	j = 10
0	1	1	0	1	1	0	0	0	0	6	0	1	2	0	0	1	1	0	0	k = 11
0	3	0	0	0	0	0	0	0	0	0	11	0	0	0	1	0	0	0	0	l = 12
0	0	1	0	1	1	0	0	0	2	1	0	3	1	0	1	3	0	1	0	m = 13
0	0	0	0	2	1	0	0	0	1	0	0	0	11	0	0	0	0	0	0	n = 14
3	2	0	0	0	0	0	0	0	0	0	1	1	0	8	0	0	0	0	0	o = 15
0	0	0	0	3	0	0	0	0	0	0	0	1	2	0	8	0	0	0	1	p = 16
1	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	10	0	0	0	q = 17
2	0	1	0	2	0	0	0	1	0	2	0	1	0	0	0	0	6	0	0	r = 18
1	1	0	0	2	0	0	0	0	1	0	0	0	0	0	0	1	0	9	0	s = 19
0	0	0	0	3	0	0	1	0	0	0	0	0	0	0	0	1	1	1	8	t = 20

Figure B.24: Confusion matrix of WPD fourth level for safari browser

4. IE browser

(a) WPD 1^{st} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.200	0.014	0.429	0.200	0.273	0.268	0.920	0.317	1
	0.800	0.014	0.750	0.800	0.774	0.762	0.975	0.712	2
	0.200	0.004	0.750	0.200	0.316	0.373	0.915	0.562	3
	0.000	0.000	0.000	0.000	0.000	0.000	0.882	0.453	4
	0.867	0.098	0.317	0.867	0.464	0.488	0.846	0.296	5
	1.000	0.042	0.556	1.000	0.714	0.729	0.979	0.556	6
	0.733	0.000	1.000	0.733	0.846	0.850	0.992	0.905	7
	0.867	0.004	0.929	0.867	0.897	0.892	0.992	0.881	8
	0.200	0.004	0.750	0.200	0.316	0.373	0.928	0.520	9
	0.933	0.175	0.219	0.933	0.354	0.403	0.869	0.215	10
	0.533	0.000	1.000	0.533	0.696	0.721	0.992	0.913	11
	0.867	0.018	0.722	0.867	0.788	0.779	0.978	0.661	12
	0.067	0.000	1.000	0.067	0.125	0.252	0.826	0.314	13
	0.667	0.021	0.625	0.667	0.645	0.626	0.960	0.526	14
	0.733	0.004	0.917	0.733	0.815	0.812	0.987	0.820	15
	1.000	0.021	0.714	1.000	0.833	0.836	0.989	0.714	16
	0.200	0.007	0.600	0.200	0.300	0.329	0.843	0.329	17
	0.000	0.000	0.000	0.000	0.000	0.000	0.946	0.823	18
	0.867	0.011	0.813	0.867	0.839	0.830	0.990	0.763	19
	0.733	0.014	0.733	0.733	0.733	0.719	0.926	0.585	20
Weighted Avg.	0.573	0.022	0.641	0.573	0.536	0.552	0.937	0.593	

Figure B.25: Accuracy of WPD first level for IE browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
3	1	0	0	2	0	0	0	0	6	0	0	0	0	1	0	0	0	2	0	a = 1
0	12	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	3	0	1	3	0	0	0	7	0	0	0	0	0	0	0	0	1	0	c = 3
1	1	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	d = 4
0	0	0	0	13	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	e = 5
0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	0	1	11	0	1	0	0	0	0	0	0	2	0	0	0	0	g = 7
0	0	1	0	0	1	0	13	0	0	0	0	0	0	0	0	0	0	0	0	h = 8
0	2	0	0	0	5	0	0	3	0	0	4	0	0	0	1	0	0	0	0	i = 9
0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	1	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	0	8	0	0	5	0	1	1	0	0	0	k = 11
0	0	0	0	0	0	0	0	0	2	0	13	0	0	0	0	0	0	0	0	l = 12
0	0	0	0	5	0	0	0	0	8	0	0	1	1	0	0	0	0	0	0	m = 13
0	0	0	0	3	0	0	0	0	0	0	0	0	10	0	0	0	0	0	2	n = 14
0	0	0	0	0	0	0	0	0	4	0	0	0	0	11	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	p = 16
1	0	0	0	4	2	0	1	0	3	0	0	0	0	0	1	3	0	0	0	q = 17
0	0	0	0	12	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	r = 18
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	s = 19
0	0	0	0	1	0	0	0	0	2	0	1	0	0	0	0	0	0	0	11	t = 20

Figure B.26: Confusion matrix of WPD first level for IE browser

(b) WPD 2^{nd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.200	0.018	0.375	0.200	0.261	0.247	0.928	0.326	1
	0.867	0.014	0.765	0.867	0.813	0.804	0.973	0.693	2
	0.467	0.007	0.778	0.467	0.583	0.587	0.927	0.710	3
	0.000	0.000	0.000	0.000	0.000	0.000	0.817	0.325	4
	0.867	0.056	0.448	0.867	0.591	0.598	0.864	0.410	5
	0.867	0.032	0.591	0.867	0.703	0.698	0.976	0.572	6
	0.867	0.007	0.867	0.867	0.867	0.860	0.993	0.839	7
	0.800	0.007	0.857	0.800	0.828	0.819	0.991	0.869	8
	0.733	0.000	1.000	0.733	0.846	0.850	0.972	0.839	9
	0.933	0.112	0.304	0.933	0.459	0.497	0.892	0.288	10
	0.800	0.004	0.923	0.800	0.857	0.853	0.989	0.854	11
	0.867	0.021	0.684	0.867	0.765	0.757	0.977	0.629	12
	0.200	0.004	0.750	0.200	0.316	0.373	0.961	0.512	13
	0.667	0.011	0.769	0.667	0.714	0.702	0.969	0.644	14
	0.800	0.007	0.857	0.800	0.828	0.819	0.983	0.771	15
	0.933	0.007	0.875	0.933	0.903	0.898	0.995	0.855	16
	0.467	0.007	0.778	0.467	0.583	0.587	0.829	0.434	17
	0.267	0.000	1.000	0.267	0.421	0.507	0.968	0.852	18
	1.000	0.018	0.750	1.000	0.857	0.858	0.991	0.750	19
	0.800	0.018	0.706	0.800	0.750	0.738	0.940	0.595	20
Weighted Avg.	0.670	0.017	0.704	0.670	0.647	0.653	0.947	0.638	

Figure B.27: Accuracy of WPD second level for IE browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
3	1	0	0	2	0	0	0	0	5	0	0	0	0	2	0	0	0	2	0	a = 1
0	13	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	7	0	0	3	1	0	0	3	0	0	0	0	0	0	0	0	1	0	c = 3
2	1	0	0	0	0	0	0	0	10	0	0	0	0	0	0	1	0	0	1	d = 4
0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	e = 5
0	1	0	0	0	13	0	0	0	0	0	1	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	0	1	13	0	0	0	0	0	0	0	0	1	0	0	0	0	g = 7
0	0	2	0	0	1	0	12	0	0	0	0	0	0	0	0	0	0	0	0	h = 8
0	0	0	0	0	1	1	1	11	0	0	1	0	0	0	0	0	0	0	0	i = 9
0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	1	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	1	12	0	1	1	0	0	0	0	0	0	k = 11
0	0	0	0	0	0	0	0	0	2	0	13	0	0	0	0	0	0	0	0	l = 12
0	0	0	0	3	0	0	0	0	6	1	1	3	1	0	0	0	0	0	0	m = 13
0	0	0	0	3	0	0	0	0	0	0	0	0	10	0	0	0	0	0	2	n = 14
2	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	1	0	o = 15
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	14	0	0	0	0	p = 16
1	0	0	0	0	2	0	1	0	3	0	0	0	1	0	0	7	0	0	0	q = 17
0	1	0	0	7	0	0	0	0	0	0	1	0	0	0	0	0	4	0	2	r = 18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	s = 19
0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	12	t = 20

Figure B.28: Confusion matrix of WPD second level for IE browser

(c) WPD 3^{rd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.667	0.018	0.667	0.667	0.667	0.649	0.964	0.584	1
	0.933	0.007	0.875	0.933	0.903	0.898	0.992	0.838	2
	0.600	0.007	0.818	0.600	0.692	0.688	0.919	0.685	3
	0.000	0.004	0.000	0.000	0.000	-0.013	0.835	0.366	4
	0.867	0.021	0.684	0.867	0.765	0.757	0.882	0.601	5
	0.933	0.025	0.667	0.933	0.778	0.776	0.989	0.746	6
	0.933	0.004	0.933	0.933	0.933	0.930	0.997	0.917	7
	0.933	0.011	0.824	0.933	0.875	0.870	0.985	0.945	8
	0.800	0.000	1.000	0.800	0.889	0.890	0.975	0.857	9
	0.867	0.049	0.481	0.867	0.619	0.623	0.931	0.490	10
	0.667	0.011	0.769	0.667	0.714	0.702	0.970	0.760	11
	0.867	0.028	0.619	0.867	0.722	0.716	0.960	0.558	12
	0.533	0.004	0.889	0.533	0.667	0.677	0.978	0.747	13
	0.867	0.007	0.867	0.867	0.867	0.860	0.980	0.787	14
	0.867	0.011	0.813	0.867	0.839	0.830	0.989	0.765	15
	1.000	0.007	0.882	1.000	0.938	0.936	0.996	0.882	16
	0.400	0.025	0.462	0.400	0.429	0.402	0.824	0.320	17
	0.400	0.000	1.000	0.400	0.571	0.623	0.971	0.894	18
	1.000	0.018	0.750	1.000	0.857	0.858	0.991	0.750	19
	0.800	0.014	0.750	0.800	0.774	0.762	0.899	0.623	20
Weighted Avg.	0.747	0.013	0.737	0.747	0.725	0.722	0.951	0.706	

Figure B.29: Accuracy of WPD third level for IE browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
10	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0	0	0	1	0	a = 1
0	14	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	9	0	0	2	1	0	0	0	0	0	0	0	0	0	2	0	1	0	c = 3
2	0	0	0	0	0	0	0	0	6	0	1	0	0	0	0	4	0	2	0	d = 4
0	0	0	0	13	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	e = 5
0	0	0	0	0	14	0	0	0	0	0	1	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	1	0	0	0	0	g = 7
0	0	0	0	0	1	0	14	0	0	0	0	0	0	0	0	0	0	0	0	h = 8
0	0	1	0	0	0	0	0	12	0	0	1	0	0	0	0	1	0	0	0	i = 9
0	0	0	0	0	1	0	0	0	13	0	0	0	0	0	1	0	0	0	0	j = 10
0	1	0	0	0	0	0	0	0	1	10	0	1	2	0	0	0	0	0	0	k = 11
0	0	0	0	0	0	0	0	0	2	0	13	0	0	0	0	0	0	0	0	l = 12
1	0	0	1	1	1	0	0	0	1	1	1	8	0	0	0	0	0	0	0	m = 13
0	0	0	0	1	0	0	1	0	0	0	0	0	13	0	0	0	0	0	0	n = 14
2	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	p = 16
0	0	1	0	0	2	0	1	0	2	1	1	0	0	0	0	6	0	0	1	q = 17
0	1	0	0	3	0	0	0	0	0	1	1	0	0	0	0	0	6	0	3	r = 18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	s = 19
0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	12	t = 20

Figure B.30: Confusion matrix of WPD third level for IE browser

(d) WPD 4^{th} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.733	0.018	0.688	0.733	0.710	0.694	0.980	0.641	1
	1.000	0.007	0.882	1.000	0.938	0.936	0.997	0.905	2
	0.600	0.004	0.900	0.600	0.720	0.724	0.920	0.729	3
	0.267	0.007	0.667	0.267	0.381	0.404	0.816	0.330	4
	0.867	0.014	0.765	0.867	0.813	0.804	0.902	0.714	5
	0.933	0.032	0.609	0.933	0.737	0.739	0.979	0.700	6
	1.000	0.007	0.882	1.000	0.938	0.936	0.996	0.882	7
	0.933	0.000	1.000	0.933	0.966	0.964	0.991	0.949	8
	0.800	0.000	1.000	0.800	0.889	0.890	0.970	0.822	9
	0.733	0.021	0.647	0.733	0.688	0.671	0.931	0.565	10
	0.867	0.004	0.929	0.867	0.897	0.892	0.970	0.869	11
	0.867	0.018	0.722	0.867	0.788	0.779	0.976	0.657	12
	0.667	0.004	0.909	0.667	0.769	0.769	0.990	0.833	13
	0.867	0.007	0.867	0.867	0.867	0.860	0.963	0.771	14
	0.733	0.011	0.786	0.733	0.759	0.747	0.989	0.719	15
	1.000	0.014	0.789	1.000	0.882	0.882	0.993	0.789	16
	0.600	0.021	0.600	0.600	0.600	0.579	0.830	0.449	17
	0.533	0.004	0.889	0.533	0.667	0.677	0.949	0.677	18
	1.000	0.011	0.833	1.000	0.909	0.908	0.995	0.833	19
	0.733	0.025	0.611	0.733	0.667	0.650	0.941	0.534	20
Weighted Avg.	0.787	0.011	0.799	0.787	0.779	0.775	0.954	0.718	

Figure B.31: Accuracy of WPD fourth level for IE browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
11	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	1	0	a = 1
0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	9	0	0	3	1	0	0	0	0	1	0	0	0	0	1	0	0	0	c = 3
2	0	0	4	1	0	0	0	0	3	0	0	0	0	0	0	4	1	0	0	d = 4
0	0	0	0	13	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	e = 5
0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	f = 6
0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	g = 7
0	0	0	0	1	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	h = 8
0	0	0	0	0	1	1	0	12	0	0	1	0	0	0	0	0	0	0	0	i = 9
0	1	0	0	0	1	0	0	0	11	0	0	0	1	0	1	0	0	0	0	j = 10
0	0	0	0	0	0	0	0	0	0	13	0	1	0	0	1	0	0	0	0	k = 11
0	0	0	0	0	1	0	0	0	1	0	13	0	0	0	0	0	0	0	0	l = 12
0	0	0	1	0	1	0	0	0	1	0	1	10	1	0	0	0	0	0	0	m = 13
0	0	0	0	1	0	0	0	0	0	0	0	0	13	0	0	0	0	0	1	n = 14
3	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	1	0	o = 15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	p = 16
0	1	1	1	0	1	0	0	0	0	1	0	0	0	0	1	9	0	0	0	q = 17
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	8	0	5	r = 18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	s = 19
0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	0	0	0	11	t = 20

Figure B.32: Confusion matrix of WPD fourth level for IE browser

5. Opera browser

    (a) WPD 1^{st} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.267	0.011	0.571	0.267	0.364	0.370	0.690	0.222	1
	0.267	0.084	0.143	0.267	0.186	0.137	0.695	0.100	2
	0.600	0.028	0.529	0.600	0.563	0.539	0.912	0.467	3
	0.267	0.011	0.571	0.267	0.364	0.370	0.808	0.308	4
	0.600	0.182	0.148	0.600	0.237	0.226	0.785	0.120	5
	0.267	0.060	0.190	0.267	0.222	0.177	0.755	0.130	6
	0.467	0.000	1.000	0.467	0.636	0.674	0.991	0.916	7
	0.267	0.032	0.308	0.267	0.286	0.252	0.910	0.322	8
	0.133	0.000	1.000	0.133	0.235	0.357	0.867	0.623	9
	0.467	0.119	0.171	0.467	0.250	0.220	0.731	0.121	10
	0.133	0.007	0.500	0.133	0.211	0.240	0.878	0.462	11
	0.267	0.035	0.286	0.267	0.276	0.239	0.826	0.197	12
	0.200	0.004	0.750	0.200	0.316	0.373	0.910	0.488	13
	0.467	0.018	0.583	0.467	0.519	0.500	0.931	0.465	14
	0.133	0.007	0.500	0.133	0.211	0.240	0.696	0.191	15
	0.467	0.032	0.438	0.467	0.452	0.422	0.967	0.443	16
	0.133	0.018	0.286	0.133	0.182	0.167	0.645	0.100	17
	0.200	0.000	1.000	0.200	0.333	0.438	0.835	0.430	18
	0.133	0.077	0.083	0.133	0.103	0.045	0.605	0.072	19
	0.333	0.011	0.625	0.333	0.435	0.437	0.914	0.427	20
Weighted Avg.	0.303	0.037	0.484	0.303	0.319	0.321	0.818	0.330	

Figure B.33: Accuracy of WPD first level for opera browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t		<-- classified as
4	0	0	0	3	1	0	0	0	2	0	1	0	0	1	0	1	0	2	0	|	a = 1
0	4	1	0	1	4	0	0	0	3	0	0	0	0	0	0	1	0	1	0	|	b = 2
1	1	9	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	|	c = 3
0	1	0	4	5	0	0	1	0	1	0	0	0	2	0	0	0	0	1	0	|	d = 4
0	0	0	0	9	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	|	e = 5
0	5	0	0	3	4	0	0	0	1	0	0	0	0	0	0	0	0	2	0	|	f = 6
0	4	1	0	1	0	7	0	0	0	0	0	1	0	0	0	0	0	1	0	|	g = 7
0	0	0	1	2	0	0	4	0	2	0	0	0	0	0	6	0	0	0	0	|	h = 8
2	0	1	0	1	3	0	1	2	3	0	0	0	0	0	0	0	0	2	0	|	i = 9
0	0	1	0	5	2	0	0	0	7	0	0	0	0	0	0	0	0	0	0	|	j = 10
0	0	0	2	5	2	0	0	0	1	2	0	0	0	0	2	0	0	1	0	|	k = 11
0	4	0	0	2	0	0	0	0	1	0	4	0	0	0	0	0	0	2	2	|	l = 12
0	4	0	0	2	2	0	0	0	1	1	0	3	0	0	0	1	0	1	0	|	m = 13
0	1	0	0	3	0	0	0	0	1	0	0	0	7	0	0	0	0	3	0	|	n = 14
0	3	0	0	2	0	0	0	0	3	0	0	0	0	2	0	2	0	3	0	|	o = 15
0	0	0	0	0	0	0	7	0	0	1	0	0	0	0	7	0	0	0	0	|	p = 16
0	0	2	0	5	1	0	0	0	2	0	0	0	1	0	1	2	0	0	1	|	q = 17
0	0	1	0	3	0	0	0	0	3	0	3	0	2	0	0	0	3	0	0	|	r = 18
0	0	1	0	5	2	0	0	0	3	0	1	0	0	1	0	0	0	2	0	|	s = 19
0	1	0	0	1	0	0	0	0	0	0	5	0	0	0	0	0	0	3	5	|	t = 20

Figure B.34: Confusion matrix of WPD first level for opera browser

(b) WPD 2^{nd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.267	0.011	0.571	0.267	0.364	0.370	0.717	0.265	1
	0.267	0.074	0.160	0.267	0.200	0.152	0.790	0.138	2
	0.733	0.018	0.688	0.733	0.710	0.694	0.925	0.654	3
	0.333	0.021	0.455	0.333	0.385	0.362	0.833	0.294	4
	0.600	0.098	0.243	0.600	0.346	0.333	0.830	0.192	5
	0.200	0.046	0.188	0.200	0.194	0.150	0.728	0.132	6
	0.667	0.000	1.000	0.667	0.800	0.809	0.989	0.940	7
	0.333	0.035	0.333	0.333	0.333	0.298	0.919	0.308	8
	0.467	0.000	1.000	0.467	0.636	0.674	0.927	0.711	9
	0.400	0.095	0.182	0.400	0.250	0.213	0.789	0.144	10
	0.467	0.004	0.875	0.467	0.609	0.627	0.870	0.598	11
	0.333	0.039	0.313	0.333	0.323	0.286	0.828	0.262	12
	0.533	0.007	0.800	0.533	0.640	0.639	0.963	0.608	13
	0.667	0.028	0.556	0.667	0.606	0.586	0.907	0.452	14
	0.200	0.028	0.273	0.200	0.231	0.199	0.712	0.131	15
	0.333	0.032	0.357	0.333	0.345	0.312	0.962	0.409	16
	0.200	0.025	0.300	0.200	0.240	0.213	0.738	0.164	17
	0.333	0.007	0.714	0.333	0.455	0.471	0.840	0.409	18
	0.133	0.053	0.118	0.133	0.125	0.076	0.600	0.077	19
	0.400	0.021	0.500	0.400	0.444	0.421	0.893	0.364	20
Weighted Avg.	0.393	0.032	0.481	0.393	0.412	0.394	0.838	0.363	

Figure B.35: Accuracy of WPD second level for opera browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
4	1	1	0	2	1	0	0	0	1	0	2	0	0	1	0	0	0	1	1	a = 1
0	4	0	0	0	3	0	0	0	5	0	0	0	0	1	0	0	0	2	0	b = 2
0	0	11	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	c = 3
0	1	0	5	3	1	0	1	0	1	0	1	0	1	0	0	0	0	1	0	d = 4
0	0	0	0	9	0	0	0	0	5	0	0	0	1	0	0	0	0	0	0	e = 5
0	4	0	0	1	3	0	0	0	2	0	0	0	1	0	0	0	0	4	0	f = 6
0	2	0	0	0	0	10	0	0	0	0	0	1	0	1	0	0	0	0	1	g = 7
0	0	0	2	0	0	0	5	0	1	0	0	0	0	0	7	0	0	0	0	h = 8
1	0	1	0	1	0	0	1	7	1	0	0	0	0	0	0	2	1	0	0	i = 9
0	2	1	0	5	1	0	0	0	6	0	0	0	0	0	0	0	0	0	0	j = 10
0	0	0	2	0	1	0	0	0	1	7	1	1	0	0	1	0	1	0	0	k = 11
0	2	0	0	0	0	0	0	0	4	0	5	0	0	0	0	0	0	1	3	l = 12
1	3	1	0	0	1	0	0	0	1	0	0	8	0	0	0	0	0	0	0	m = 13
0	1	0	0	3	0	0	0	0	1	0	0	0	10	0	0	0	0	0	0	n = 14
0	5	0	0	0	0	0	0	0	1	0	1	0	0	3	0	3	0	2	0	o = 15
0	0	0	1	0	0	0	8	0	0	1	0	0	0	0	5	0	0	0	0	p = 16
1	0	1	0	3	2	0	0	0	0	0	0	0	2	1	1	3	0	1	0	q = 17
0	0	0	1	3	0	0	0	0	2	0	1	0	3	0	0	0	5	0	0	r = 18
0	0	0	0	4	3	0	0	0	1	0	2	0	0	2	0	0	0	2	1	s = 19
0	0	0	0	2	0	0	0	0	0	0	3	0	0	2	0	0	0	2	6	t = 20

Figure B.36: Confusion matrix of WPD second level for opera browser

(c) WPD 3^{rd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.333	0.039	0.313	0.333	0.323	0.286	0.831	0.248	1
	0.267	0.060	0.190	0.267	0.222	0.177	0.696	0.113	2
	0.933	0.021	0.700	0.933	0.800	0.797	0.944	0.692	3
	0.400	0.018	0.545	0.400	0.462	0.444	0.765	0.299	4
	0.667	0.077	0.313	0.667	0.426	0.416	0.845	0.254	5
	0.533	0.025	0.533	0.533	0.533	0.509	0.844	0.390	6
	0.867	0.000	1.000	0.867	0.929	0.928	0.994	0.952	7
	0.200	0.025	0.300	0.200	0.240	0.213	0.929	0.339	8
	0.800	0.000	1.000	0.800	0.889	0.890	0.952	0.851	9
	0.200	0.049	0.176	0.200	0.188	0.142	0.796	0.140	10
	0.467	0.007	0.778	0.467	0.583	0.587	0.873	0.500	11
	0.533	0.046	0.381	0.533	0.444	0.417	0.859	0.347	12
	0.533	0.007	0.800	0.533	0.640	0.639	0.939	0.558	13
	0.400	0.014	0.600	0.400	0.480	0.469	0.901	0.464	14
	0.267	0.032	0.308	0.267	0.286	0.252	0.778	0.192	15
	0.533	0.042	0.400	0.533	0.457	0.429	0.963	0.413	16
	0.267	0.025	0.364	0.267	0.308	0.281	0.784	0.214	17
	0.333	0.007	0.714	0.333	0.455	0.471	0.840	0.390	18
	0.333	0.049	0.263	0.333	0.294	0.254	0.712	0.151	19
	0.467	0.021	0.538	0.467	0.500	0.477	0.926	0.397	20
Weighted Avg.	0.467	0.028	0.511	0.467	0.473	0.454	0.859	0.395	

Figure B.37: Accuracy of WPD third level for opera browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
5	0	1	0	0	0	0	0	0	0	0	1	0	0	4	0	1	0	2	1	a = 1
0	4	0	0	1	3	0	0	0	3	0	1	0	1	1	0	0	0	1	0	b = 2
0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	c = 3
0	1	0	6	2	0	0	0	0	0	1	2	0	0	0	1	0	0	2	0	d = 4
0	1	0	0	10	0	0	0	0	2	0	0	0	1	0	0	1	0	0	0	e = 5
0	4	0	0	1	8	0	0	0	1	0	0	0	0	0	0	0	0	1	0	f = 6
0	1	0	0	0	0	13	0	0	0	0	0	1	0	0	0	0	0	0	0	g = 7
0	0	0	2	1	0	0	3	0	0	0	0	0	0	0	9	0	0	0	0	h = 8
2	0	1	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	i = 9
0	2	0	0	5	3	0	0	0	3	0	1	0	0	0	0	0	1	0	0	j = 10
0	0	0	2	1	0	0	1	0	1	7	0	1	0	0	2	0	0	0	0	k = 11
0	0	0	0	2	0	0	0	0	2	0	8	0	0	0	0	0	0	1	2	l = 12
0	3	3	0	0	0	0	0	0	0	0	0	8	0	1	0	0	0	0	0	m = 13
0	1	0	0	5	0	0	0	0	2	0	0	0	6	0	0	0	1	0	0	n = 14
2	4	0	0	0	0	0	0	0	0	0	1	0	0	4	0	1	0	3	0	o = 15
0	0	0	1	0	0	0	6	0	0	0	0	0	0	0	8	0	0	0	0	p = 16
4	0	1	0	1	0	0	0	0	0	1	0	0	1	1	0	4	0	1	1	q = 17
0	0	0	0	2	0	0	0	0	1	0	4	0	1	0	0	1	5	0	1	r = 18
2	0	0	0	1	1	0	0	0	2	0	0	0	0	2	0	1	0	5	1	s = 19
1	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	3	7	t = 20

Figure B.38: Confusion matrix of WPD third level for opera browser

(d) WPD 4^{th} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.067	0.053	0.063	0.067	0.065	0.014	0.824	0.169	1
	0.267	0.056	0.200	0.267	0.229	0.184	0.754	0.124	2
	0.933	0.007	0.875	0.933	0.903	0.898	0.940	0.860	3
	0.533	0.007	0.800	0.533	0.640	0.639	0.836	0.535	4
	0.667	0.067	0.345	0.667	0.455	0.443	0.881	0.300	5
	0.533	0.042	0.400	0.533	0.457	0.429	0.865	0.330	6
	0.867	0.000	1.000	0.867	0.929	0.928	0.999	0.977	7
	0.467	0.028	0.467	0.467	0.467	0.439	0.955	0.428	8
	0.733	0.004	0.917	0.733	0.815	0.812	0.927	0.835	9
	0.400	0.039	0.353	0.400	0.375	0.341	0.816	0.187	10
	0.333	0.014	0.556	0.333	0.417	0.408	0.896	0.428	11
	0.467	0.021	0.538	0.467	0.500	0.477	0.874	0.444	12
	0.600	0.004	0.900	0.600	0.720	0.724	0.969	0.684	13
	0.867	0.021	0.684	0.867	0.765	0.757	0.927	0.606	14
	0.267	0.035	0.286	0.267	0.276	0.239	0.865	0.213	15
	0.467	0.039	0.389	0.467	0.424	0.393	0.963	0.413	16
	0.400	0.032	0.400	0.400	0.400	0.368	0.822	0.271	17
	0.267	0.007	0.667	0.267	0.381	0.404	0.820	0.322	18
	0.400	0.042	0.333	0.400	0.364	0.328	0.803	0.186	19
	0.400	0.014	0.600	0.400	0.480	0.469	0.904	0.443	20
Weighted Avg.	0.497	0.026	0.539	0.497	0.503	0.485	0.882	0.438	

Figure B.39: Accuracy of WPD fourth level for opera browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
1	1	0	0	2	0	0	0	0	0	0	1	0	0	5	0	3	0	1	1	a = 1
0	4	0	0	1	5	0	0	0	1	0	1	0	1	0	0	0	0	2	0	b = 2
0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	c = 3
0	0	0	8	1	2	0	0	0	2	1	0	0	0	0	1	0	0	0	0	d = 4
0	0	0	0	10	0	0	0	0	3	0	0	0	2	0	0	0	0	0	0	e = 5
0	4	0	0	0	8	0	0	0	0	0	0	0	1	1	0	0	0	1	0	f = 6
0	2	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	g = 7
0	0	0	0	0	0	0	7	1	0	0	0	0	0	0	7	0	0	0	0	h = 8
2	0	0	0	1	0	0	0	11	1	0	0	0	0	0	0	0	0	0	0	i = 9
0	2	0	0	4	2	0	0	0	6	0	0	0	0	0	0	1	0	0	0	j = 10
0	0	0	2	2	0	0	1	0	0	5	0	1	0	1	3	0	0	0	0	k = 11
1	2	0	0	1	0	0	0	0	0	0	7	0	0	0	0	0	0	3	1	l = 12
0	1	1	0	0	0	0	0	0	0	1	0	9	0	0	0	2	0	0	1	m = 13
0	1	0	0	1	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	n = 14
5	3	0	0	0	0	0	0	0	0	0	0	0	0	4	0	1	0	2	0	o = 15
0	0	0	0	0	0	0	7	0	0	1	0	0	0	0	7	0	0	0	0	p = 16
1	0	1	0	1	0	0	0	0	0	1	0	0	1	2	0	6	1	0	1	q = 17
0	0	0	0	5	0	0	0	0	2	0	2	0	1	0	0	1	4	0	0	r = 18
2	0	0	0	0	2	0	0	0	2	0	1	0	0	1	0	1	0	6	0	s = 19
4	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	3	6	t = 20

Figure B.40: Confusion matrix of WPD fourth level for opera browser

6. Tor browser

(a) WPD 1^{st} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.800	0.021	0.667	0.800	0.727	0.715	0.982	0.624	1
	0.667	0.007	0.833	0.667	0.741	0.734	0.976	0.676	2
	0.267	0.011	0.571	0.267	0.364	0.370	0.881	0.410	3
	0.067	0.000	1.000	0.067	0.125	0.252	0.753	0.251	4
	0.733	0.158	0.196	0.733	0.310	0.322	0.866	0.185	5
	0.333	0.032	0.357	0.333	0.345	0.312	0.827	0.233	6
	0.267	0.000	1.000	0.267	0.421	0.507	0.939	0.697	7
	0.600	0.025	0.563	0.600	0.581	0.558	0.908	0.444	8
	0.200	0.000	1.000	0.200	0.333	0.438	0.851	0.665	9
	0.533	0.260	0.098	0.533	0.165	0.134	0.694	0.086	10
	0.400	0.000	1.000	0.400	0.571	0.623	0.959	0.924	11
	0.400	0.021	0.500	0.400	0.444	0.421	0.917	0.354	12
	0.133	0.000	1.000	0.133	0.235	0.357	0.906	0.727	13
	0.067	0.014	0.200	0.067	0.100	0.090	0.799	0.156	14
	0.467	0.018	0.583	0.467	0.519	0.500	0.931	0.442	15
	1.000	0.035	0.600	1.000	0.750	0.761	0.984	0.625	16
	0.133	0.014	0.333	0.133	0.190	0.186	0.629	0.134	17
	0.000	0.000	0.000	0.000	0.000	0.000	0.624	0.225	18
	0.267	0.025	0.364	0.267	0.308	0.281	0.825	0.208	19
	0.333	0.011	0.625	0.333	0.435	0.437	0.890	0.358	20
Weighted Avg.	0.383	0.032	0.575	0.383	0.383	0.400	0.857	0.421	

Figure B.41: Accuracy of WPD first level for tor browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
12	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	a = 1
1	10	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	b = 2
1	0	4	0	2	0	0	0	0	5	0	0	0	0	3	0	0	0	0	0	c = 3
0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	5	1	0	1	3	d = 4
0	0	0	0	11	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	e = 5
0	0	0	0	1	5	0	0	0	7	0	2	0	0	0	0	0	0	0	0	f = 6
0	0	1	0	7	0	4	0	0	1	0	0	0	0	0	0	2	0	0	0	g = 7
0	0	1	0	2	0	0	9	0	1	0	0	0	0	0	1	0	0	1	0	h = 8
0	0	1	0	0	0	0	3	3	7	0	0	0	0	0	0	0	0	1	0	i = 9
0	0	0	0	5	2	0	0	0	8	0	0	0	0	0	0	0	0	0	0	j = 10
1	0	0	0	1	1	0	0	0	3	6	0	0	1	0	0	0	0	2	0	k = 11
0	0	0	0	2	0	0	0	0	7	0	6	0	0	0	0	0	0	0	0	l = 12
1	1	0	0	3	0	0	0	0	6	0	0	2	1	0	0	1	0	0	0	m = 13
0	0	0	0	4	0	0	1	0	8	0	0	0	1	0	1	0	0	0	0	n = 14
1	0	0	0	2	0	0	0	0	5	0	0	0	0	7	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	p = 16
0	0	0	0	4	0	0	2	0	4	0	0	0	0	1	1	2	0	1	0	q = 17
1	0	0	0	6	5	0	0	0	2	0	0	0	1	0	0	0	0	0	0	r = 18
0	0	0	0	1	0	0	0	0	6	0	1	0	1	0	2	0	0	4	0	s = 19
0	1	0	0	5	0	0	0	0	1	0	2	0	0	0	0	0	0	1	5	t = 20

Figure B.42: Confusion matrix of WPD first level for tor browser

(b) WPD 2^{nd} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.867	0.011	0.813	0.867	0.839	0.830	0.993	0.788	1
	0.800	0.032	0.571	0.800	0.667	0.656	0.949	0.573	2
	0.400	0.007	0.750	0.400	0.522	0.532	0.883	0.500	3
	0.200	0.000	1.000	0.200	0.333	0.438	0.866	0.399	4
	0.800	0.137	0.235	0.800	0.364	0.385	0.875	0.237	5
	0.267	0.046	0.235	0.267	0.250	0.208	0.844	0.190	6
	0.533	0.004	0.889	0.533	0.667	0.677	0.942	0.691	7
	0.600	0.021	0.600	0.600	0.600	0.579	0.920	0.486	8
	0.400	0.000	1.000	0.400	0.571	0.623	0.873	0.677	9
	0.467	0.137	0.152	0.467	0.230	0.200	0.725	0.117	10
	0.600	0.000	1.000	0.600	0.750	0.767	0.942	0.932	11
	0.467	0.042	0.368	0.467	0.412	0.380	0.904	0.361	12
	0.267	0.000	1.000	0.267	0.421	0.507	0.920	0.744	13
	0.067	0.011	0.250	0.067	0.105	0.107	0.788	0.159	14
	0.667	0.018	0.667	0.667	0.667	0.649	0.950	0.556	15
	1.000	0.042	0.556	1.000	0.714	0.729	0.979	0.556	16
	0.333	0.007	0.714	0.333	0.455	0.471	0.801	0.370	17
	0.000	0.004	0.000	0.000	0.000	-0.013	0.754	0.227	18
	0.200	0.025	0.300	0.200	0.240	0.213	0.911	0.272	19
	0.400	0.021	0.500	0.400	0.444	0.421	0.923	0.350	20
Weighted Avg.	0.467	0.028	0.580	0.467	0.462	0.468	0.887	0.459	

Figure B.43: Accuracy of WPD second level for tor browser

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t   <-- classified as
 13  1  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0 |  a = 1
  0 12  0  0  0  0  0  0  0  2  0  1  0  0  0  0  0  0  0  0 |  b = 2
  0  0  6  0  1  1  0  0  0  4  0  0  0  0  3  0  0  0  0  0 |  c = 3
  0  1  1  3  0  1  0  0  0  0  0  2  0  0  0  4  1  0  1  1 |  d = 4
  0  0  0  0 12  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0 |  e = 5
  0  1  0  0  0  4  0  0  0  6  0  3  0  0  0  1  0  0  0  0 |  f = 6
  0  0  0  0  6  0  8  0  0  0  0  0  0  0  0  0  0  1  0  0 |  g = 7
  0  0  0  0  1  0  0  9  0  2  0  1  0  0  0  2  0  0  0  0 |  h = 8
  0  0  1  0  0  0  0  2  6  3  0  0  0  0  0  0  0  0  1  2 |  i = 9
  0  0  0  0  3  3  0  1  0  7  0  0  0  0  0  0  0  0  0  1 |  j = 10
  1  0  0  0  0  1  0  0  0  1  9  1  0  0  0  0  0  0  2  0 |  k = 11
  0  2  0  0  1  1  0  0  0  2  0  7  0  0  0  2  0  0  0  0 |  l = 12
  1  1  0  0  5  0  0  0  0  2  0  0  4  1  0  0  1  0  0  0 |  m = 13
  0  0  0  0  7  1  0  0  0  5  0  0  0  1  0  1  0  0  0  0 |  n = 14
  0  1  0  0  0  0  0  0  0  4  0  0  0  0 10  0  0  0  0  0 |  o = 15
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 15  0  0  0  0 |  p = 16
  1  0  0  0  3  0  0  2  0  1  0  0  0  0  1  1  5  0  1  0 |  q = 17
  0  0  0  0  6  5  1  0  0  0  0  2  0  1  0  0  0  0  0  0 |  r = 18
  0  0  0  0  2  0  0  1  0  3  0  2  0  1  0  1  0  0  3  2 |  s = 19
  0  2  0  0  4  0  0  0  0  1  0  0  0  0  0  0  0  0  2  6 |  t = 20
```

Figure B.44: Confusion matrix of WPD second level for tor browser

(c) WPD 3^{rd} level

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.933    0.004    0.933      0.933    0.933      0.930    0.992     0.892     1
                 0.867    0.011    0.813      0.867    0.839      0.830    0.946     0.766     2
                 0.600    0.004    0.900      0.600    0.720      0.724    0.945     0.713     3
                 0.467    0.014    0.636      0.467    0.538      0.525    0.923     0.446     4
                 0.800    0.084    0.333      0.800    0.471      0.480    0.922     0.330     5
                 0.667    0.046    0.435      0.667    0.526      0.509    0.802     0.365     6
                 0.733    0.007    0.846      0.733    0.786      0.777    0.978     0.744     7
                 0.867    0.014    0.765      0.867    0.813      0.804    0.979     0.693     8
                 0.667    0.004    0.909      0.667    0.769      0.769    0.932     0.780     9
                 0.333    0.060    0.227      0.333    0.270      0.229    0.769     0.168     10
                 0.600    0.000    1.000      0.600    0.750      0.767    0.938     0.881     11
                 0.800    0.025    0.632      0.800    0.706      0.694    0.959     0.616     12
                 0.533    0.004    0.889      0.533    0.667      0.677    0.968     0.731     13
                 0.133    0.014    0.333      0.133    0.190      0.186    0.810     0.269     14
                 0.800    0.021    0.667      0.800    0.727      0.715    0.958     0.615     15
                 1.000    0.014    0.789      1.000    0.882      0.882    0.995     0.833     16
                 0.400    0.028    0.429      0.400    0.414      0.384    0.793     0.326     17
                 0.133    0.004    0.667      0.133    0.222      0.284    0.765     0.245     18
                 0.467    0.046    0.350      0.467    0.400      0.368    0.889     0.287     19
                 0.533    0.004    0.889      0.533    0.667      0.677    0.964     0.676     20
Weighted Avg.    0.617    0.020    0.672      0.617    0.615      0.611    0.911     0.569
```

Figure B.45: Accuracy of WPD third level for tor browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	a = 1
0	13	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	b = 2
0	0	9	0	0	0	0	0	0	1	0	0	0	0	1	0	2	0	2	0	c = 3
0	0	0	7	0	1	0	0	0	1	0	1	0	0	0	1	2	0	2	0	d = 4
0	0	0	0	12	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	e = 5
0	1	0	0	0	10	0	0	0	1	0	0	0	0	1	1	0	0	1	0	f = 6
0	0	0	0	3	0	11	0	1	0	0	0	0	0	0	0	0	0	0	0	g = 7
0	0	0	0	0	1	0	13	0	0	0	0	0	0	0	0	0	0	1	0	h = 8
0	0	0	0	0	0	0	1	10	2	0	0	0	0	0	0	0	0	2	0	i = 9
0	1	0	0	4	3	0	2	0	5	0	0	0	0	0	0	0	0	0	0	j = 10
1	0	0	0	0	0	0	0	0	2	9	1	1	0	0	0	0	0	1	0	k = 11
0	0	0	0	0	1	0	0	0	1	0	12	0	0	0	0	0	0	1	0	l = 12
0	0	0	0	2	0	0	0	0	0	0	0	8	3	0	0	1	1	0	0	m = 13
0	0	0	0	6	2	0	0	0	4	0	0	0	2	0	0	1	0	0	0	n = 14
0	1	0	0	0	0	0	0	0	2	0	0	0	0	12	0	0	0	0	0	o = 15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	p = 16
0	0	0	4	1	1	0	1	0	0	0	0	0	0	2	0	6	0	0	0	q = 17
0	0	0	0	5	4	2	0	0	1	0	0	0	1	0	0	0	2	0	0	r = 18
0	0	1	0	0	0	0	0	0	0	0	4	0	0	0	2	0	0	7	1	s = 19
0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	0	2	0	2	8	t = 20

Figure B.46: Confusion matrix of WPD third level for tor browser

(d) WPD 4^{th} level

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.933	0.007	0.875	0.933	0.903	0.898	0.993	0.892	1
	0.933	0.007	0.875	0.933	0.903	0.898	0.986	0.885	2
	0.733	0.018	0.688	0.733	0.710	0.694	0.961	0.605	3
	0.467	0.007	0.778	0.467	0.583	0.587	0.890	0.555	4
	0.867	0.053	0.464	0.867	0.605	0.610	0.948	0.444	5
	0.867	0.039	0.542	0.867	0.667	0.665	0.882	0.520	6
	0.800	0.007	0.857	0.800	0.828	0.819	0.984	0.789	7
	0.800	0.018	0.706	0.800	0.750	0.738	0.962	0.653	8
	0.867	0.004	0.929	0.867	0.897	0.892	0.972	0.838	9
	0.200	0.021	0.333	0.200	0.250	0.229	0.833	0.204	10
	0.800	0.000	1.000	0.800	0.889	0.890	0.939	0.912	11
	0.667	0.025	0.588	0.667	0.625	0.605	0.916	0.533	12
	0.600	0.000	1.000	0.600	0.750	0.767	0.947	0.742	13
	0.067	0.014	0.200	0.067	0.100	0.090	0.797	0.138	14
	0.867	0.025	0.650	0.867	0.743	0.736	0.972	0.599	15
	1.000	0.004	0.938	1.000	0.968	0.967	0.998	0.938	16
	0.467	0.035	0.412	0.467	0.437	0.407	0.846	0.320	17
	0.333	0.007	0.714	0.333	0.455	0.471	0.922	0.514	18
	0.667	0.042	0.455	0.667	0.541	0.522	0.888	0.365	19
	0.667	0.007	0.833	0.667	0.741	0.734	0.974	0.690	20
Weighted Avg.	0.680	0.017	0.692	0.680	0.667	0.661	0.931	0.607	

Figure B.47: Accuracy of WPD fourth level for tor browser

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t   <-- classified as
 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0 |  a = 1
  0 14  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0 |  b = 2
  0  0 11  0  0  0  0  1  0  0  0  0  0  0  1  0  1  0  1  0 |  c = 3
  0  0  1  7  0  1  0  0  0  0  0  1  0  0  0  1  1  0  3  0 |  d = 4
  0  0  0  0 13  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0 |  e = 5
  0  0  0  0  0 13  0  0  0  0  0  0  0  0  1  0  0  0  1  0 |  f = 6
  0  0  1  0  0  0 12  0  1  0  0  0  0  0  0  0  1  0  0  0 |  g = 7
  0  0  1  0  0  2  0 12  0  0  0  0  0  0  0  0  0  0  0  0 |  h = 8
  0  0  0  0  1  0  0  0 13  0  0  0  0  0  0  0  0  0  1  0 |  i = 9
  0  0  0  0  5  3  0  3  0  3  0  0  0  0  1  0  0  0  0  0 |  j = 10
  1  0  0  0  1  0  0  0  0  0 12  0  0  0  0  0  1  0  0  0 |  k = 11
  0  1  0  0  0  2  0  0  0  0  0 10  0  0  0  0  0  0  2  0 |  l = 12
  0  0  0  0  1  0  0  0  0  0  0  0  9  3  1  0  0  1  0  0 |  m = 13
  0  0  0  0  4  2  0  0  0  4  0  0  0  1  1  0  1  1  0  1 |  n = 14
  0  1  0  0  0  0  0  0  0  0  0  0  0  0 13  0  1  0  0  0 |  o = 15
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 15  0  0  0  0 |  p = 16
  0  0  1  2  0  0  0  1  0  0  0  1  0  0  2  0  7  0  0  1 |  q = 17
  1  0  0  0  1  1  2  0  0  0  0  0  0  1  0  0  2  5  2  0 |  r = 18
  0  0  1  0  0  0  0  0  0  0  0  3  0  0  0  0  1  0 10  0 |  s = 19
  0  0  0  0  2  0  0  0  0  0  0  1  0  0  0  0  0  0  2 10 |  t = 20
```

Figure B.48: Confusion matrix of WPD fourth level for tor browser

APPENDIX C : Detailed Performance Metrics For MLP classifier and WPD method

1. Chrome browser

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.467	0.028	0.467	0.467	0.467	0.439	0.928	0.564	1
	0.800	0.018	0.706	0.800	0.750	0.738	0.961	0.810	2
	0.867	0.014	0.765	0.867	0.813	0.804	0.988	0.918	3
	0.733	0.021	0.647	0.733	0.688	0.671	0.939	0.634	4
	0.933	0.007	0.875	0.933	0.903	0.898	1.000	0.992	5
	0.800	0.011	0.800	0.800	0.800	0.789	0.975	0.843	6
	0.933	0.004	0.933	0.933	0.933	0.930	1.000	0.996	7
	0.933	0.021	0.700	0.933	0.800	0.797	0.997	0.953	8
	0.800	0.004	0.923	0.800	0.857	0.853	0.998	0.965	9
	0.467	0.018	0.583	0.467	0.519	0.500	0.841	0.400	10
	0.733	0.011	0.786	0.733	0.759	0.747	0.993	0.856	11
	1.000	0.007	0.882	1.000	0.938	0.936	0.999	0.970	12
	0.800	0.021	0.667	0.800	0.727	0.715	0.923	0.825	13
	0.600	0.004	0.900	0.600	0.720	0.724	0.930	0.769	14
	0.733	0.025	0.611	0.733	0.667	0.650	0.969	0.762	15
	0.800	0.004	0.923	0.800	0.857	0.853	0.969	0.890	16
	0.667	0.011	0.769	0.667	0.714	0.702	0.925	0.569	17
	0.400	0.021	0.500	0.400	0.444	0.421	0.945	0.620	18
	0.867	0.007	0.867	0.867	0.867	0.860	0.971	0.890	19
	0.667	0.011	0.769	0.667	0.714	0.702	0.951	0.600	20
Weighted Avg.	0.750	0.013	0.754	0.750	0.747	0.736	0.960	0.791	

Figure C.1: Accuracy of WPD for chrome browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
7	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	1	0	0	2	a = 1
1	12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	b = 2
0	0	13	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	c = 3
1	0	1	11	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	d = 4
0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	e = 5
0	1	0	0	0	12	0	1	0	1	0	0	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	0	0	14	0	0	0	1	0	0	0	0	0	0	0	0	0	g = 7
0	0	0	1	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	h = 8
0	0	0	0	0	0	0	1	12	0	0	0	2	0	0	0	0	0	0	0	i = 9
0	2	0	0	1	1	0	1	0	7	0	2	0	0	0	0	0	0	0	1	j = 10
0	0	2	0	0	0	0	0	0	0	11	0	0	0	0	0	0	2	0	0	k = 11
0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	l = 12
0	0	0	0	0	0	0	0	0	1	1	0	12	0	0	0	0	1	0	0	m = 13
1	0	0	0	1	1	0	0	0	0	0	0	1	9	0	0	1	0	1	0	n = 14
2	0	0	0	0	0	0	0	0	0	0	0	0	1	11	0	1	0	0	0	o = 15
0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	12	0	0	0	0	p = 16
0	1	0	3	0	0	0	0	0	0	0	0	0	0	1	0	10	0	0	0	q = 17
0	0	1	1	0	0	1	2	1	0	1	0	2	0	0	0	0	6	0	0	r = 18
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	13	0	s = 19
3	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	10	t = 20

Figure C.2: Confusion matrix of WPD for chrome browser

2. Firefox browser

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.400	0.042	0.333	0.400	0.364	0.328	0.894	0.327	1
	0.933	0.011	0.824	0.933	0.875	0.870	0.997	0.969	2
	0.667	0.011	0.769	0.667	0.714	0.702	0.986	0.826	3
	0.467	0.018	0.583	0.467	0.519	0.500	0.947	0.635	4
	0.867	0.004	0.929	0.867	0.897	0.892	0.996	0.959	5
	0.933	0.004	0.933	0.933	0.933	0.930	0.988	0.948	6
	0.867	0.014	0.765	0.867	0.813	0.804	0.963	0.904	7
	0.733	0.032	0.550	0.733	0.629	0.613	0.966	0.699	8
	1.000	0.007	0.882	1.000	0.938	0.936	1.000	0.991	9
	0.733	0.011	0.786	0.733	0.759	0.747	0.930	0.787	10
	0.867	0.011	0.813	0.867	0.839	0.830	0.968	0.903	11
	0.867	0.000	1.000	0.867	0.929	0.928	0.967	0.922	12
	0.933	0.004	0.933	0.933	0.933	0.930	0.999	0.986	13
	0.867	0.000	1.000	0.867	0.929	0.928	0.956	0.918	14
	0.733	0.014	0.733	0.733	0.733	0.719	0.972	0.762	15
	0.867	0.004	0.929	0.867	0.897	0.892	0.981	0.902	16
	0.600	0.025	0.563	0.600	0.581	0.558	0.921	0.673	17
	0.800	0.000	1.000	0.800	0.889	0.890	0.945	0.890	18
	0.933	0.011	0.824	0.933	0.875	0.870	0.998	0.964	19
	0.600	0.011	0.750	0.600	0.667	0.656	0.842	0.690	20
Weighted Avg.	0.783	0.011	0.795	0.783	0.785	0.776	0.961	0.833	

Figure C.3: Accuracy of WPD for Firefox browser

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
6	0	1	1	0	0	1	0	0	0	0	0	0	0	3	0	0	0	1	2	a = 1
0	14	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	b = 2
0	0	10	0	0	0	2	0	0	0	0	0	0	0	0	0	3	0	0	0	c = 3
1	1	2	7	0	0	0	0	1	0	2	0	0	0	0	0	1	0	0	0	d = 4
0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	e = 5
1	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = 6
0	0	0	0	0	0	13	1	1	0	0	0	0	0	0	0	0	0	0	0	g = 7
2	0	0	0	0	0	0	11	0	2	0	0	0	0	0	0	0	0	0	0	h = 8
0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	i = 9
0	0	0	0	0	0	0	4	0	11	0	0	0	0	0	0	0	0	0	0	j = 10
0	0	0	1	0	0	0	1	0	0	13	0	0	0	0	0	0	0	0	0	k = 11
1	1	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	l = 12
0	0	0	0	0	0	0	0	0	0	1	0	14	0	0	0	0	0	0	0	m = 13
0	0	0	0	0	1	0	0	0	0	0	0	1	13	0	0	0	0	0	0	n = 14
4	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	o = 15
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	13	0	0	0	0	p = 16
2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	9	0	1	1	q = 17
0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	12	0	0	r = 18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	14	0	s = 19
1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	9	t = 20

Figure C.4: Confusion matrix of WPD for Firefox browser

3. IE browser

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.733	0.011	0.786	0.733	0.759	0.747	0.975	0.604	1
	0.933	0.007	0.875	0.933	0.903	0.898	0.997	0.969	2
	0.667	0.018	0.667	0.667	0.667	0.649	0.929	0.778	3
	0.400	0.014	0.600	0.400	0.480	0.469	0.900	0.538	4
	0.867	0.004	0.929	0.867	0.897	0.892	0.971	0.932	5
	1.000	0.021	0.714	1.000	0.833	0.836	0.999	0.987	6
	0.867	0.000	1.000	0.867	0.929	0.928	0.923	0.884	7
	0.933	0.011	0.824	0.933	0.875	0.870	0.987	0.948	8
	0.933	0.004	0.933	0.933	0.933	0.930	0.983	0.945	9
	0.800	0.014	0.750	0.800	0.774	0.762	0.978	0.838	10
	0.600	0.004	0.900	0.600	0.720	0.724	0.988	0.840	11
	0.867	0.000	1.000	0.867	0.929	0.928	0.974	0.894	12
	0.867	0.011	0.813	0.867	0.839	0.830	0.996	0.931	13
	0.867	0.014	0.765	0.867	0.813	0.804	0.980	0.900	14
	0.867	0.032	0.591	0.867	0.703	0.698	0.983	0.823	15
	1.000	0.007	0.882	1.000	0.938	0.936	0.993	0.786	16
	0.733	0.011	0.786	0.733	0.759	0.747	0.798	0.735	17
	0.867	0.004	0.929	0.867	0.897	0.892	0.991	0.795	18
	0.867	0.007	0.867	0.867	0.867	0.860	0.997	0.938	19
	0.733	0.000	1.000	0.733	0.846	0.850	0.951	0.869	20
Weighted Avg.	0.820	0.009	0.830	0.820	0.818	0.813	0.965	0.847	

Figure C.5: Accuracy of WPD for IE browser

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.733	0.011	0.786	0.733	0.759	0.747	0.975	0.604	1
	0.933	0.007	0.875	0.933	0.903	0.898	0.997	0.969	2
	0.667	0.018	0.667	0.667	0.667	0.649	0.929	0.778	3
	0.400	0.014	0.600	0.400	0.480	0.469	0.900	0.538	4
	0.867	0.004	0.929	0.867	0.897	0.892	0.971	0.932	5
	1.000	0.021	0.714	1.000	0.833	0.836	0.999	0.987	6
	0.867	0.000	1.000	0.867	0.929	0.928	0.923	0.884	7
	0.933	0.011	0.824	0.933	0.875	0.870	0.987	0.948	8
	0.933	0.004	0.933	0.933	0.933	0.930	0.983	0.945	9
	0.800	0.014	0.750	0.800	0.774	0.762	0.978	0.838	10
	0.600	0.004	0.900	0.600	0.720	0.724	0.988	0.840	11
	0.867	0.000	1.000	0.867	0.929	0.928	0.974	0.894	12
	0.867	0.011	0.813	0.867	0.839	0.830	0.996	0.931	13
	0.867	0.014	0.765	0.867	0.813	0.804	0.980	0.900	14
	0.867	0.032	0.591	0.867	0.703	0.698	0.983	0.823	15
	1.000	0.007	0.882	1.000	0.938	0.936	0.993	0.786	16
	0.733	0.011	0.786	0.733	0.759	0.747	0.798	0.735	17
	0.867	0.004	0.929	0.867	0.897	0.892	0.991	0.795	18
	0.867	0.007	0.867	0.867	0.867	0.860	0.997	0.938	19
	0.733	0.000	1.000	0.733	0.846	0.850	0.951	0.869	20
Weighted Avg.	0.820	0.009	0.830	0.820	0.818	0.813	0.965	0.847	

Figure C.6: Confusion matrix of WPD for IE browser

4. Opera browser

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.067	0.046	0.071	0.067	0.069	0.022	0.774	0.143	1
	0.267	0.039	0.267	0.267	0.267	0.228	0.800	0.190	2
	0.867	0.011	0.813	0.867	0.839	0.830	0.940	0.929	3
	0.467	0.007	0.778	0.467	0.583	0.587	0.774	0.579	4
	0.333	0.039	0.313	0.333	0.323	0.286	0.923	0.364	5
	0.400	0.039	0.353	0.400	0.375	0.341	0.913	0.479	6
	0.933	0.018	0.737	0.933	0.824	0.819	0.976	0.942	7
	0.200	0.018	0.375	0.200	0.261	0.247	0.792	0.223	8
	0.733	0.011	0.786	0.733	0.759	0.747	0.920	0.810	9
	0.133	0.056	0.111	0.133	0.121	0.071	0.753	0.181	10
	0.533	0.028	0.500	0.533	0.516	0.490	0.911	0.454	11
	0.600	0.035	0.474	0.600	0.529	0.505	0.927	0.559	12
	0.533	0.007	0.800	0.533	0.640	0.639	0.927	0.774	13
	0.667	0.032	0.526	0.667	0.588	0.568	0.925	0.517	14
	0.333	0.039	0.313	0.333	0.323	0.286	0.827	0.305	15
	0.467	0.032	0.438	0.467	0.452	0.422	0.898	0.329	16
	0.333	0.025	0.417	0.333	0.370	0.343	0.841	0.496	17
	0.533	0.025	0.533	0.533	0.533	0.509	0.920	0.588	18
	0.400	0.049	0.300	0.400	0.343	0.307	0.869	0.271	19
	0.200	0.028	0.273	0.200	0.231	0.199	0.854	0.311	20
Weighted Avg.	0.450	0.029	0.459	0.450	0.447	0.422	0.873	0.472	

Figure C.7: Accuracy of WPD for Opera browser

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.067	0.046	0.071	0.067	0.069	0.022	0.774	0.143	1
	0.267	0.039	0.267	0.267	0.267	0.228	0.800	0.190	2
	0.867	0.011	0.813	0.867	0.839	0.830	0.940	0.929	3
	0.467	0.007	0.778	0.467	0.583	0.587	0.774	0.579	4
	0.333	0.039	0.313	0.333	0.323	0.286	0.923	0.364	5
	0.400	0.039	0.353	0.400	0.375	0.341	0.913	0.479	6
	0.933	0.018	0.737	0.933	0.824	0.819	0.976	0.942	7
	0.200	0.018	0.375	0.200	0.261	0.247	0.792	0.223	8
	0.733	0.011	0.786	0.733	0.759	0.747	0.920	0.810	9
	0.133	0.056	0.111	0.133	0.121	0.071	0.753	0.181	10
	0.533	0.028	0.500	0.533	0.516	0.490	0.911	0.454	11
	0.600	0.035	0.474	0.600	0.529	0.505	0.927	0.559	12
	0.533	0.007	0.800	0.533	0.640	0.639	0.927	0.774	13
	0.667	0.032	0.526	0.667	0.588	0.568	0.925	0.517	14
	0.333	0.039	0.313	0.333	0.323	0.286	0.827	0.305	15
	0.467	0.032	0.438	0.467	0.452	0.422	0.898	0.329	16
	0.333	0.025	0.417	0.333	0.370	0.343	0.841	0.496	17
	0.533	0.025	0.533	0.533	0.533	0.509	0.920	0.588	18
	0.400	0.049	0.300	0.400	0.343	0.307	0.869	0.271	19
	0.200	0.028	0.273	0.200	0.231	0.199	0.854	0.311	20
Weighted Avg.	0.450	0.029	0.459	0.450	0.447	0.422	0.873	0.472	

Figure C.8: Confusion matrix of WPD for Opera browser

5. Tor browser

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.933	0.004	0.933	0.933	0.933	0.930	0.991	0.948	1
	0.933	0.011	0.824	0.933	0.875	0.870	0.994	0.959	2
	1.000	0.007	0.882	1.000	0.938	0.936	1.000	1.000	3
	0.467	0.014	0.636	0.467	0.538	0.525	0.965	0.574	4
	0.867	0.018	0.722	0.867	0.788	0.779	0.987	0.810	5
	0.533	0.011	0.727	0.533	0.615	0.606	0.924	0.623	6
	0.800	0.011	0.800	0.800	0.800	0.789	0.919	0.839	7
	0.733	0.018	0.688	0.733	0.710	0.694	0.962	0.752	8
	1.000	0.007	0.882	1.000	0.938	0.936	0.996	0.871	9
	0.467	0.018	0.583	0.467	0.519	0.500	0.953	0.646	10
	0.600	0.011	0.750	0.600	0.667	0.656	0.941	0.676	11
	0.800	0.025	0.632	0.800	0.706	0.694	0.961	0.710	12
	0.667	0.011	0.769	0.667	0.714	0.702	0.957	0.758	13
	0.133	0.028	0.200	0.133	0.160	0.128	0.745	0.174	14
	0.867	0.025	0.650	0.867	0.743	0.736	0.987	0.880	15
	0.933	0.011	0.824	0.933	0.875	0.870	0.997	0.940	16
	0.667	0.039	0.476	0.667	0.556	0.536	0.928	0.658	17
	0.800	0.011	0.800	0.800	0.800	0.789	0.992	0.834	18
	0.667	0.011	0.769	0.667	0.714	0.702	0.970	0.725	19
	0.667	0.004	0.909	0.667	0.769	0.769	0.842	0.732	20
Weighted Avg.	0.727	0.014	0.723	0.727	0.718	0.707	0.951	0.755	

Figure C.9: Accuracy of WPD for Tor browser

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.933	0.004	0.933	0.933	0.933	0.930	0.991	0.948	1
	0.933	0.011	0.824	0.933	0.875	0.870	0.994	0.959	2
	1.000	0.007	0.882	1.000	0.938	0.936	1.000	1.000	3
	0.467	0.014	0.636	0.467	0.538	0.525	0.965	0.574	4
	0.867	0.018	0.722	0.867	0.788	0.779	0.987	0.810	5
	0.533	0.011	0.727	0.533	0.615	0.606	0.924	0.623	6
	0.800	0.011	0.800	0.800	0.800	0.789	0.919	0.839	7
	0.733	0.018	0.688	0.733	0.710	0.694	0.962	0.752	8
	1.000	0.007	0.882	1.000	0.938	0.936	0.996	0.871	9
	0.467	0.018	0.583	0.467	0.519	0.500	0.953	0.646	10
	0.600	0.011	0.750	0.600	0.667	0.656	0.941	0.676	11
	0.800	0.025	0.632	0.800	0.706	0.694	0.961	0.710	12
	0.667	0.011	0.769	0.667	0.714	0.702	0.957	0.758	13
	0.133	0.028	0.200	0.133	0.160	0.128	0.745	0.174	14
	0.867	0.025	0.650	0.867	0.743	0.736	0.987	0.880	15
	0.933	0.011	0.824	0.933	0.875	0.870	0.997	0.940	16
	0.667	0.039	0.476	0.667	0.556	0.536	0.928	0.658	17
	0.800	0.011	0.800	0.800	0.800	0.789	0.992	0.834	18
	0.667	0.011	0.769	0.667	0.714	0.702	0.970	0.725	19
	0.667	0.004	0.909	0.667	0.769	0.769	0.842	0.732	20
Weighted Avg.	0.727	0.014	0.723	0.727	0.718	0.707	0.951	0.755	

Figure C.10: Confusion matrix of WPD for Tor browser

Vitae

Name : Majdi Saeed Mohammed Bin Salman

Nationality : Yemeni

Date of Birth : 02/11/1980

Email : msbsalman@gmail.com

Address : Yemen- Mukalla – Hadramout

Academic Background

- Received Bachelor degree in Computer Science from Al-Ahgaff University, Yemen in 2004 with a GPA of 4.06/5.
- Joined Yemeni Fish Company in Yemen as IT supervisor from 2004-2010.
- Joint the Information and Computer science dept as full time student at King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia in September 2010.
- Complete Master in Computer Science, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia with a GPA of 3.30/4.

Publications

1. "Using SMCD to Develop Consistent Misuse Case Models: An Industrial Case Study", M. El-Attar, M. S. Bin Salman, 21st International Conference on Software Engineering and Data Engineering (SEDE'12), Los Angeles, USA. 2012.
2. "Fingerprinting Tor Protocol through Wavelet Packet Decomposition", Majdi Saeed Bin Salman, Sami Zhioua, Md. Rafiul Hassan, Fist International Conference on Anti-Cybercrimes (ICACC),Riyadh, Saudi Arabia. 2015.