University of Bedfordshire

“A Traffic Classification Method using Machine Learning Algorithm”

ASSIGNMENT TITLE: PROJECT REPORT

UNIT TITLE: MSC PROJECT 2013

SUPERVISOR: Dr Gregory Epiphaniou

Student ID: 1202428

Contents

List of Figures

List of Tables

Acknowledgement

I would like to take this opportunity to thanks my supervisor Dr Gregory Epiphaniou for the help and guidance he has provided throughout this thesis. I would also like to thanks my parents whom invaluable financial and moral support throughout my life helped me to achieve my aims.

CHAPTER 1

Abstract

Applying concepts of attack investigation in IT industry, this idea has been developed to design a Traffic **Classification Method** using Data **Mining techniques** at the intersection of Machine Learning Algorithm, Which will classify the normal and malicious traffic. This classification will help to learn about the unknown attacks faced by IT industry. The notion of traffic classification is not a new concept; plenty of work has been done to classify the network traffic for heterogeneous application nowadays. Existing techniques such as (payload based, port based and statistical based [3, 18, 20]) have their own pros and cons which will be discussed in this literature later, but classification using Machine Learning techniques [7] is still an open field to explore and has provided very promising results up till now.

1.0 Introduction

Honeypots are quite effective when it comes to prevent an attacker to sabotage the live networks. “Adisson wasley” in [1] define honeypot as “*resource whose value is in being probed, attacked or compromised*”. But in the past few years technology development bring new threats which are not easy to dealt with, sophisticated attacks of the present century have ability to evade firewalls, filters and honeypots [2]. Looking from the research perspective these tools can be very helpful to understand the theory of attacks and mindset of an attacker. Traditional use of honeypot is to prevent an attack as mentioned in literature and clear from the definition, but it can be used to study the security domain and can be a great help to develop new tools which would be capable of dealing with unknown sophisticated attacks. Such a honeypot would not rely on any attacker’s traffic but, filtered traffic from a sophisticated system which will only feed the honeypot with unknown traffic for the sole purpose of research and study of malicious packets.

In order to build such an environment, there is a need to classify the traffic before it enters to the research honeypot. This research is based on to accomplish such a classifying model which will classify the network traffic before it can be studied using honeypot. Generally characterization or classification gives the idea of traffic dynamics and helps to optimize the utilization parameters such as quality of service, bandwidth planning coping with security constraints and many more. But in the last few years development of new application and increase in the network traffic made classification a very challenging task and grabbed great attention from the researchers and system engineers. Port independent applications are contributing in the major part of internet traffic such as bitorrent and kazza (P2P) [3]. Traditional classification techniques like port based and payload based are not fully effective now, because of dynamic port allocation facilities and federal policies on cryptographic content [4, 5]. Latest trends are use of machine learning techniques [3, 4], which are very effective and use the distinctive flow characteristics of the traffic to classify. To the best of my knowledge this collaboration of research honeypot and traffic classification using machine learning algorithm is still wide open for the research community and has much to offer, which motivated myself to dig in and investigate.

1.1 Aim

Designing an internet Traffic Classification Method by implementing suitable machine learning Algorithms, in order to minimize the processing time of classification and increase the accuracy using open source tools.

1.2 Objectives

- Literature Review
    - To revise existing methods and provide extensive comparison among them.
- Creating a Test-bed Dataset

    - Test-bed parameter includes virtual machines, Traffic generation tools [41] and simulation tools [37]. Traffic generation to capture data for experimentation.
- Feature Selection
    - In order to build a classifier we need to define certain features extracted from the raw data. Maximum number of features will take maximum processing time, which contradicts our aim so feature selection will be critical part of this work.
- Data Formatting and Class (normal or abnormal) assigning
    - It is important to use Format of data which is acceptable by “Weka” this is “.arrf”. Furthermore defining the classes for the sampled data in order to define rules for classifier.
- Testing the Classifier and discussing simulation Results

The summary of this work has been shown in the form of pyramid structure below. Each step shown in the pyramid has a pre requisite, so the adopted methodology is waterfall model in this case.

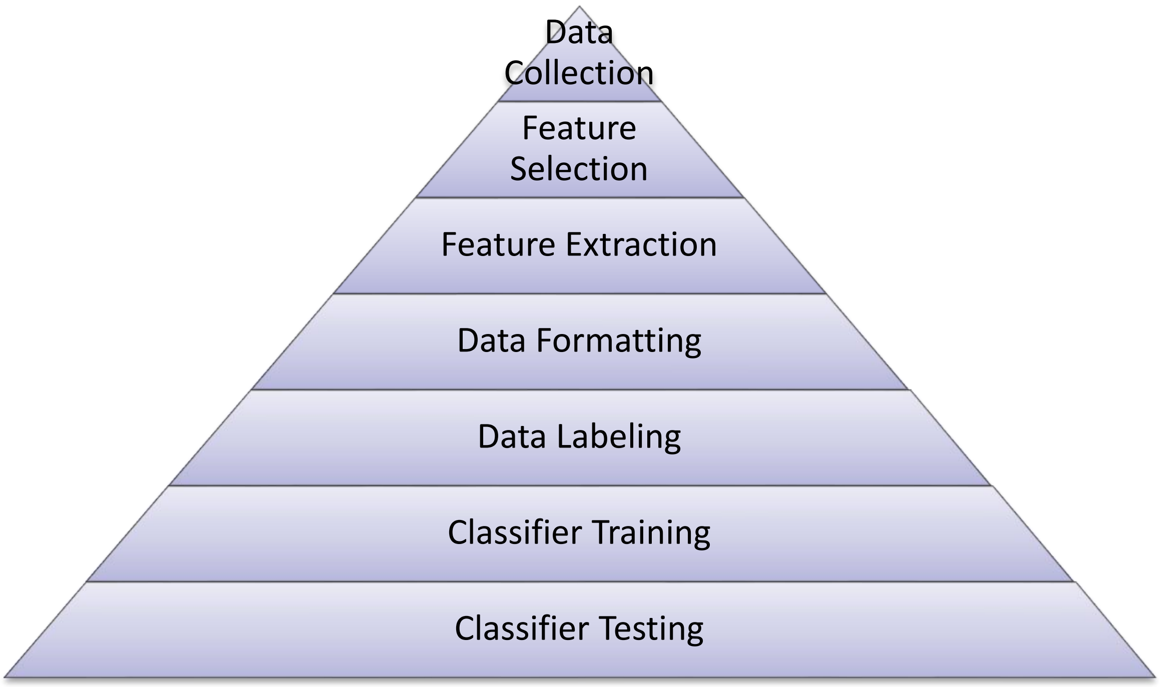


Figure 3.0 Pyramid representation of data classification method.

The rest of the work has been divided into three chapters; chapter 2 will present the literature review for internet Traffic classification and critically analyze the previous and current techniques used for classification. Chapter 3 will describe the designed method and its implementation using open source tools; last but not least chapter 4 will discuss the results and conclusion for this report.

CHAPTER 2

2.0 Literature Review

This chapter will highlight the previous and current state of the art work in the domain of internet traffic classification, furthermore it will investigate the reason that why classification is important in real time networks.

2.1 Port based classification

In order to send or receive any data on internet one needs transport protocol, which works on the third layer of TCP/IP stack or the fourth layer of OSI model. The most common transport protocols used for this purpose are TCP (Transport control protocol) which is connection oriented and UDP (User datagram) protocol which is connection less protocol. Both of these protocols uses logical concept of ports to distinguish the connections between two same end points. Traditionally many applications uses well known ports for communication with the host, classifier just needs to look at the first TCP SYN packet to grab this information and match the port with IANA [6] directory to classify the application. In UDP case the process is the same but without any connection establishment and also it does not maintain the state of connection [7].

IANA is responsible for registering the ports for dynamic applications it ranges from 0 to 65535 and have been divide into three subcategories:

- Well Known ports (0-1023)
- Registered Ports (1024-49151)
- Private ports (49152-65535)

Well know ports are commonly used by well know application and can only be altered by system administrator. Whereas registered ports does not require administrator privileges and are used by common user processes, however private ports are not and cannot be registered by IANA and just used for temporary purposes [8].

However there are limitations to this approach with the increase of internet traffic over the past few years and new P2P applications such as bittorrent, Napster and kaaza [9] may not use registered ports with IANA. Furthermore, due to the known vulnerabilities of different application, network administrators prefer to use different port numbers rather than registered ports for particular applications. This brings difficulty for the classification tools [10] which use ports as reference for classification. Authors in [11] observe up to 70% efficiency while using port based classification using IANA list matching. Similarly A. Madhukar and C.Williamson in [12] showed the limitations of port based approach as it was unable to classify 30-70% of the total internet flows they investigated in their research, different studies proved limitations of port-based due to the increase in P2P traffic on internet. This inability forced the researcher to come up with new classification techniques.

2.2 Payload based Classification

Payload-based classification often used as DPI (Deep Packet Inspection), it uses application level information from the coming packets in order generate and match signatures. It is quite reliable approach as each packet content is inspected in order to reconstruct the session and application information as mention in [4]. Sen et al. [13] showed significance increase in the accuracy by using payload based classification for P2P traffic, The work includes five different well know P2P protocols demonstrating signature based application classification.

To make signature based classification less resource consuming, key is to search for the specific string or byte pattern in the packet header. It uses predefined signatures to match the particular application traffic some of the examples of such technique are mentioned in table 2.2

[14]. This method of capturing traffic and then matching the pattern has been defined in these papers [15, 16].

Table 2.2 P2P protocols and signature strings.

Protocol	String
eDonkey	“\xe3\x38”
BitTorrent	"0x13Bit"
www	“\GET”

.

However DPI increases the classification accuracy but it has issues like complexity which increase the resources consumption, some protocols are encrypted and it is not easy task to decipher them in [13] Authors mentioning the significance of this approach on port based also predicted that future protocols will use encryption techniques to avoid signature based identification and this approach will hold the same state as port-based holding today. Furthermore, it is very difficult to store the string pattern of every protocol in the classifier which might occur on network, different countries have strict privacy restrictions. DPI uses application level information for classification so it is not ideal method in such conditions. However, after all these issues this technique is still widely used as it is most reliable among other of its kind [14], but it does not scale well for large networks, the reason is increased processing time and decreased accuracy.

2.3 Statistical Properties Classification

In contrast to payload and port based classification newer techniques rely on traffic statistics instead, to identify dynamic applications. The idea behind this approach is that traffic at the network layer has unique statistical properties such as packet inter-arrival time, packet length; flow idle time and flow duration which can be used to classify the traffic on the basis of applications as these parameters are different in each case [14]. Also Sen et al. [13] mentioned this as their future work for classification, as it was predicted that future protocols will be using encryption to hide their identities. Furthermore, it was suggested that unique characteristics (inter-arrival packets time, packet size and flow rate) of packets at network layer can be used to classify traffic.

An example of such work has been shown in [17] where authors tried to justify this relationship between class and statistical parameters, but studies found that WAN traffic cannot be modeled in statistical sense, only the simple models can be constructed which would give a reasonable approximation model. The reason behind this is traffic characteristics are changed between different sites and different hardware but similar protocols show quite similar behavior which can be used to construct approximation models. Some more examples of such work have been presented in [18, 19, and 20]. In [18] K. claffy presented his PHD thesis which includes an extensive research on internet traffic characterization, probably one of the earliest works of its kind. It realizes the limitation of port based application classification and discusses some quality of service issues for upcoming multimedia applications. Furthermore, it discusses the limitations of current statistics collection techniques of the time. Similarly [19] and [20] evaluates the statistical parameters by modeling two online gaming applications traffic that is Half life and Quake3, this is important to realize that these application are time sensitive and involves QOS issues so can be used as good examples for modeling sensitive traffic. They used ns2 simulator for evaluating results and found that characterization parameters like packet size, flow rate and inter arrival packet time are good source for guessing applications but one cannot be sure about the results. As the simulation graphs showed slight difference in these parameters while using different hardware. For example one can tell that it is multimedia traffic but cannot be sure if it is for Skype or Msn.

The results of these studies have given a new face to the classification techniques which are based on statistical properties plus artificial intelligence for these reasons researchers started applying data mining and machine learning methods for traffic classification.

2.4 Background on Machine Learning

Machine learning is the subset of algorithms developed in the discipline of Artificial Intelligence and these algorithms use different features to learn a set of rules in order to identify different classes [21]. Z.shi in [22] described "*One of the defining features of intelligence is the ability to learn*." It is a study of learning new knowledge and skills while reorganizing the existing one.

Machine learning has wide range of applications as mention in [14] search engines, image screening, marketing , forecasting, medical science, text and hand writing are few among many. The input of a machine learning process is a dataset of instances or examples, these instances are derived from the features also called discriminators (statistical parameters in case of networking) and a data set is a matrix of instances versus discriminators. Output of such process is the knowledge learnt by the machine.

2.4.1 Types of Learning

There are two major types of machine learning in context of network traffic classification.

- Supervised (classification)
- Unsupervised (clustering)

In this research the focus will be on supervised learning, for the sake of understanding the notion of machine learning a brief introduction and state of the art research work is presented here.

2.4.1.1 Supervised Machine Learning

It creates knowledge based structures which than help to classify the new instances of different classes [23]. Supervise machine learning models the input/output relationship for classification, sample instances are provided during the learning process which are pre-classified into classes and output of such a process depends on these generalized instances.

The dataset provided for training is labeled and at this stage of process the time does not matter that is how long it takes to process the sampled flows. More the number of attributes or feature more will be the time to process them and better will be the accuracy of classifier, different algorithm have different set of rules developed from the provided dataset and their performance varies under different circumstances. There is plenty of work published achieving high efficiency using these techniques for example in [24] authors are not using machine learning algorithm but they have used the same technique which provides the basis for supervised learning, they call it statistical fingerprinting technique first they train the classifier by providing statistical signatures for known traffic flows and called them fingerprints and then they use that learned knowledge to classify the traffic. In [25] classification has been discussed using three different machine learning algorithms to automatically generate the application signatures which can be later used for online classification, all information for constructing signatures has been grabbed during the beginning of communication. While in [26] authors used Bayesian analysis a pure supervised machine learning algorithm to classify dynamic traffic, with the very basic implementation authors were able to achieve 65% accuracy. Furthermore by implementing kernel estimation, accuracy has been increased up to 95% which proves the effectiveness of machine learning algorithms in this field. In 2006 Juhang Park and team implemented tree based classifiers to classify the live ISP traffic and suggested pre-classification to avoid collision errors [27].

High accuracy in field of network classification means low positive false rate (classifying flow "a" as "y" class while it does not actually belongs to y but "x") these are called the performance metric to evaluate the system efficiency. Table 2.4 shows the evaluation metrics of such machine learning algorithms [27].

Table 2.4 Evaluation metrics.

Belongs to →	X	Y
X	True positive	False negative
Y	False positive	True negative

If it is a class “X” in which we are interested then the accuracy with these parameters is measured as:

False positive→percentage of members belonging to Y but classified as X.

True positive→percentage of members of class X correctly classified as X.

False negative→percentage of members belonging to X incorrectly classified as Y.

True Negative→percentage of members not belonging to X and correctly classified as Y.

There are two more metrics which are often used as Machine learning evaluation metrics:

- **Recall**: percentage of members belonging to X and correctly classified.
- **Precision**: percentage of member classified as X truly belongs to X.

More details on evaluation metrics and class identification can be find in ji yang , Wang , Dong and Cheng work here [30]. The most common machine learning algorithms applied to traffic classification are C4.5 Decision Tree [31] , Naïve Bayes, Naïve Bayes Kernal Estimation, Bayesian Network K-NN , Neural network and SVM (support vector machines) [32]. Most successful results have been obtained from C4.5 and SVM.

2.4.2 Machine Learning Algorithms

Here for the sake of understanding let us have a look on couple of most used algorithms.

C 4.5 Decision Tree

It is an algorithm developed by Ross Quinlan in 1992 [33] as an extension of ID3.

- ID3

“decision tree algorithm written by J. Ross Quinlann in 1975, the process of building tree depends on provided examples and then this tree is use to classify future instances. Provided examples have different attributes which belongs to certain classes. Selection of these attributes depends on information gain, the attribute with the most information which can be easily separated for different classes are selected. More about ID3 can be found in [42].

C4.5 algorithm uses and generate tree based structure which can be used for Classification that is why it is also called statistical algorithm. It uses concept of entropy theory for classification for example we have data set S= {s1, s2…….sx.} where s1, s2 ….sx Represents the training samples of the data set which are characterized by different features, let say {X1, X2…} are the corresponding features consisting target class. Now C4.5 selects particular feature of the data set on each node, which is used to split these samples into different classes. The idea of selecting the feature depends on the normalized gain information from the samples, feature with the highest normalized gain is selected and the decision is made [34].

Some **advantages** of using decision trees are:

- Self Explanatory and easy to follow
- Can handle both numeric and nominal input attributes
- Can handle a data set with many errors including missing values

However, most DTs require the target variable to only have discrete values; they tend to perform well with non complex attributes. Furthermore, they are very sensitive to the training data sets any corrupt values close to the root node can change the whole structure of the tree.

Bayesian Network

It is a probabilistic graphical model [35, 36] often called belief network, it uses DAC (directed acyclic graph) to represent the conditional dependencies. Each node represents the random variables while the disconnected nodes are conditionally independent and the edges of nodes represent probabilistic dependencies among those random variables of corresponding nodes which are estimated using statistical and computational methods [34]. Learning in this algorithm is consisting of two parts, first is learning the network structure and second is learning probability tables. There are different methods uses for these two steps but a very famous tool "WeKa" which will be used for this thesis as well offers following approaches to accomplish the tasks [34].

- Local score metrics
- Conditional Independence test
- Global score metrics
- Fixed structure

"Weka" provides different search algorithms for each of these approaches once the network structure is identified than the probabilistic tables can be easily estimated using these tools. Various learned Bayesian classifiers has been discussed in [36] and results shows that CI (conditional independent) algorithms are quite efficient. It has already been discussed that 95% accuracy can be achieved using Bayesian analysis techniques in [26], however it does not discuss the training and computational time for the processes. Authors in [36] proved that using Bayesian classifiers saves much of processing and training time as compared to other machine learning algorithm, but it does not provide the data size information and type of applications, rather presented study by authors is more general and discusses the effectiveness of this approach in diversified fields.

This research will try to focus on the usefulness of these approaches in the field of internet traffic classification.

SVM (Support vector Machine)

SVM are powerful algorithms used to solve classification and regression problems. In order to classify the algorithm transform the input data to a high dimensional hyper plane, where it becomes more separable compared to the original form. This is done by using non linear kernel functions, and then linear classifiers are used to construct maximum margin hyper planes to separate the different classes in training data. Two hyper planes are constructed both sides of the hyper plane separating the data which tends to maximize the space between two parallel hyper planes. The assumption made is larger the distance between parallel hyper planes the better the generalization errors of the classifier will be [48].

SVM's learns through historic cases in the form of data point which contribute to very accurate classification, another **advantage** of these algorithms is they can handle missing values and noise effectively. However, these are complex and demands high memory requirements.

2.4.1.2 Unsupervised Learning

This type of learning is out of the scope of this work, so just a brief introduction is given here.

It is different from the supervised learning as it does not need labeled data for input alternatively, these techniques finds a way to naturally groups the data sets also called clusters, but still these clusters needs to be labeled by an expert. The advantage of clustering is if it is unknown it can be investigated later on [21]. Earliest work done on unsupervised technique by McGregor [38] uses Expectation maximization algorithm for IP traffic classification based on application such as HTTP, FTP, SMTP, IMAP and DNS results were impressive.

Issues with unsupervised learning are that clusters do not map 1:1to applications, in ideal case number of clusters formed are equal to the number of application to be mapped [7], but practically that is not the case. As mention in [39, 40] the number of clusters are often greater than the number of applications that is one single application might dominate over the number of clusters or application might spread over but do not dominate any of the clusters [7].

CHAPTER 3

3.0 IMPLEMENTATION AND DESIGN

This chapter will discuss the design methodology for classification task; furthermore it will highlight the approaches used to achieve the aim. Step by step flow chart diagram has been given below for Traffic Classification method.

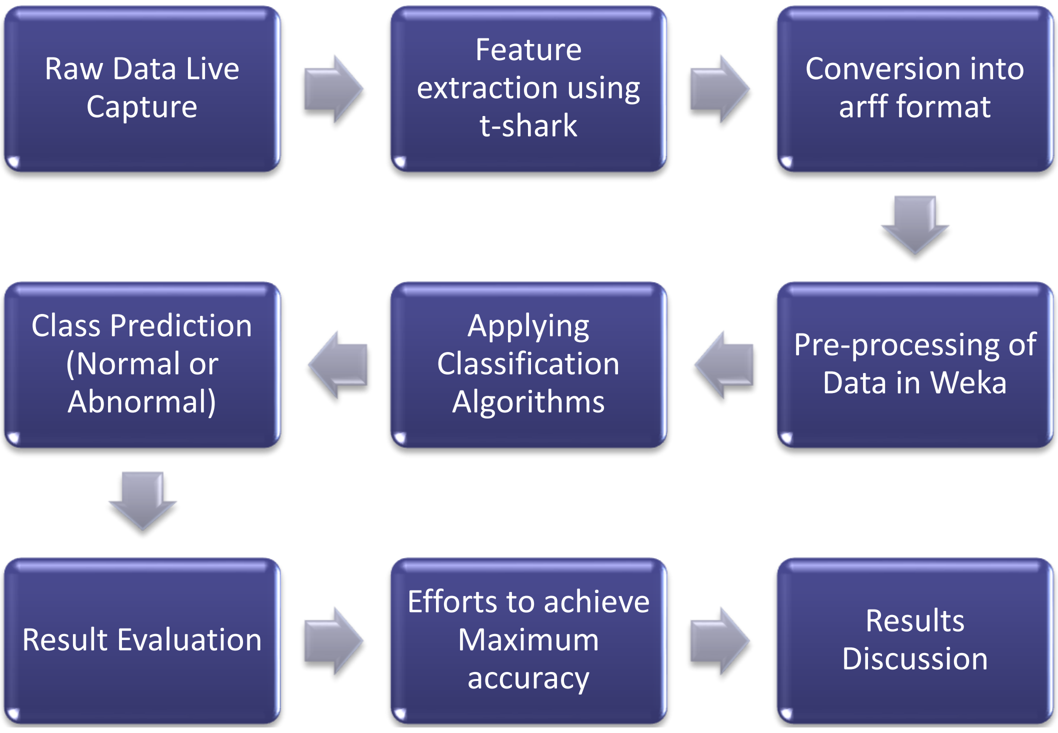

Raw Data Live Capture
Feature extraction using t-shark
Conversion into arff format
Pre-processing of Data in Weka
Applying Classification Algorithms
Class Prediction (Normal or Abnormal)
Result Evaluation
Efforts to achieve Maximum accuracy
Results Discussion

Figure 4.1: Flow diagram for Classification Task.

3.1 Data Collection

Data collection and feature extraction which is usually called preprocessing, are the most difficult steps of any research work as it consumes most time of whole process. This research is based on offline classification of benign (normal) and malicious (abnormal) traffic on the network. There are different methods to achieve this task, but it depends upon certain factors such as convenient resources, time and aim of research. Here it is important to learn that data collection itself is not difficult but collecting the relevant data for defined objectives makes it head scratching task, for example there is huge amount of data flowing through a single network due to large number of applications and protocols but if we are interested in all that data? When implementing machine learning algorithms for classification it is crucial to collect relevant data for classifiers learning process, now collecting this data is a challenge.

In the perspective of this work, collection of relevant data can be achieved using following approaches:

- Exporting data from large repositories online such as [43]. These datasets are maintained and provided by research organization for example DARPA and individual volunteers for research purposes. One can find labeled and unlabelled data sets in different formats.
- Intrusion detection logs from different IDSs and firewalls can provide malicious data and signatures which can be used for classification purposes.

- Honeypots are one of the best and reliable means of collecting malicious traffic which can be labeled and then used to train the classifiers.
- Traffic generated by pentesting tools during penetration can be learned by the classifier.
- There are number of tools available with capabilities to generate real time traffic or replay the captured real time traffic for example tcpreplay [41].

3.1.1 Defining Malicious Traffic

For the matter of fact one can ask that, how we are going to define abnormal or Malicious Traffic? Well, this is crucial but easy at the same time thanks to IT Security industry for providing such tools which only generate attack Traffic. But it does not mean that this traffic is only use to sabotage the network security on the other hand it can be used to increase the security of a network, but it is not the discussion here. It is very difficult to capture the live traffic on the network and then manually study the packets to differentiate abnormal or attack traffic, we would need some firewalls and IDSs for this purpose rather it is more easy to use security tools provided by different operating systems such as Backtrack [44] to generate attack traffic this will provide us with guarantee of having malicious traffic dataset for this study. Particular attribute extraction from these datasets will provide an opportunity to examine the difference between anomalies of benign and malicious traffic.

3.1.2 Wireshark Live Capture

For this work keeping in mind the factors involved that is time, aim and resources chosen approach includes data capture on a small residential wireless network to collect the traces of benign traffic. Tool used for this purpose is open source Wireshark, main focus of the study is on TCP traces as most of the internet traffic is consist of TCP protocol [7]. Due to the presence of firewall and Norton antivirus running on the workstation it can be **assumed** that more than 95 percent of the captured traffic have normal behavior and can be used to train the classifier.

This assumption about the traffic to be benign at this point sounds quite vague, because we have not done any deep packet inspection for the live capture. As we know that payload of the incoming packets is seen at application level and Wireshark captures packets at network layer, but to justify the assumption it can be said that post processing of captured tcp packets did not raised any alarm alerts on the work station as the Norton Antivirus fully updated version was running on the test machine. This fact can justify the assumption made about 95% of the packets captured. Dataset used for this particular case is not as critical as it would not be used for work like framework development for IDS testing system, because here we are trying to prove the effectiveness of Machine learning algorithms for Internet traffic Classification and how it can incorporate for research studies in the field of security. There are many datasets available online for research purposes, but the notion of data collection for this study is to get the in depth knowledge of this critical process and to get aware of critical factors involved in such process. Wireshark captured traffic file format is “.pcap” which will be converted into “.arrf”, it will be discussed later.

In order to collect the malicious traffic for training and classifying purposes pen-testing tools can be used and exploits available in backtrack5 [44] are quite handy and can be helpful for this task. Nmap is one of the most popular security tool [45] used by hackers and pen-testers, as main focus of this study is on TCP protocol for this reason the most used and popular nmap scan is used to generate malicious traffic form a Linux box called stealth scan. The reason for being most used among security community is that, it does not create unwanted traffic on the network by not completing the tcp handshake rather it just send the SYN packet and wait for the response to generate the required information about open, closed and filtered ports [45], and again Wireshark is used to capture these packets. Furthermore, to get some taste of a real exploit “Armitage” GUI version of Metasploit also available in Backtrack is used to hack a virtual Machine. This virtual machine is running windows XP as an operating system and famous vulnerability called **MS08-067 NetApi** is exploited using Metasploit. This vulnerability provides an attacker system privileges and has been known as a high risk to these systems until programmers have fixed this but in new window patches, more about this vulnerability can be found in [51] government database for known vulnerabilities. A successful attack on the system has been achieved and attack traffic has been captured through Wireshark which will be used to train the classifier. No payload inspection has been done as it is clear that this generated traffic is purely attack traffic. Furthermore, this classification is purely based on header information there will be no payload inspection reasons for this has been discussed in literature review that, what are the hurdles or difficulties one have to face with payload inspection of data.

3.2 Feature Selection

Selection of features which have unique properties among different applications is no doubt the vital part of any classification effort ever made, as it includes minute details and deep investigation of data packets. A single data packet has large number of features which can be studied, but not all of them are useful for classification. Furthermore, it is important to know the aim one wants to achieve, for example classification between different applications requires different features, to be extracted. Terry Brugger's in 2004 [46] has published a survey on Data "Mining Methods for Network Intrusion Detection", which discusses feature selection process in great details and also presented the work done for this task from number of other authors. The main task here is to select the attributes which have different values for normal and malicious traffic. The number of studies presented in Brugger's work shows that the most common features used for particular classification as ours are:

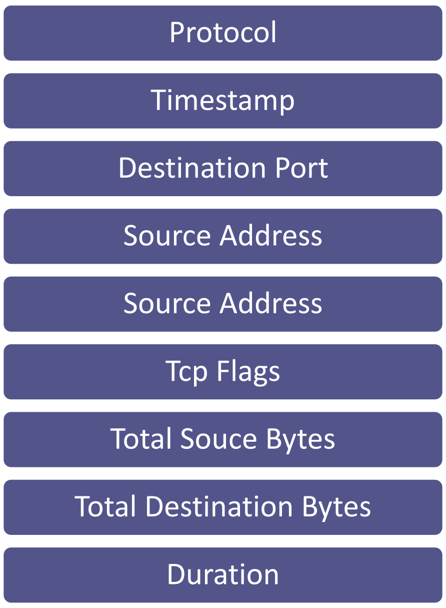


Figure 3.1 Features used the most for abnormal traffic classification.

Later studies [5, 7] have shown that features depending on time are not feasible for such classification as they vary from network to network and hardware to hardware. So from the learned knowledge of such studies, some features have been selected for extraction from the data collected for classification. This selection as discussed earlier is based on the previous studied, as these are the ones most commonly used by researches especially in security domain. Furthermore, manual observation from the captured data using wireshark makes more sense and compliments the use of these features by other researchers. However, there is no hard and fast rule for such selection, so at this stage of the project these reasons provide enough justification for mentioned selection. Later on we can validate the selection if we would be able to achieve the aim. Short detail of these features of a TCP packet has been discussed below.

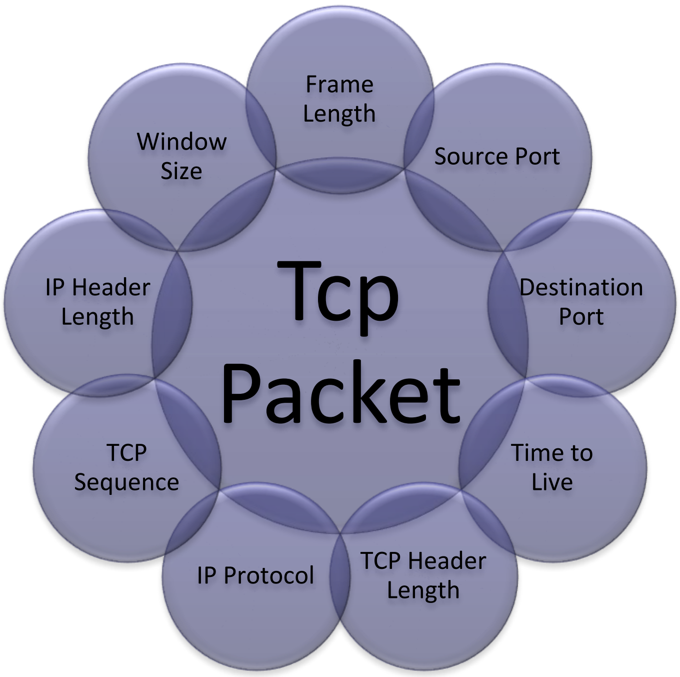


Figure 3.2 Selected Features for Classification Method.

- Frame Length represents the single packet size of IPv4 stream on the wire.
- Source Port is the logical number assigned to client IP.
- Destination Port is the logical number assigned to Server IP
- Time to live defines the life of a data on wire and after that it is discarded.
- TCP Header Length is the TCP protocol header size in bytes.
- IP Header Length is the IP header size in wire in bytes.
- IP Protocol represents type of communication protocol used; in this case it is TCP.
- TCP Sequence number is a 32-bit number used to keep the track of TCP data steam.
- Window size is an important feature while analyzing TCP packets it represents the data size in bytes that can be received by the receiver in TCP header.

More about these features can be found online [46, 47], all these attributes plays a role in determining the anomalies of network traffic. Furthermore, importance of these will be discussed in coming chapter.

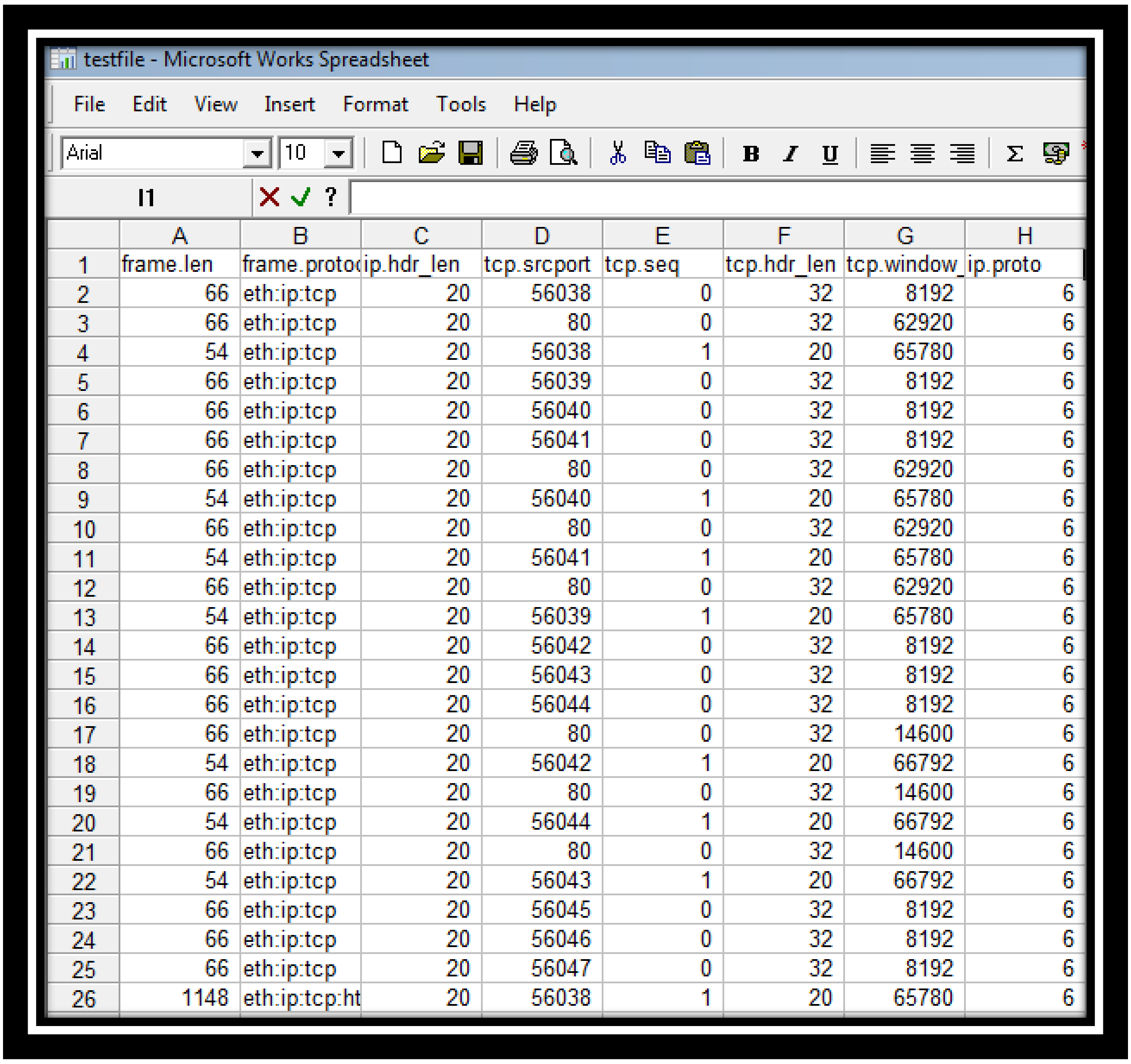
3.3 Feature Extraction

After the selection of particular features, it is time to extract them for next step. Extraction is a straight forward task if one has the required knowledge of the tools and their use. Here it is important to remember these extracted features are basic components of classification process as the classifier will be trained through these features in order to classify the unseen data. For this purpose wireshark command line utility can be used called as tshark. It serves the same purpose as wirehsark GUI but the process is faster if one is familiar with the right commands.

For this study wireshark GUI is used to capture and analyse the traffic so the right features can be examined and selected as it is more convenient. After capturing the data it is saved in the form of ".Pcap" format and then the following command is used to extract the selected features.

```
tshark -r capture.pcap -T fields -e frame.len -e tcp.dstport  -e ip.src  -e ip.dst -e ip.hdr_len -e
tcp.srcport -e tcp.seq -e tcp.hdr_len -e tcp.window_size -e ip.proto -E header=y -E separator=, -E
quote=d -E occurrence=f  > capture.csv
```

This command is calling function tshark which will "-r" read in file "capture.pcap" data file which has been saved earlier using wireshark, "-T" as a text file with "-e" fields mentioned here which represents the feature we have selected for extraction form the raw data. "-E" is the field print options how we want our data to be printed. This command will extract these features from the raw data and will print them off on the screen but if we want to convert this data in "csv" comma separated version we just need to add “>” and name of the output file as shown above this will redirect the output of this command to the mentioned captured.csv file which can be read by any common text file reader or if one wants to see this in the form of columns and rows which is more convenient it can be open in spread sheet viewer. Further information on the command synopsis can be found on the manual page of wireshark.org [48]. As an example Screen short of such a file has been provided below.



	A	B	C	D	E	F	G	H
1	frame.len	frame.protoc	ip.hdr_len	tcp.srcport	tcp.seq	tcp.hdr_len	tcp.window_	ip.proto
2	66	eth:ip:tcp	20	56038	0	32	8192	6
3	66	eth:ip:tcp	20	80	0	32	62920	6
4	54	eth:ip:tcp	20	56038	1	20	65780	6
5	66	eth:ip:tcp	20	56039	0	32	8192	6
6	66	eth:ip:tcp	20	56040	0	32	8192	6
7	66	eth:ip:tcp	20	56041	0	32	8192	6
8	66	eth:ip:tcp	20	80	0	32	62920	6
9	54	eth:ip:tcp	20	56040	1	20	65780	6
10	66	eth:ip:tcp	20	80	0	32	62920	6
11	54	eth:ip:tcp	20	56041	1	20	65780	6
12	66	eth:ip:tcp	20	80	0	32	62920	6
13	54	eth:ip:tcp	20	56039	1	20	65780	6
14	66	eth:ip:tcp	20	56042	0	32	8192	6
15	66	eth:ip:tcp	20	56043	0	32	8192	6
16	66	eth:ip:tcp	20	56044	0	32	8192	6
17	66	eth:ip:tcp	20	80	0	32	14600	6
18	54	eth:ip:tcp	20	56042	1	20	66792	6
19	66	eth:ip:tcp	20	80	0	32	14600	6
20	54	eth:ip:tcp	20	56044	1	20	66792	6
21	66	eth:ip:tcp	20	80	0	32	14600	6
22	54	eth:ip:tcp	20	56043	1	20	66792	6
23	66	eth:ip:tcp	20	56045	0	32	8192	6
24	66	eth:ip:tcp	20	56046	0	32	8192	6
25	66	eth:ip:tcp	20	56047	0	32	8192	6
26	1148	eth:ip:tcp:ht	20	56038	1	20	65780	6

Figure 3.3 Screen shot of “.csv” converted file.

It can be seen that number of columns are representing the features and rows represents the frame number as expected; now it is very easy to analyze each frame and the corresponding feature value.

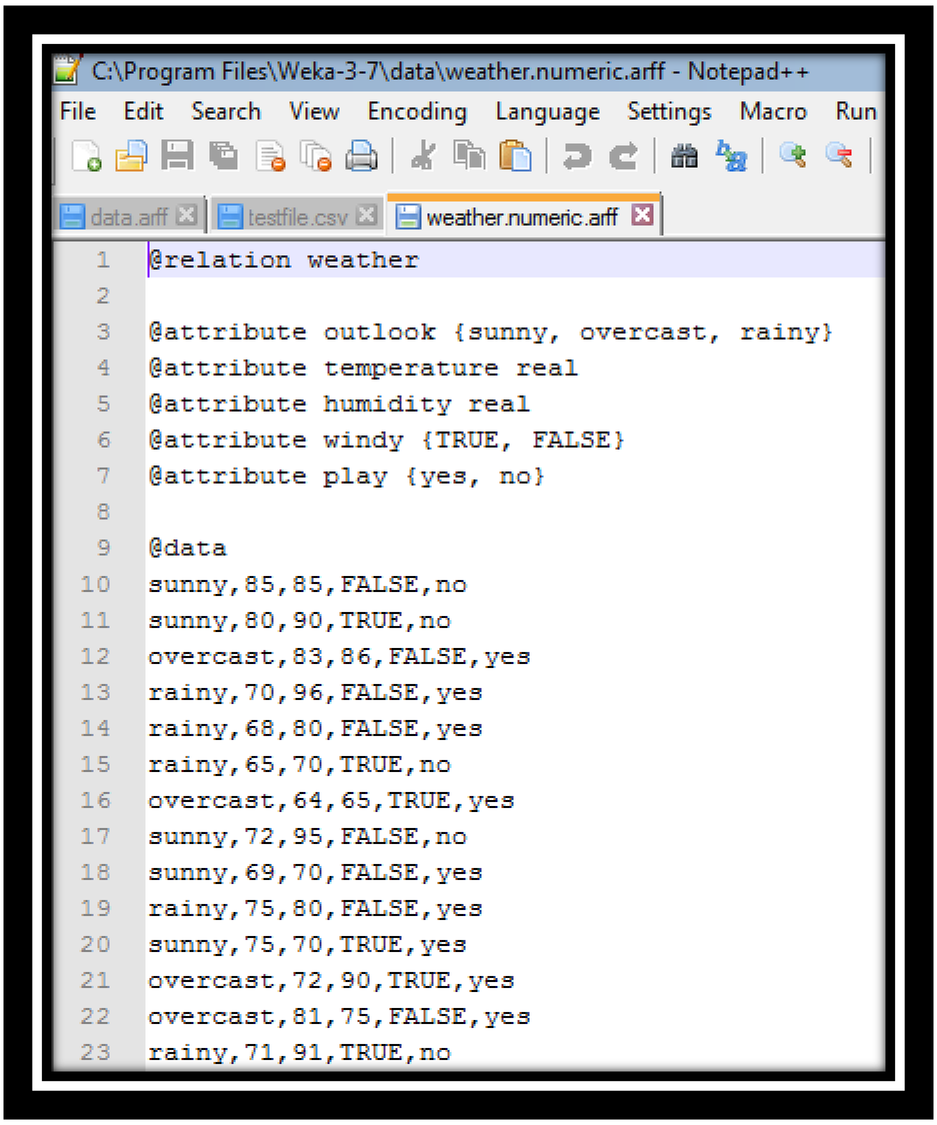
3.4 Data Formatting

When dealing with machine learning techniques we need to specify objective plans, that what aim is to be achieved, what data is to be used, what tools needed to process data are prominent among others. Aim of this study is classification of internet traffic data, for this purpose the tool selected is **Weka** the most popular known open source tool available easily and readily. Another reason behind this selection is its ability to implement large number of famous supervised, unsupervised algorithms and tree structures as discussed in the literature review. Furthermore, it is written in the widely use java language and has its own java API for implementation and research purposes.

As it is known that weka accepts “.arrf” format to process datasets, we have already converted the raw data to “.csv” now it is very easy to convert this file to “.arrf” format either by writing manually or by online conversion tools [49].

3.4.1 Structure of an ARRF file

Weka ARRF book version define Attribute Relation File Format as “An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes” [37]. Instances can be defined as Rows and attributes as columns in a dataset as shown in figure 3.3. To understand the structure of “.arrf” let us consider an example which is provided in the weka library.



```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

Figure 3.4 Example of an “.arrf” file.

Figure 3.4 shows and simple example of a training dataset in .arrf format. There are two main parts of an ARRF file **Header** part and **Data** part. Header part consist of all the features, last feature represents the class which determines whether the value is true or false for particular instance.

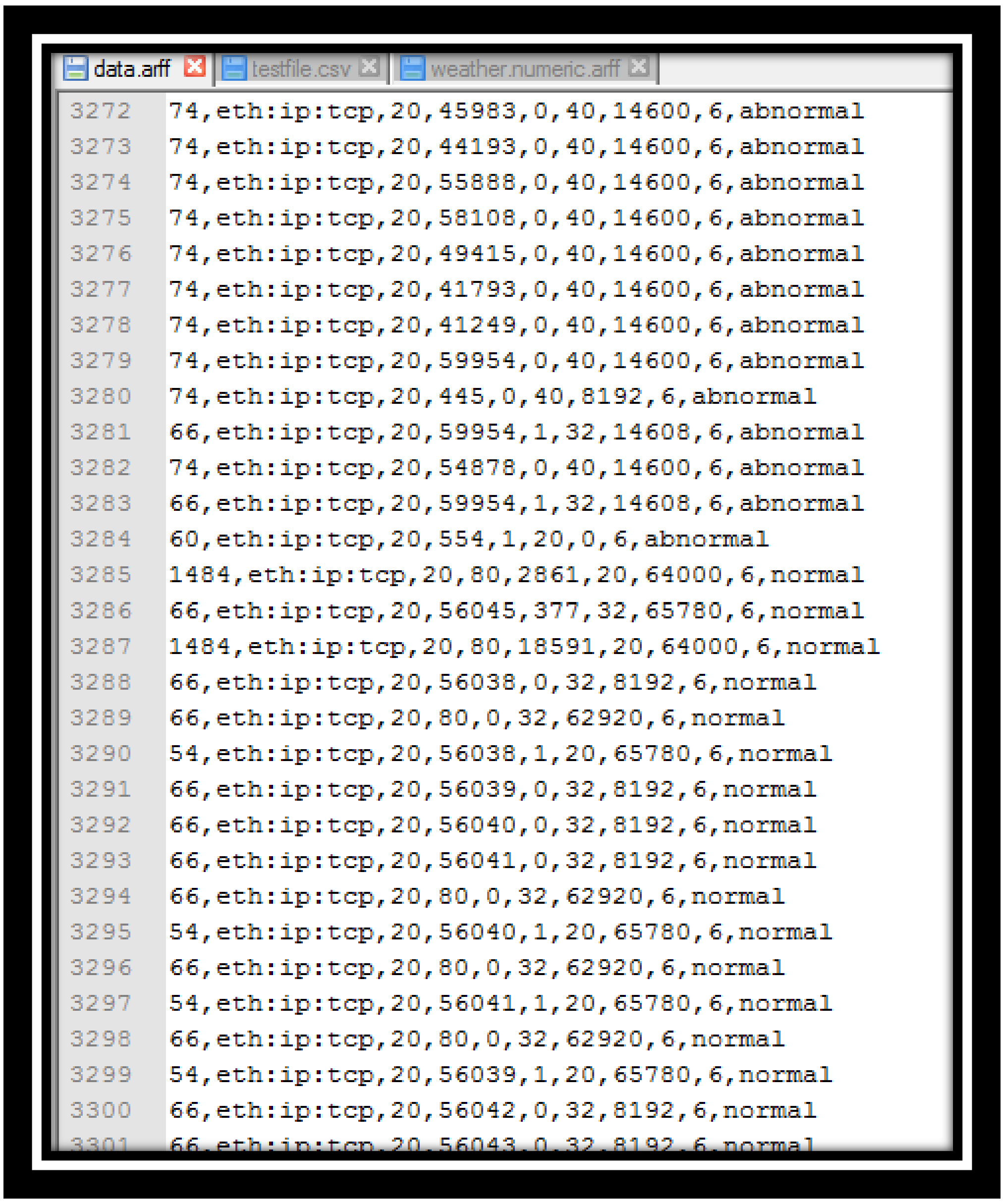
- @relation represents what kind of data is used in the dataset; in this case it is dealing with weather data.
- @attribute describes the unique features extracted from the raw data for classification.
- @data represents the data part of the ARRF file, above example shows a training dataset as it is labeled with class attributes. It is important to know that last attribute in the header part must be class attribute.

The aim of this example is to generate a set of rules to decide whether to play or not depending upon the four feature values, which a classifier takes as an input, the beauty of Weka is it can take input in different forms to generate such rules as it can be seen in this example dataset is consist of both numeric and nominal values at the same time.

3.5 Data Labeling

Data labeling is a process of adding a class feature to the extracted data, as in figure 3.4 the last column in data part is class and represents two values for each row “Yes” or “No”. There could be more than two values for class attribute but it depends on the type of data or classification used. In reference to this study we are dealing with IP traffic and aiming to classify the malicious traffic from benign, for this purpose it has two class nominal values “normal” and “abnormal” where normal represents benign and abnormal represents malicious traffic.

Now in order to add a class attribute to the last column of our collected data, it is suggested in various online articles explaining “.arff” files that one can simply edit the file in notepad or spread sheet editor and add another column. As authenticity of these articles could not be verified so they are not referenced in this work, as a matter of fact this technique has been applied to the collected data but unfortunately it did not work. To accomplish the task help has been taken from Weka.org [37] one can find in manual pages of Weka help that how to add a class attribute to unclassified data. Weka provides command line plus GUI tools to do so, after adding a class attribute values can be easily set using Weka editor. An example has been shown in figure 3.6. It can be seen that the last column of the dataset is showing two nominal values “normal” and “abnormal” for each row of dataset.

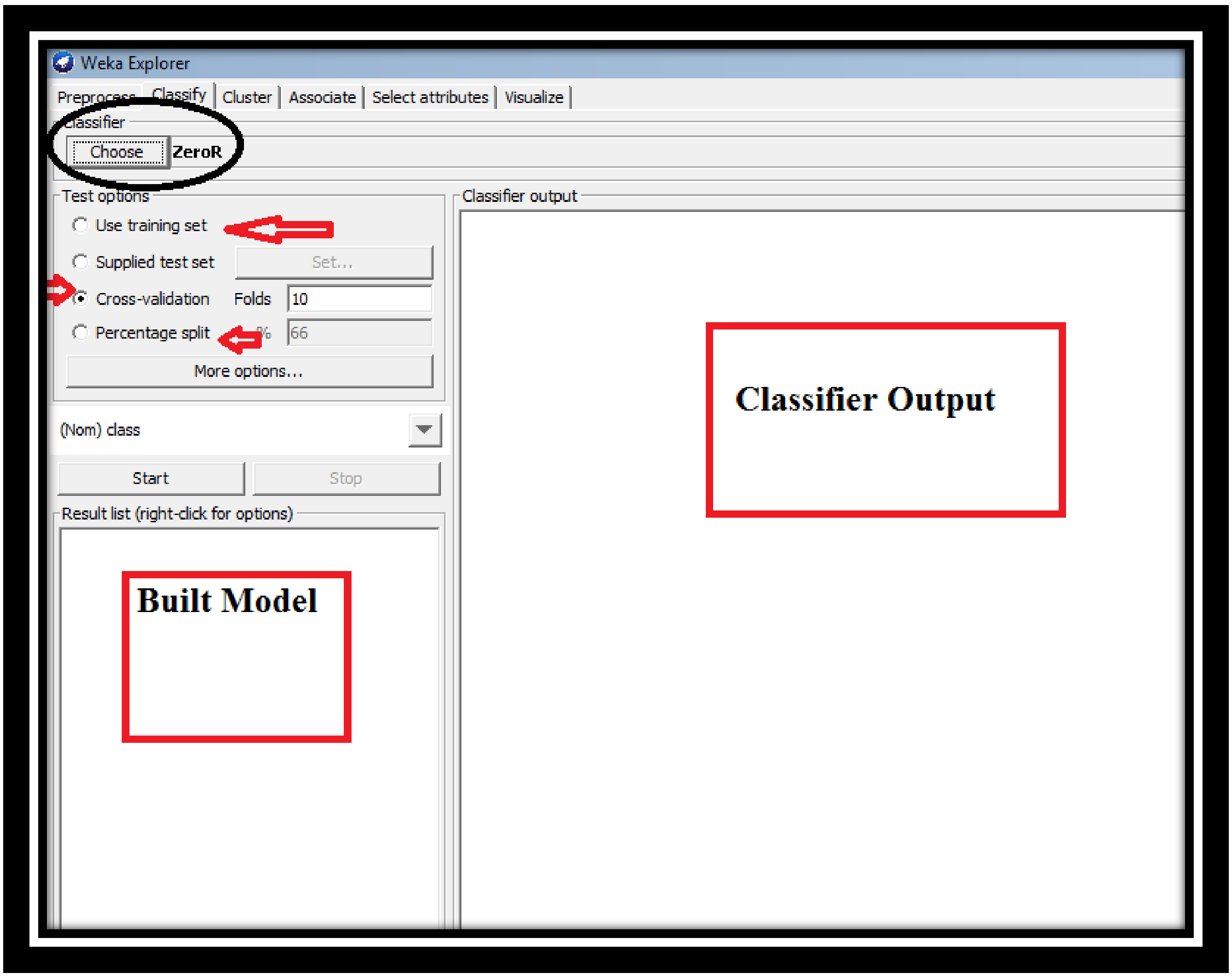


```
74,eth:ip:tcp,20,45983,0,40,14600,6,abnormal
74,eth:ip:tcp,20,44193,0,40,14600,6,abnormal
74,eth:ip:tcp,20,55888,0,40,14600,6,abnormal
74,eth:ip:tcp,20,58108,0,40,14600,6,abnormal
74,eth:ip:tcp,20,49415,0,40,14600,6,abnormal
74,eth:ip:tcp,20,41793,0,40,14600,6,abnormal
74,eth:ip:tcp,20,41249,0,40,14600,6,abnormal
74,eth:ip:tcp,20,59954,0,40,14600,6,abnormal
74,eth:ip:tcp,20,445,0,40,8192,6,abnormal
66,eth:ip:tcp,20,59954,1,32,14608,6,abnormal
74,eth:ip:tcp,20,54878,0,40,14600,6,abnormal
66,eth:ip:tcp,20,59954,1,32,14608,6,abnormal
60,eth:ip:tcp,20,554,1,20,0,6,abnormal
1484,eth:ip:tcp,20,80,2861,20,64000,6,normal
66,eth:ip:tcp,20,56045,377,32,65780,6,normal
1484,eth:ip:tcp,20,80,18591,20,64000,6,normal
66,eth:ip:tcp,20,56038,0,32,8192,6,normal
66,eth:ip:tcp,20,80,0,32,62920,6,normal
54,eth:ip:tcp,20,56038,1,20,65780,6,normal
66,eth:ip:tcp,20,56039,0,32,8192,6,normal
66,eth:ip:tcp,20,56040,0,32,8192,6,normal
66,eth:ip:tcp,20,56041,0,32,8192,6,normal
66,eth:ip:tcp,20,80,0,32,62920,6,normal
54,eth:ip:tcp,20,56040,1,20,65780,6,normal
66,eth:ip:tcp,20,80,0,32,62920,6,normal
54,eth:ip:tcp,20,56041,1,20,65780,6,normal
66,eth:ip:tcp,20,80,0,32,62920,6,normal
54,eth:ip:tcp,20,56039,1,20,65780,6,normal
66,eth:ip:tcp,20,56042,0,32,8192,6,normal
66,eth:ip:tcp,20,56043,0,32,8192,6,normal
```

Figure 3.6 Example of a classified training dataset.

3.6 Training Classifier

After preprocessing of the data one single file has been produced in the format which is acceptable by the tool used for building classification model. This file contains the data which will be used to train and test the classifier. There are three different ways provided by weka GUI to train a particular classifier to understand this process figure 3.7 has been provided below.



Classifier
Choose ZeroR

Test options
○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

Classifier Output

Built Model

Figure 3.7 Weka explorer Classification window GUI

The above figure is a screen shot of Weka explorer GUI, it is important to understand certain tabs and their functions when working with Weka tool. On the top left corner circled Tab showing choose classifier allows user to choose from range of classifiers based on number of algorithms such as Bayesian filters, SVM and trees as discussed in literature review. After choosing the classifier there are training and testing options on the left side as highlighted using arrows. In order to train the classifier one can provide a separate training set which will built a trained model showing in bottom left corner. Then one can provide a test set to test the already built model and results will be shown on the right side pane called classifier output, this is one way of training and testing [37].

Another way of doing the same procedure is using cross validation with “n-folds”, where “n” is the number of iterations used to train the classifier. By default the value of “n” is 10 it means that weka will split the training data in 10 equal parts and use 9/10 for training and 1/10 for testing the classifier, the process repeats 10 times until all the training sets has been used to train and test the classifier and output of the process it the best average model. This model can be than saved and used to test the different data sets.

Similarly the third option is the percentage split, the specified value will split the data accordingly for example if the percentage value is selected 66 percent than weka will use 66 percent of data for training the classifier and 34 percent for testing the performance of trained model [37].

CHAPTER 4

This chapter will test the trained classifier and will discuss the results, in order to understand the results it is important define some performance parameters which will help to assess the gains and limitations of this work. Weka output returns various statistics and calculations as results to evaluate the models prediction accuracy and performance. These techniques not only indicate the performance of a classifier, but can also be used as the basis of comparison to other classifiers.

4.0 Performance Parameters

- **Number of Correctly/Incorrectly classified instances** output displays the number of instances classified correctly and the number of instances those are not.
- **Accuracy** is the overall prediction accuracy which can be measured as:-

  Accuracy = number of correctly classified instances ÷ Total number of instances

- **Error rate** if the classifier predicts the class of an instance correctly, it is counted as a success if not it is an error. The error rate is the proportion of errors made over a whole set of instances which could be used to measure the overall performance of the classifier.
- **Confusion Matrix** A single prediction can have four outcomes namely True Positives (TP), True Negative (TN), False Positive (FP) and False Negative (FN).TP and TN are correct classifications where class ‘A’ is predicted as ‘A’ and class ‘B’ is predicted as ‘B’ where as FP is when class ‘A’ is predicted as ‘B’ and FN is when class ‘B’ is predicted as ‘A’. A confusion matrix is displayed as a table with a row and column for each class. The row denotes the actual value of a class where as the column denoted the predicted value of a class. Ideal results would have large numbers down the main diagonal and small or 0 on the off-diagonal.
- **True Positives (TP) and False Positive (FP) Rate** TP rate is TP divided by the total number of positives where as FP rate is FP divided by the total number of negatives. Ideally a good performing model would have a higher TP rate and a low FP rate.
- **Receiver Operating Characteristic (ROC) curve** The ROC curves plot the TP rate on the vertical axis against the FP rate on the horizontal axis to form bowl shape curve. The area under the ROC curve (AUC) denotes the classifiers performance. The bigger the area, the better the performance of the classifier therefore a well performing classifier would have a ROC curve pointing towards the top right [37].

These were the few parameters which will help to evaluate the performance of a classifier model, more on parameters can be found Weka official website.

4.1 Testing Classifier

Now in order to train and test the classifier an Algorithm is needed, first choice made for this study is **Naïve bayes** (Bayes Family) as it has been explained briefly that how these algorithms works in previous chapters. It is time to see the practical implementation of Naïve Bayes, it is probability based algorithm.

Naïve bayes

In this study we have nine extracted attributes from the raw data excluding the nominal attribute which is known as class, let us consider that these nine attributes can be denoted as A_x= {a1, a2, …..a_x} and two known classes in this case normal and abnormal let us say $\overset{i}{C}$= {C^n, C^a} now for each observed data attribute there is a known class that is $\overset{i}{C}$= a1→C^n. In order to predict the probability of unseen instance A_x the posterior probability of Naïve Bayes is given as [26]:

$$\Pr [\overset{i}{C} \mathrm{I} A_x] = \Pr(\overset{i}{C}) * f(A_x \mathrm{I} \overset{i}{C}) \div \sum \Pr(\overset{i}{C}) * f(A_x \mathrm{I} \overset{i}{C})$$

$\sum$ represents the summation of all probabilities of independence class $\overset{i}{C}$, $f(A_x \mathrm{I} \overset{i}{C})$ is the distribution function and denominator act as a normalization constant. Probability of class $\overset{i}{C}$ given that A_x, depends upon the Gaussian (normal) distribution of whole data. Concept of Gaussian distribution is given by the expression below [26]

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

This expression gives the distribution density for any value of "X" where in our case X is represented by "A_x". "σ" represents the standard deviation and "σ^2" represents the variance, "μ" is the mean of the distribution density. In order to calculate the posterior probability we need to find variance and means for the independent features, which is calculated from the given training set. In the form of mathematical expression it can be written as

$$f A_x \mathrm{I} \dot{C} \, (A_x, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(Ax-\mu)^2}{2\sigma^2}}$$

Where x= 1,2…9

In order to calculate the values of "σ^2" and "μ" for "A_x" which are not the actual values but the estimated values for these variables, are given by the prior results of maximum likelihood of events calculated in the training data.

$$\mu = \sum_{Ax \sim \dot{C}} \frac{Ax}{n\dot{C}}$$

This expression gives the likelihood of A_x belong to class $\overset{i}{C}$ when a certain value of Ax is divided by the total number of event occurrence of certain Class, in our case there are 2 classes C1 and Cs2. So in order to calculate first mean put x=1 and i=1.similarly variance can be calculated as [26]

$$\sigma^2 = \sum_{Ax \sim \dot{C}} (Ax - \mu)^2 \frac{1}{n\dot{C} - 1}$$

Now the final expression of posterior probability with the normal or Gaussian distribution can be given as

$$p(\dot{C} \sim Ax) = f A_x \mathrm{I} \dot{C} \, (A_x, \mu, \sigma^2) * p(\dot{C}) * \frac{1}{N}$$

Where N is the Normalization constant and probability for any value of A_x will be calculated from the area under the Gaussian curve which would look like this

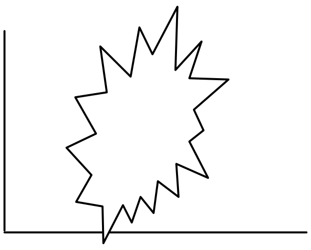


Figure 4.1a: unstructured data

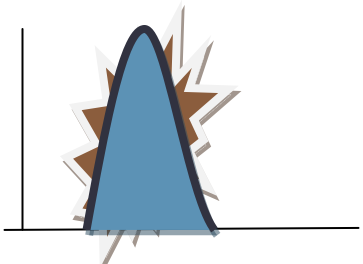


Figure 4.1b Gaussian (Normalized form) distribution function.

In figure "a" a rough representation of data instances has been shown before applying the Gaussian distribution. After applying the Gaussian distribution it can be seen that almost 90 percent of the data instances have been covered by the Gaussian curve and their estimated probabilities can be calculated using above equations, one should keep in mind that these are the estimated values that is why called probabilities and that is all we need rather that calculating the exact values. The centre of the peak curve in figure “b” gives us the mean which is “μ” and mean + and mean – will give the standard deviation [52]. More number of discrete instances gives clear (normal) Gaussian distribution, so the standard deviation can be easily calculated. This relationship of Gaussian distribution and Naïve bayes theorem gives Naïve bayes an edge when there is large number of data instances are used. Naïve Bayes assumes that each feature in the data is independent of another, though this assumption does not hold for most of the cases but Naives performance is not affected for this reason which is a good part.

Now in order to apply these finding to the data set, it has been split into two parts 66 percent of the data set has been used to train and built the classifier model and 34 percent of the data has been used to test the performance of built model.

Table4.1 Performance Evaluation of Naive Bayes

True positive	False Positive	Roc Area	Precision	Recall	Class
0.921	0.034	0.984	0.981	0.921	Normal
0.966	0.079	0.984	0.869	0.966	Abnormal

The time taken to test the built model which contains 3322 instances is 0.41 sec. Table above gives the performance evaluation of the Naïve Bayes. Correctly classified instances divided by the total number of instances gives the accuracy of the built model, which in this case is 93.70 percent. The accuracy achieved clearly depicts that machine learning has the ability to perform better than the existing approaches which gives us the maximum accuracy of 70-80 percent such that Port-based classification discussed in literature Review. Another important feature to access the performance of an Algorithm is its confusion matrix, in case of Naives Bayes the matrix achieved is

```
=== Confusion Matrix ===

    a    b   <-- classified as

 1988  170 |    a = normal

   39 1125 |    b = abnormal
```

The value “aa” of the matrix this is first row first column gives us the number of correctly classified instances belongs to normal class, similarly “bb” gives the number of correctly classified instances belongs to abnormal class. Performance of this matrix can be increased by increasing the values of first diagonal and decreasing the values in second. It can be noticed that there are still 209 instances which are classified as a wrong class, which suggests that the accuracy of the model is not maximum and needs improvement!!

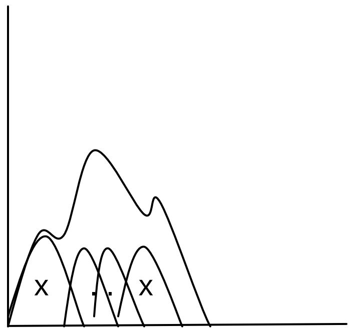
Now if we recall authors of [26] suggested and proved that accuracy of the naïve bayes can be increased by applying “**Kernel estimations**”. In order to improve our models accuracy concept of kernel estimation has been applied which is explained here.

Naïve Bayes Kernel Estimation

The major difference between normal distribution and kernel density is that normal distribution in naïve bayes fits the Gaussian distribution over the whole data set, where as kernel estimation estimates the Gaussian distribution for each kernel or instance. There are two important factors related to kernel estimation one is the shape selection and second is bandwidth window. The shape selection is normally the Gaussian curve distribution as shown earlier due to the fact it has finite end points which gives a good estimation of the density, bandwidth selection plays an important role towards the accuracy of the model [52].

$$f(t \sim \overset{i}{C}) = \frac{1}{n\overset{i}{C}h} \sum_{Ax \sim \overset{i}{C}} K(\frac{t - Ax}{h})$$

Where h is the bandwidth and K(t) is any non negative kernel such that $\int_{-\infty}^{\infty} K(t)dt = 1$ as we know for Gaussian distribution K(t) [26] is the exponential function that is $\frac{1}{\sqrt{2pi}} e^{(-t^{2}\frac{1}{2})}$. So distribution expression represents a linear combination of shifted kernels. Which can be visualize as



In this figure “x” and “.” Represents the data instances for each data instance separate kernel has estimated and sum of all those kernels gives the kernel density estimation represented by largest bump or kernel [54].

Figure 4.2: Gaussian Kernel estimation.

As it can be seen that the peak of estimated kernel curve is taking weight from the second instance “.”. But it is also taking weights from the declining kernels of relative instances. So what kernel estimation does is, reduce the assumption made by naïve bayes normal density distribution and get weights from all the instances in order to compute the commutative density. Does that help to increase the accuracy or minimize the error!! Well visually it looks a little bit more efficient if compared with the normal distribution. After running the naïve bayes with kernel estimation the results we got are shown in table 4.2.

Table4. 2 Naive bayes with kernel estimation

True positive	False Positive	Roc Area	Precision	Recall	Class
0.995	0.015	0.999	0.992	0.995	Normal
0.985	0.005	0.999	0.985	0.985	Abnormal

It can be seen that there is a prominent difference between the two test models; Naïve bayes with kernel estimation has achieved accuracy up to 99 percent. Incorrectly classified instance has been reduced significantly up to 28, and can be seen in confusion matrix of kernel estimation.

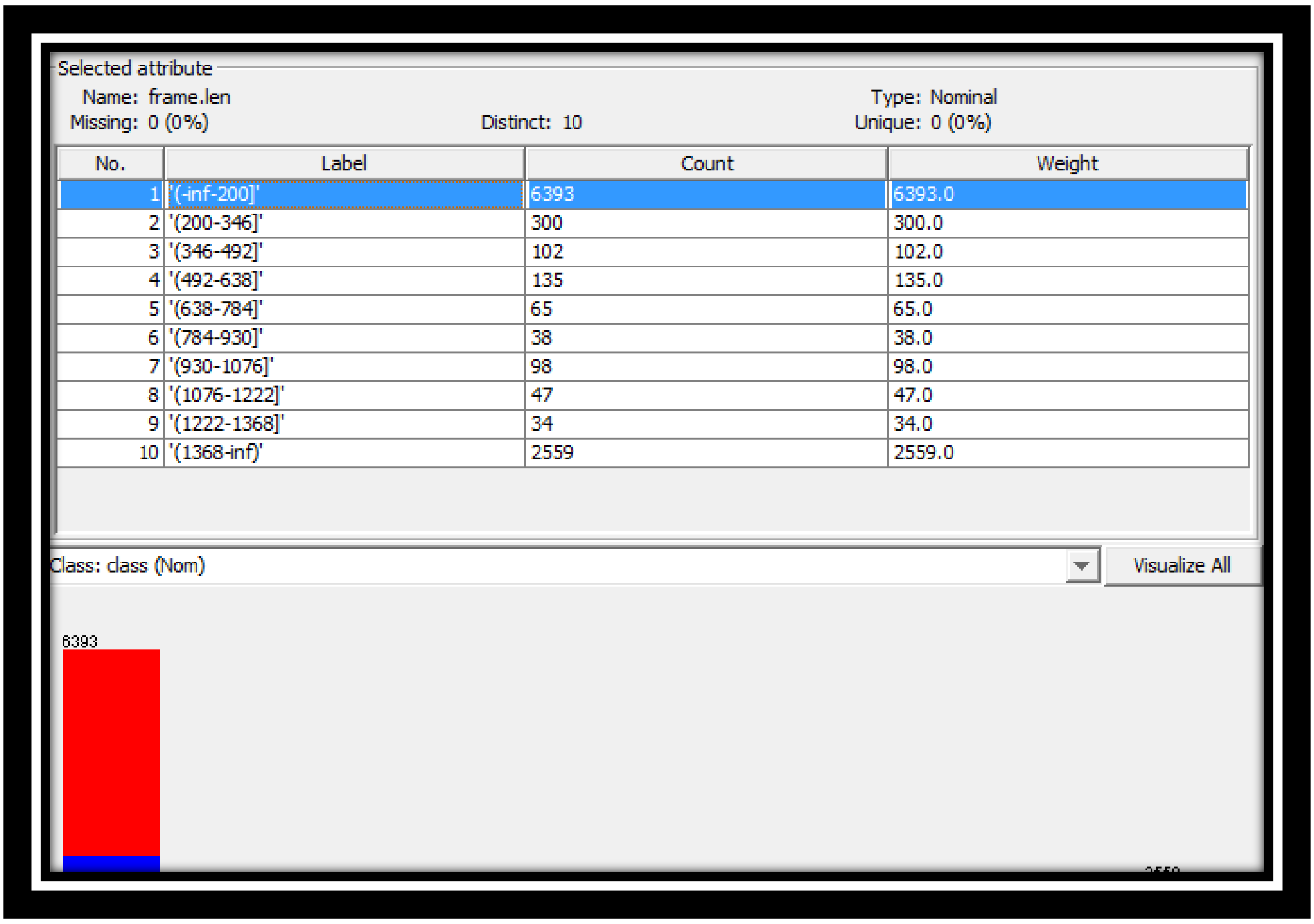
```
=== Confusion Matrix ===

    a    b   <-- classified as
 2147   11 |    a = normal
   17 1147 |    b = abnormal
```

These results are quite promising but have been achieved on the computational expanse. Normal distribution computes the density function just once for the whole data. However, kernel density distribution computes the weights “n” time hence increasing the computation process and memory usage. Furthermore, time to test the data set has been increased up to 1.22 seconds but an important observation has been made here which implies that, if we increase the test runs then the algorithm becomes smarter and after three tests runs the time taken to test the model was reduced to 0.66 seconds. This proves the efficiency and capability of machine learning algorithms, that they can become smart and smart after having certain experience just like humans!

Applying Discretization Concept

Authors in [55] have applied concepts of discretization to biomedical data and achieved better classification than its continuous form. Furthermore, form the research in the field of machine learning it has been proved that algorithms which handles nominal values performs well with the numeric datasets if the concepts of discretization is applied in the preprocess of datasets [56]. [55] Defines that “discretization is a process of converting continuous data to discrete intervals” and claims supervised discretization works well for classification purposes. In this study the focus is on supervised discretization as we are dealing with supervised learning, it will use the label class information from the training data to the target value of discretize data. In Weka the concept of discretizing data is developed by using a feature called bins [56], numerical data is divided into different parts representing bins and each bin has its new label in reference to the corresponding numeric data.

This concept would work well with naïve bayes this assumption is based on the naives capability of handling nominal class values. Applying discretization in weka changes the data in nominal form shown in figure 4.3. This is a data from a single attribute that is frame length of an IP packet, it can be noticed that the type has been changed from numeric to nominal and data has been split in to 10 different parts because the value selected for bin by weka was 10 by default in this case. The values highlighted here are showing that from “- ∞ to 200” total number of instances for particular feature is 6393 and so on so forth.



Selected attribute

Name: frame.len
Type: Nominal
Missing: 0 (0%)
Distinct: 10
Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-200]'	6393	6393.0
2	'(200-346]'	300	300.0
3	'(346-492]'	102	102.0
4	'(492-638]'	135	135.0
5	'(638-784]'	65	65.0
6	'(784-930]'	38	38.0
7	'(930-1076]'	98	98.0
8	'(1076-1222]'	47	47.0
9	'(1222-1368]'	34	34.0
10	'(1368-inf)'	2559	2559.0

Class: class (Nom)

Figure 4.3 Weka Discretization of numeric instances.

Data has been labeled according to the continuity in its nature of occurrence by even visualizing these labels it becomes clear that, it is easy to remember this format. After training naive bayes with this data, results achieved are quite promising and surprising as shown in table 4.3.

Table4. 3 Naive bayes with discretization

True positive	False Positive	Roc Area	Precision	Recall	Class
1.000	0.001	1.000	1.000	1.000	Normal
0.999	0.000	1.000	0.999	0.999	Abnormal

This can be observed that the precision has been increased up to 100 percent and accuracy of naïve bayes with discretize data has been improved to 99.938. Two important observations have been documented here, first is that the testing time for the built model has been surprisingly decreased up t0 0.23 seconds which is better than both previous cases and the second thing is the time taken to build the model is increased up to 0.35 seconds, which is more than the previous cases. These finding implies that with discretization preprocessing time of the data is increased as we have to apply extra filters to change the numeric values to nominal. Furthermore, algorithm learning time is also increased as the labels has been increased by the factor of 10 (bins) for each attribute. Confusion matrix gives us the number of incorrectly classified instances which are reduced to just “2” in this case.

```
=== Confusion Matrix ===


    a    b   <-- classified as

 2157    1 |    a = normal

   1 1163 |    b = abnormal
```

4.2 Findings from Naïve bayes Experimentation

Naïve bayes experimentation showed that the simplicity and efficiency of the algorithm are the major factors of its fame among the research community, the assumptions made by the algorithm sounds vague but really works well in practice. As the results in this study have proved that the capabilities of the algorithm are not limited and results can be boosted with simple refinements such as **Kernel estimation** and **discretization**. Though there are some tradeoffs has to be done to achieve higher accuracy, but they are negotiable! Kernel estimation improves the model accuracy at the expanse of computational complexity, but the development in the field of IT where we have 3D transistors available can ease the computational process. Furthermore, being a machine learning algorithm it has ability to make himself smarter by making number of runs on the test data. Naïve bayes works really well with large datasets, which makes it more reliable choice for IP classification. As it has been noticed that the algorithm has a great ability to work with nominal attributes, which means that if the data is in the continuous form it can be discretized which helps to enhance the accuracy of the algorithm. Unfortunately no renowned work has been published in the field of Internet traffic classification with applied Discretization. As from the experimentation it has been found that naïve bayes gives best results if the data is discretized, comparison among the basic naïve bayes algorithm and its refinements has been shown statistically in table 4.4.

Table 4. 4 Performance Comparison Naïve Bayes Algortihm

Algorithm	Accuracy	Built time	Test time
Naïve Bayes	93.70	0.12	0.41
Kernel estimation	99.13	0.15	1.22
Discretization	99.93	0.35	0.23

4.3 Limitations

Every research has its own limitations due to different critical factors involved such as time and resources. Best efforts have been made to achieve aim in short period of time with minimum limitations. The major limitation of this study could be the dataset used, due to the minimum resources and short time period; the dataset used does incorporate all protocols and malicious traces of internet traffic. However, dataset used includes all ingredients essential to test any algorithm performance in reference to machine learning techniques.

4.4 Conclusion

Classification of normal and normal traffic is not easy in the modern world of internet; current techniques have limitations which are being exploited by network intruders. This critical issue has taken a war situation among good and bad, in order to meet the basic security interests on internet traffic these issues should be dealt with serious efforts. The field of machine learning has shown some promising results which can be used to fight against cyber crime. This study is an effort to explore such aspects of machine learning, which can be used to study most sophisticated attacks on internet. It has been seen that applying concepts of data mining with intersection machine learning can achieve highest accuracies in classifying internet traffic. In this study it has been shown that how a simplest machine algorithm like Naïve bayes can be used to achieve maximum classification accuracy, which makes us wondering about the complex algorithms like SVMs, Tree based and neural network performance in this domain. It has also been noticed that the features selection from the raw data was quite accurate as the accuracy of the algorithm depends upon the data used, the most distinct features among used set were “time to live” and “frame length” which can be observed manually or by applying information gain filter in Weka. Kernel estimation performs very well with Naïve bayes as shown in previous studies as well. The most important observation made from the results is that, internet traffic data depicts continuous form so concepts of discretization can easily be applied to improve the task of classification. So by applying discretization to the dataset used in this study results were overwhelming, as the accuracy achieved was almost 100 percent in this case. This study proves machine has a great potential to overcome the limitations of current techniques for internet classification and it will be the upcoming trend to follow by the IT manufactures.

4.5 Future Work

In future, this study can be extended to incorporate all protocols by capturing traffic on large networks for longer period of time to read the anomalies. By creating a state of the art dataset, performance evaluation can be done using more complex Algorithms. Some research on the tree structures was done during this study, but due to the short time period did not documented. Tree based research will be extended as it is believed that tree based structures have most capabilities to generate highly effective rules to classify data, which can be later used in current IDSs infrastructures. After achieving the best classifier the aim is to apply that model on online traffic, which will give a clear insight to the task of real time classification.

References

[1] N. Provos and T. Holz, Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Addison-Wesley, 2008.

[2] Qassrawi, M.T.; Hongli Zhang, "Client honeypots: Approaches and challenges," *New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on* , vol., no., pp.19,25, 11-13 May 2010.

[3] Arthur Callado, Carlos Kamienski ,Géza Szabó, Balázs Péter Ger Ýo, Judith Kelner,Stênio Fernandes ,and Djamel Sadok. "A Survey on Internet Traffic Identification," IEEE Communications Survey & tutorials, Vol. 11, No. 3, pp. 37-52, Third Quarter 2009.

[4] Thuy T.T. Nguyen and Grenville Armitage. "A Survey of Techniques for Internet Traffic Classification using Machine Learning,"IEEE Communications Survey & tutorials,Vol. 10, No. 4, pp. 56-76, Fourth Quarter 2008.

[5] Runyuan Sun, Bo Yang, Lizhi Peng, Zhenxiang Chen, Lei Zhang, and Shan Jing. "Traffic Classification Using Probabilistic Neural Network," in Sixth International Conference on Natural Computation (ICNC 2010), 2010, pp. 1914-1919.

[6] http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml accessed on 4/04/13.

[7] Nguyen, T.T.T.; Armitage, G., "A survey of techniques for internet traffic classification using machine learning," *Communications Surveys & Tutorials, IEEE* , vol.10, no.4, pp.56,76, Fourth Quarter 2008doi: 10.1109/SURV.2008.080406

[8] http://www.ncftp.com/ncftpd/doc/misc/ephemeral_ports.html accessed on 04/04/13.

[9] M. Roughan, S. Sen, O. Spatscheck and N. Duffield"Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification"*Proc. ACM/SIGCOMM Internet Measurement Conference (IMC) 2004, 2004.*

[10] CoralReef. http://www.caida.org/tools/measurement/coralreef accessed on 04/04/13.

[11] A. Moore and K. Papagiannaki"Toward the accurate identification of network applications"*Proc. Passive and Active Measurement Workshop (PAM2005), 2005.*

[12] A. Madhukar and C. Williamson"A longitudinal study of P2P traffic classification"*14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006.*

[13] S. Sen, O. Spatscheck and D. Wang"Accurate, scalable in network identification of P2P traffic using application signatures"*WWW2004, 2004.*

[14] Callado, A.; Kamienski, C.; Szabo, G.; Gero, B.; Kelner, J.; Fernandes, S.; Sadok, D., "A Survey on Internet Traffic Identification," *Communications Surveys & Tutorials, IEEE* , vol.11, no.3, pp.37,52, 3rd Quarter 2009.

[15] T. Karagiannis, A. Broido, N. Brownlee, K. Clay, and M. Faloutsos.File-sharing in the Internet: A characterization of P2P trafic in the backbone. University of California, Riverside, USA, Tech. Rep\ , 2003.

[16] A. Moore and K. Papagiannaki. Toward the Accurate Identification of Network Applications. In Passive & Active Measurement Workshop Springer, 2005.

[17] V. Paxson "Empirically derived analytic models of wide-area TCP connections" *IEEE/ACM Trans. Networking, vol. 2, no. 4, pp. 316-336, 1994.*

[18] K. Claffy "Internet traffic characterization" *1994.*

[19] T. Lang, G. Armitage, P. Branch and H.-Y. Choo "A synthetic traffic model for Half-life"*Proc. Australian Telecommunications Networks and Applications Conference 2003 ATNAC2003, 2003.*

[20] T. Lang, P. Branch and G. Armitage "A synthetic traffic model for Quake 3"*Proc. ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE2004), 2004.*

[21] Orial Mula-Valls "A practical retraining mechanism for network traffic classification in operational environments" june 2011.

[22] Z. Shi Principles of Machine Learning *1992, International Academic Publishers.*

[23] Y. Reich and J. S. Fenves Fisher, D. H. and Pazzani, M. J. (editors), Concept Formation: Knowledge and Experience in Unsupervised Learning *1991, Morgan Kaufman.*

[24] M. Crotti and F. Gringoli. Traffic classification through simple statistical fingerprinting. ACM SIGCOMM Computer Communication Review,37(1):5{16, 2007.

[25] P. Haner, S. Sen, O. Spatscheck, and D. Wang. ACAS: automated construction of application signatures. In Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data, pages 197{202. ACM New York, NY, USA, 2005.

[26] A. W. Moore and D. Zuev. Internet traffic classification using Bayesian analysis techniques.ACM SIGMETRICS Performance Evaluation Review, 33(1):50{60, 2005.

[27] J. Park, H.-R. Tyan, and C. C. J. Kuo. GA-Based Internet Traffic Classification Technique for QoS Provisioning. In Proceedings of the2006 International Conference on Intelligent Information Hiding and Multimedia, IIH-MSP '06, pages 251{254, Washington, DC, USA, 2006.IEEE Computer Society.

[28] M. Roughan, S. Sen, O. Spatscheck, and N. Dueld. Class-of-Service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification. InInIMC'04, 2004.

[29] G. Szabo, I. Szabo, and D. Orincsay. Accurate Traffic Classification. In World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a, pages 1{8, 2007.

[30] Jie Yang; Yixuan Wang; Chao Dong; Gang Cheng, "The evaluation measure study in network traffic multi-class classification based on AUC," *ICT Convergence (ICTC), 2012 International Conference on* , vol., no., pp.362,367, 15-17 Oct. 2012.

[31] J. R. Quinlan. C4. 5: Programs for Machine Learning. Morgan Kauf-mann, 1993.

[32] K. P. Bennett and C. Campbell. Support vector machines: hype or hallelujah? SIGKDD Explor. Newsl. 2(2):1 13, December 2000.

[33] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[34] Singh, K.; Agrawal, S., "Comparative analysis of five machine learning algorithms for IP traffic classification," *Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on* , vol., no., pp.33,38, 22-24 April 2011.

[35] Ian H, Witten and Eibe Frank.(2005) Data Mining: Practical Machine Learning Tools and Techniques,2th edition, Morgan Kaufmann Publishers, San Francisco, CA.

[36] Jie Cheng and Russell Greiner. Learning Bayesian Belief Network Classifiers: Algorithms and System. Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada.

[37] Weka website (2011) http://www.cs.waikato.ac.nz/ml/weka/ accessed on 05/04/2013.

[38] A. Mcgregor, M. Hall, P. Lorier, and J. Brunskill. Flow Clustering UsingMachine Learning Techniques. In *In PAM, pages* 205 214, 2004.

[39] S. Zander, T. Nguyen and G. Armitage "Automated traffic classification and application identification using machine learning" *IEEE 30th Conference on Local Computer Networks (LCN 2005), 2005.*

[40] J. Erman, A. Mahanti, M. Arlitt and C. Williamson "Identifying and discriminating between web and peer-to-peer traffic in the network core" *WWW \'07: Proc. 16th international conference on World Wide Web, pp. 883-892, 2007.*

[41] http://wiki.wireshark.org/Tools accessed on 14/04/2013. Accessed on 15/04/2013.

[42] http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm. Accessed on 16/04/2013.

[43] http://www.netresec.com/?page=PcapFiles Accessed on 25/04/2013.

[44] http://www.backtrack-linux.org/ Accessed on 25/04/2013.

[45] http://nmap.org Accessed on 25/04/2013.

[46] Terry Brugger's "Data Mining Methods for Network Intrusion Detection" 2004.

[47] http://packetlife.net Accessed on 26/04/2013.

[48] http://www.wireshark.org/ Accessed on 26/04/2013.

[49] http://slavnik.fe.uni-lj.si/markot/csv2arff/csv2arff.php Accessed on 27/04/2013.

[50] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, "A Practical Guide to Support Vector Classification", Technical report, National Taiwan University. Taipei, 2004.

[51] http://web.nvd.nist.gov/view/vuln/statistics Accessed on 5/11/2013

[52] "Introduction to Gaussian distribution" https://www.youtube.com/watch?v=iYiOVISWXS4 Accessed on 05/16/2013.

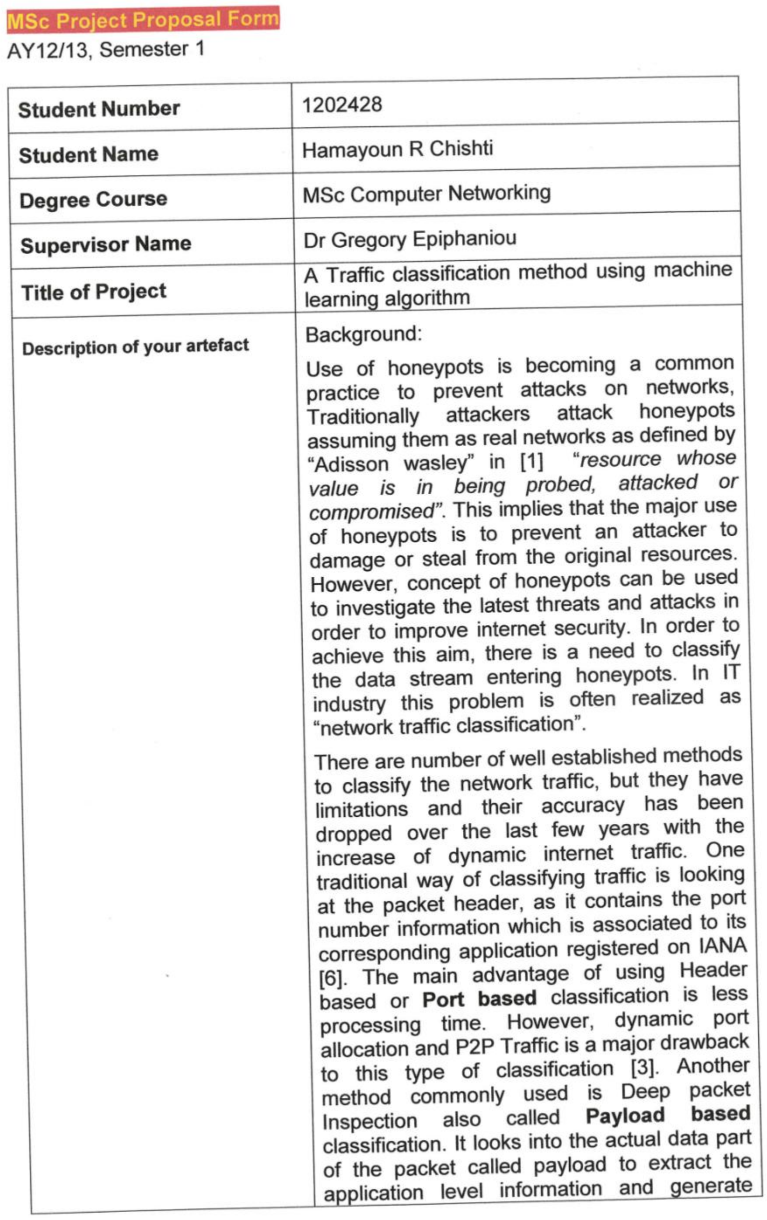
[53] "9.1 Kernel Density Estimation | 9 Unsupervised Learning | Pattern Recognition Class 2012" http://hci.iwr.uni-heidelberg.de/MIP/Teaching/pr/ Accessed on 05/17/2013.

[54] "POLS 506: Bayesian and Nonparametric Stat. - Lecture 7 - Kernel Regression & Density Estimation" https://www.youtube.com/watch?v=30yDDzLGviM Accessed on 5/17/2013.

[55] Jonathan L. Lustgarten, MS, Vanathi Gopalakrishnan, PhD, Himanshu Grover, MS, and Shyam Visweswaran, MD, PhD "Improving Classification Performance with Discretization on Biomedical Datasets" 2008.

[56] "Data processing in Weka" http://maya.cs.depaul.edu/classes/ect584/WEKA/preprocess.html -- WEKA 3.4.1 Accessed on 5/17/2013.

Appendix



MSc Project Proposal Form

AY12/13, Semester 1

Student Number	1202428
Student Name	Hamayoun R Chishti
Degree Course	MSc Computer Networking
Supervisor Name	Dr Gregory Epiphaniou
Title of Project	A Traffic classification method using machine learning algorithm
Description of your artefact	Background: Use of honeypots is becoming a common practice to prevent attacks on networks, Traditionally attackers attack honeypots assuming them as real networks as defined by "Adisson wasley" in [1] "*resource whose value is in being probed, attacked or compromised*". This implies that the major use of honeypots is to prevent an attacker to damage or steal from the original resources. However, concept of honeypots can be used to investigate the latest threats and attacks in order to improve internet security. In order to achieve this aim, there is a need to classify the data stream entering honeypots. In IT industry this problem is often realized as "network traffic classification". There are number of well established methods to classify the network traffic, but they have limitations and their accuracy has been dropped over the last few years with the increase of dynamic internet traffic. One traditional way of classifying traffic is looking at the packet header, as it contains the port number information which is associated to its corresponding application registered on IANA [6]. The main advantage of using Header based or **Port based** classification is less processing time. However, dynamic port allocation and P2P Traffic is a major drawback to this type of classification [3]. Another method commonly used is Deep packet Inspection also called **Payload based** classification. It looks into the actual data part of the packet called payload to extract the application level information and generate

signatures to match those against the stored signatures of well known applications [3]. The main limitation is its heuristic nature. Furthermore packet fragmentation can add complexity to classification process, it can only classify for known applications signatures and it is very difficult to parse all protocols.

Problem Statement:

Applying the concept of investigating the attack and its theory, this idea has been developed to design a Traffic **classification Method** using **Data Mining** techniques at the intersection of Machine learning algorithm, which will classify the normal and malicious traffic. This classification will help to learn about the unknown attacks. The notion of traffic classification is not a new concept; a lot of work has been done in order to classify the network traffic for heterogeneous applications nowadays. Existing techniques (such as payload based, port based, statistical based [3] [18] [20]) have their pros and cons which will be discussed later in literature review, but classification using machine learning techniques [7] is still an open research field for researcher and system engineers providing promising results and accuracy.

Aim:

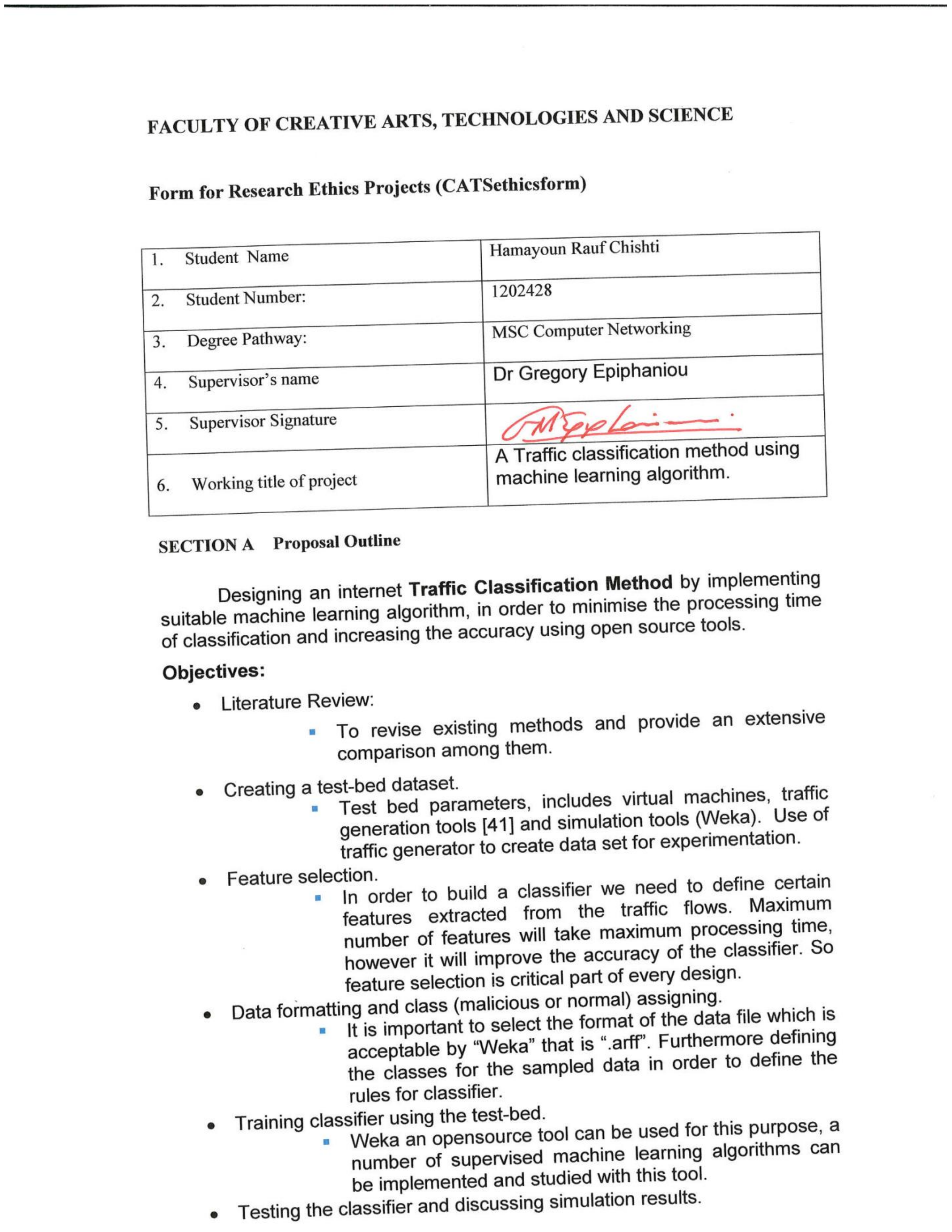
Designing an internet **Traffic Classification Method** by implementing suitable machine learning algorithm, in order to minimise the processing time of classification and increasing the accuracy using open source tools.

Objectives:

- Literature Review:
    - To revise existing methods and provide an extensive comparison among them.
- Creating a test-bed dataset.
    - Test bed parameters, includes virtual machines, traffic generation tools [41] and simulation tools (Weka). Use of traffic

	generator to create data set for experimentation. • Feature selection. ▪ In order to build a classifier we need to define certain features extracted from the traffic flows. Maximum number of features will take maximum processing time, however it will improve the accuracy of the classifier. So feature selection is critical part of every design. • Data formatting and class (malicious or normal) assigning. ▪ It is important to select the format of the data file which is acceptable by “Weka” that is “.arff”. Furthermore defining the classes for the sampled data in order to define the rules for classifier. • Training classifier using the test-bed. ▪ Weka [37] an opensource tool can be used for this purpose, a number of supervised machine learning algorithms can be implemented and studied with this tool. • Testing the classifier and discussing simulation results.
What methodology (structured process) will you be following to realise your artefact?	Adopted methodology will be waterfall also called sequential Model it contains four stages: 1.Requirements 2.Design 3.Implementation 4.Testing Each stage depends on the previous stage

	it is not possible to run them in parallel manner.
How does your project relate to your degree course and build upon the units/knowledge you have studied/acquired	This is purely network security based research project, which directly relates to the knowledge gained from different modules in this course: such as Network Administration Management which covers advance techniques for managing Computer networks, moreover modules like Computer security and Advance security gives a firm understanding of system hardening and security.
Resources	Course books, Peer reviewed articles and papers from databases such as IEEE and ACM will be used as academic research for literature review.
Have you completed & submitted your ethics form?	Yes



FACULTY OF CREATIVE ARTS, TECHNOLOGIES AND SCIENCE

Form for Research Ethics Projects (CATSethicsform)

1. Student Name	Hamayoun Rauf Chishti
2. Student Number:	1202428
3. Degree Pathway:	MSC Computer Networking
4. Supervisor's name	Dr Gregory Epiphaniou
5. Supervisor Signature	
6. Working title of project	A Traffic classification method using machine learning algorithm.

SECTION A Proposal Outline

Designing an internet **Traffic Classification Method** by implementing suitable machine learning algorithm, in order to minimise the processing time of classification and increasing the accuracy using open source tools.

Objectives:

- Literature Review:
  - To revise existing methods and provide an extensive comparison among them.
- Creating a test-bed dataset.
  - Test bed parameters, includes virtual machines, traffic generation tools [41] and simulation tools (Weka). Use of traffic generator to create data set for experimentation.
- Feature selection.
  - In order to build a classifier we need to define certain features extracted from the traffic flows. Maximum number of features will take maximum processing time, however it will improve the accuracy of the classifier. So feature selection is critical part of every design.
- Data formatting and class (malicious or normal) assigning.
  - It is important to select the format of the data file which is acceptable by "Weka" that is ".arff". Furthermore defining the classes for the sampled data in order to define the rules for classifier.
- Training classifier using the test-bed.
  - Weka an opensource tool can be used for this purpose, a number of supervised machine learning algorithms can be implemented and studied with this tool.
- Testing the classifier and discussing simulation results.

SECTION B Check List

Please answer the following questions by circling **YES** or **NO** as appropriate.

1. Does the study involve vulnerable participants or those unable to give informed consent (e.g. children, people with learning disabilities, your own students)?

   YES [✓ **NO**]

2. Will the study require permission of a gatekeeper for access to participants (e.g. schools, self-help groups, residential homes)?

   YES [✓ **NO**]

3. Will it be necessary for participants to be involved without consent (e.g. covert observation in non-public places)?

   YES [✓ **NO**]

4. Will the study involve sensitive topics (e.g. obtaining information about sexual activity, substance abuse)?

   YES [✓ **NO**]

5. Will blood, tissue samples or any other substances be taken from participants?

   YES [✓ **NO**]

6. Will the research involve intrusive interventions (e.g. the administration of drugs, hypnosis, physical exercise)?

   YES [✓ **NO**]

7. Will financial or other inducements be offered to participants (except reasonable expenses or small tokens of appreciation)?

   YES [✓ **NO**]

8. Will the research investigate any aspect of illegal activity (e.g. drugs, crime, underage alcohol consumption or sexual activity)?

   YES [✓ **NO**]

9. Will participants be stressed beyond what is considered normal for them?

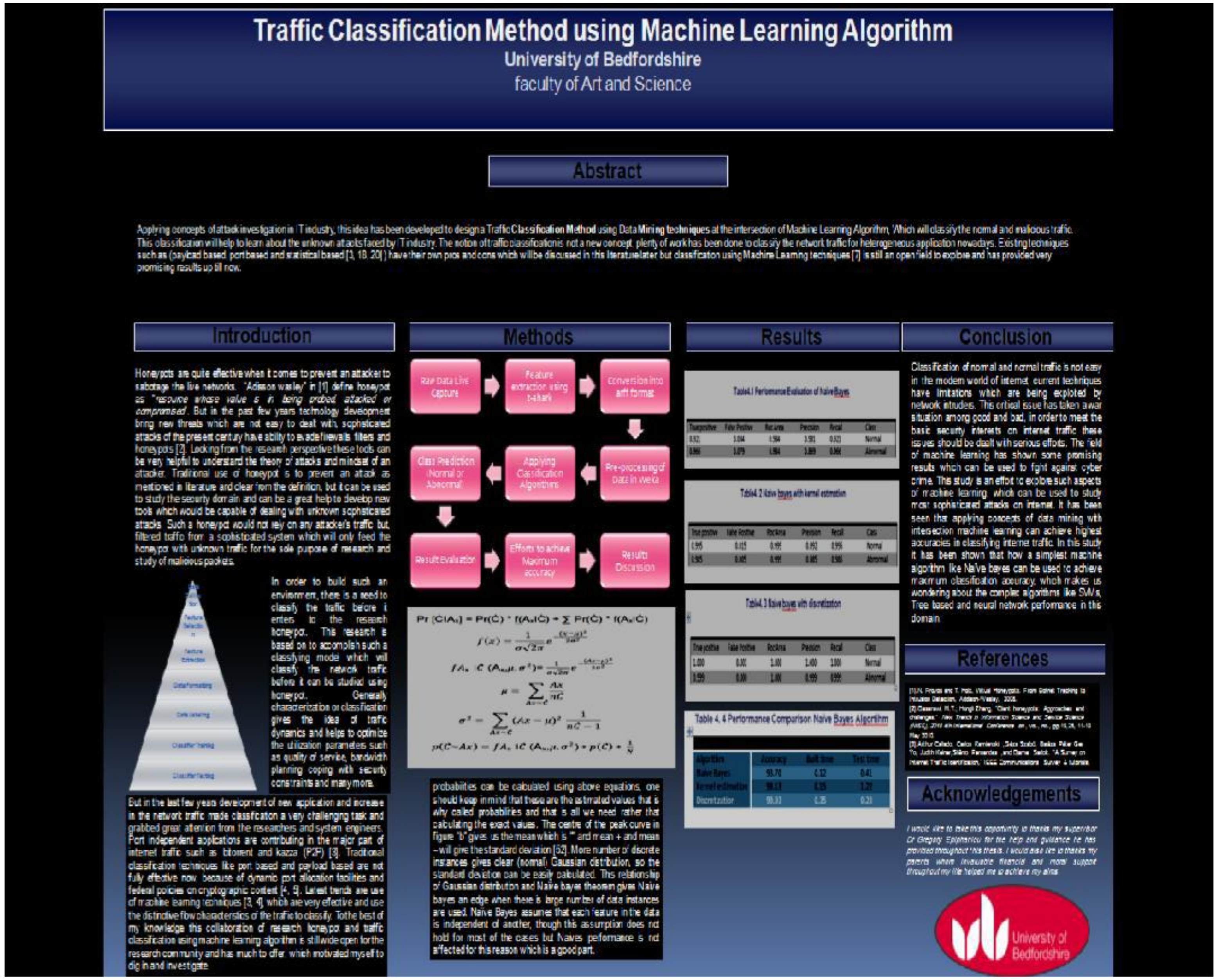
   YES [✓ **NO**]

10. Will the study involve participants from the NHS (patients or staff) or will data be obtained from NHS premises?

    YES [✓ **NO**]

If the answer to any of the questions above is "Yes", or if there are any other significant ethical issues, then further ethical consideration is required. Please document carefully how these issues will be addressed.

Signed (student): Hamayoun Date: 17/04/2013
Countersigned (Supervisor): Date: 8/03/2017


Raw Data Live Capture
Feature extraction using t-shark
Conversion into arff format
Pre-processing of Data in Weka
Applying Classification Algorithms
Class Prediction (Normal or Abnormal)
Result Evaluation
Efforts to achieve Maximum accuracy
Results Discussion
Feature Selection
Feature Extraction
Data Formatting
Data Labeling
Classifier Training
Classifier Testing

Traffic Classification Method using Machine Learning Algorithm

University of Bedfordshire
faculty of Art and Science

Abstract

Applying concepts of attack investigation in IT industry, this idea has been developed to design a Traffic **Classification Method** using Data **Mining techniques** at the intersection of Machine Learning Algorithm, Which will classify the normal and malicious traffic. This classification will help to learn about the unknown attacks faced by IT industry. The notion of traffic classification is not a new concept plenty of work has been done to classify the network traffic for heterogeneous application nowadays. Existing techniques such as (payload based, port based and statistical based [1, 18, 20]) have their own pros and cons which will be discussed in this literature later but classification using Machine Learning techniques [7] is still an open field to explore and has provided very promising results up till now.

Introduction

Honeypots are quite effective when it comes to prevent an attacker to sabotage the live networks. "Adisson wasley" in [1] define honeypot as *"resource whose value is in being probed, attacked or compromised"*. But in the past few years technology development bring new threats which are not easy to deal with, sophisticated attacks of the present century have ability to evade firewalls filters and honeypots [2]. Looking from the research perspective these tools can be very helpful to understand the theory of attacks and mindset of an attacker. Traditional use of honeypot is to prevent an attack as mentioned in literature and clear from the definition, but it can be used to study the security domain and can be a great help to develop new tools which would be capable of dealing with unknown sophisticated attacks. Such a honeypot would not rely on any attacker's traffic but, filtered traffic from a sophisticated system which will only feed the honeypot with unknown traffic for the sole purpose of research and study of malicious packets.

In order to build such an environment, there is a need to classify the traffic before it enters to the research honeypot. This research is based on to accomplish such a classifying model which will classify the network traffic before it can be studied using honeypot. Generally characterization or classification gives the idea of traffic dynamics and helps to optimize the utilization parameters such as quality of service, bandwidth planning coping with security constraints and many more.

But in the last few years development of new application and increase in the network traffic made classification a very challenging task and grabbed great attention from the researchers and system engineers. Port independent applications are contributing in the major part of internet traffic such as bitorrent and kazza (P2P) [3]. Traditional classification techniques like port based and payload based are not fully effective now because of dynamic port allocation facilities and federal policies on cryptographic content [4, 5]. Latest trends are use of machine learning techniques [3, 4] which are very effective and use the distinctive flow characteristics of the traffic to classify. To the best of my knowledge this collaboration of research honeypot and traffic classification using machine learning algorithm is still wide open for the research community and has much to offer which motivated myself to dig in and investigate.

Methods

$$\Pr[C|A_i] = \Pr(\dot{C}) * f(A_i|\dot{C}) + \sum \Pr(\dot{C}) * f(A_i|\dot{C})$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$fA_x\ C\ (A_{x};\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(Ax-\mu)^2}{2\sigma^2}}$$

$$\mu = \sum_{Ax=c} \frac{Ax}{nC}$$

$$\sigma^2 = \sum_{Ax=c} (Ax-\mu)^2 \frac{1}{nC-1}$$

$$p(C \sim Ax) = fA_x \setminus C\ (A_{x};\mu,\sigma^2) * p(C) * \frac{1}{N}$$

probabilities can be calculated using above equations, one should keep in mind that these are the estimated values that is why called probabilities and that is all we need rather that calculating the exact values. The centre of the peak curve in figure "b" gives us the mean which is " and mean + and mean – will give the standard deviation [52]. More number of discrete instances gives clear (normal) Gaussian distribution, so the standard deviation can be easily calculated. This relationship of Gaussian distribution and Naïve bayes theorem gives Naïve bayes an edge when there is large number of data instances are used. Naïve Bayes assumes that each feature in the data is independent of another, though this assumption does not hold for most of the cases but Naives peformance is not affected for this reason which is a good part.

Results

Table4.1 Performance Evaluation of NaiveBayes

True positive	False Positive	Roc Area	Precision	Recall	Class
0.92	0.034	0.984	0.951	0.921	Normal
0.966	0.079	0.984	0.889	0.966	Abnormal

Table4. 2 Naive bayes with kernal estimation

True positive	False Positive	Roc Area	Precision	Recall	Class
0.985	0.015	0.995	0.992	0.996	Normal
0.985	0.005	0.995	0.985	0.986	Abnormal

Table4. 3 Naive bayes with discretization

True positive	False Positive	Roc Area	Precision	Recall	Class
1.000	0.001	1.000	1.000	1.000	Normal
0.999	0.000	1.000	0.999	0.999	Abnormal

Table 4. 4 Performance Comparison Naive Bayes Algorithm

Algorithm	Accuracy	Built time	Test time
Naive Bayes	93.70	0.12	0.41
Kernel estimation	98.13	0.15	1.20
Discretization	99.33	0.25	0.21

Conclusion

Classification of normal and normal traffic is not easy in the modern world of internet current techniques have limitations which are being exploited by network intrudes. This critical issue has taken a war situation among good and bad, in order to meet the basic security interests on internet traffic these issues should be dealt with serious efforts. The field of machine learning has shown some promising results which can be used to fight against cyber crime. This study is an effort to explore such aspects of machine learning which can be used to study most sophisticated attacks on internet. It has been seen that applying concepts of data mining with intersection machine learning can achieve highest accuracies in classifying internet traffic. In this study it has been shown that how a simplest machine algorithm like Naïve bayes can be used to achieve maximum classification accuracy, which makes us wondering about the complex algorithms like SVM's, Tree based and neural network performance in this domain.

References

[1] N. Provos and T. Holz, Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Addison-Wesley, 2008.

[2] Qassrawi, M.T.; Hongli Zhang, "Client honeypots: Approaches and challenges," *New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on*, vol., no., pp.19,26, 11-13 May 2010.

[3] Arthur Callado, Carlos Kamienski, Géza Szabó, Balázs Péter Gerő, Judith Kelner, Stênio Fernandes, and Djamel Sadok, "A Survey on Internet Traffic Identification," IEEE Communications Surveys & tutorials

Acknowledgements

I would like to take this opportunity to thanks my supervisor Dr Gregory Epiphaniou for the help and guidance he has provided throughout this thesis. I would also like to thanks my parents whose invaluable financial and moral support throughout my life helped me to achieve my aims.

University of Bedfordshire