ABSTRACT

With drastic increase in number of internet users and thereby internet traffic over the last few years, the IP traffic classification has gained significant importance for the research community as well as for various internet service providers for optimization of their network performance and the governmental intelligence organizations. Today, traditional IP traffic classification techniques such as port number and payload based direct packet inspection techniques are rarely used because of use of dynamic port number instead of static port number in packet headers. Current trends are using Machine Learning (ML) techniques for IP traffic classification.

Real time internet traffic dataset has been developed by capturing information in the form of audio, video, text data using packet capturing tool and pre-processed for removing redundant features and modifying the dataset in order to be trained and tested. Five ML algorithms SVM, KNN, Random Forest, Decision Tree and Naïve Bayes are employed for IP traffic classification for dataset. The experimental analysis shows that Random Forest is an effective ML technique for near real time and online IP traffic classification with reduction in packet capture duration and reduction in number of features characterizing each application sample with Correlation based Feature Selection Algorithm.

Organization of Thesis

LIST OF ABBREVIATIONS:

IP	:	Internet Protocol
DPI	:	Deep Packet Inspection
KNN	:	K-Nearest Neighbor
SVM	:	Support Vector Machine
TCP	:	Transport Control Protocol
UDP	:	User data Protocol
NAPT	:	Network Address rotocol Traslation
GUI	:	Graphical User Interface
AI	:	Artificial Intelligence
SSH	:	Secure Shell
ML	:	Machine Learning
SMTP	:	Simple Mail Transfer Protocol
FTP	:	File Transfer Protocol
HTTP	:	Hyper Text Transfer Protocol
DHCP	:	Dynamic Host Configaration Protocol
API	:	Application Program Interface
PPP	:	Point to Point Protocol
LTE	:	Long Term Evolution
CSV	:	Comma Separated Value
ASCII	:	American Standard Code for Information Exchange
ARP	:	Address Resolution Protocol
ICMP	:	Internet Control Message Protocol
TFTP	:	Trivial File Transfer Protocol
BOOTP	:	Bootstrap Protocol
SNMP	:	Simple Network Management Protocol
SLIP	:	Serial Line Internet Protocol
DNS	:	Domain Name System

CHAPTER 1
INTRODUCTION

Network traffic classification creates challenge for the complete network, by increase in the load of network which leads to decrease in efficiency of the system. Each network carries data for different applications, which have different usage. Therefore the information about the quality level provided requires knowledge of what kind of data is flowing in the network at the present time. Generally, traffic classification methods use a concept of flow defined as a group of packets having the same end IP addresses, using the same transport protocol, and its port numbers. Flows are considered bidirectional-packets going from the local machine to the remote server and from the remote server to the local machine are part of the same flow. Using application ports for traffic classification is a very simple idea and is widely used by network administrators to limit traffic generated by worms and unwanted services. This method is fast and can be applied to almost all the routers existing in the market. This method is good to classify some protocols operating on fixed port numbers, but this method is not applicable for dynamic ports e.g skype, gaming etc. because it operates on dynamic port number and one cannot detect that application. Deep Packet Inspection (DPI) solutions are quite slow and they relay on inspecting the user data and therefore privacy and confidentiality issues can appear because user data is always in private manner existing methods like Decision tree, Random Forests, Naive Bayes, KNN and SVM have much wider coverage. They can be used in any point of the network, providing very fast statistical detection of the application, to which the traffic belongs. Achievable detection rate correctness is over 95. The goal of machine learning is to design and develop algorithms that allow systems to use empirical data, experience, and training to evolve and adapt to changes that occur in their environment. A major focus of Machine Learning research is to automatically induce models, such as rules and patterns, from the training data it analyses.

There are many classification methods of network traffic, they all try to classify network traffic accurately but precision of classification is less. Classification based on well-known TCP or UDP ports is becoming increasingly less effective when growing numbers of networked applications have port-mobility(allocating dynamic ports as needed), end users are deliberately using non-standard ports to hide their traffic, and use of network address port translation (NAPT) is widespread .This approach is limited by the fact that classification rules must be updated whenever an application implements even a trivial protocol change, and privacy laws and encryption can effectively make the payload inaccessible. In existing tool like Wireshark, it is a cross-platform analyser that does deep inspection of hundreds of protocols. It does live capture and capture save (for offline browsing), which can be viewed in Graphical User Interface. When we use wireshark, we have to stop all running applications.. In our proposed system we are using accurate data sets for both training and testing the boosted classifier. The classifier is able to distinguish traffic which appears to be similar, like web browser traffic, audio and video streamed via a web page.

CHAPTER 2
MACHINE LEARNING

2.1 What is Machine Learning?

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

2.2 Importance of Machine Learning:

Machine learning is needed for tasks that are too complex for humans to code directly. Some tasks are so complex that it is impractical, if not impossible, for humans to work out all of the nuances and code for them explicitly. So instead, we provide a large amount of data to a machine learning algorithm and let the algorithm work it out by exploring that data and searching for a model that will achieve what the programmers have set it out to achieve.

How well the model performs is determined by a cost function provided by the programmer and the task of the algorithm is to find a model that minimises the cost function.

The most intuitive examples of this are the various applications for reinforcement learning, such as self driving cars and self-flying helicopters. In reinforcement learning the algorithm is trained like a dog, receiving rewards when it gets things right and punishments when it gets it wrong. This method was used by Stanford students nearly a decade ago to train a radio controlled helicopter to do all sorts of amazing stunts, including flips and hovering upside down. You simply can’t explicitly code that sort of thing because there are too many variables to consider and too many nuances in the model.

Learning problems fall into a few categories:

- **Supervised Learning:** In supervised learning, the data comes with additional attributes that we want to predict. This problem can be either:
    - **<u>Classification</u>:** samples belong to two or more classes and we want to learn from already labelled data how to predict the class of unlabelled data. An example of a classification problem would be handwritten digit recognition, in which the aim is to assign each input vector to one of a finite number of discrete categories. Another way to think of classification is as a discrete (as opposed to continuous) form of supervised learning where one has a limited number of categories and for each of the n samples provided, one is to try to label them with the correct category or class.
    - **<u>Regression</u>:** If the desired output consists of one or more continuous variables, then the task is called regression. An example of a regression problem would be the prediction of the length of a salmon as a function of its age and weight.

- **Unsupervised learning:** Unsupervised learning, in which the training data consists of a set of input vectors x without any corresponding target values. The goal in such problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization.

CHAPTER 3

DIFFERENT CLASSIFIERS

Types of classification algorithms in machine learning:

1. Linear Classifiers: Logistic Regression, Naive Bayes Classifier
2. Support Vector Machines
3. Decision Trees
4. Random Forest
5. K-Nearest Neighbor

In this project we are some of these algorithms to acquire the required results.

3.1 RANDOM FOREST:

The Random forest algorithm is a supervised learning algorithm. It can be used both for classification as well as regression. It is the most flexible and easy to use algorithm. In general, a forest consists of trees.

It is said to be more robust when it has more number of trees. Decision trees are usually created on randomly selected data. It samples, gets prediction from each tree. Later, it selects the best solution by means of voting. It is a pretty good indicator of the feature importance. Random forest algorithm has wide range of applications and among them few were recommendation engines, image classification and feature selection. It is used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which helps in selects important features in a dataset.
According to layman's this algorithm can be explained by the below example. For suppose you want to go on a trip and visit a place so that you will enjoy. So what do you do to find a place that you will like? You have many options like search online, read reviews on travel blogs and portals, or you can also ask your friends. Let's suppose you have decided to ask your friends and you made a discussion about their previous travel experiences. We will also be getting few recommendations from your friends and now you have to make a list of all

those recommended places. Then, you ask them to vote from the list of recommended places you made in order to select the best place. The place with good number of votes will be your final choice for the trip.

From the above discussion its clearly understood that there are two parts. First, asking your friends about their individual travel experience and getting recommendations on the places they have visited. This part uses the decision tree algorithm. Here, each friend makes a selection of the places he has visited so far.

The second part is the voting procedure. Here after collecting all the recommendations the voting has to be held in order to select the best place in the list of recommendations. This entire process of getting recommendations from friends and then voting on them to find the best place is known as the random forest algorithm. It is technically an ensemble method (based on the divide-and-conquer approach) of decision trees generated on a randomly split dataset. A forest is a collection of decision tree classifier. Each decision tree is generated using an attribute selection indicator such as information gain, gain ratio, and Gini index for each attribute. Each tree depends on an independent random sample. In a classification problem, each tree votes and final result comprises of the most popular class is chosen. In the case of regression, the average of all the tree outputs is considered as final result. It is simpler and more powerful when compared to other non-linear classification algorithms.
It includes four steps:

1. Selection of random samples from a given dataset.
2. Construction of   decision tree for each sample and knowing the prediction result from each decision tree
3. Voting have to be done for each predicted result.
4. Selection of the prediction result with the best number of votes as the final prediction.

Advantages:

- This algorithm is considered as a highly accurate and robust method because of the number of decision trees participating in the process.
- It does not suffer from the overfitting problem. The very reason is that it takes the average of all the predictions, which cancels the biases.
- Random Forest algorithm can be used in both classification and regression problems.

- It can also handle missing values. There are two ways to handle these: replacing continuous variables can be done using median values and computing the proximity-weighted average of missing values.
- We obtain the relative feature importance, which helps in selecting the most contributing features for the classifier.

Disadvantages:

- As it has multiple decision trees it is slow in generating predictions. Whenever it makes a prediction, all the trees in the forest for the same given input has to make prediction and then perform voting on it. This is a time-consuming process.
- When compared to decision tree this model is difficult to interpret, where you can easily make a decision by following the path in the tree.

3.2 Support Vector Machine (SVM):

Support vector machine is a machine learning algorithm mostly used for classification (classifies the data according to the category) and rarely for regression (statistical method to find relation between variables). In this method we plot each data item as a point in n-dimensional plane. Then we find hyper plane and perform classification

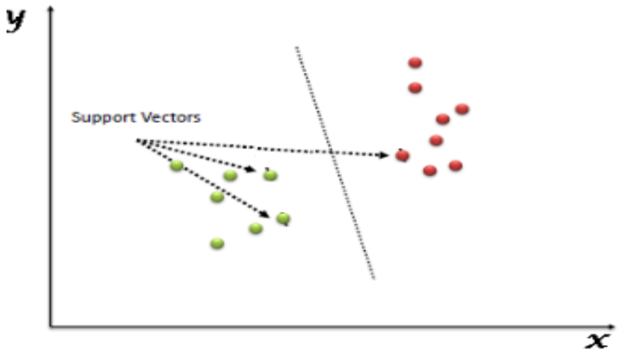


Figure 3.1:Data points in SVM classifier.

Support Vectors are coordinates of each individual observation whereas Support Vector Machine is frontier that segregates the classes

Working:

Identify the right hyper plane:

We need to identify the right hyper plane.

Select the right hyper plane which segregates the two classes accurately

Classify the classes in below scenario:

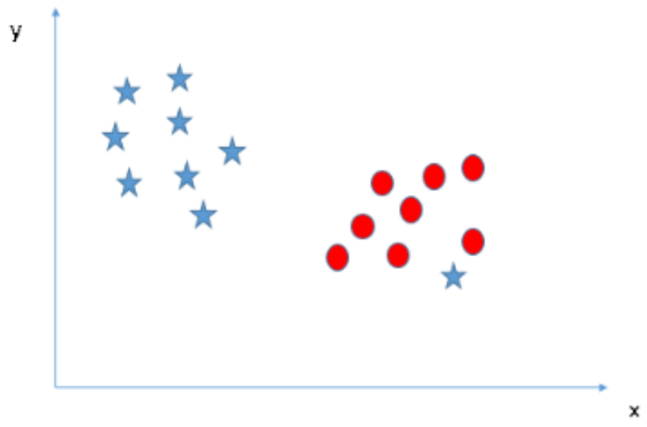


Figure 3.3: Data to be classified by SVM classifier

In the above scenario the straight line does segregate the classes accurately because one of the classes in in another territory .But Support Vector Machine as a feature to ignore outliners and segregate classes with maximum margin so we can say that Suppport Vector Machine in robust to outliners.

The figure below segregates the classes by avoiding the ouliners.

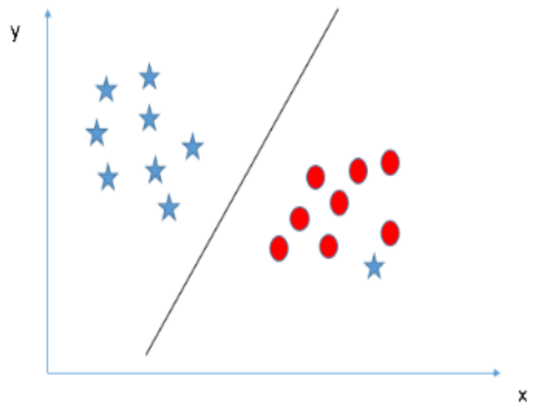


Figure 3.4:data classified after SVM classification

Advantages:

- It works well with clear margin of separation.
- It is has effective separation in high dimensional spaces.
- It is effective in samples where number of dimensions is greater than the number of samples.
- It uses a subset of training points in the decision function, so it is also memory efficient.

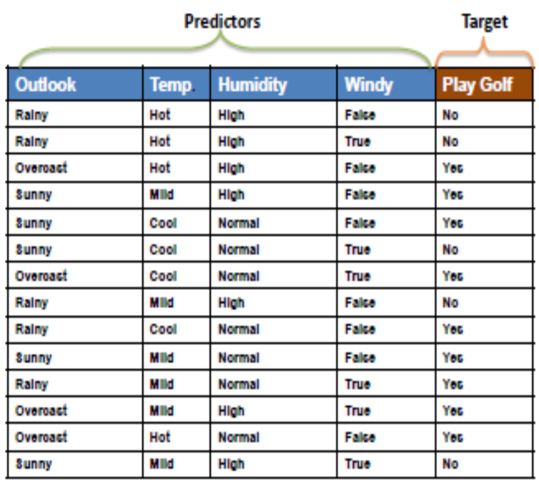
Disadvantages:

- It doesn’t perform well, when we have large data set
- It also doesn’t perform very well, when the data set has more noise (target classes are overlapping)
- It doesn’t directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

3.3 DECISION TREE CLASSIFIER:

Decision tree algorithm is one of the classification algorithms which belong to the family of supervised learning algorithms. Decision tree algorithm is used for regression and classification problems too, unlike every other supervised algorithm.

The understanding level of Decision Trees algorithm is so easy compared with other classification algorithms in supervised learning. It breaks down a dataset into smaller subsets which may be again broke down into subsets. At the same time an associated decision tree is incrementally developed. The result of the final decision tree is a tree with a root node, decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label. Decision tree algorithm can handle both categorical and numerical data.


Predictors
Target

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

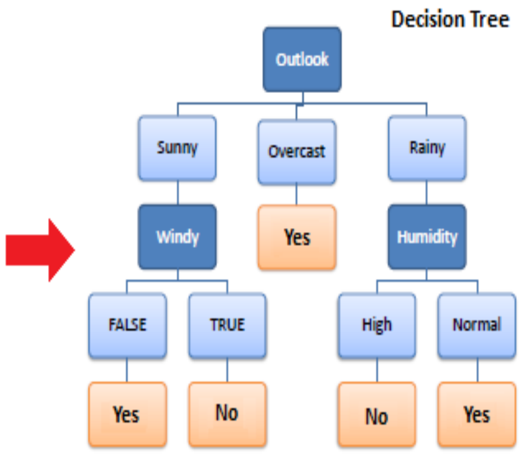


Table1

Decision Tree Algorithm:

In machine learning, decision trees were being used for decades as easily understandable classifiers. The most important and well-being used decision tree algorithms among a number of decision tree algorithms are

- Iterative Dichotomiser 3 (ID3)
- C4.5
- CART

ID3, C4.5, and CART all adopt a top-down, recursive, divide-and-conquer approach to decision tree induction.

The core algorithm for building decision tree algorithm is ID3 by J.R.Quinlan which employs greedy search through the space of possible branches with no backtracking.
If a dataset consists of n attributes, the most complicated test is deciding which attribute to be placed as the root node or at the different levels as internal nodes. We can't solve the issue by randomly selecting any node. That random selection of nodes leads to low accuracy of results. The popular attribute selection measures are Entropy, Information Gain and gini index. These measures will calculate the values for each attribute in the dataset. The values are sorted and attributes are placed at the root following the order. To construct decision tree,

ID3 uses entropy and Information Gain.C4.5 algorithm uses gain ratio and CART algorithm uses gini index.

Advantages:

- Decision tree follows same approach that humans follow while making decisions.
- Decision Trees results in set of rules which are easy to explain and understand.
- Interpretation of a complex Decision Tree model can be simplified by its visualizations.

Disadvantages:

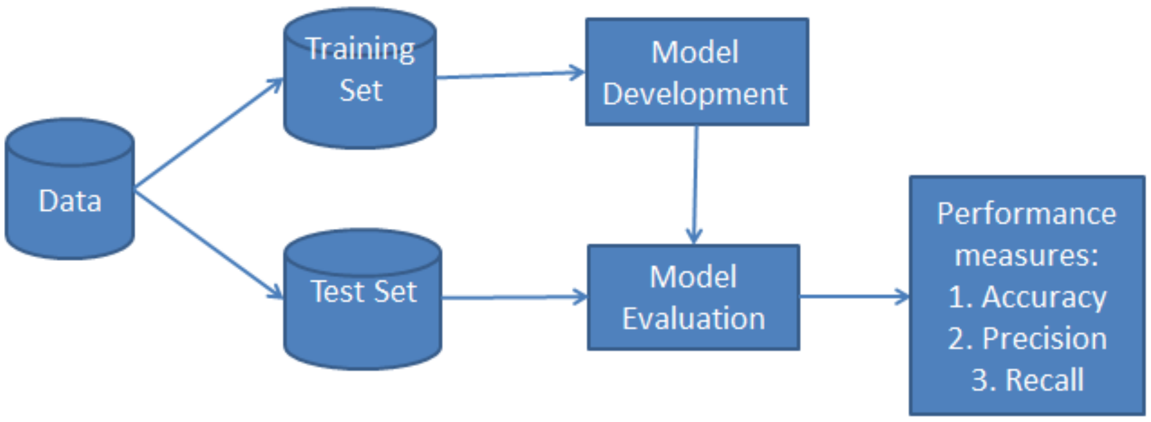
- There is a high probability of overfitting in Decision Tree.
- Information gain in a decision tree with categorical variables gives a biased response for attributes with more no. of categories.
- With more class labels, calculations become very complex.

3.4 NAIVE BAYES:

Naive Bayes is one of the most straightforward and fast classification algorithms, which is suitable for a large chunk of data. This classifier is successfully used in various applications such as spam filtering, text classification, sentiment analysis, and recommender systems. For prediction of unknown class, it uses Bayes theorem probability

Workflow:

While performing classification, the main step is to understand the problem and identify potential features and label. Features are those characteristics which affect the results of the label. Let's consider an example in the case of a loan distribution, bank manager's identify customer's occupation, income, age, location, previous loan history, transaction history, and credit score. All these attributes are known as features that help the model classify customer. The classification has two phases. First, learning phase, and the second evaluation phase. classifier trains its model on a given dataset in the learning phase and it tests the classifier performance in the evaluation phase. Calculation of performance can be done on the basis of various parameters such as accuracy, error, precision, and recall.



Definition: Naive Bayes Classifier

Naive Bayes is a statistical classification technique which is based on Bayes Theorem. It is the simplest supervised learning algorithms. Compared to others Naive Bayes classifier is the fast, accurate and reliable algorithm. These classifiers have high accuracy and speed on large datasets.

Naive Bayes classifier assumes that particular feature's effect in a class is independent of other features. Let us consider an example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location. Although these features are interdependent, these features are still considered independent, this assumption simplifies computation, and hence it is considered as naive. This assumption is named as class conditional independence.

$$P(h|D)=P(D|h)*P(h)/P(D)$$

- P(h): probability of hypothesis (h) being true (regardless of the data). It is known as the prior probability of h.
- P(D): probability of data (regardless of the hypothesis). It is known as the prior probability.
- P (h |D): probability of hypothesis(h) given the data D. It is known as posterior probability.
- P (D| h): probability of data (d) given that the hypothesis(h) was true. It is known as posterior probability.

Advantages:

- It is a simple, fast and accurate approach
- Low computation cost.
- Works efficiently on a large dataset.
- When compared to the continuous variable, it performs well in case of discrete response variable
- Used in multiple class prediction problems.
- Performs well in case of text analytics problems.
- Naive Bayes classifier performs better compared to other models like logistic regression.

Disadvantages:

- The features are assumed to be independent.
- Zero posterior probability occurs if there is no tuple. In such case, the model is unable to make predictions and is known as Zero Probability/Frequency Problem.

3.5 K-NEAREST NEIGHBORS ALGORITHM:

The k-nearest neighbors is a simple algorithm. It is easy to implement supervised machine learning algorithm. It is used to solve both regression and classification problem

This algorithm takes data and differentiates new data points based on a similarity. It may consider distance measurements. Classification is done majorly due to its neighbors. The data is allocated to the class which has the nearest neighbors. If you increase the number of nearest neighbors, then value of k accuracy increases. Similar things are near to each other is it's an assumption.

Predictions Using KNN :

- It makes predictions using the training dataset directly.
- Prediction is made for a new data (x) by searching the whole training data set for the K most similar data and gets the output variable for those K instances.
- For regression it calculates mean output variable while classification this might be the mode or most common class value.

- To determine which of the K instances in the training set are similar to a new input any distance measure can be used.
- For real value input variables, the most used distance measure is Euclidean distance.

  Euclidean distance is calculated by the square root of the sum of the squared differences between a new point (x) and the existing point (xi) across all input attributes j.

Euclidean Distance(x, xi) = sqrt(sum ((xj– xi)^2))

Other popular distance measures include:

- Hamming Distance: it is based on calculating the distance between binary vectors
- Manhattan Distance: it calculates the distance between real vectors using the sum of their absolute difference. It is also known as the City Block Distance.
- Minkowski Distance: it is combination of both Euclidean and Manhattan distance.

You can use the best distance measure based on the properties of your data. If you are unsure, you can test with different distance metrics and different values of K together and see the results and take most accurate models.

Euclidean distance is a good distance measure which we can use if the input variables are similar in type for example for all measured widths and heights. Manhattan distance is better if the input variables are not similar in type.

Parameter Selection:

- The best value of k depends upon the data. Generally, larger k values decrease effect of the noise on the classification. But it makes the boundaries between classes less distinct.
- The special case is that the class is predicted to be the class of the closest training sample when k = 1 is known as the nearest neighbor algorithm.
- The accuracy of the KNN algorithm can be decreased by noisy or irrelevant features or if feature scales are not consistent with their importance.
- In binary classification problems, it is better to choose k to be an odd number.
- One best way of choosing the optimal k in this setting is using bootstrap method.
- The value for K can be get by algorithm varying. It is better to try different values for K from 1 to 21 and see for better accuracy.

KNN for Regression :

When KNN is used for regression problems, the output is mostly based on the mean or the median of the k most similar instances.

KNN for Classification :

When KNN is used for classification, the output is calculated as the class having the most frequency from k most similar instances. The class with the most repeated is taken as the prediction.

Class probabilities is calculated as the normalized frequency samples that belongs to the class in the set of the k most closest instances for a new data. For a binary classification problem (class is 0 or 1) :-

p(class=0) = count(class=0) / (count(class=0)+count(class=1))

If you use k and it has an even number of classes. it is a better to choose a k value with an odd number to avoid a tie. Similarly, use an even number for K when you have an odd number of classes.

ADVANTAGES

- It is easy to implement.
- This algorithm is so simple.
- It is not necessary to build a model, change several parameters, or make additional assumptions.
- It is used for classification, regression, and search.

DISADVANTAGES:

- The algorithm gets slower if the number of examples or independent variables increase.
- Performance of this algorithm depends on the number of dimensions .

CHAPTER 4
4G LTE AND WIRESHARK APPLICATION

4.1 4G LTE:

LTE is a 4G wireless communications standard developed by the 3rd Generation Partnership Project designed to provide up to ten times the speed of the 3G networks for devices such as smart phones, wireless hotspots, tablets etc. 4G technologies are designed to provide IP-based data, voice and multimedia with speed of atleast 100 Mega Bits Per Second and also upto as fast as 1 Giga Bit Per Second.

Procotol:

Protocol is a set of rules or procedures for transmitting data between electronic devices, such as computers. In order for computers to exchange information, there must be a pre-existing agreement as to how the information will be structured and how each side will send and receive it. Without a protocol, a transmitting computer, for example, could be sending its data in 8-bit packets while the receiving computer might expect the data in 16 bit packets.

Protocols are established by international or industry wide organizations. Perhaps the most important computer protocol is OSI (Open Systems Interconnection), a set of guidelines for implementing networking communications between computers

TCP/IP:

Internet Protocol (IP) is the principal set (or communications protocol) of digital message formats and rules for exchanging messages between computers across a single network or a series of interconnected networks, using the Internet Protocol Suite (often referred to as TCP/IP). Messages are exchanged as datagrams, also known as data packets or just packets.

HTTP:

HyperText Transfer Protocol (HTTP) is an application-layer protocol used primarily on the World Wide Web. HTTP uses a client-server model where the web browser is the client and communicates with the webserver that hosts the website. The browser uses HTTP, which is carried over TCP/IP to communicate to the server and retrieve Web content for the user. HTTP is a widely used protocol and has been rapidly adopted over the Internet because of its simplicity.

DNS:

Domain name system (DNS) is a hierarchical naming system built on a distributed database. This system transforms domain names to IP addresses and makes it possible to assign domain names to groups of Internet resources and users, regardless of the entities' physical location.

SMTP:

Simple Mail Transfer Protocol (SMTP) is the standard protocol for email services on a TCP/IP network. SMTP provides the ability to send and receive email messages. SMTP is an application-layer protocol that enables the transmission and delivery of email over the Internet. SMTP is created and maintained by the Internet Engineering Task Force (IETF).

Simple Mail Transfer Protocol is also known as RFC 821 and RFC 2821.

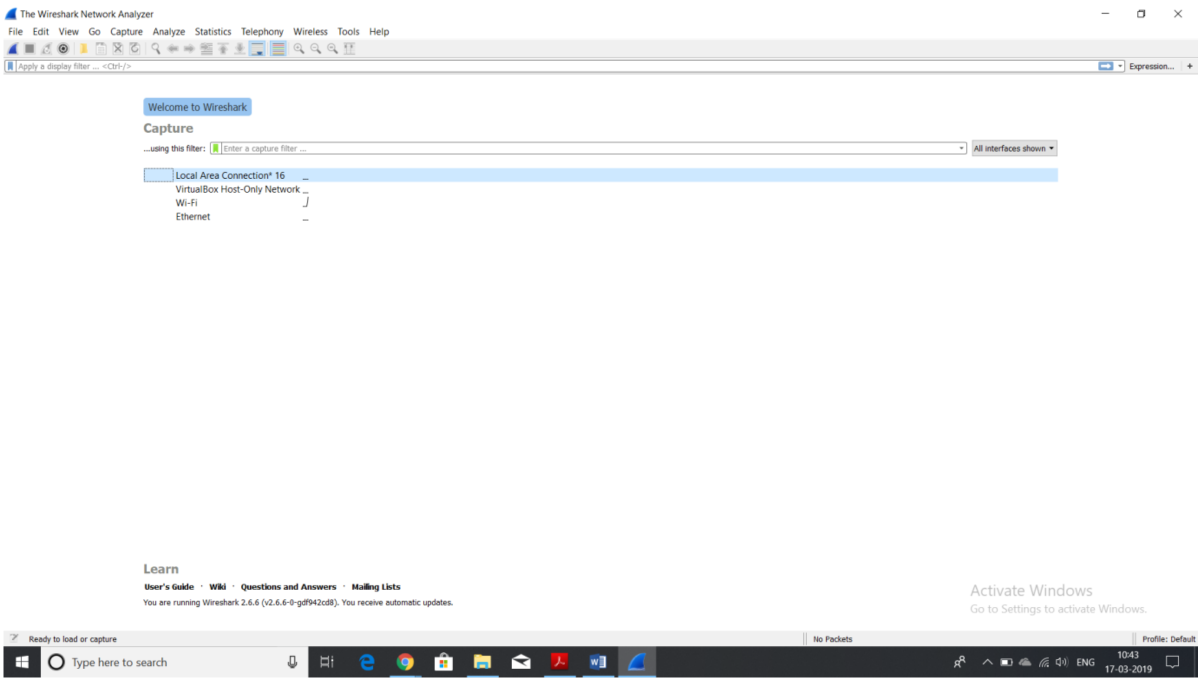
Network Traffic:

Network traffic tells about the amount of data moving across a network at a given point of time. The proper organization of network traffic helps in ensuring the quality of service in a given network.

Network traffic is also called as data traffic. Network traffic is the main component for bandwidth measurement and management. Moreover, various topologies of the network can only be implemented based on the amount of network traffic in the system.

4.2 WIRESHARK:

Wireshark is free open-source network protocol analyser. It captures real time network packets and displays them in human readable format. It has many advanced features including live capture, offline analysis, three-pane packet browsing, colour rules for analysis. It is used for communication protocol analysis and troubleshooting. Here we are using Wireshark for data capture and make a dataset through it.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
Activate Windows
Go to Settings to activate Windows.
Ready to load or capture
No Packets
Profile: Default
Type here to search
ENG
10:43
17-03-2019

Local Area Connection* 16
VirtualBox Host-Only Network
Wi-Fi
Ethernet

You are running Wireshark 2.6.6 (v2.6.6-0-gdf942cd8). You receive automatic updates.

Figure 4.1: Wireshark in windows 10

Figure 4.1 is User interface after installation in windows 10 and here we have to select the required interface so that we can capture flow of data through that network.

Background:

TCP/IP Network Stack

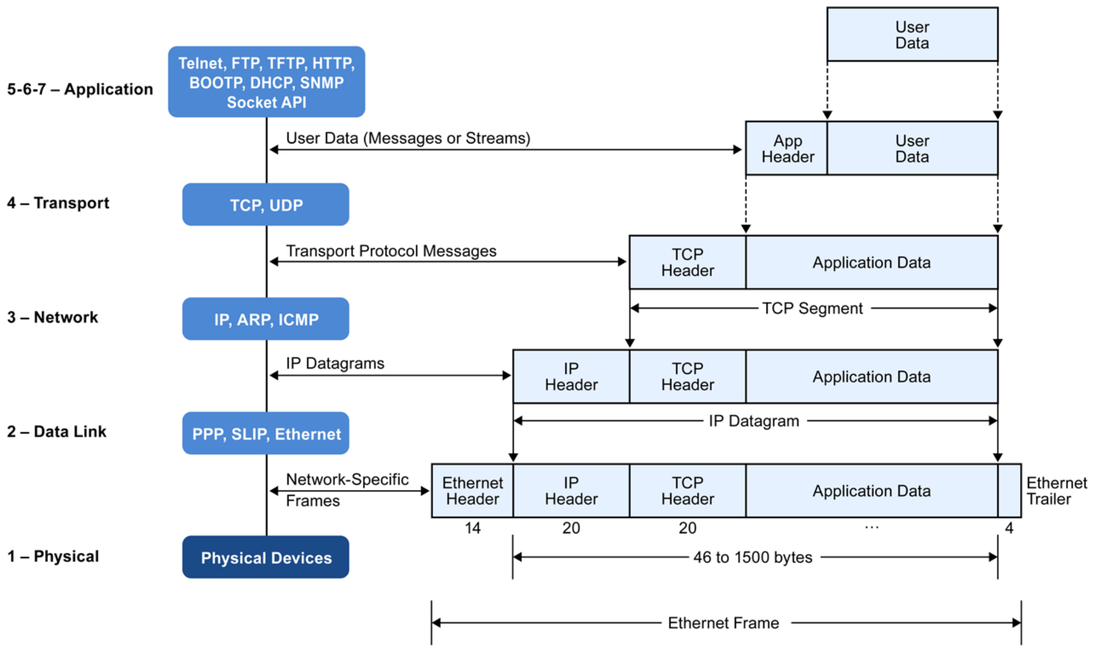


Figure 4.2: Encapsulation of Data in the TCP/IP network stack

TCP/IP(Transmission Control Protocol/Internet Protocol) protocol is the most commonly used network model for internet services. These were the first defined network protocols defined in the standard. It has many layers that include Application layer, Transport layer, Network layer and Datalink layer.

- Application layer: This includes the protocols used by most applications for providing user services. Examples of application layer protocols are Hyper Text Transfer Protocol(HTTP), Secure Shell(SSH), File Transfer Protocol(FTP) and Simple Mail Transfer Protocol(SMTP).

- Transport Layer: Transport layer establishes process to process connectivity and provides end to end services that are independent of underlying user data. To implement process to process communication the protocol introduces a concept of port. Examples of transport protocol are TCP, User Datagram Protocol (UDP). TCP provides flow control and connection establishment and reliable transmission of data, while UDP is a connectionless Transmission model

- Internet Layer: This layer is responsible for sending packets across networks. It has 2 functions
  1) Host identification using IP addressing System
  2) Packet routing from source to Destination. Examples of Internet layer protocols are Internet Protocol, internet Control Message protocol and Address Resolution Protocol (ARP).

- Link Layer: The link layer defines the networking methods within the scope of the local network link. It is used to move the packets between two hosts on the same link. Common example of link layer protocols is Ethernet.

Packet Sniffer:

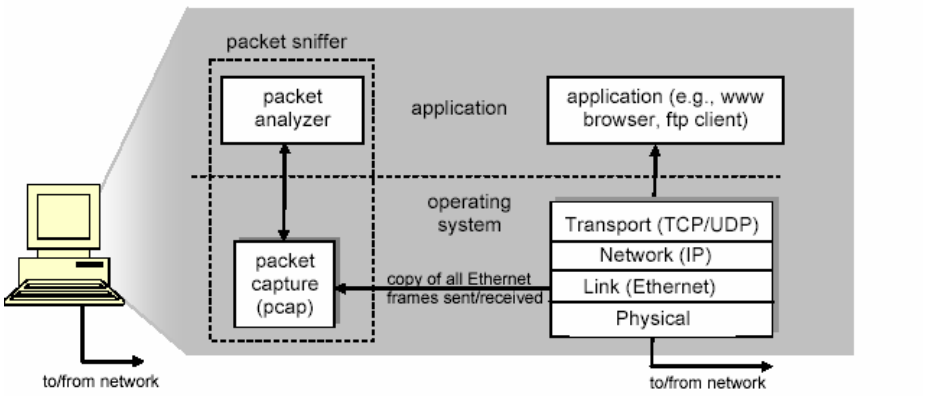


Figure 4.3: Packet Sniffer Structure

Packet sniffer is a basic tool for observing network packet exchanges in a computer. As the name itself suggests, a packet sniffer captures packets being sent/received from/by your computer. It will also typically store and/or display the contents of the various protocol fields in these captured packets. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Figure 4.3 shows the structure of a packet sniffers, different protocols and applications used in the computer . The packet sniffer, shown within the dashed rectangle in Figure 3 is an addition to the usual software in your computer, and consists of two parts. The packet capture library receives a copy of every link-layer frame that is sent from or received by your computer. Messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable. In Figure 4.1, the assumed physical media is an Ethernet, and so all upper-layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you access to all messages sent/received from/by all protocols and applications executing in your computer. The second component of a packet sniffer is the packet analyser , which displays the contents of all fields within a protocol message. In order to do so, the packet analyser

Getting Wireshark:

The Windows 10 has Wireshark installed. You can just launch the Windows 10

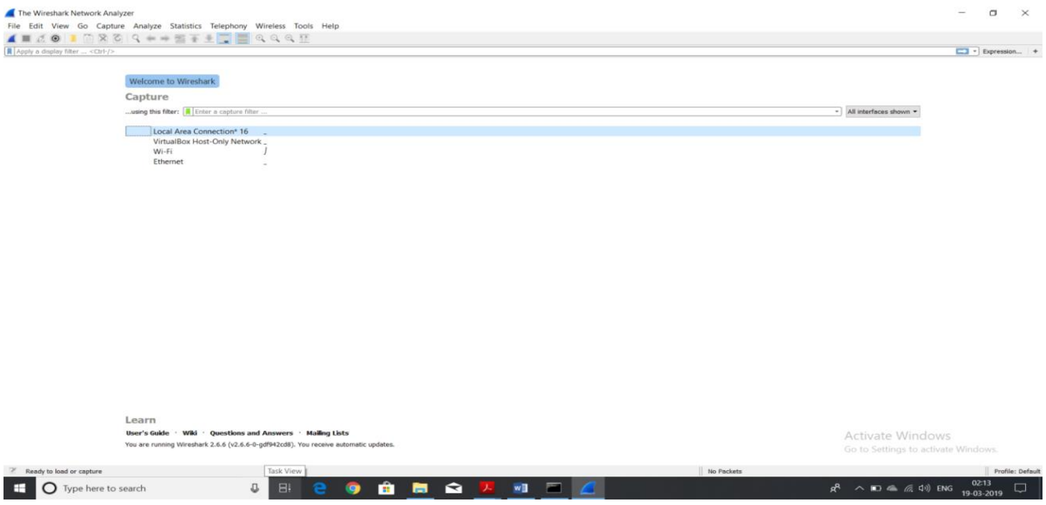
and open Wireshark there. Wireshark can also be downloaded from here:

https://www.wireshark.org/download.html

Starting Wireshark:

When you run the Wireshark program, the Wireshark graphic user interface will

be shown as **Figure** 4.4. Currently, the program is not capturing the packets.

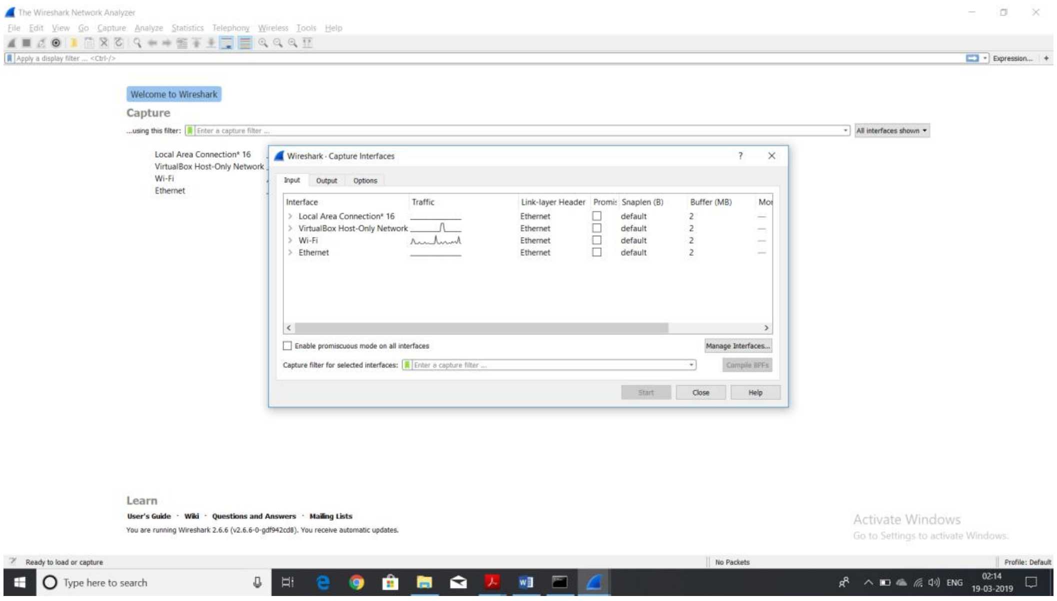

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
Activate Windows
Go to Settings to activate Windows.
Ready to load or capture
Task View
No Packets
Profile: Default
Type here to search
ENG
02:13
19-03-2019

Local Area Connection* 16
VirtualBox Host-Only Network
Wi-Fi
Ethernet

You are running Wireshark 2.6.6 (v2.6.6-0-gdf942cd8). You receive automatic updates.

Figure 4.4: Initial Graphic User Interface of Wireshark

Then, you need to choose an interface. If you are running the Wireshark on your laptop, you need to select WiFi interface. If you are at a desktop, you need to select the Ethernet interface being used. Note that there could be multiple interfaces. In general, you can select any interface but that does not mean that traffic will flow through that interface. The network interfaces (i.e., the physical connections) that your computer has to the network are shown. The attached Figure 4.5 was taken from my computer. After you select the interface, you can click start to capture the packets as shown in Figure 4.6.



Wireshark · Capture Interfaces

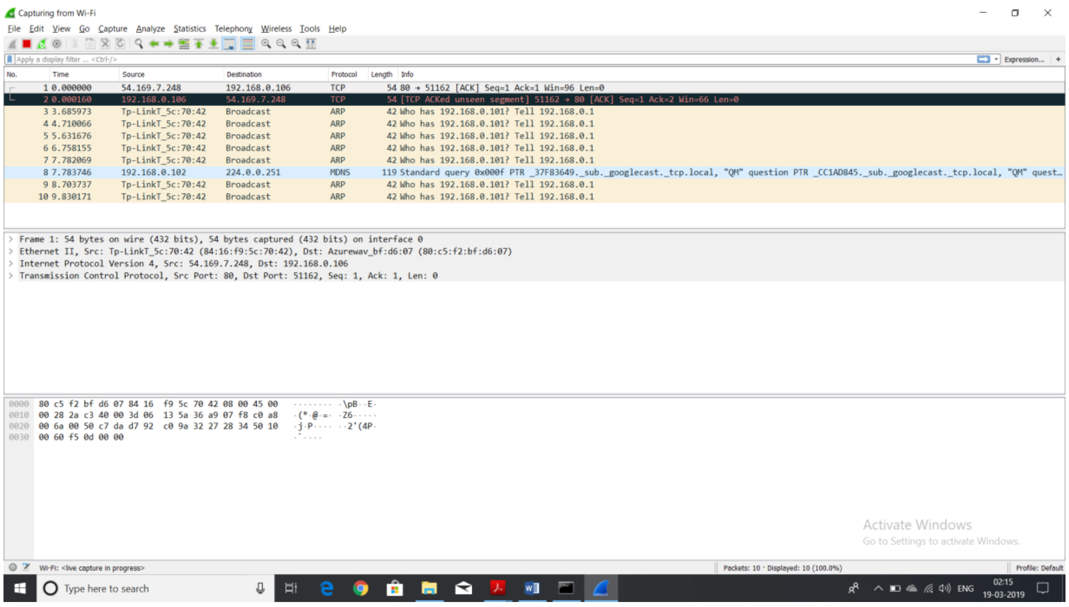
Interface	Traffic	Link-layer Header	Promis	Snaplen (B)	Buffer (MB)	Mo
Local Area Connection* 16		Ethernet	☐	default	2	—
VirtualBox Host-Only Network		Ethernet	☐	default	2	—
Wi-Fi		Ethernet	☐	default	2	—
Ethernet		Ethernet	☐	default	2	—

☐ Enable promiscuous mode on all interfaces

Capture filter for selected interfaces: Enter a capture filter ...

You are running Wireshark 2.6.6 (v2.6.6-0-gdf942cd8). You receive automatic updates.

Figure 4.6: Capture Interfaces in Wireshark.



Capturing from Wi-Fi

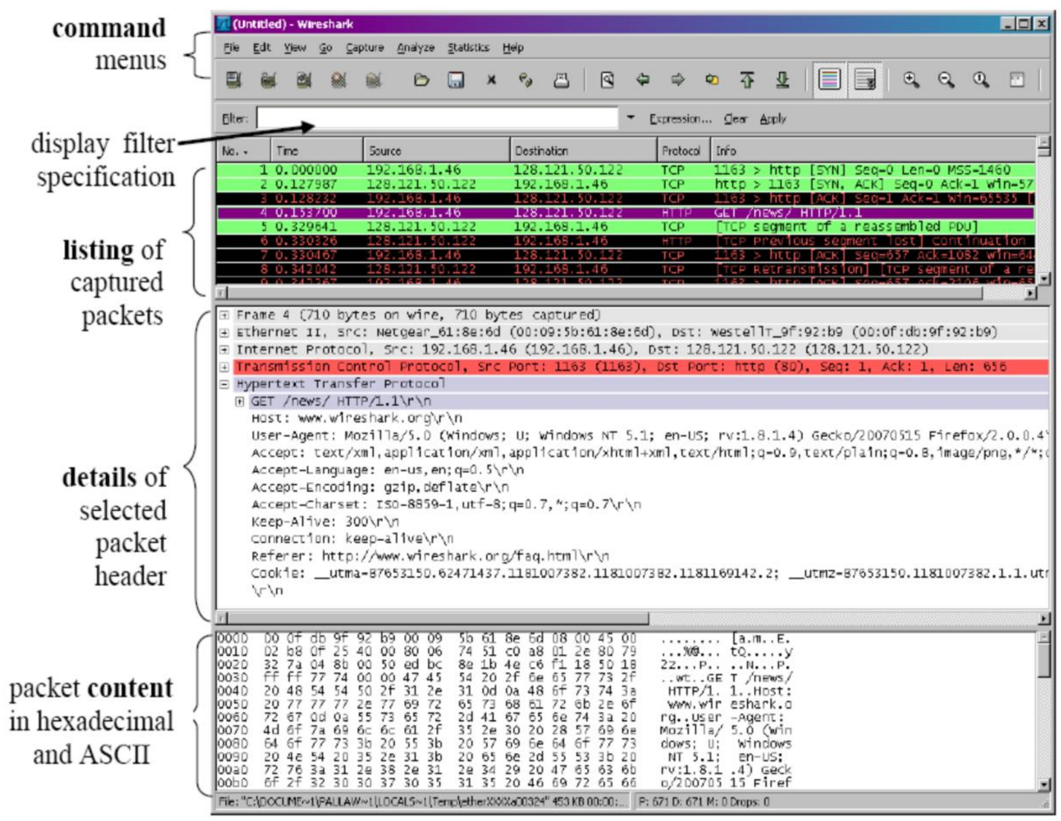
```
No.  Time      Source             Destination    Protocol Length Info
  1 0.000000   54.169.7.248       192.168.0.106  TCP        54 80 → 51162 [ACK] Seq=1 Ack=1 Win=96 Len=0
  2 0.000160   192.168.0.106      54.169.7.248   TCP        54 [TCP ACKed unseen segment] 51162 → 80 [ACK] Seq=1 Ack=2 Win=66 Len=0
  3 3.685973   Tp-LinkT_5c:70:42  Broadcast      ARP        42 Who has 192.168.0.101? Tell 192.168.0.1
  4 4.710066   Tp-LinkT_5c:70:42  Broadcast      ARP        42 Who has 192.168.0.101? Tell 192.168.0.1
  5 5.631676   Tp-LinkT_5c:70:42  Broadcast      ARP        42 Who has 192.168.0.101? Tell 192.168.0.1
  6 6.758155   Tp-LinkT_5c:70:42  Broadcast      ARP        42 Who has 192.168.0.101? Tell 192.168.0.1
  7 7.782069   Tp-LinkT_5c:70:42  Broadcast      ARP        42 Who has 192.168.0.101? Tell 192.168.0.1
  8 7.783746   192.168.0.102      224.0.0.251    MDNS      119 Standard query 0x000f PTR _37F83649._sub._googlecast._tcp.local, "QM" question PTR _CC1AD845._sub._googlecast._tcp.local, "QM" quest…
  9 8.703737   Tp-LinkT_5c:70:42  Broadcast      ARP        42 Who has 192.168.0.101? Tell 192.168.0.1
 10 9.830171   Tp-LinkT_5c:70:42  Broadcast      ARP        42 Who has 192.168.0.101? Tell 192.168.0.1

> Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: Tp-LinkT_5c:70:42 (84:16:f9:5c:70:42), Dst: Azurewav_bf:d6:07 (80:c5:f2:bf:d6:07)
> Internet Protocol Version 4, Src: 54.169.7.248, Dst: 192.168.0.106
> Transmission Control Protocol, Src Port: 80, Dst Port: 51162, Seq: 1, Ack: 1, Len: 0

0000  80 c5 f2 bf d6 07 84 16  f9 5c 70 42 08 00 45 00   ········ ·\pB··E·
0010  00 28 2a c3 40 00 3d 06  13 5a 36 a9 07 f8 c0 a8   ·(*·@·=· ·Z6·····
0020  00 6a 00 50 c7 da d7 92  c0 9a 32 27 28 34 50 10   ·j·P···· ··2'(4P·
0030  00 60 f5 0d 00 00                                  ·`····
```

Wi-Fi: <live capture in progress>   Packets: 10 · Displayed: 10 (100.0%)   Profile: Default

Figure 4.6: Capturing Packets in Wireshark.


command menus
display filter specification
listing of captured packets
details of selected packet header
packet content in hexadecimal and ASCII
(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. -  Time        Source            Destination       Protocol  Info
     1 0.000000    192.168.1.46      128.121.50.122    TCP       1163 > http [SYN] Seq=0 Len=0 MSS=1460
     2 0.127987    128.121.50.122    192.168.1.46      TCP       http > 1163 [SYN, ACK] Seq=0 Ack=1 Win=57
     3 0.128232    192.168.1.46      128.121.50.122    TCP       1163 > http [ACK] Seq=1 Ack=1 Win=65535 [
     4 0.153700    192.168.1.46      128.121.50.122    HTTP      GET /news/ HTTP/1.1
     5 0.329641    128.121.50.122    192.168.1.46      TCP       [TCP segment of a reassembled PDU]
     6 0.330326    128.121.50.122    192.168.1.46      HTTP      [TCP Previous segment lost] Continuation
     7 0.330467    192.168.1.46      128.121.50.122    TCP       1163 > http [ACK] Seq=657 Ack=1082 Win=64
     8 0.342042    128.121.50.122    192.168.1.46      TCP       [TCP Retransmission] [TCP segment of a re
```

```
⊞ Frame 4 (710 bytes on wire, 710 bytes captured)
⊞ Ethernet II, Src: Netgear_61:8e:6d (00:09:5b:61:8e:6d), Dst: WestellT_9f:92:b9 (00:0f:db:9f:92:b9)
⊞ Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 128.121.50.122 (128.121.50.122)
⊞ Transmission Control Protocol, Src Port: 1163 (1163), Dst Port: http (80), Seq: 1, Ack: 1, Len: 656
⊟ Hypertext Transfer Protocol
  ⊞ GET /news/ HTTP/1.1\r\n
    Host: www.wireshark.org\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4
    Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Referer: http://www.wireshark.org/faq.html\r\n
    Cookie: __utma=87653150.62471437.1181007382.1181007382.1181169142.2; __utmz=87653150.1181007382.1.1.utr
    \r\n
```

```
0000  00 0f db 9f 92 b9 00 09  5b 61 8e 6d 08 00 45 00   ........ [a.m..E.
0010  02 b8 0f 25 40 00 80 06  74 51 c0 a8 01 2e 80 79   ...%@... tQ.....y
0020  32 7a 04 8b 00 50 ed bc  8e 1b 4e c6 f1 18 50 18   2z...P.. ..N...P.
0030  ff ff 77 74 00 00 47 45  54 20 2f 6e 65 77 73 2f   ..wt..GE T /news/
0040  20 48 54 54 50 2f 31 2e  31 0d 0a 48 6f 73 74 3a    HTTP/1. 1..Host:
0050  20 77 77 77 2e 77 69 72  65 73 68 61 72 6b 2e 6f    www.wir eshark.o
0060  72 67 0d 0a 55 73 65 72  2d 41 67 65 6e 74 3a 20   rg..User -Agent: 
0070  4d 6f 7a 69 6c 6c 61 2f  35 2e 30 20 28 57 69 6e   Mozilla/ 5.0 (Win
0080  64 6f 77 73 3b 20 55 3b  20 57 69 6e 64 6f 77 73   dows; U;  Windows
0090  20 4e 54 20 35 2e 31 3b  20 65 6e 2d 55 53 3b 20    NT 5.1;  en-US; 
00a0  72 76 3a 31 2e 38 2e 31  2e 34 29 20 47 65 63 6b   rv:1.8.1 .4) Geck
00b0  6f 2f 32 30 30 37 30 35  31 35 20 46 69 72 65 66   o/200705 15 Firef
```

```
File: "C:\DOCUME~1\PALLAW~1\LOCALS~1\Temp\etherXXXXa00324" 453 KB 00:00:...   P: 671 D: 671 M: 0 Drops: 0
```

Figure 4.7: Wireshark Graphical User Interface on Microsoft Windows.

The Wireshark interface has five major components:

The command menus are standard pull down menus located at the top of the window. Of interest to us now is the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture. The packet-listing window displays a one-line summary for each packet captured, including the packet number ,the time at which the packet was captured, the packet’s source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

The packet-header details window provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet's one-line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the right pointing or down-pointing arrowhead to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided. The packet-contents window displays the entire contents of the captured frame, in both ASCII and hexadecimal format. Towards the top of the Wireshark graphical user interface, is the packet display filter field, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

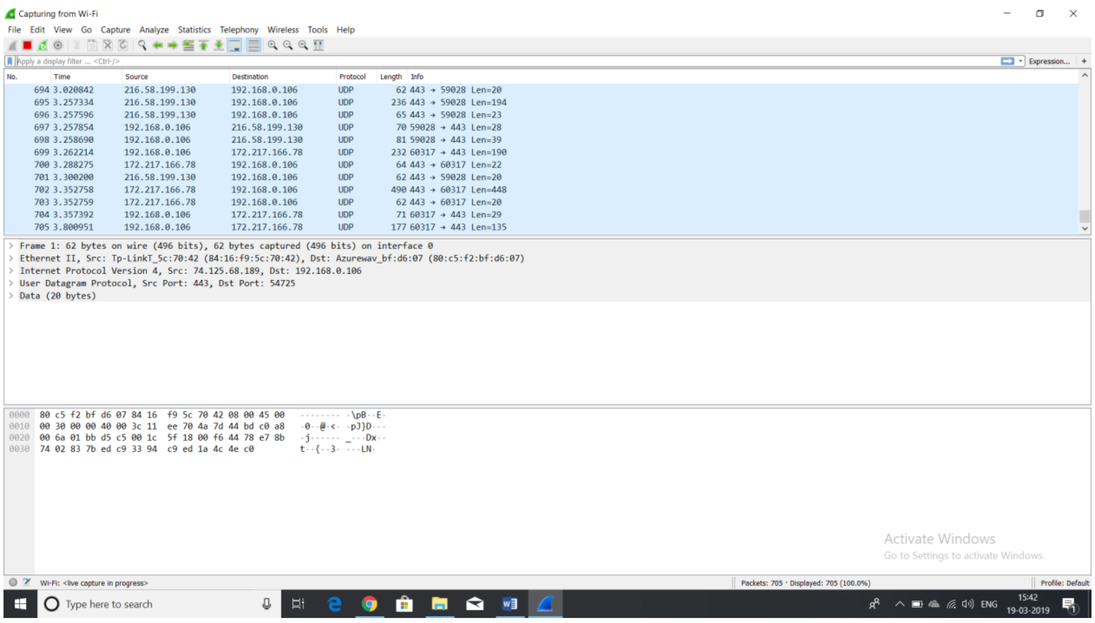
Capturing Packets:

After downloading and installing Wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface.

Test Run

Do the following steps:

1. Start up the Wireshark program (select an interface and press start to capture packets).
2. Start up your favourite browser.
3. In your browser, go to youtube homepage by typing www.youtube.com.
4. After your browser has displayed the http://www.youtube.com page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture see image below:



```
No.  Time        Source          Destination     Protocol Length Info
 694 3.020842    216.58.199.130  192.168.0.106   UDP          62 443 → 59028 Len=20
 695 3.257334    216.58.199.130  192.168.0.106   UDP         236 443 → 59028 Len=194
 696 3.257596    216.58.199.130  192.168.0.106   UDP          65 443 → 59028 Len=23
 697 3.257854    192.168.0.106   216.58.199.130  UDP          70 59028 → 443 Len=28
 698 3.258690    192.168.0.106   216.58.199.130  UDP          81 59028 → 443 Len=39
 699 3.262214    192.168.0.106   172.217.166.78  UDP         232 60317 → 443 Len=190
 700 3.288275    172.217.166.78  192.168.0.106   UDP          64 443 → 60317 Len=22
 701 3.300200    216.58.199.130  192.168.0.106   UDP          62 443 → 59028 Len=20
 702 3.352758    172.217.166.78  192.168.0.106   UDP         490 443 → 60317 Len=448
 703 3.352759    172.217.166.78  192.168.0.106   UDP          62 443 → 60317 Len=20
 704 3.357392    192.168.0.106   172.217.166.78  UDP          71 60317 → 443 Len=29
 705 3.800951    192.168.0.106   172.217.166.78  UDP         177 60317 → 443 Len=135
```

```
> Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
> Ethernet II, Src: Tp-LinkT_5c:70:42 (84:16:f9:5c:70:42), Dst: Azurewav_bf:d6:07 (80:c5:f2:bf:d6:07)
> Internet Protocol Version 4, Src: 74.125.68.189, Dst: 192.168.0.106
> User Datagram Protocol, Src Port: 443, Dst Port: 54725
> Data (20 bytes)
```

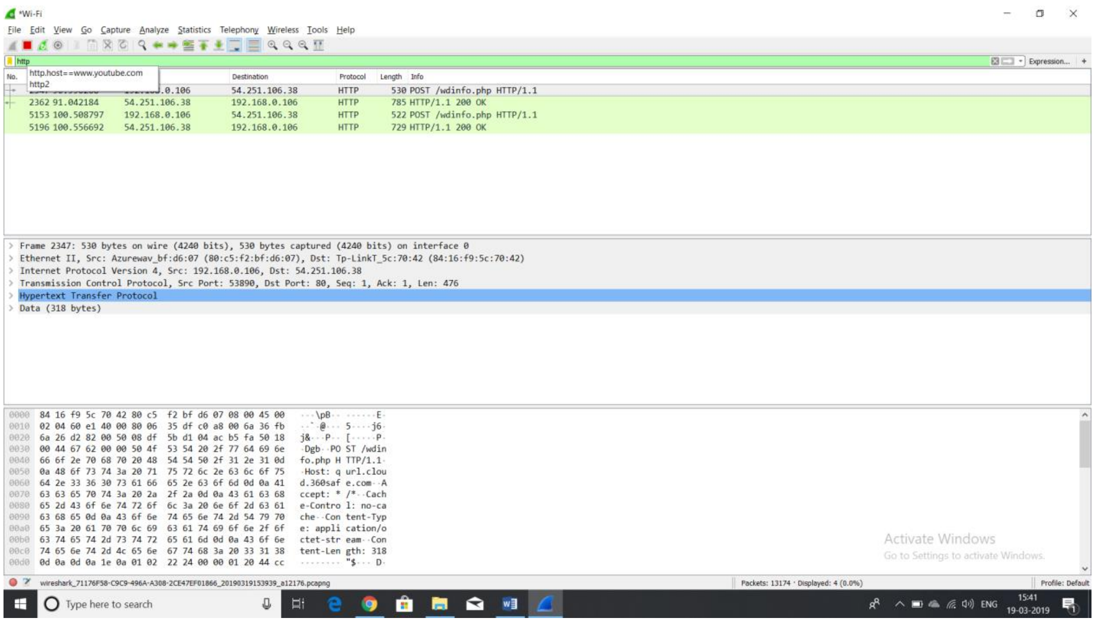
```
0000  80 c5 f2 bf d6 07 84 16  f9 5c 70 42 08 00 45 00   ········ ·\pB··E·
0010  00 30 00 00 40 00 3c 11  ee 70 4a 7d 44 bd c0 a8   ·0··@·<· ·pJ}D···
0020  00 6a 01 bb d5 c5 00 1c  5f 18 00 f6 44 78 e7 8b   ·j······ _···Dx··
0030  74 02 83 7b ed c9 33 94  c9 ed 1a 4c 4e c0         t··{··3· ···LN·
```

Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Activate Windows
Go to Settings to activate Windows.
Wi-Fi: <live capture in progress>
Packets: 705 · Displayed: 705 (100.0%)
Profile: Default
Type here to search
ENG
15:42
19-03-2019

Figure 4.8:Capturing all the data packets.

5. Colour Coding: You'll probably see packets highlighted in green, blue, and black. Wireshark uses colours to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

6. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! However, as you will notice the HTTP messages are not clearly shown because there are many other packets included in the packet capture. Even though the only action you took was to open your browser, there are many other programs in your computer that communicate via the network in the background. To filter the connections to the ones we want to focus on, we have to use the filtering functionality of Wireshark by typing "http" in the filtering field as shown below:



```
http
http.host==www.youtube.com
http2

No.                                  Destination      Protocol  Length Info
                           .0.106    54.251.106.38    HTTP         530 POST /wdinfo.php HTTP/1.1
     2362 91.042184   54.251.106.38  192.168.0.106    HTTP         785 HTTP/1.1 200 OK
     5153 100.508797  192.168.0.106  54.251.106.38    HTTP         522 POST /wdinfo.php HTTP/1.1
     5196 100.556692  54.251.106.38  192.168.0.106    HTTP         729 HTTP/1.1 200 OK
```

```
> Frame 2347: 530 bytes on wire (4240 bits), 530 bytes captured (4240 bits) on interface 0
> Ethernet II, Src: Azurewav_bf:d6:07 (80:c5:f2:bf:d6:07), Dst: Tp-LinkT_5c:70:42 (84:16:f9:5c:70:42)
> Internet Protocol Version 4, Src: 192.168.0.106, Dst: 54.251.106.38
> Transmission Control Protocol, Src Port: 53890, Dst Port: 80, Seq: 1, Ack: 1, Len: 476
> Hypertext Transfer Protocol
> Data (318 bytes)
```

```
0000  84 16 f9 5c 70 42 80 c5  f2 bf d6 07 08 00 45 00   ···\pB·· ······E·
0010  02 04 60 e1 40 00 80 06  35 df c0 a8 00 6a 36 fb   ··`·@··· 5····j6·
0020  6a 26 d2 82 00 50 08 df  5b d1 04 ac b5 fa 50 18   j&···P·· [·····P·
0030  00 44 67 62 00 00 50 4f  53 54 20 2f 77 64 69 6e   ·Dgb··PO ST /wdin
0040  66 6f 2e 70 68 70 20 48  54 54 50 2f 31 2e 31 0d   fo.php H TTP/1.1·
0050  0a 48 6f 73 74 3a 20 71  75 72 6c 2e 63 6c 6f 75   ·Host: q url.clou
0060  64 2e 33 36 30 73 61 66  65 2e 63 6f 6d 0d 0a 41   d.360saf e.com··A
0070  63 63 65 70 74 3a 20 2a  2f 2a 0d 0a 43 61 63 68   ccept: * /*··Cach
0080  65 2d 43 6f 6e 74 72 6f  6c 3a 20 6e 6f 2d 63 61   e-Contro l: no-ca
0090  63 68 65 0d 0a 43 6f 6e  74 65 6e 74 2d 54 79 70   che··Con tent-Typ
00a0  65 3a 20 61 70 70 6c 69  63 61 74 69 6f 6e 2f 6f   e: appli cation/o
00b0  63 74 65 74 2d 73 74 72  65 61 6d 0d 0a 43 6f 6e   ctet-str eam··Con
00c0  74 65 6e 74 2d 4c 65 6e  67 74 68 3a 20 33 31 38   tent-Len gth: 318
00d0  0d 0a 0d 0a 1e 0a 01 02  22 24 00 00 01 20 44 cc   ········ "$··· D·
```

```
wireshark_71176F58-C9C9-496A-A308-2CE47EF01866_20190319153939_a12176.pcapng    Packets: 13174 · Displayed: 4 (0.0%)    Profile: Default
```

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
15:41
19-03-2019

Figure 4.9:Capturing packets of a single website by setting the filter.

Notice that we now view only the packets that are of protocol HTTP. However, we also still do not have the exact communication we want to focus on because using HTTP as a filter is not descriptive enough to allow us to find our connection. We need to be more precise if we want to capture the correct set of packets.

CHAPTER 5
PYTHON

Introduction:

Python was created by Guido van Rossum in the late 1980's at Centrum Wiskunde and Informatica (CWI) in the Netherlands as a successor to the ABC language which is capable of exception handling and interfacing with the Amoeba operating system

Python is both object oriented and functional oriented programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming. Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution, which binds method and variable names during program execution. The standard library has two modules that implement functional tools borrowed from Haskell and Standard ML.

History:

Its implementation began in December 1989.Van Rossum's long time influence on Python is reflected in the title given to him by the Python community: Benevolent Dictator For Life (BDFL) – a post from which he gave himself permanent vacation on July 12, 2018.

Python 2.0 was released on 16th October 2000 with many major new features including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3rd December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6 and 2.7 version series. Releases of Python 3 include the utility, which automates the translation of Python 2 code to Python 3.

There are many releases of python as we are working on machine learning, we have to install anaconda so that we can work easily on machine learning algorithms.

Anaconda installation:

As we are working in windows 10,

1. Go to https://www.anaconda.com/distribution/ and download the appropriate exe file.
2. Double click on the .exe file in the downloads and allow python for installation
3. Then click on I agree in the installation part

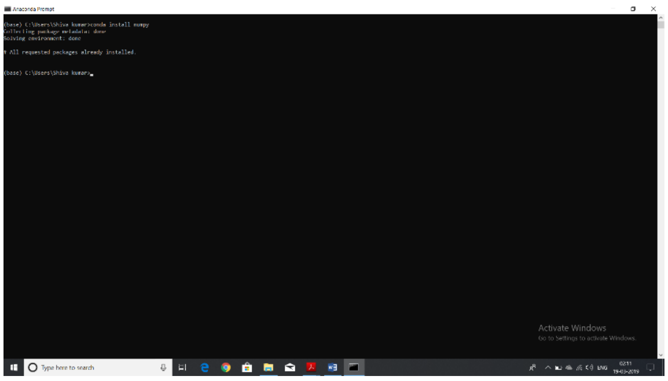
4. Give appropriate path for installation it may take few minutes to install and the finish it.
5. Now, we can use anaconda by typing idle in the windows search.
6. For new libraries use anaconda prompt by typing anaconda prompt in the windows search.

CHAPTER 6
LIBRARIES

6.1 **NumPy:**

NumPy is a library for the Python programming language that (among other things) provides support for large, multi-dimensional arrays. Why is that important? Using 1 mathematical functions, we can quickly perform numerical analysis on an image.
NumPy, we can express images as multi-dimensional arrays. Representing images as NumPy arrays is not only computational and resource efficient, but many other image processing and machine learning libraries use NumPy array representations as well. Furthermore, by using NumPy's built-in high-level.
Installation command:

conda install numpy



```
(base) C:\Users\Shiva kumar>conda install numpy
Collecting package metadata: done
Solving environment: done

# All requested packages already installed.


(base) C:\Users\Shiva kumar>
```

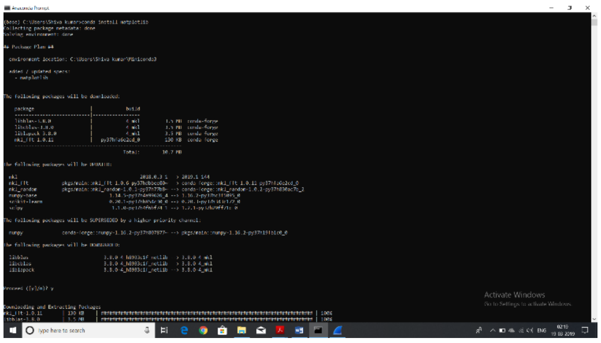
Figure 6.1:NumPy installation

6.2 Matplotlib:

Simply put, matplotlib is a plotting library. If you've ever used MATLAB before, you'll probably feel very comfortable in the matpotlib environment. When analyzing images, we'll make use of matplotlib, whether plotting the overall accuracy of search systems or simply viewing the image itself, matplotlib is a great tool to have in your toolbox.

Installation command:

conda install matplotlib



```
(base) C:\Users\Shiva kumar>conda install matplotlib
Collecting package metadata: done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\Shiva kumar\Miniconda3

  added / updated specs:
    - matplotlib


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    libblas-3.8.0              |            4_mkl         1.5 MB  conda-forge
    libcblas-3.8.0             |            4_mkl         1.5 MB  conda-forge
    liblapack 3.8.0            |            4_mkl         3.5 MB  conda forge
    mkl_fft 1.0.11             |  py37hfa6e2cd_0         130 KB  conda forge
    ------------------------------------------------------------
                                           Total:        10.7 MB

The following packages will be UPDATED:

  mkl                                     2018.0.3-1  --> 2019.1-144
  mkl_fft          pkgs/main::mkl_fft-1.0.6-py37hdbbee80-- --> conda-forge::mkl_fft-1.0.11-py37hfa6e2cd_0
  mkl_random       pkgs/main::mkl_random-1.0.1-py37h77b8-- --> conda-forge::mkl_random-1.0.2-py37h830ac7b_2
  numpy-base                        1.14.5-py37h4a99626_4 --> 1.16.2-py37hc315895_0
  scikit-learn                      0.20.1-py37h8b5dc98_0 --> 0.20.1-py37h343c172_0
  scipy                             1.1.0-py37h4f6bf74_1 --> 1.2.1-py37h29ff71c_0

The following packages will be SUPERSEDED by a higher-priority channel:

  numpy            conda-forge::numpy-1.16.2-py37h807877-- --> pkgs/main::numpy-1.16.2-py37h19fb1c0_0

The following packages will be DOWNGRADED:

  libblas                          3.8.0-4_h8933c1f_netlib --> 3.8.0-4_mkl
  libcblas                         3.8.0-4_h8933c1f_netlib --> 3.8.0-4_mkl
  liblapack                        3.8.0-4_h8933c1f_netlib --> 3.8.0-4_mkl


Proceed ([y]/n)? y


Downloading and Extracting Packages
mkl_fft-1.0.11       | 130 KB    | ############################################ | 100%
libblas-3.8.0        | 3.5 MB    | ############################################ | 100%
```

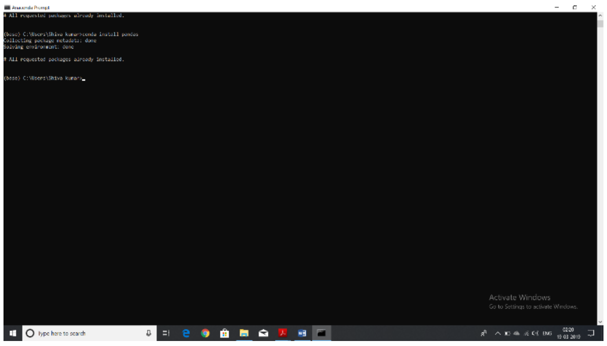
Figure 6.2:Matplotlib installation

6.3 Pandas:

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Installation command:

conda install pandas



```
# All requested packages already installed.


(base) C:\Users\Shiva kumar>conda install pandas
Collecting package metadata: done
Solving environment: done

# All requested packages already installed.


(base) C:\Users\Shiva kumar>_
```

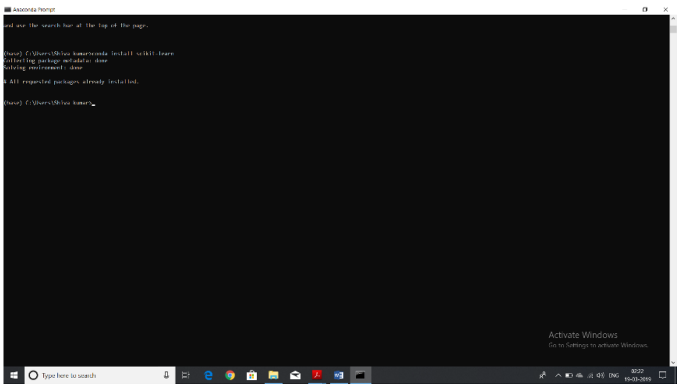
Figure 6.3: Pandas installation

6.4 Scikit-learn:

Scikit-learn is probably the most useful library for machine learning in Python. It is on NumPy, SciPy and matplotlib, this library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. scikit-learn is used to build models. It should not be used for reading the data, manipulating and summarizing it. There are better libraries for that (e.g. NumPy, Pandas etc.)

Installation command:

conda install scikit-learn



```
and use the search bar at the top of the page.


(base) C:\Users\Shiva kumar>conda install scikit-learn
Collecting package metadata: done
Solving environment: done

# All requested packages already installed.


(base) C:\Users\Shiva kumar>
```

Figure 6.4:scikit-learn installation

CHAPTER 7
PROJECT IMPLEMENTATION

7.1 Implementation

Creating CSV file:

1. Initially Connect Laptop to 4G LTE network hotspot and make sure that no other applications are running in background. The source and destination ports have to be IPv6 addresses in order to determine if the obtained network is 4G.
2. Open Wireshark application and google chrome.
3. In google chrome, open any mail, video, audio and web and browse them simultaneously start capturing packets using Wireshark.
4. Now go to file and save by exporting the captured data into a Comma Separated Value (CSV).
5. In the obtained CSV file, there are six columns: Timestamp, source, destination, protocol, information and packet size.
6. The pre-processing of the data set is done and it is downsized to four columns and a Target labelled as Audio, Video, Web and Mail.
7. The classification is performed using various algorithms like KNN, SVM, Random Forest, Decision Tree and Naïve Bayes.
8. The results are simulated for Accuracy, Prediction time, Training time and Feature importance of the above mentioned algorithms. A statistical analysis is provided for the same.

Implementation of classifiers:

Import the Libraries :

We will get started by importing the necessary libraries required to implement the required Algorithm. We should import the numpy libraries for scientific calculation. Next, we should import the matplotlib.pyplot library for plotting the graph.

We will import libraries for required Classifier from sklearn library and implement them .We should import accuracy score from sklearn.metrics for accuracy score.

Fetch the Data:

We will fetch the data using ‘pandas_datareader’. We store this in the data frame ‘df’. Then we should drop all the missing values from the data using ‘dropna’ function.

Define Predictor Variable:

Predictor variable is also known as an independent variable. It can be used to determine the value of the target variable. We can use ‘Open-Close’ and ‘High-Low’ as a predictor variable. We should drop NaN values and store the predictor variables in ‘X’.

Define Target Variables:

The target variable is also known as dependent variable. These values are to be predicted by predictor variables. We will be storing the target variable in a variable ’Y’.

Pre-processing Data:

Pre-processing means transformation applied to our data before feeding it to the algorithm. We need data pre-processing to achieve better results from the applied model in machine learning. In the above data set collected we have applied pre-processing to four different columns .

For example: Let us consider a data set consisting of 1000 rows and some n number of columns .If a column has 5 different groups , on applying pre-processing , each group will be assigned with a machine understandable value.

Split the Dataset:

We should split the dataset into training set and test set. In this we will use 90% of our data to train and the rest 10% to test. For this, we shall create a split parameter which can divide the data frame in a 90-10 ratio. You can change the split percentage as per your choice. It is advisable to allot at least 60% data as train data for better results.

‘X_train’ and ‘Y_train’ are called as train dataset.

‘X_test’ and ‘Y_test’ are called as test dataset.

Instantiate Model:

After completion of splitting the dataset into training and test dataset, we should instantiate classifier.

We can notice the change in result by changing classifiers. Next, we should fit the train data by using ‘fit’ function. Then, we should calculate the train and test accuracy by using the ‘accuracy score’ function.

Here, We are instantiating all different algorithms and finding their accuracies.

Plotting Graphs

After the training algorithm and testing, we plot the results using Matplot Library to analyze .

FINAL CODE:

```
import pandas as pd

import numpy as np

from sklearn import datasets

import matplotlib.pyplot as plt

from matplotlib.colors import ListedColormap

video=pd.read_csv('new.csv')

size=video.shape

print(size)

target=video['target']

cols_to_drop=['Time','Info','No.']

video_feature = video.drop(cols_to_drop,axis=1)

from sklearn import preprocessing

label_encoder = preprocessing.LabelEncoder()

video_feature['Protocol'] = label_encoder.fit_transform(video_feature['Protocol'])

video_feature['Source'] = label_encoder.fit_transform(video_feature['Source'])
```

```
video_feature['Destination'] = label_encoder.fit_transform(video_feature['Destination'])

video_feature['target'] = label_encoder.fit_transform(video_feature['target'])

cols_to_drop = ['target']

video_feature = video_feature.drop(cols_to_drop,axis=1)

seed=7 #To generate same sequence of random numbers

import sklearn

from sklearn.model_selection import train_test_split

#Splitting the data for training and testing(90% train,10% test)

train_data,test_data, train_label, test_label = train_test_split(video_feature, target,
test_size=.1,random_state=seed)

from sklearn.naive_bayes import GaussianNB

import time as t

classifier=GaussianNB()

t0=t.time()

classifier = classifier.fit(train_data, train_label)

nbtt=round(t.time()-t0, 5)

print("training time nbc:",nbtt,"s")

t1=t.time()

video_predicted_target=classifier.predict(test_data)

nbpt=round(t.time()-t1, 5)

print("predict time nbc :",nbpt, "s")

score1= classifier.score(test_data, test_label)

print('Naive Bayes : ',score1)

from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n_jobs=2, random_state=5)

t0=t.time()

classifier = classifier.fit(train_data, train_label)
```

```
rftt=round(t.time()-t0, 5)
print("training time rfc:", rftt,"s")
t1=t.time()
video_predicted_target=classifier.predict(test_data)
rfpt=round(t.time()-t1, 5)
print("predict time rfc:",rfpt, "s")
score2 = classifier.score(test_data, test_label)
print('Random Forest Classifier : ',score2)
label = ['Source port', 'Destionation port', 'Protocol', 'Packet Size']
class_training = classifier.feature_importances_
def plot_bar_x():
 index = np.arange(len(label))
 plt.bar(index, class_training)
 plt.xlabel('Features', fontsize=10)
 plt.ylabel('Importance of feature', fontsize=10)
 plt.xticks(index, label, fontsize=10, rotation=30)
 plt.title('Feature importance in Random forest')
 plt.show()
 plot_bar_x()
 from sklearn import tree
 decision_tree = tree.DecisionTreeClassifier(criterion='gini')
 classifier=decision_tree.fit(train_data, train_label)
 print('The accuracy of the Decision Tree classifier on test data is
{:.2f}'.format(decision_tree.score(test_data, test_label)))
 t0=t.time()
 classifier = classifier.fit(train_data, train_label)
 dttt=round(t.time()-t0, 5)
```

```
score3 = classifier.score(test_data, test_label)

print("training time of Decision tree :",dttt ,"s")

t1=t.time()

video_predicted_target=classifier.predict(test_data)

dtpt=round(t.time()-t1, 5)

print("predict time of decision tree:",dtpt , "s")

label = ['Source port', 'Destionation port', 'Protocol', 'Packet Size']

class_training = classifier.feature_importances_

def plot_bar_x():

index = np.arange(len(label))

plt.bar(index, class_training)

plt.xlabel('Features', fontsize=10)

plt.ylabel('Importance of feature', fontsize=10)

plt.xticks(index, label, fontsize=10, rotation=30)

plt.title('Feature importance in Decision Tree')

plt.show()

plot_bar_x()

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors = 7, p = 2, metric='minkowski')

classifier=knn.fit(train_data, train_label)

print('The accuracy of the Knn classifier on test data is {:.2f}'.format(knn.score(test_data,
test_label)))

t0=t.time()

classifier = classifier.fit(train_data, train_label)

kntt=round(t.time()-t0, 5)

print("training time of knn :",kntt,"s")

t1=t.time()
```

```
 video_predicted_target=classifier.predict(test_data)

 score4 = classifier.score(test_data, test_label)

 knpt=round(t.time()-t1, 5)

print("predict time of knn:",knpt , "s")

from sklearn.svm import SVC

svm = SVC(kernel='rbf', random_state=0, gamma=.10, C=1.0)

classifier=svm.fit(train_data, train_label)

print('The accuracy of the SVM classifier on test data is {:.2f}'.format(svm.score(test_data,
test_label)))

t0=t.time()

classifier = classifier.fit(train_data, train_label)

svtt=round(t.time()-t0, 5)

print("training time of SVM :",svtt,"s")

t1=t.time()

video_predicted_target=classifier.predict(test_data)

svpt=round(t.time()-t1, 5)

score5 = classifier.score(test_data, test_label)

print("predict time of SVM :",svpt, "s")

label = ['Naive bayees', 'Random forest', 'decision tree', 'knn', 'svm']

class_training = [

nbtt,

rftt,

dttt,

kntt,

svtt

]

def plot_bar_x():
   index = np.arange(len(label))
   plt.bar(index, class_training)
   plt.xlabel('classifiers', fontsize=10)
   plt.ylabel('time', fontsize=10)
   plt.xticks(index, label, fontsize=10, rotation=30)
   plt.title('Training time')
   plt.show()
  plot_bar_x()
 label = ['Naive bayees', 'Random forest', 'decision tree', 'knn', 'svm']
 class_prediction = [
 nbpt,
 rfpt,
 dtpt,
 knpt,
 svpt
]
 def plot_bar_x():
 index = np.arange(len(label))
  plt.bar(index, class_prediction)
  plt.xlabel('classifiers', fontsize=10)
  plt.ylabel('time', fontsize=10)
  plt.xticks(index, label, fontsize=10, rotation=30)
  plt.title('Predict time')
 plt.show()
 plot_bar_x()
label = ['Naive bayees', 'Random forest', 'decision tree', 'knn', 'svm']
class_score = [
score1*100,
score2*100,
score3*100,
score4*100,
score5*100
]
  def plot_bar_x():
   index = np.arange(len(label))
  plt.bar(index, class_score)
  plt.xlabel('classifiers', fontsize=10)
  plt.ylabel('percentage', fontsize=10)
 plt.xticks(index, label, fontsize=10, rotation=30)
 plt.title('Accuracy Score')
 plt.show()
 plot_bar_x()
```

7.2 Results and Discussion:

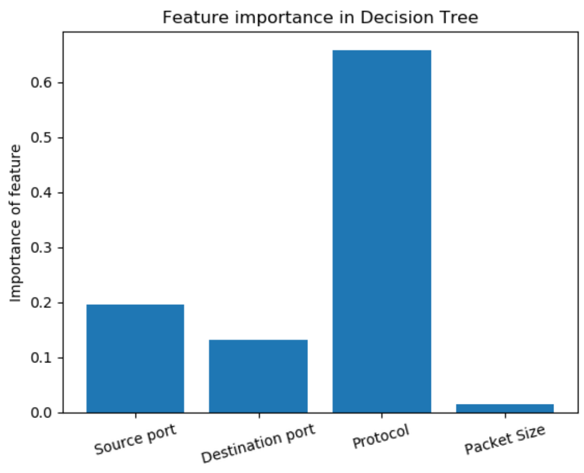


Figure 7.1: Feature importance in Decision tree

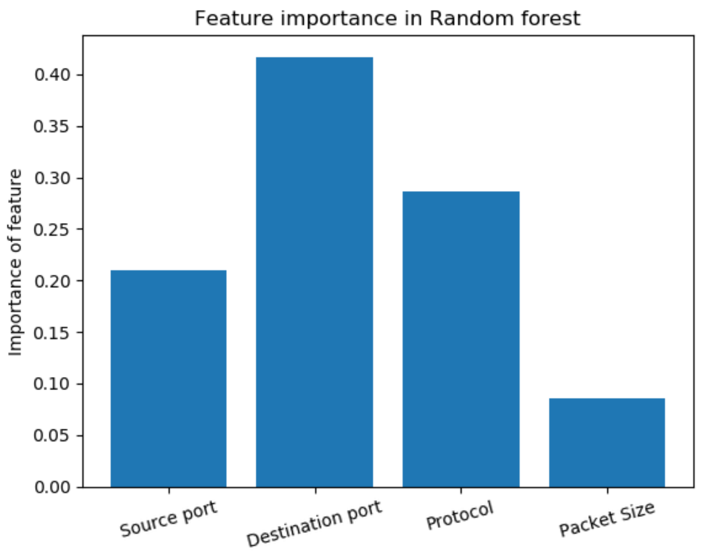


Figure 7.2:Feature importance in Random Forest

Figure 7.1 and Figure 7.2 shows that when the Decision tree a and Random forest algorithms are used , different importance to different features are given before predicting the test data.

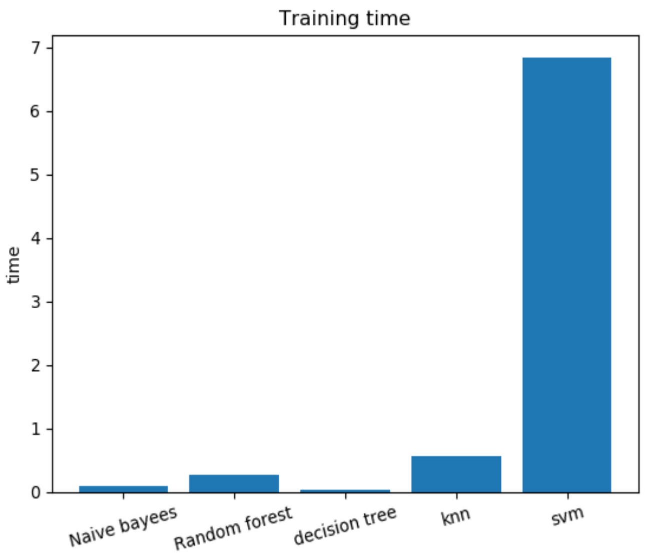


Figure 7.3:Training time of different algorithms

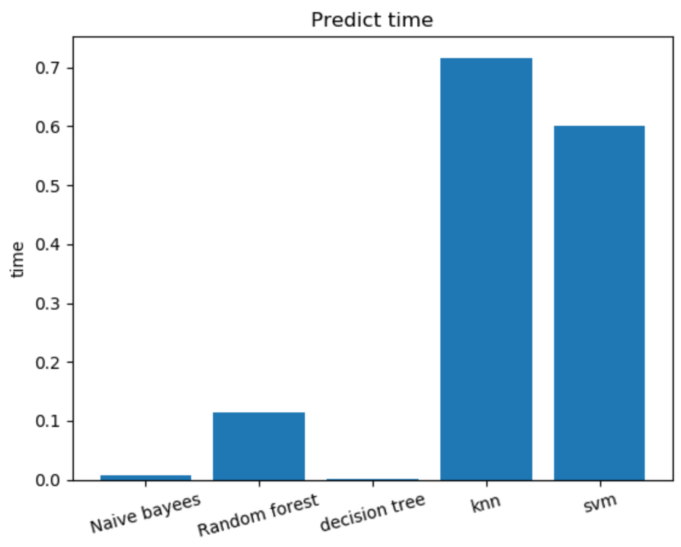


Figure 7.4: Predict time of different algorithms

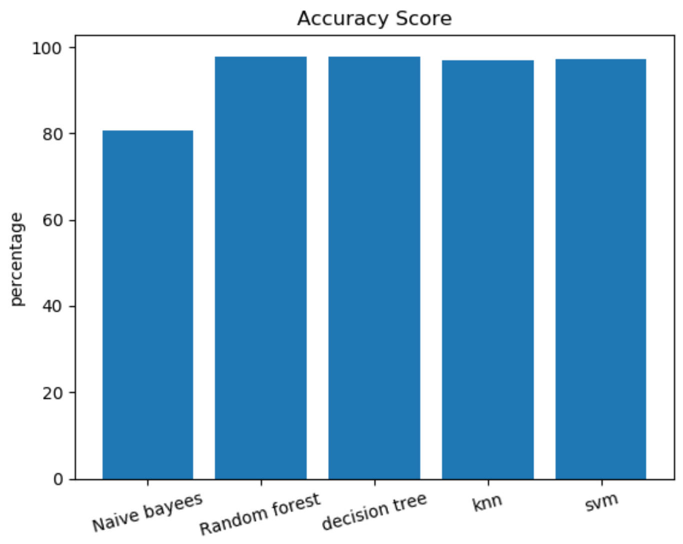


Figure 7.5: Classification Score

The different features we obtained while training the data are Source, Destination, Protocol and Packet size. From the above results, it is observed that while predicting test data least importance is given to packet size and highest importance is given to Protocol in Decision Tree Algorithm. In Random Forest Algorithm highest importance is given to destination port and least importance is given to packet length while predicting.

Training time and Prediction time are least for Decision Tree and highest for SVM classifier. Accuracy is least for Naïve Bayes and highest for Random Forest Algorithm.

CHAPTER 8

CONCLUSION

In this project, first internet traffic has been captured using Wireshark software for packet capture for different sites and created a CSV file. After that, Internet traffic from this dataset is classified using five ML classifiers.

Results show that Random forest gives better performance with classification accuracy of 97.7%. Using this dataset, performance of five ML classifiers has been analysed based on training time, prediction time and accuracy. Results show that Random forest classifier gives better performance among all other classifiers in terms classification accuracy of 97.7%, training time of ML algorithms and recall and precision values of individual internet applications. Thus, it is evident that Random forest is an effective Machine Learning techniques for IP traffic classification.

Secondly, this internet traffic dataset can be extended for many other internet applications which internet users use in their day to day life and it can also be captured from various different real time environments such as university or college campus, offices, home environments and other work stations etc.

REFERENCES

Books

[1] Machine Learning in Action 1st Edition by Peter Harrington

[2] Learning scikit-learn:Machine Learning in Python Paperback by Raul Garreta and Guillermo Moncecchi
[3] Learn Data Analysis with Python by A.J Henley and Dave Wolf

Links

https://www.wireshark.org/docs/wsug_html_chunked/

https://pythonprogramming.net/machine-learning-tutorial-python-introduction/

https://www.pythonprogramming.in/matplotlib.html