SMART INTERNET PROBING: SCANNING USING ADAPTIVE MACHINE LEARNING

**ABSTRACT**

Network scanning is widely used to assess security postures of hosts/networks, discover vulnerabilities, and study Internet trends. However, scans can generate large amounts of traffic, and efficient probing of IPv6 hosts (where global scans are infeasible) is an outstanding problem. In this chapter, we develop a framework for efficient Internet scans using machine learning, by preemptively detecting and avoiding the scanning of inactive hosts. We evaluate this framework over global scans of the IPv4 space over 20 ports, and show that using location and ownership information we can reduce the bandwidth of scans by 26.7-72.0%, while discovering 90-99% of active hosts. We then evaluate a sequential method by gradually adding information obtained from scanned ports to adaptively predict the remaining port responses, yielding 47.4-83.5% of bandwidth savings at the same true positive rates. Our framework can be used to lower the bandwidth consumption of scans and increase their hit rate, thereby reducing their intrusive nature and enabling efficient discovery of active devices.

**1.INTRODUCTION**

Network scanning is a widely studied topic, ranging from partial scans of the Internet, to global scans of the IPv4 address space. This has lead to the development of network scanning tools such as ZMap and NMap which have provided researchers with large amounts of information on arbitrary Internet hosts. Data resulting from network scans have been used in a wide range of security studies, e.g., to probe and characterize machines utilized in the Mirai botnet, to gauge the security posture of networks for cyber-risk forecasting and to study hosts susceptible to the Heartbleed vulnerability. Internet scanning is a crucial tool for giving visibility into the security of Internet-connected entities, as it can measure the attack surface of networks by revealing (potentially misconfigured/vulnerable) networked devices accessible on the public Internet. Additionally, network scanning has been used in many Internet measurement studies, including studies for examining trends and adoption rates of different technologies , to detect discoverable hosts and to categorize them (e.g., IoT devices) , and to map network topologies.

**2.LITERATURE SURVEY**

# 2.1 Selecting representative IP addresses for Internet topology studies

**Authors: X. Fan and J. Heidemann**

**Abstract:** An Internet hitlist is a set of addresses that cover and can represent the the Internet as a whole. Hitlists have long been used in studies of Internet topology, reachability, and performance, serving as the destinations of traceroute or performance probes. Most early topology studies used manually generated lists of prominent addresses, but evolution and growth of the Internet make human maintenance untenable. Random selection scales to today’s address space, but most random addresses fail to respond. In this paper we present what we believe is the first automatic generation of hitlists informed censuses of Internet addresses. We formalize the desirable characteristics of a hitlist: responsiveness, each representative responds to pings; completeness, they cover all the allocated IPv4 address space; and stability, list evolution is minimized when possible. We quantify the accuracy of our automatic hitlists, showing that only one-third of the Internet allows informed selection of representatives. Of informed representatives, 50–60% are likely to respond three months later, and we show that causes for non-responses are likely due to dynamic addressing (so no stable representative exists) or firewalls. In spite of these limitations, we show that the use of informed hitlists can add 1.7 million edge links (a 5% growth) to traceroute-based Internet topology studies Our hitlists are available free-of-charge and are in use by several other research projects.

# 2.2 Target generation for Internet-wide IPv6 scanning

**Authors: A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson**

**Abstract:** Fast IPv4 scanning has enabled researchers to answer a wealth of new security and measurement questions. However, while increased network speeds and computational power have enabled comprehensive scans of the IPv4 address space, a brute-force approach does not scale to IPv6. Systems are limited to scanning a small fraction of the IPv6 address space and require an algorithmic approach to determine a small set of candidate addresses to probe. In this paper, we first explore the considerations that guide designing such algorithms. We introduce a new approach that identifies dense address space regions from a set of known “seed” addresses and generates a set of candidates to scan. We compare our algorithm 6Gen against Entropy/IP—the current state of the art—finding that we can recover between 1–8 times as many addresses for the five candidate datasets considered in the prior work. However, during our analysis, we uncover widespread IP aliasing in IPv6 networks. We discuss its effect on target generation and explore preliminary approaches for detecting aliased regions.

# 2.3 ZMap: Fast Internetwide scanning and its security applications

**Authors: Z. Durumeric, E. Wustrow, and J. A. Halderman**

**Abstract:** Internet-wide network scanning has numerous security applications, including exposing new vulnerabilities and tracking the adoption of defensive mechanisms, but probing the entire public address space with existing tools is both difficult and slow. We introduce ZMap, a modular, open-source network scanner specifically architected to perform Internet-wide scans and capable of surveying the entire IPv4 address space in under 45 minutes from user space on a single machine, approaching the theoretical maximum speed of gigabit Ethernet. We present the scanner architecture, experimentally characterize its performance and accuracy, and explore the security implications of high speed Internet-scale network surveys, both offensive and defensive. We also discuss best practices for good Internet citizenship when performing Internet-wide surveys, informed by our own experiences conducting a long-term research survey over the past year.

# 2.4 A search engine backed by Internet-wide scanning

# 

**Authors: Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman**

**Abstract:** Fast Internet-wide scanning has opened new avenues for security research, ranging from uncovering widespread vulnerabilities in random number generators to tracking the evolving impact of Heartbleed. However, this technique still requires significant effort: even simple questions, such as, "What models of embedded devices prefer CBC ciphers?", require developing an application scanner, manually identifying and tagging devices, negotiating with network administrators, and responding to abuse complaints. In this paper, we introduce Censys, a public search engine and data processing facility backed by data collected from ongoing Internet-wide scans. Designed to help researchers answer security-related questions, Censys supports full-text searches on protocol banners and querying a wide range of derived fields (e.g., 443.https.cipher). It can identify specific vulnerable devices and networks and generate statistical reports on broad usage patterns and trends. Censys returns these results in sub-second time, dramatically reducing the effort of understanding the hosts that comprise the Internet. We present the search engine architecture and experimentally evaluate its performance. We also explore Censys's applications and show how questions asked in recent studies become simple to answer.

**3.SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM:**

Network scanning is widely used to assess security postures of hosts/networks, discover vulnerabilities, and study Internet trends. However, scans can generate large amounts of traffic, and efficient probing of IPv6 hosts (where global scans are infeasible) is an outstanding problem.

**3.1.1 DISADVANTAGES OF EXISTING SYSTEM:**

1. efficient probing of IPv6 hosts is an outstanding problem

# 3.1Proposed System:

In this chapter, we develop a framework for efficient Internet scans using machine learning, by preemptively detecting and avoiding the scanning of inactive hosts. We evaluate this framework over global scans of the IPv4 space over 20 ports, and show that using location and ownership information we can reduce the bandwidth of scans while discovering of active hosts. We then evaluate a sequential method by gradually adding information obtained from scanned ports to adaptively predict the remaining port responses, yielding 47.4-83.5% of bandwidth savings at the same true positive rates.

# Advantages of proposed system:

1. Our framework can be used to lower the bandwidth consumption of scans and increase their hit rate, thereby reducing their intrusive nature and enabling efficient discovery of active devices.

**3.3 SYSTEM REQUIREMENTS:**

**SOFTWARE REQUIREMENTS**

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

* **Python idel 3.7 version (or)**
* **Anaconda 3.7 ( or)**
* **Jupiter (or)**
* **Google colab**

**HARDWARE REQUIREMENTS**

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

* **Operating system : windows, linux**
* **Processor : minimum intel i3**
* **Ram : minimum 4 gb**
* **Hard disk : minimum 250gb**

### **FUNCTIONAL REQUIREMENTS**

1.Data Collection

2.Data Preprocessing

3.Training And Testing

4.Modiling

5.Predicting

### **3.5 NON FUNCTIONAL REQUIREMENTS**

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *“how fast does the website load?”* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

* Usability requirement
* Serviceability requirement
* Manageability requirement
* Recoverability requirement
* Security requirement
* Data Integrity requirement
* Capacity requirement
* Availability requirement
* Scalability requirement
* Interoperability requirement
* Reliability requirement
* Maintainability requirement
* Regulatory requirement
* Environmental requirement

**3.6 SYSTEM STUDY**

**FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**4.SYSTEM DESIGN**

**4.1 SYSTEM ARCHITECTURE:**

Run Logistic Regression Algorithm

Dataset Preprocessing

Upload Internet Port Scan Dataset

Train decision tree, random forest, gradient boosting, XGBOOST, deep learning neural network and SVM

Comparison Graph

**DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

**Upload Internet Port Scan Dataset**

**VERIFY**

**NO PROCESS**

**Yes NO**

**Dataset Preprocessing**

**Run Logistic Regression Algorithm**

**Train decision tree, random forest, gradient boosting, XGBOOST, deep learning neural network and SVM**

**Comparison graph**

**End process**

**4.3 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

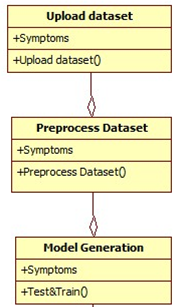
**Use case diagram:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



**Class diagram:**

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



**Activity diagram:**

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

Upload Internet Port Scan Dataset

Dataset pre-processing

Run logistic regression algorithm

Train decision tree, random forest, gradient boosting, XGBOOST, deep learning neural network and SVM

Comparison graph

**Sequence diagram:**

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".



**Collaboration diagram:**

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.



**Component diagram:**

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.



**Deployment diagram:**

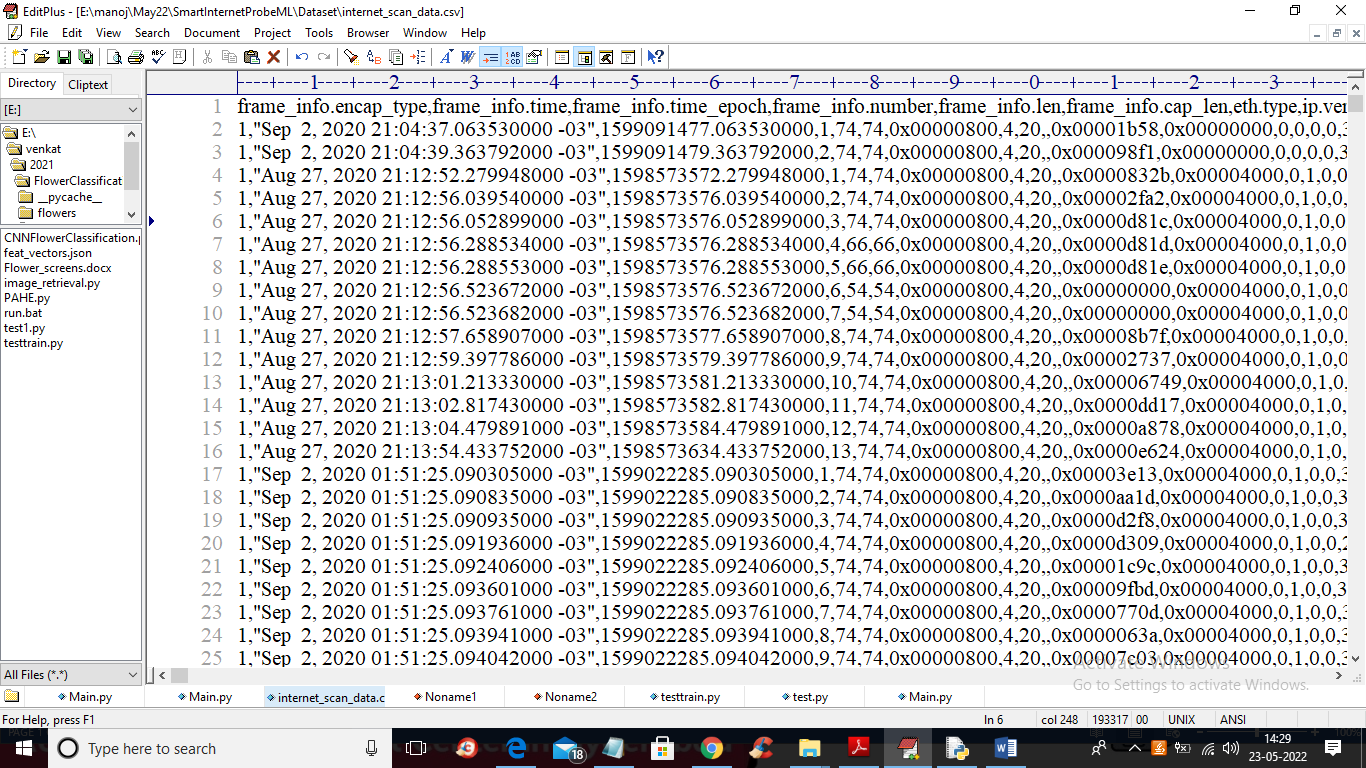
The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

****

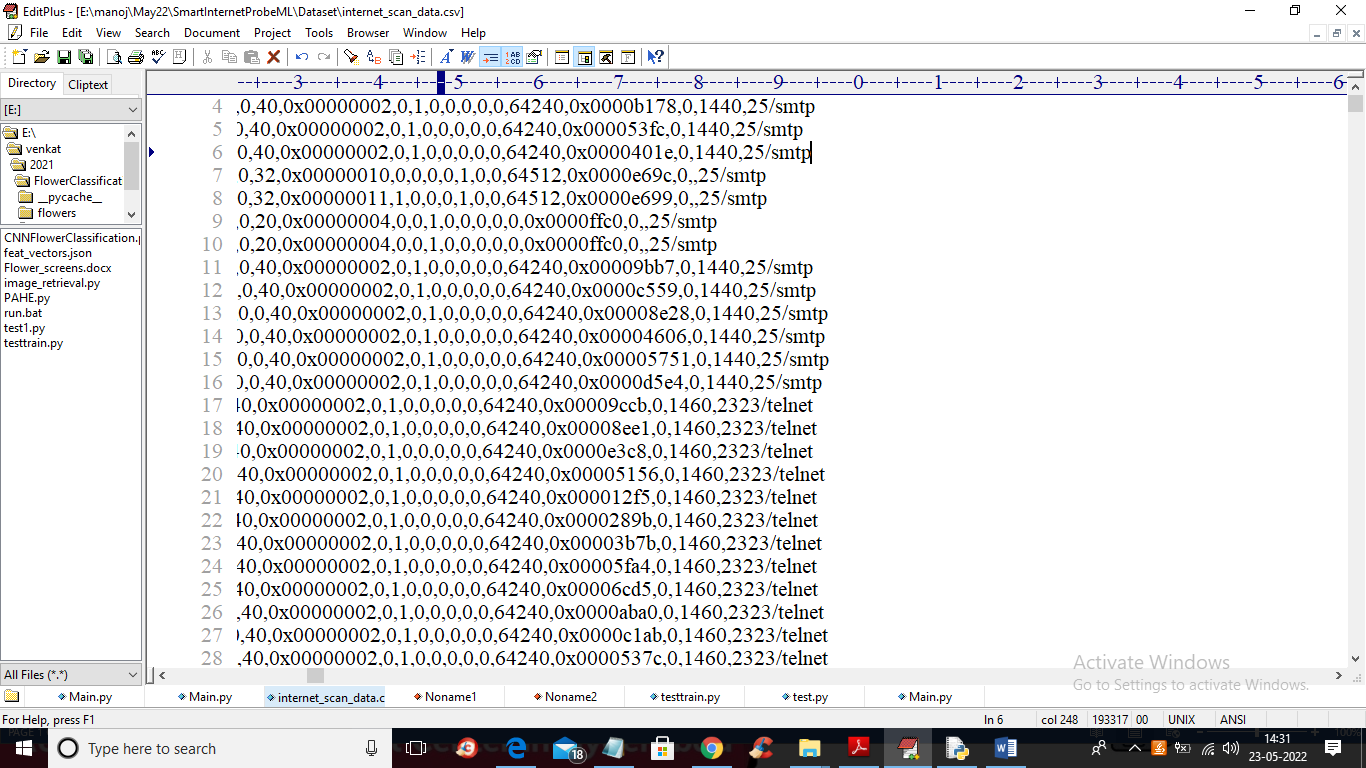
* 1. **IMPLEMENTATION:**

On internet all services are running on servers and this servers runs on IP address and PORT numbers and to access this services client will probe or scan PORT numbers to check servers are running or not. Sometime some servers will not run and probing such servers will increase network bandwidth and consume resources and to avoid such wastage of bandwidth and resources author of this paper employing machine learning algorithms to predict servers are running or not. Author saying PORTS and other details of all servers will share similar features and if one server is not running then machine learning can predict servers with similar features may also not run. Prediction of working port prior to scan/probe may save resources and bandwidth.

In propose paper author is using CENSYS dataset which scan 23 different PORTS and we are using same dataset. Below screen showing dataset details



In above dataset screen first row contains dataset column names and remaining rows contains dataset values and in last columns we can see labels as PORT and in below screen you can see those ports



In above screen in last column we can see smtp and telnet with port numbers are the labels used to predict weather port with similar features are working or not.

In propose paper for port classification author is using various machine learning algorithms called SVM, Logistic Regression, Decision Tree, Random Forest, Deep Learning Neural Network, Gradient Boosting and XGBOOST. Author evaluating each algorithm performance in terms of confusion matric and TPR (true positive rate). Algorithm with higher TPR will be consider as best. This algorithms can scan port PARALLELY and SEQUENTIALLY.

MODULES:

To implement this project we have designed following modules

1. Upload Internet Port Scan Dataset: using this module we will upload dataset to application and then display types of PORTS found in dataset
2. Dataset Preprocessing: dataset contains non-numeric and numeric data and machine learning algorithms only accept numeric data so by using module we will encode non-numeric data into numeric and then split dataset into train and test where application used 80% dataset for training and 20% for testing
3. Run Logistic Regression Algorithm: using this module we will train logistic regression by using 80% dataset and then apply regression model on 20% dataset to calculate TPR value. The higher the TPR the better is the algorithm
4. Similarly we will train decision tree, random forest, gradient boosting, XGBOOST, deep learning neural network and SVM and then calculate its TPR.
5. Comparison Graph: using this module we will plot TPR graph of all algorithms.

ALGORITHMS:

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is **used for predicting the categorical dependent variable using a given set of independent variables**. Logistic regression predicts the output of a categorical dependent variable.

Support Vector Machine(SVM) is **a supervised machine learning algorithm used for both classification and regression**. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

Decision trees **use multiple algorithms to decide to split a node into two or more sub-nodes**. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable.

Random forest is **a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems**. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Gradient boosting algorithm **can be used for predicting not only continuous target variable (as a Regressor) but also categorical target variable (as a Classifier)**. When it is used as a regressor, the cost function is Mean Square Error (MSE) and when it is used as a classifier then the cost function is Log loss.

XGBoost is **a scalable and highly accurate implementation of gradient boosting that pushes the limits of computing power for boosted tree algorithms**, being built largely for energizing machine learning model performance and computational speed.

Deep learning Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function f ( ⋅ ) : R m → R o by training on a dataset, where is the number of dimensions for input and is the number of dimensions for output.

**SAMPLE CODE:**

**%config** IPCompleter.greedy=True

**import** pandas **as** pd

**import** seaborn **as** sns

**import** numpy **as** np

**import** matplotlib **as** matplot

**import** matplotlib.pyplot **as** plt

**%matplotlib** inline

**from** IPython.core.interactiveshell **import** InteractiveShell

InteractiveShell**.**ast\_node\_interactivity **=** "all"

**import** warnings

warnings**.**filterwarnings("ignore")

**from** keras **import** Sequential

**from** keras.models **import** Model, load\_model

**from** keras.layers **import** **\***

**from** keras.callbacks **import** ModelCheckpoint

**from** keras **import** regularizers

**from** sklearn.metrics **import** **\***

**from** sklearn.tree **import** DecisionTreeClassifier

**from** sklearn.ensemble **import** RandomForestClassifier, ExtraTreesClassifier, VotingClassifier

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.preprocessing **import** LabelEncoder,normalize

**import** xgboost**,** lightgbm

**from** mlxtend.classifier **import** EnsembleVoteClassifier

train **=** pd**.**read\_csv('https://raw.githubusercontent.com/Nir-J/ML-Projects/master/UNSW-Network\_Packet\_Classification/UNSW\_NB15\_training-set.csv')

test **=** pd**.**read\_csv('https://raw.githubusercontent.com/Nir-J/ML-Projects/master/UNSW-Network\_Packet\_Classification/UNSW\_NB15\_testing-set.csv')

combined\_data **=** pd**.**concat([train, test])**.**drop(['id'],axis**=**1)

*# Contaminsation mean pollution (outliers) in data*

tmp **=** train**.**where(train['attack\_cat'] **==** "Normal")**.**dropna()

contamination **=** round(1 **-** len(tmp)**/**len(train), 2)

print("train contamination ", contamination)

tmp **=** test**.**where(test['attack\_cat'] **==** "Normal")**.**dropna()

print("test contamination ", round(1 **-** len(tmp)**/**len(test),2),'\n')

**if** contamination **>** 0.5:

print(f'contamination is {contamination}, which is greater than 0.5. Fixing...')

contamination **=** round(1**-**contamination,2)

print(f'contamination is now {contamination}')

le1 **=** LabelEncoder()

le **=** LabelEncoder()

vector **=** combined\_data['attack\_cat']

print("attack cat:", set(list(vector))) *# use print to make it print on single line*

combined\_data['attack\_cat'] **=** le1**.**fit\_transform(vector)

combined\_data['proto'] **=** le**.**fit\_transform(combined\_data['proto'])

combined\_data['service'] **=** le**.**fit\_transform(combined\_data['service'])

combined\_data['state'] **=** le**.**fit\_transform(combined\_data['state'])

vector **=** combined\_data['attack\_cat']

print('\nDescribing attack\_type: ')

print("min", vector**.**min())

print("max", vector**.**max())

print("mode",vector**.**mode(), "Which is,", le1**.**inverse\_transform(vector**.**mode()))

print("mode", len(np**.**where(vector**.**values**==**6)[0])**/**len(vector),"%")

*## OMITTED: For statistical feature removal*

lowSTD **=** list(combined\_data**.**std()**.**to\_frame()**.**nsmallest(6, columns**=**0)**.**index)

*# this is stupid. suppose a feature has a 1.0 (spearman or pearson) correlation, OR conditional probability, when not 0.... That a very useful feature*

lowCORR **=** list(combined\_data**.**corr()**.**abs()**.**sort\_values('attack\_cat')['attack\_cat']**.**nsmallest(3)**.**index) *# .where(lambda x: x < 0.005).dropna()*

*# This might be stupid. A Deep MLP (feed forward neural net) may see patterns*

drop **=** set( lowCORR **+** lowSTD)

drop **=** {'ackdat', 'ct\_ftp\_cmd', 'djit', 'is\_ftp\_login', 'is\_sm\_ips\_ports', 'response\_body\_len', 'sjit', 'synack', 'tcprtt'}

*# print(f'Before {combined\_data.shape}')*

combined\_data\_reduced**=**combined\_data *# .drop(drop,axis=1)*

*# print(f'After {combined\_data.shape}')*

**5.SOFTWARE ENVIRONMENT**

# What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like Kivy, Tkinter, PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

### Advantages of Python :-

Let’s see how Python dominates over other languages.

#### 1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

#### 2. Extensible

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

#### 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

#### 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### 6. Simple and Easy

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

#### 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

#### 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

#### 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

### **Advantages of Python Over Other Languages**

#### 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

#### 3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### **Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**History of Python : -**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

**What is Machine Learning : -**

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

**Categories Of Machine Leaning :-**

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

## Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

# How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### **How to start learning ML?**

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### (a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### (b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### (c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/), [TensorFlow](https://www.tensorflow.org/), [Scikit-learn](https://scikit-learn.org/stable/), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python) available Free on GeeksforGeeks.

### **Step 2 – Learn Various ML Concepts**

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (a) Terminologies of Machine Learning

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### (b) Types of Machine Learning

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### **Advantages of Machine learning :-**

#### 1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### 2. No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### 3. Continuous Improvement

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### 5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### **Disadvantages of Machine Learning :-**

#### 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

#### 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### 4. High error-susceptibility

[**Machine Learning**](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [**https://www.python.org**](https://www.python.org/)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**.

**6.SYSTEM TESTING**

**6.1 TESTING STRATEGIES**

**6.1.1 UNIT TESTING**

Unit testing, a testing technique using which individual modules are tested to determine if there are issues by the developer himself.. it is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Unit Testing Techniques:

Black Box Testing - Using which the user interface, input and output are tested.

White Box Testing –Used to test each one of those functions behavior is tested.

**6.1.2 DATA FLOW TESTING**

Data flow testing is a family of testing strategies based on selecting paths through the program’s control flow in order to explore sequence of events related to the status of Variables or data object. Dataflow Testing focuses on the points at which variables receive and the points at which these values are used.

**6.1.3 INTEGRATION TESTING**

Integration Testing done upon completion of unit testing, the units or modules are to be integrated which gives raise too integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

**6.1.4 BIG BANG INTEGRATION TESTING**

Big Bang Integration Testing is an integration testing Strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

**6.1.5 USER INTERFACE TESTING**

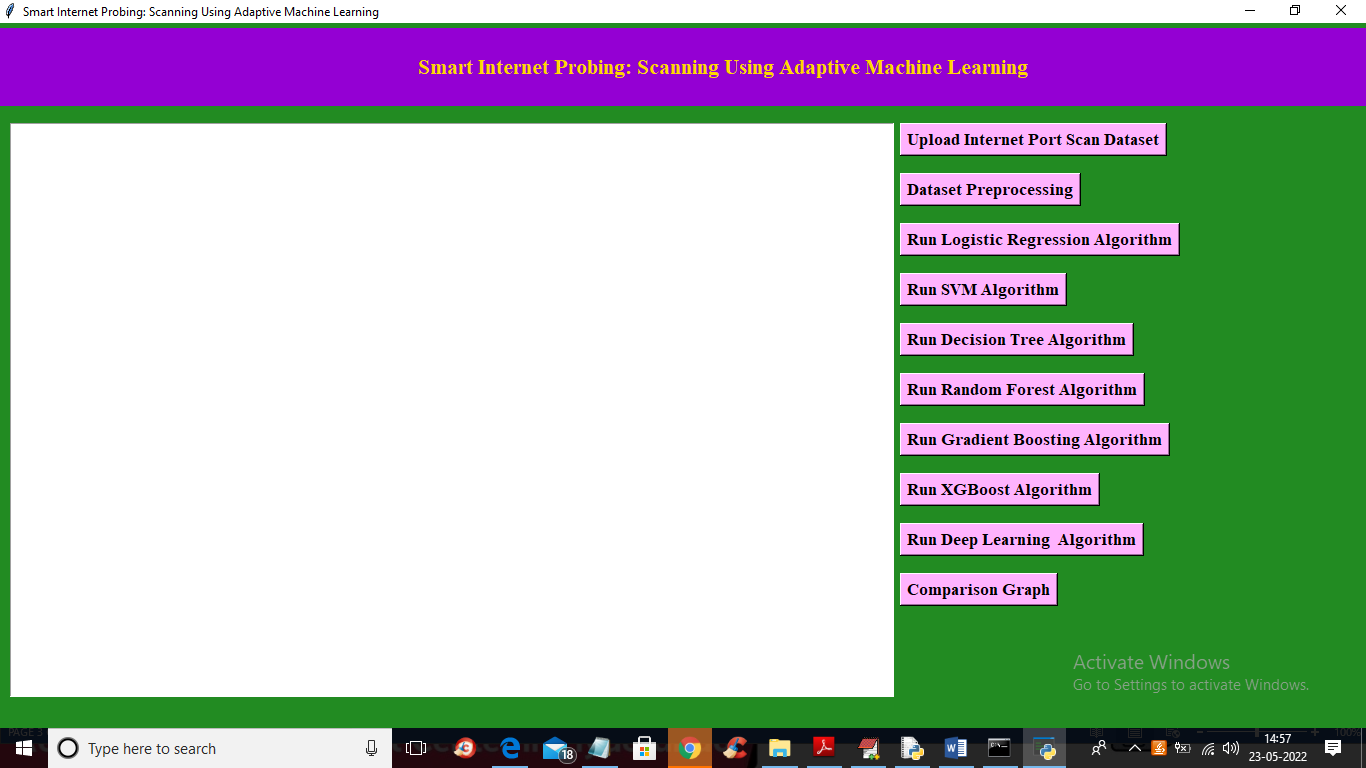
User interface testing, a testing technique used to identify the presence of defects is a product/software under test by Graphical User interface [GUI].

**6.2 TEST CASES:**

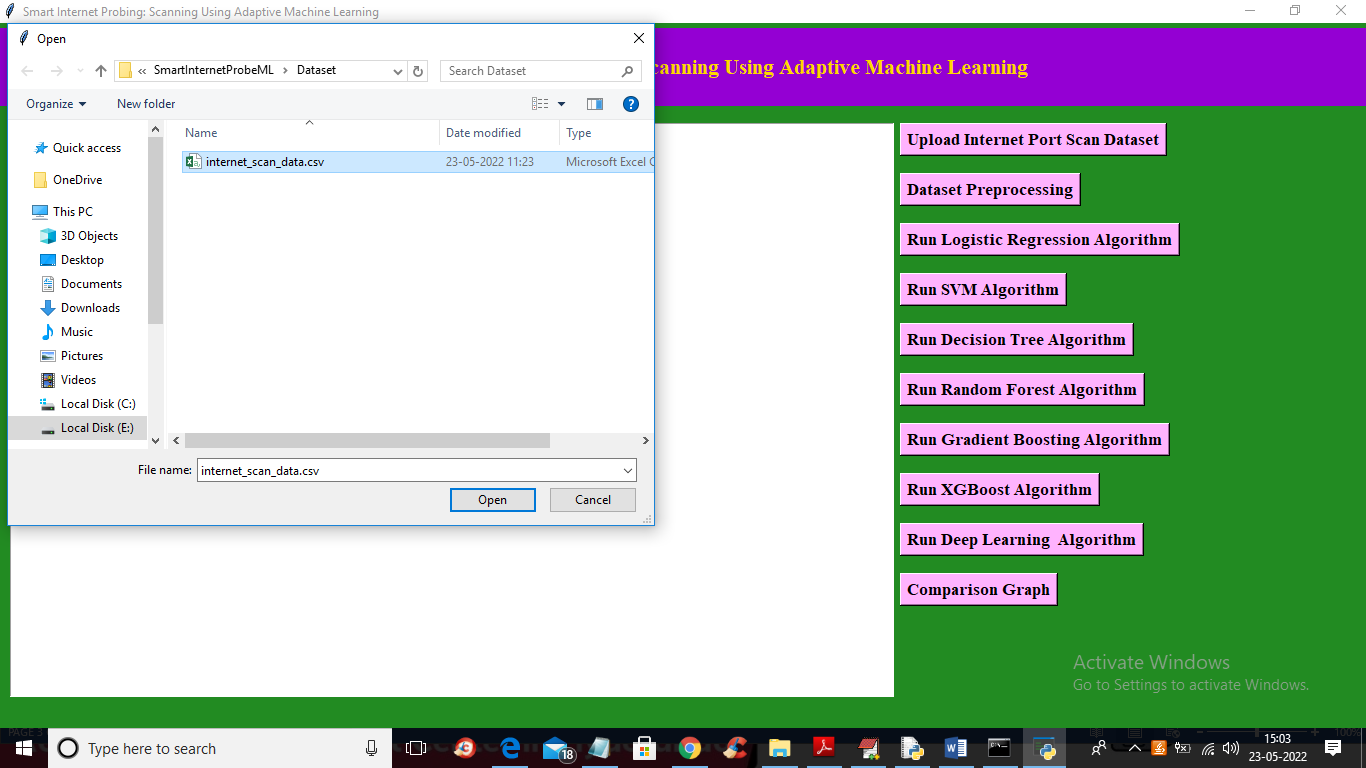
|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **INPUT** | **If available** | **If not available** |
| 1 | Upload Internet Port Scan Dataset | Dataset loaded | There is no process |
| 2 | Data preprocessing | will encode non-numeric data into numeric and then split dataset into train and test | There is no process |
| 3 | Run Logistic Regression Algorithm | calculate TPR value | There is no process |
| 4 | Run SVM Algorithm | calculate TPR value | There is no process |
| 5 | Run Decision tree Algorithm | calculate TPR value | There is no process |
| 6 | Run Random forest Algorithm | calculate TPR value | There is no process |
| 7 | Run Gradient boosting Algorithm | calculate TPR value | There is no process |
| 8 | Run XGBoost Algorithm | calculate TPR value | There is no process |
| 9 | Run Deep learning Algorithm | calculate TPR value | There is no process |
| 10 | Comparison graph | Graph dispayed | There is no process |

**7.SCREENSHOTS**

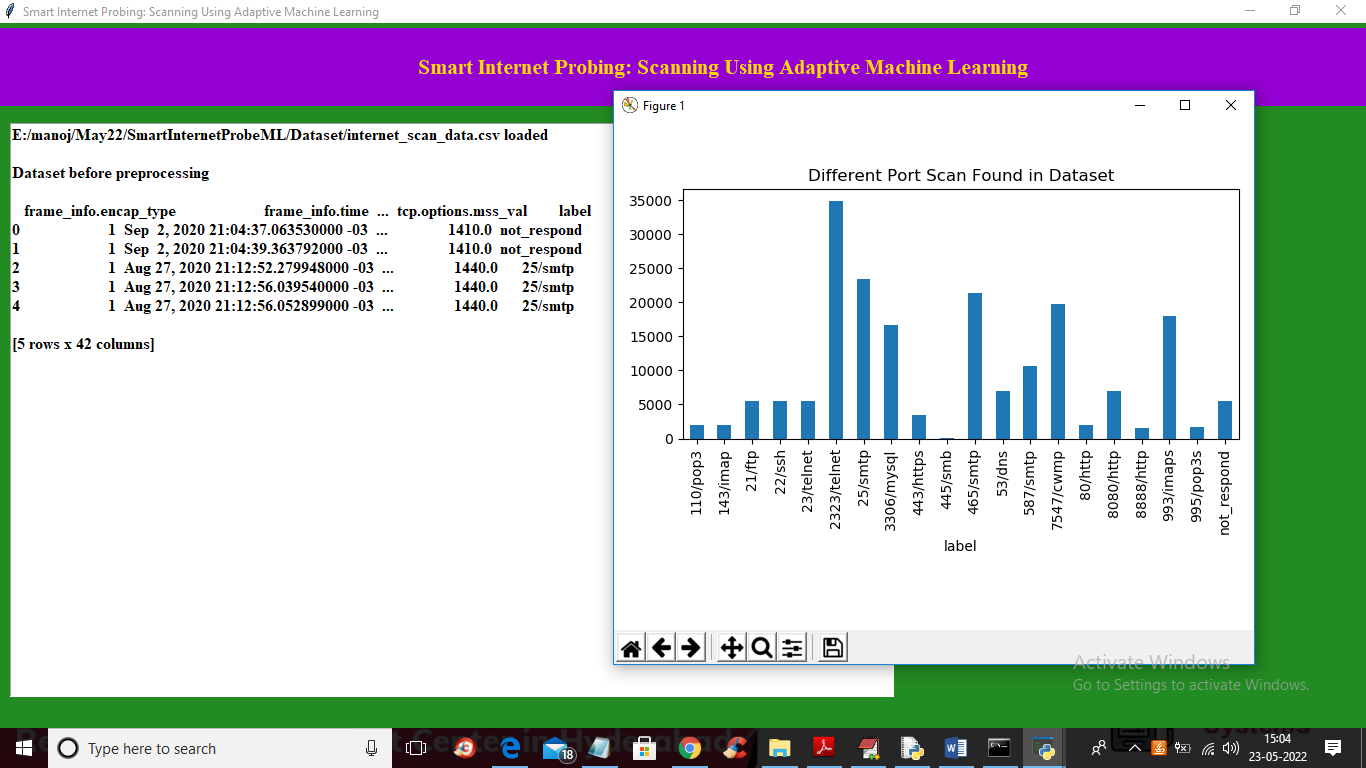
To run project double click on ‘run.bat’ file to get below output



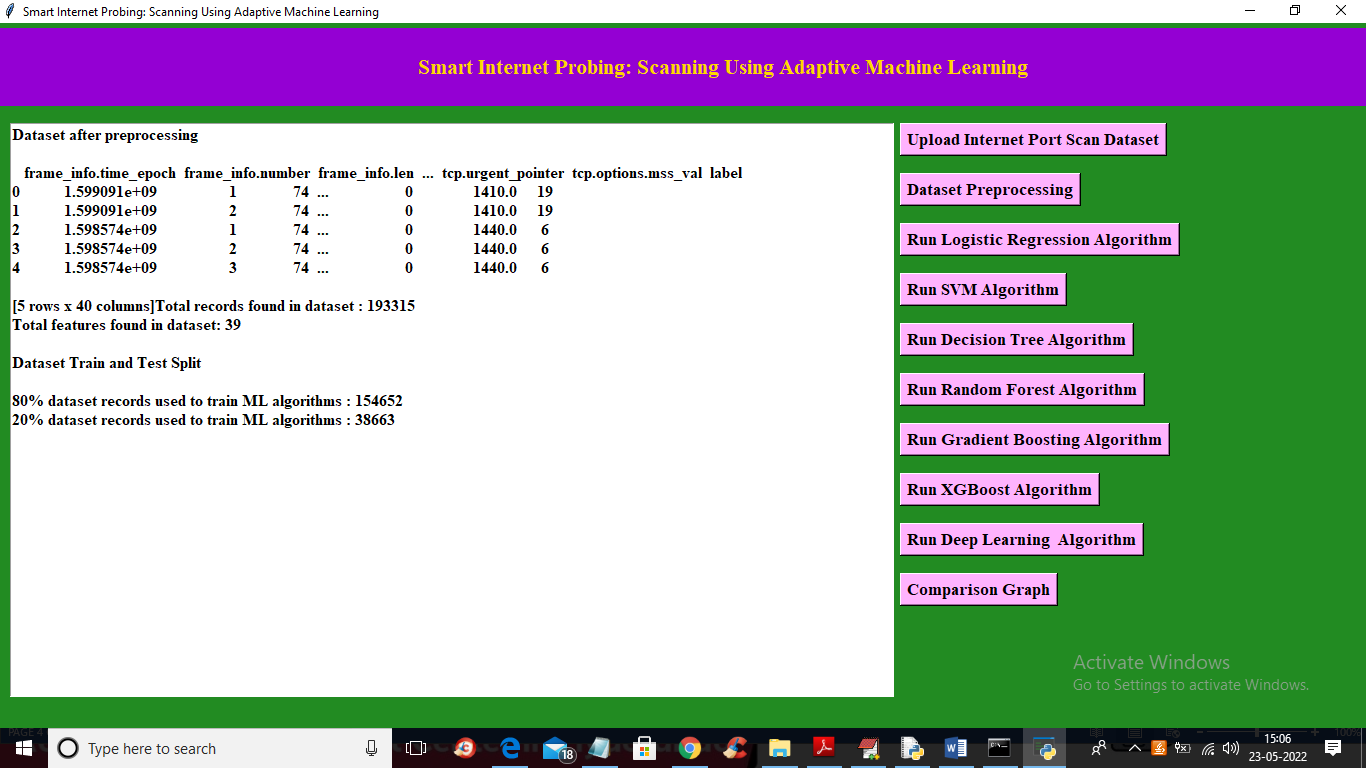
In above screen click on ‘Upload Internet Port Scan Dataset’ button to upload dataset and get below screen



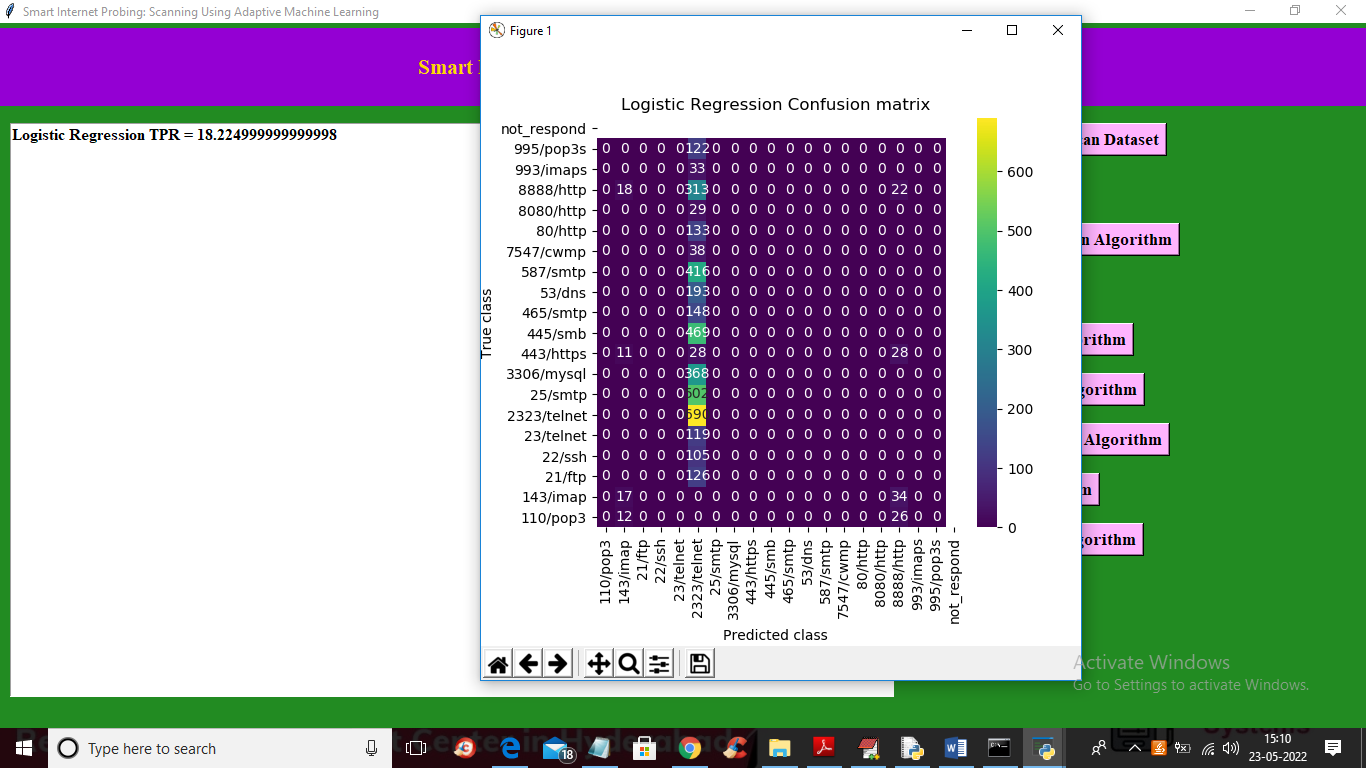
In above screen selecting and uploading dataset and then click on ‘Open’ button to load dataset and get below output



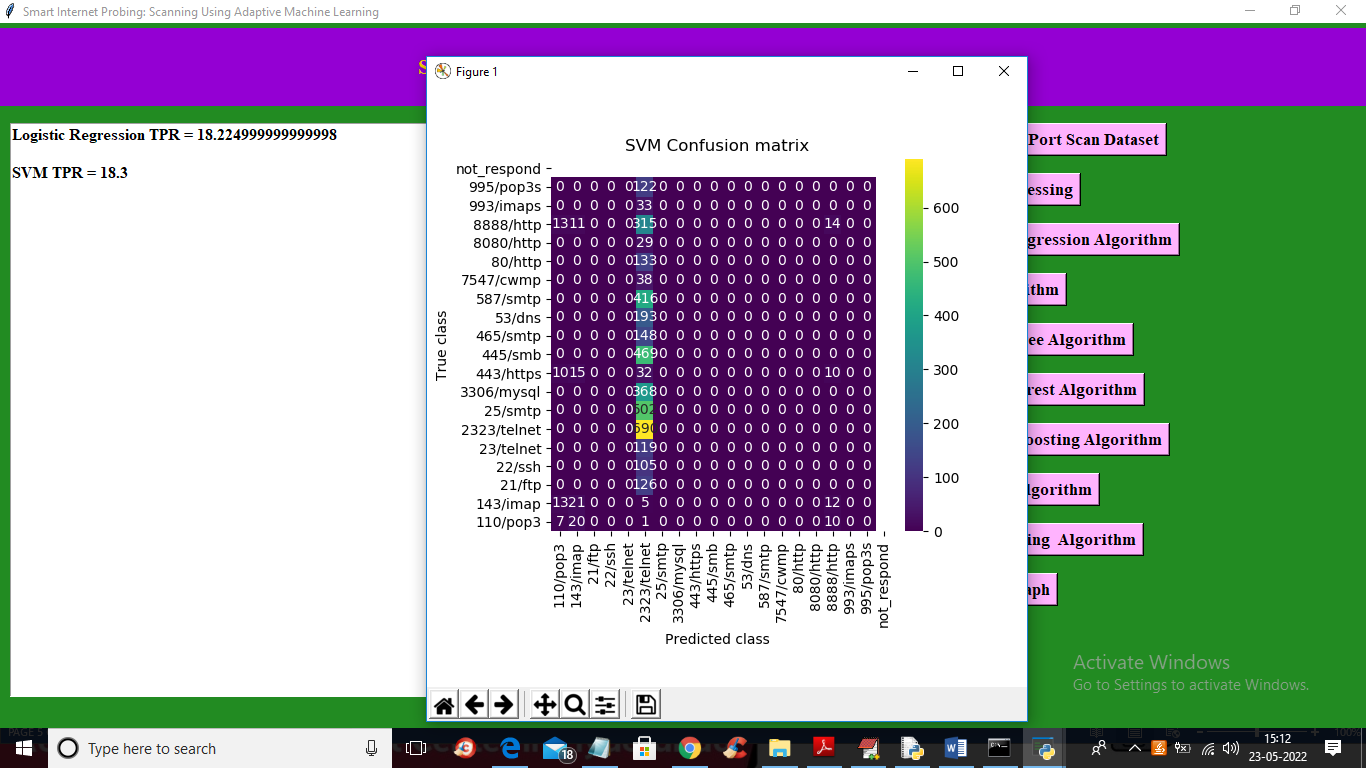
In above screen we can see dataset loaded and in dataset we can see it contains numeric and non-numeric data so we need to encode it to numeric by applying data Preprocessing. In above graph X-AXIS contains different ports and y-axis contains count of each port in dataset and now close above graph and then click on ‘Dataset Preprocessing’ button to process dataset and get below output



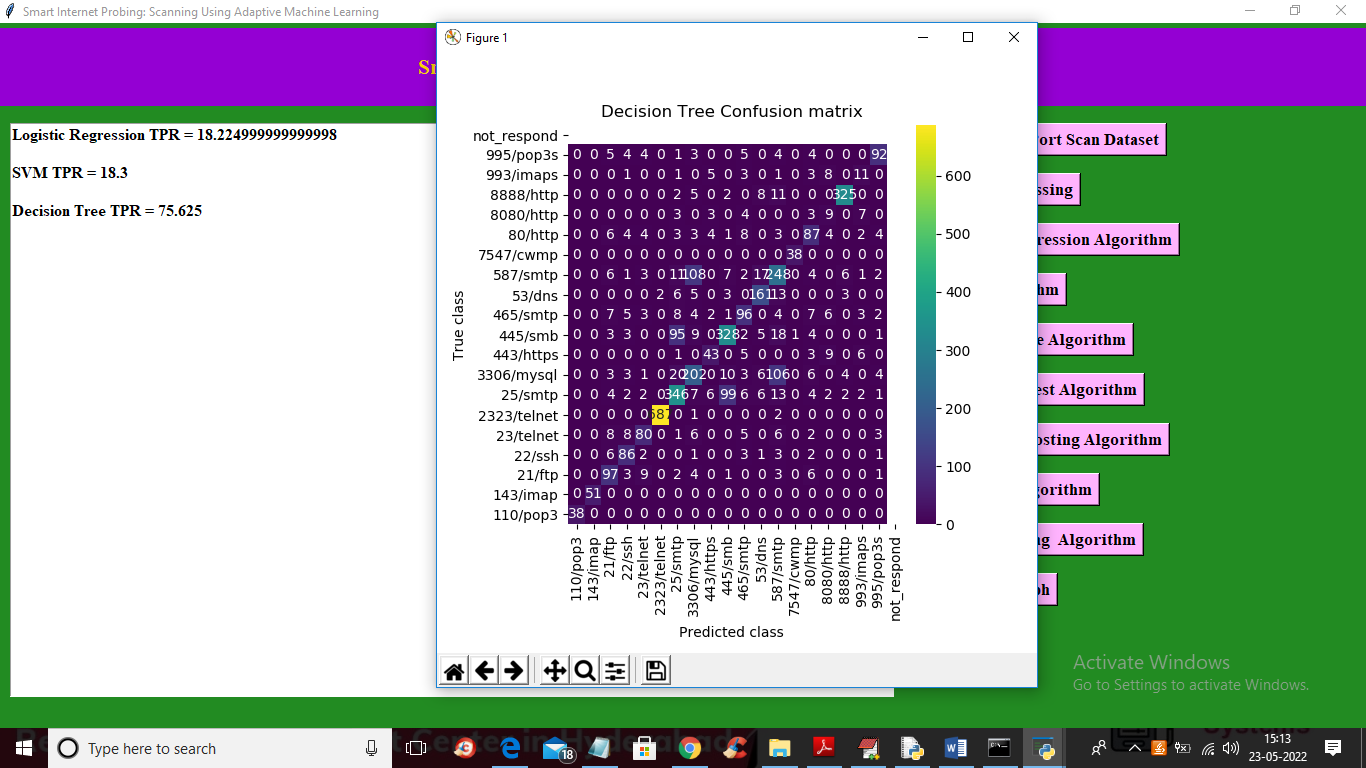
In above screen we can see all dataset values converted to numeric and we can see total records and column found in dataset and then we can see 80% training records size from dataset and 20% for testing. Now train and test data is ready and now click on ‘Run Logistic Regression Algorithm’ button to train algorithm and get below output



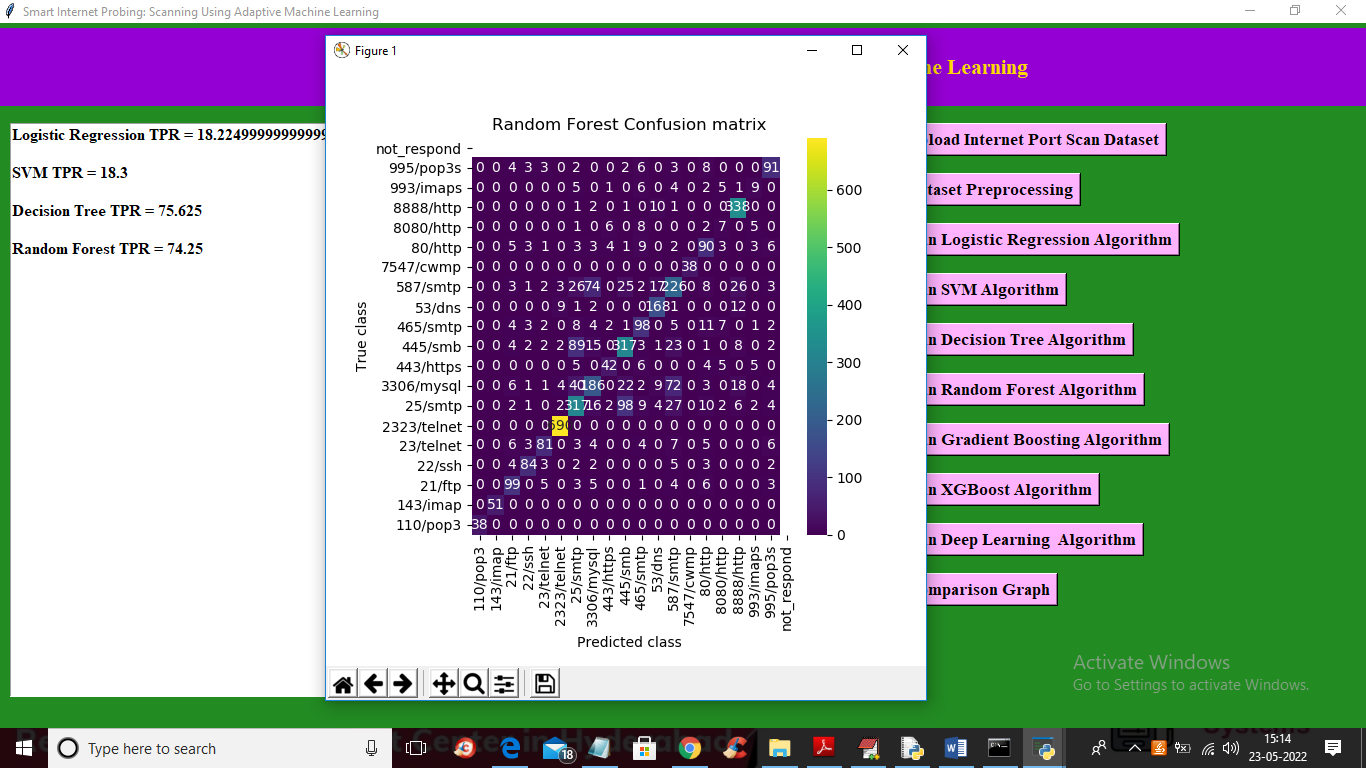
In above screen with logistic regression we got TPR as 18% which is not good and in confusion matrix x-axis represents predicted labels and y-axis represents TRUE labels and we can see all labels are predicted only in one class and now close above graph and then click on ‘Run SVM Algorithm’ to train SVM



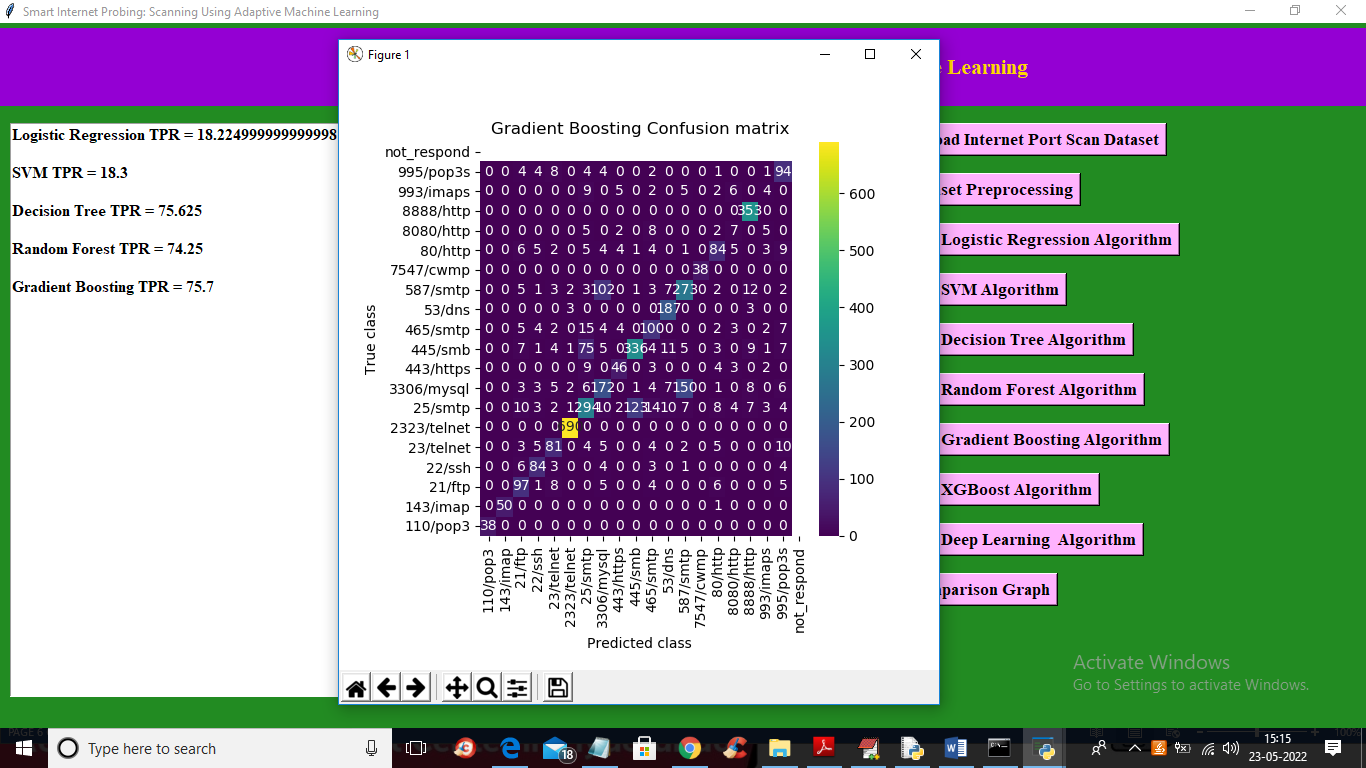
In above screen with SVM also we got only 18% TPR and now run decision tree algorithm and get below output



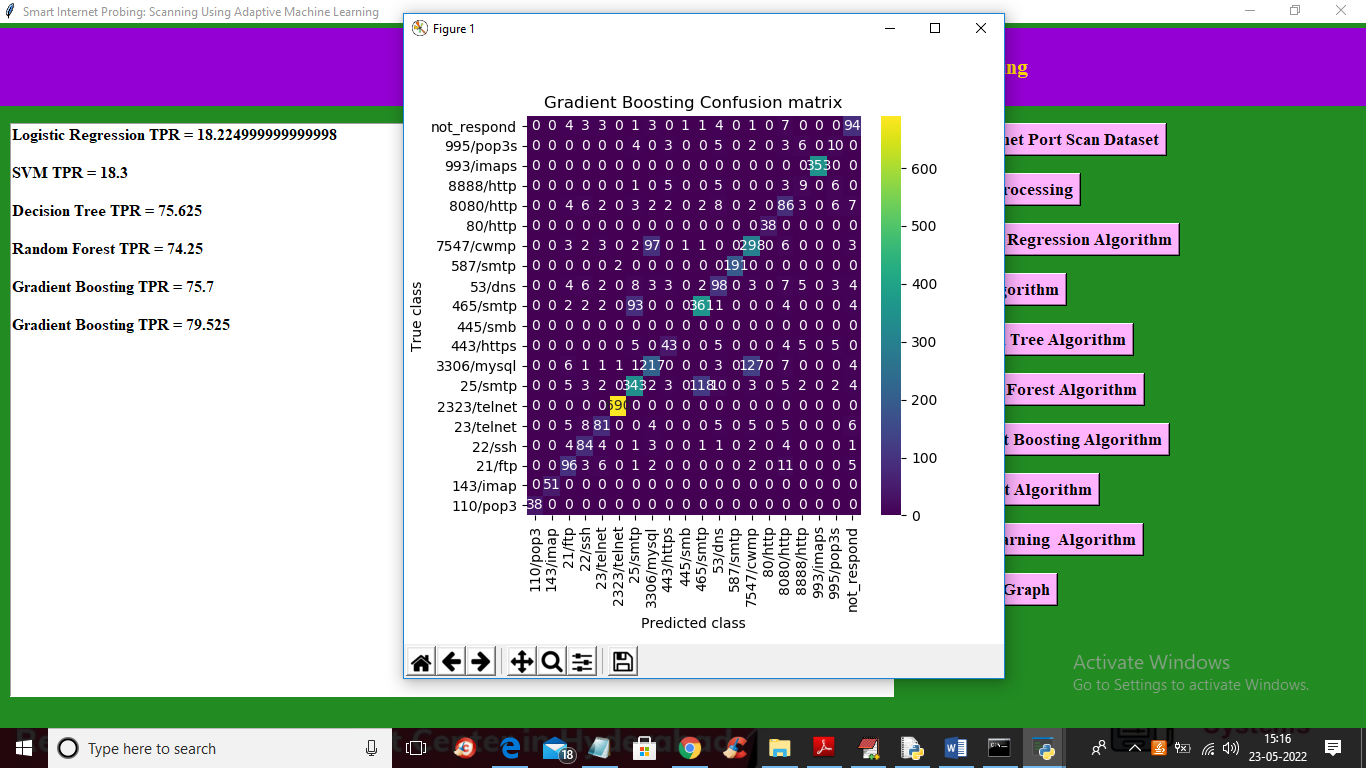
In above screen with decision tree we got 75% TPR and in confusion matrix in diagnol you can see more number of predicted and true classes are correct so its TPR is 75%. Now run Random Forest algorithm



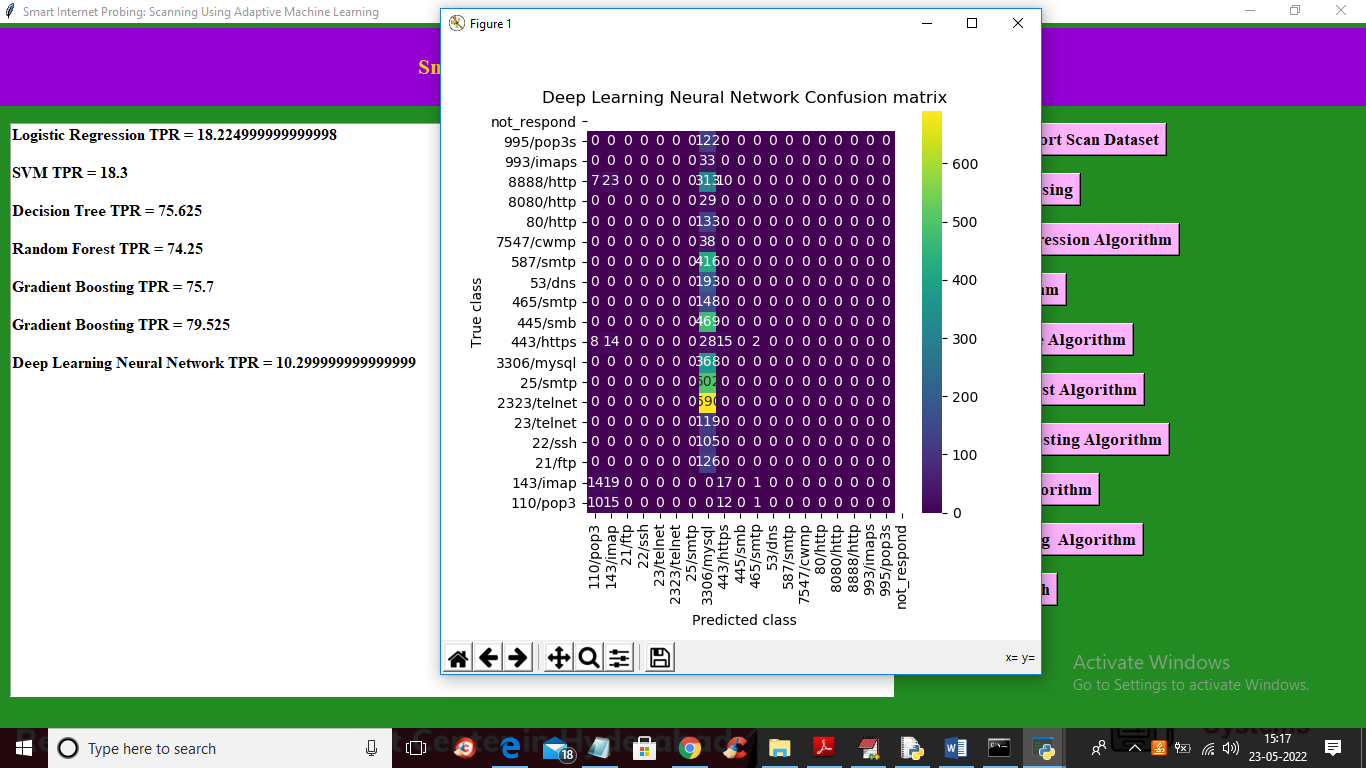
In above screen with random forest we got 74% TPR and now run gradient boosting algorithm



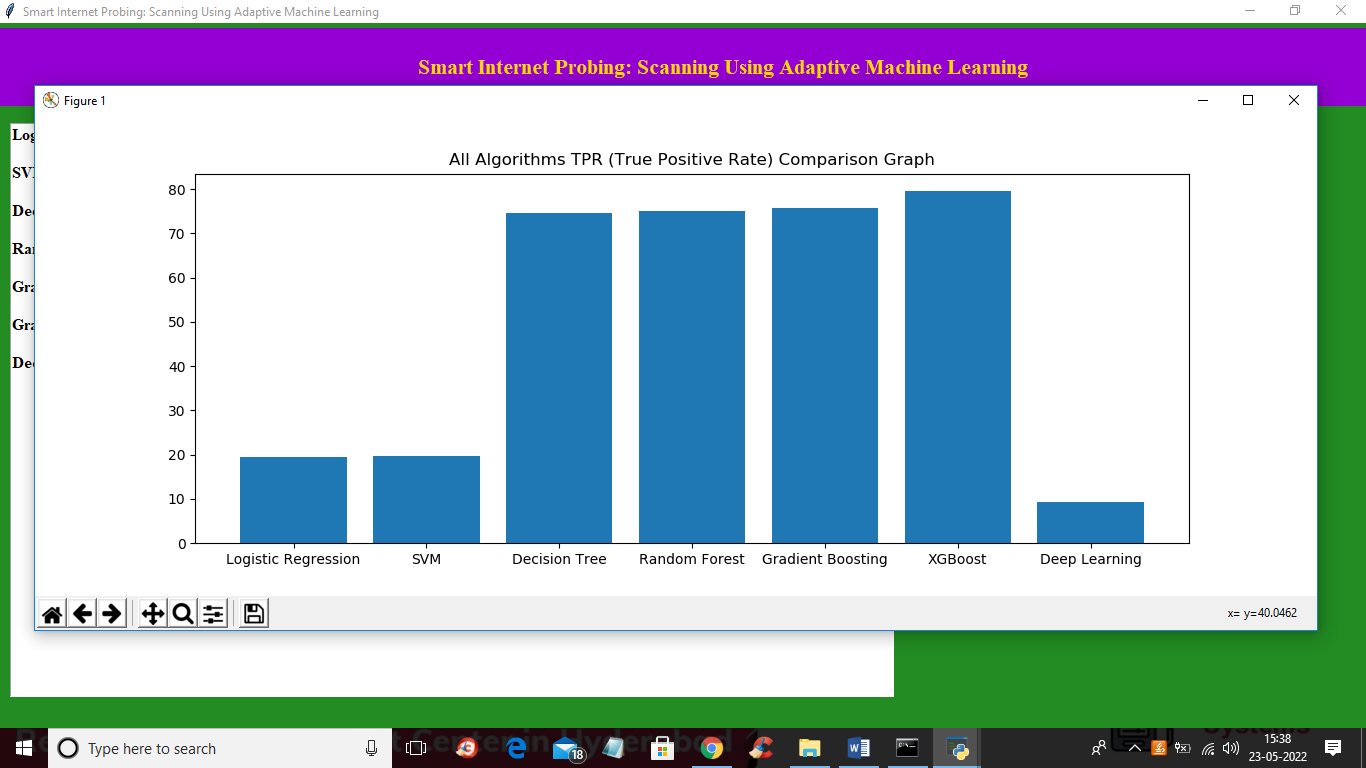
In above screen with gradient boosting we got 75.7% TPR and now run ‘XGBoost algorithm’



In above screen with XGBOOST we got 79% TPR and run Deep learning MLP algorithm



In above screen with deep learning algorithm we got 10% TPR and now click on ‘Comparison Graph’ button to get below graph



In above graph x-axis contains algorithm names and y-axis contains TPR and in all algorithms XGBOOST has got highest TPR. So by employing ML algorithms we can predict working port without probing/scanning

**8.CONCLUSION**

In this chapter, we developed and evaluated a framework for reducing the bandwidth of network scans by predicting whether a host will respond to requests on a certain port, using location and ownership (AS) properties, as well as cross-protocol information. We demonstrated that using only location and AS features we can achieve overall bandwidth savings of 26.7-72.0% at 90-99% true positive rates for detecting active/open ports, averaged over 20 port scans from the Censys database. Moreover, we developed a novel technique for finding an optimal order for scanning ports and training a sequence of classifiers, appending the responses of scanned ports for predicting active IPs over subsequent scans. We show that using this technique we can increase the bandwidth savings of Internet scans to 47.4-83.5% at 90-99% coverage levels. This reduction in bandwidth is due to the high dependency between the responses of certain ports, for instance ports corresponding to mail servers. We further show that our technique can be applied on top of current scanning tools with little computational overhead, providing blacklists in order to refrain from sending probes to certain IP/port pairs. We compared our methodology to other strategies for conducting machine learning enabled scans, concluding that ignoring location and AS properties results in poor performance, while using the full set of features from stateful scans only provides marginal benefits, while significantly increasing computational requirements. We also showed that scans have consistent coverage along vulnerable and misconfigured subpopulations, and are therefore appropriate for efficient and accurate assessment of the attack surface of networks.

FUTURE SCOPE:

We intend to apply our developed techniques to develop smart scanners that can efficiently scan IPv4 networks and IPv6 hitlists, increasing the hit rate for discovering active IPs. This allows scanners to scan IPv4 faster and less intrusively compared to exhaustive scans, while covering larger histlists for discovering more devices on IPv6. Additionally, using other sources of information, e.g., historical data, and local patterns in how devices are placed on the Internet (for example some networks might tend to put active devices at the start of their allocated IP blocks, while others might use random placement) can also help improve the efficiency of scans. Note that for sequential scans we are using a static order for probing different ports. However, it might be more efficient to change the order of scans for different networks, for instance scanning modem/router protocols first for consumer networks, while prioritizing web protocols for hosting networks. Using a dynamic order for scans is another direction for future work.

**9. REFERENCES**

[1] X. Fan and J. Heidemann, \Selecting representative IP addresses for Internet topology studies," in ACM SIGCOMM Conference on Internet Measurement. ACM, 2010, pp. 411{423.

[2] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson, \Target generation for Internet-wide IPv6 scanning," in Internet Measurement Conference. ACM, 2017, pp. 242{253.

[3] Z. Durumeric, E. Wustrow, and J. A. Halderman, \ZMap: Fast Internetwide scanning and its security applications." in USENIX Security Symposium, vol. 8, 2013, pp. 47{53.

[4] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, \A search engine backed by Internet-wide scanning," in ACM Conference on Computer and Communications Security. ACM, 2015, pp. 542{553.

[5] D. Leonard and D. Loguinov, \Demystifying service discovery: Implementing an Internet-wide scanner," in ACM SIGCOMM Conference on Internet Measurement. ACM, 2010, pp. 109{122.

[6] G. F. Lyon, Nmap network scanning: The official Nmap project guide to network discovery and security scanning. Insecure, 2009.

[7] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis et al., \Understanding the Mirai botnet," in USENIX Security Symposium, 2017, pp. 1092{1110.

[8] Y. Liu, A. Sarabi, J. Zhang, P. Naghizadeh, M. Karir, M. Bailey, and M. Liu, \Cloudy with a chance of breach: Forecasting cyber security incidents," in USENIX Security Symposium, 2015, pp. 1009{1024.

[9] Z. Durumeric, F. Li, J. Kasten, J. Amann, J. Beekman, M. Payer, N. Weaver, D. Adrian, V. Paxson, M. Bailey et al., \The matter of heartbleed," in Internet Measurement Conference. ACM, 2014, pp. 475{488.

[10] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, \Measuring HTTPS adoption on the web," in USENIX Security Symposium, 2017, pp. 1323{1338.