**A Mini Project Report On**

**DETECTION OF CYBER ATTACK IN NETWORK USING MACHINE LEARNING TECHNIQUES**

### Capstone project-II report submitted in partial fulfillment of the Requirements for the Award of the Degree of MASTER OF TECHNOLOGY

**In**

**CYBER SECURITY**

### Submitted by

**P . SUHAS 24EP03008**

### Under the Guidance of

**Dr.P.Dhanavanthini ,**

**Assistant Professor**



### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**SIDDHARTHA ACADEMY OF HIGHER EDUCATION**

**(Deemed to be University)**

**Kanuru, Vijayawada-520007, AP**

**SIDDHARTHA ACADEMY OF HIGHER EDUCATION**

(Deemed to be University)

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the CAPSTONE PROJECT- II course report entitled **”** DETECTION OF CYBER ATTACK IN NETWORK USING MACHINE LEARNING TECHNIQUES**”** being submitted by P. Suhas (24EP03008) in partial fulfilment for the award of the Degree of Master of Technology in Cyber Security to the Siddhartha Academy of Higher Education (Deemed to be University), is a record of bonafide work carried out during the period from 2024 - 2025.

**Dr.P.Dhanavanthini, Dr.D.Rajeswara Rao, M.Tech, Ph.D**

### Assistant Professor & Guide Professor & HOD, CSE

**DECLARATION**

I hereby declare that the Capstone Project-II entitled “ Detection of Cyber Attacks Using Machine Learning Techniques ” submitted for the M.Tech Degree is our original work and the dissertation has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles.

Place: Vijayawada

Date: 11-12-2024 P.Suhas (24EP03008)

**ACKNOWLEDGEMENT**

Behind every achievement lies an unfathomable sea of gratitude to those who activated it,without whom it would ever have come into existence. To them I lay the words of gratitude imprinted with us.

I would like to thank to the **professor P. Venkateswara Rao**, Vice-Chancellor and **Dr. A. V. Ratna Prasad**, Pro Vice-Chancellor of Siddhartha Academy of Higher Education for the facilities provided during the course of Capstone Project-I.

I have been bestowed with the privilege of thanking **Dr. D. Rajeswara Rao**, Professor and Head of the Department for his moral and material support. I would like to express my deep gratitude to my guide **Dr.P.Dhanavanthini** , Assistant professor for his persisting encouragement, everlasting patience and keen interest in discussion and for his numerous suggestions which I had at every phase of this project. I owe my acknowledgements to an equally long list

of people who helped us in this project work

Place: Vijayawada

Date: 19-12-2024 P.Suhas(24EP03008)

**ABSTRACT**

In today's digitally connected world, cyber attacks have become increasingly sophisticated, posing significant threats to network security. Traditional rule-based intrusion detection systems often fail to detect novel or evolving attack patterns. This project focuses on enhancing network security by leveraging machine learning techniques to detect cyber attacks effectively. The system is trained using labeled datasets containing various types of network traffic, including both normal and malicious activities. Different supervised machine learning algorithms—such as Decision Trees, Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN)—are applied and compared to evaluate their effectiveness in identifying threats. Performance metrics like accuracy, precision, recall, and F1-score are used to assess model efficiency. The results demonstrate that machine learning models can significantly improve the detection rate of cyber attacks, offering a scalable and adaptive approach to network intrusion detection.

**Keywords:** Cybersecurity, Intrusion Detection System (IDS), Machine Learning, Network Traffic Analysis, Cyber Attack Detection, Supervised Learning, Classification Algorithms, Anomaly Detection, Network Security, Threat Intelligence.

**TABLE OF CONTENTS**

**[Abstract](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960589)** [x](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960589)

[1 Introduction 3](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960590)

[1.1 Background : 3](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960591)

[1.2 Binary Classification: 2](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960592)

[1.3 Existing System: 5](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960593)

[1.4 Decision Trees: 4](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960595)

[1.6 Random Forest: 5](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960596)

[1.7 Support Vector Machine: 6](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960597)

1.8 K-Nearest Neighbors (KNN)………………………………………………..7

[2 Literature Survey 10](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960601)

2.1 Hardware Requirements………………………….……………..………………………………….12

2.2 Related Work :……………………………………………………………….13

[3 Methodology 18](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960602)

[3.1 Existing System: 12](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960603)

[3.1.1 Functionality: 12](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960604)

[3.1.2 Detection Mechanisms: 12](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960605)

[3.1.3 Use Cases: 12](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960606)

[3.1.4 Key Principles of IDS Systems: 12](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960607)

[3.1.5 Conclusion: 13](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960608)

[3.2 Proposed System: 14](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960609)

[3.2.1 Key Components: 14](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960610)

[3.2.2 Proposed Workflow: 15](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960611)

[3.2.3 Benefits: 16](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960612)

[4 Requirements 20](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960613)

[4.1 Hardware Requirements 30](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960614)

[4.1.1 Processor (CPU): 31](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960615)

[4.1.2 Memory (RAM):](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960616) 32

[4.1.3 Storage: 33](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960617)

[4.1.4 Network: 34](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960618)

[4.2 Software Requirements 45](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960619)

[5 Testing 59](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960627)

[5.1 Software Testing 59](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960628)

[6 Results and Discussion 68](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960648)

[7 Conclusion 74](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960657)

[8 References 75](file:///C:\\Users\\sri\\Downloads\\final_report.docx" \l "_Toc164960664)

# 

# 1 INTRODUCTION

Contrasted with the past, improvements in PC and correspondence innovations have given broad and propelled changes. The use of new innovations give incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults.Right now, learning the bolster support vector machine (SVM) calculations were utilized to recognize port sweep endeavors dependent on the new CICIDS2017 dataset with 97.80%, 69.79% precision rates were accomplished individually. Rather than SVM we can introduce some other algorithms like random forest, CNN, ANN where these algorithms can acquire accuracies like SVM – 93.29, CNN – 63.52, Random Forest – 99.93, ANN – 99.11.

## 1.1 Back ground :

The use of new innovations give incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created tomaintain a strategic distance from digital assaults.

## 1.2 Significance :

The use of new innovations give incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, t protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults.protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults.

## 1.3Existing System:

Blameless Bayes and Principal Component Analysis (PCA) were been used with the KDD99 dataset by Almansob and Lomte [9].Similarly, PCA, SVM, and KDD99 were used Chithik and Rabbani for IDS [10]. In Aljawarneh et al's. Paper, their assessment and examinations were conveyed reliant on the NSL-KDD dataset for their IDS model

[11] Composing inspects show that KDD99 dataset is continually used for IDS [6]– [10].There are 41 highlights in KDD99 and it was created in 1999. **Consequently, KDD99 is old and doesn't give any data about cutting edge new assault types**, example, multi day misuses and so forth. In this manner we utilized a cutting-edge and new CICIDS2017 dataset [12] in our investigation.

## Limitations of existing system:

* Strict Regulations
* Difficult to work with for non-technical users
* Restrictive to resources
* Constantly needs Patching
* Constantly being attacked

## Objectives:

Objective of this project is to detect cyber attacks by using machine learning algorithms like

* + ANN
  + CNN
  + Random fores

## Outcomes:

These predictions can be done by four algorithms like SVM, ANN, RF, CNN this paper helps to identify which algorithm predicts the best accuracy rates which helps to predict best results to identify the cyber attacks happened or not.

## Proposed System:

important steps of the algorithm are given in below. 1) Normalization of every dataset. 2) Convert that dataset into the testing and training. 3) Form IDS models with the help of using RF, ANN, CNN and SVM algorithms. 4) Evaluate every model’s performances

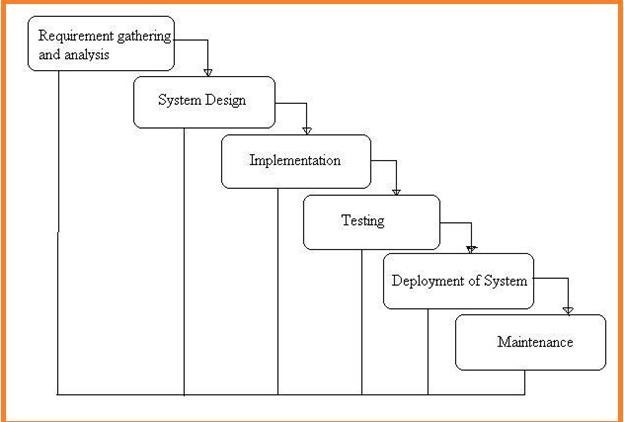
## Advantages:

* Protection from malicious attacks on your network.
* Deletion and/or guaranteeing malicious elements within a preexisting network.
* Prevents users from unauthorized access to the network.
* Deny's programs from certain resources that could be infected.

## Securing confidential information

## 

#### STRUCTURE OF PROJECT (SYSTEMANALYSIS)



**1.6.0 SYSTEMDESIGN**

In System Design has divided into three types like GUI Designing, UML Designing with avails in development of project in facile way with different actor and its utilizer case by utilizer case diagram, flow of the project utilizing sequence, Class diagram gives information about different class in the project with methods that have to be utilized in the project if comes to our project our UML Will utilizable in this way The third and post import for the project in system design is Data base design where we endeavor to design data base predicated on the number of modules in ourproject

#### 

#### 1.6.1 IMPLEMENTATION

The Implementation is Phase where we endeavor to give the practical output of the work done in designing stage and most of Coding in Business logic lay coms into action in this stage its main and crucial part of the project

#### 1.6.4TESTING UNITTESTING

It is done by the developer itself in every stage of the project and fine-tuning the bug and module predicated additionally done by the developer only here we are going to solve all the runtime errors

#### MANUAL TESTING

As our Project is academic Leave, we can do any automatic testing so we follow manual testing by endeavor and error methods

#### 1.6.1 DEPLOYMENT OF SYSTEM AND MAINTENANCE

Once the project is total yare, we will come to deployment of client system in genuinely world as its academic leave we did deployment i our college lab only with all need Software’s withhaving Windows OS.

The Maintenance of our Project is one-time process only

#### FUNCTIONALREQUIREMENTS

1.Data Collection 2.DataPreprocessing

3.Training And Testing 4.Modiling 5.Predicting

#### NON FUNCTIONALREQUIREMENTS

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 functionalrequirement.

* Usabilityrequirement
* Serviceabilityrequirement
* Manageabilityrequirement
* Recoverabilityrequirement
* Securityrequirement
* Data Integrity requirement
* Capacityrequirement
* Availabilityrequirement
* Scalabilityrequirement
* Interoperabilityrequirement
* Reliabilityrequirement
* Maintainabilityrequirement
* Regulatoryrequirement
* Environmentalrequirement

#### EXAMPLES OF NON-FUNCTIONALREQUIREMENTS

Here, are some examples of non-functional requirement:

* + - 1. Users must upload dataset
      2. The software should be portable. So moving from one OS to other OS does not createany problem.
      3. Privacy of information, the export of restricted technologies, intellectual property rights, etc. should beaudited.

Benefits/pros of Non-functional testing are:

* The nonfunctional requirements ensure the software system follow legal andcompliance rules.
* They ensure the reliability, availability, and performance of the softwaresystem
* They ensure good user experience and ease of operating thesoftware.
* They help in formulating security policy of the softwaresystem.

#### DISADVANTAGES OF NON-FUNCTIONALREQUIREMENT

Cons/drawbacks of Non-function requirement are:

* None functional requirement may affect the various high-level softwaresubsystem
* They require special consideration during the software architecture/high-level design phase which increasescosts.
* Their implementation does not usually map to the specific softwaresub-system,
* It is tough to modify non-functional once you pass the architecturephase.

#### DISADVANTAGES OF NON-FUNCTIONALREQUIREMENT

Cons/drawbacks of Non-function requirement are:

* None functional requirement may affect the various high-level softwaresubsystem
* They require special consideration during the software architecture/high-level design phase which increasescosts.
* Their implementation does not usually map to the specific softwaresub-system,

It is tough to modify non-functional once you pass the architecturephase

**2 LITERATURE SURVEY**

##### R. Christopher, “Port scanning techniques and the defense against them,” SANS Institute, 2001

Port Scanning is one of the most popular techniques attackers use to discover services that they can exploit to break into systems. All systems that are connected to a LAN or the Internet via a modem run services that listen to well- known and not so well-known ports. By port scanning, the attacker can find the following information about the targeted systems: what services are running, what users own those services, whether anonymous logins are supported, and whether certain network services require authentication. Port scanning is accomplished by sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can be probed for further weaknesses. Port scanners are important to network security technicians because they can reveal possible security vulnerabilities on the targeted system. Just as port scans can be ran against your systems, port scans can be detected and the amount of information about open services can be limited utilizing the proper tools. Every publicly available system has ports that are open and available for use. The object is to limit the exposure of open ports to authorized users and to deny access to the closed ports.

##### S. Staniford, J. A. Hoagland, and J. M. McAlerney, “Practical automated detection of stealthy portscans,” Journal of Computer Security, vol. 10, no. 1-2,

**pp. 105–136, 2002**

Portscanning is a common activity of considerable importance. It is often used by computer attackers to characterize hosts or networks which they are considering hostile activity against. Thus it is useful for system administrators and other network defenders to detect portscans as possible preliminaries to a more serious attack. It is also widely used by network defenders to understand and find vulnerabilities in their own networks. Thus it is of considerable interest to attackers to determine whether or not the defenders of a network are portscanning it regularly. However, defenders will not usually wish to hide their portscanning, while attackers will. For definiteness, in the remainder of this paper, we will speak of the attackers scanning the network, and the defenders trying to detect the scan. There are several legal/ethical debates about portscanning which break out regularly on Internet mailing lists and newsgroups. One concerns whether portscanning of remote networks without permission from the owners is itself a legal and ethical activity. This is presently a grey area in most jurisdictions. However, our experience from following up on unsolicited remote portscans we detect in practice is that almost all of them turn out to have come from compromised hosts and thus are very likely to be hostile. So we think it reasonable to consider a portscan as at least potentially hostile, and to report it to the administrators of the remote network from whence it came. However, this paper is focussed on the technical questions of how to detect portscans, which are independent of what significance one imbues them with, or how one chooses to respond to them. Also, we are focussed here on the problem of detecting a portscan via a network intrusion detection system (NIDS). We try to take into account some of the more obvious ways an attacker could use to avoid detection, but to remain with an approach that is practical to employ on busy networks. In the remainder of this section, we first define portscanning, give a variety of examples at some length, and discuss ways attackers can try to be stealthy. In the next section, we discuss a variety of prior work on portscan detection. Then we present the algorithms that we propose to use, and give some very preliminary data justifying our approach. Finally, we consider possible extensions to this work, along with other applications that might be considered.

Throughout, we assume the reader is familiar with Internet protocols, with basic ideas about network intrusion detection and scanning, and with elementary probability theory, information theory, and linear algebra. There are two general purposes that an attacker might have in conducting a portscan: a primary one, and a secondary one.

The primary purpose is that of gathering information about the reachability and status of certain combinations of IP address and port (either TCP or UDP). (We do not directly discuss ICMP scans in this paper, but the ideas can be extended to that case in an obvious way.) The secondary purpose is to flood intrusion detection systems with alerts, with the intention of distracting the network defenders or preventing them from doing their jobs. In this paper, we will mainly be concerned with detecting information gathering portscans, since detecting flood portscans is easy. However, the possibility of being maliciously flooded with information will be an important consideration in our algorithm design. We will use the term scan footprint for the set of port/IP combinations which the attacker is interested in characterizing. It is helpful to conceptually distinguish the footprint of the scan, from the script of the scan, which refers to the time sequence in which the attacker tries to explore the footprint. The footprint is independent of aspects of the script, such as how fast the scan is, whether it is randomized, etc. The footprint represents the attacker’s information gathering

requirements for her scan, and she designs a scan script that will meet those requirements, and perhaps other non-information-gathering requirements (such as not being detected by an NIDS). The most common type of portscan footprint at present is a horizontal scan. By this, we mean that an attacker has an exploit for a particular service, and is interested in finding any hosts that expose that service. Thus she scans the port of interest on all IP addresses in some range of interest. Also at present, this is mainly being done sequentially on TCP port 53 (DNS)

#### 2.1 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**

# 2.3 Related Work :

This segment presents different late achievements around here. It ought to be noticed that we just examine the work that have utilized the NSL-KDD dataset for their perfor mance benchmarking. Subsequently, any dataset alluded from here on out ought to be considered as NSL-KDD. This methodology permits a more exact examination of work with other found in the writing. Another restriction is the utilization of preparing information for both preparing and testing by most work. At long last, we examine a couple of profound learning based methodologies that have been attempted so far for comparable sort of work. One of the most punctual work found in writing utilized ANN with improved strong back-spread for the plan of such an IDS [6]. This work utilized just the preparation dataset for preparing (70%), approval (15%) and testing (15%). As expected, utilization of unlabelled information for testing brought about a reduction of execution. A later work utilized J48 choice tree classifier with 10-overlay cross-approval for testing on the preparation dataset [4]. This work utilized a decreased list of capabilities of 22 highlights rather than the full arrangement of 41 highlights. A comparable work assessed different well known regulated tree-based classifiers and tracked down that Random Tree model performed best with the most extensive level of exactness alongside a decreased bogus alert rate [5]. Numerous 2- level characterization approaches have likewise been master presented. One such work utilized Discriminative Multinomial Naive Bayes (DMNB) as a base classifier and Nominal-to Binary directed separating at the second level alongside 10-crease cross approval for testing [9]. This work was hide the reached out to utilize Ensembles of Balanced Nested Dichotomies (END) at the main level and Random Forest at the second level [10]. True to form, this upgrade resulted in an improved location rate and a lower bogus positive rate. Another 2-level execution utilized head segment examination (PCA) for the list of capabilities decrease and afterward SVM (utilizing Radial Basis Function) for last classification, brought about a high recognition precision with just the preparation dataset and full 41 highlights set. A decrease in features set to 23 came about in far better location exactness in a portion of the assault classes, however the general execution was diminished [11]. The creators improved their work by utilizing data gain to rank the highlights and afterward a conduct based element determination to lessen the list of capabilities to

20. This brought about an improvement in detailed precision utilizing the preparation dataset [12]. The subsequent class to take a gander at, utilized both the preparation and test dataset. An underlying endeavour in this classification utilized fluffy characterization with hereditary calculation and came about in a detection precision of 80%+ with a low bogus positive rate [13]. Another significant work

# PURPOSE :

The DARPA's program for ID assessment of 1998 was overseen and arranged by Lincoln Labs of MIT. The primary target of this is to investigate and lead research in ID. A normalized dataset was arranged, which included different sorts of interruptions which imitated a military climate and was made freely accessible. The KDD interruption location challenge's dataset of 1999 was an all around refined rendition of this. The DARPA's ID assessment bunch, amassed network based information of IDS by reenactment of an aviation based armed forces base LAN by over 1000s of UNIX hubs and for ceaselessly 9 weeks, 100s of clients at a given time in Lincoln Labs which was then partitioned into 7 and fourteen days of preparing and testing individually to remove the crude dump information TCP. MIT's lab with broad monetary help from DARPA and AFRL, utilized Windows and UNIX hubs for practically the entirety of the inbound interruptions from an estranged LAN dissimilar to other OS hubs. With the end goal of dataset, 7 unmistakable situations and 32 particular assaults which totals up to 300 assaults were recreated. Since the time of arrival of KDD-'99' dataset, it is the most tremendously used information for assessing a few IDSs. This dataset is gathered by right around 4,900,000 individual associations which incorporates a component check of 41

#### UML DIAGRAMS

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and externalinterfaces.

##### Global Use Case Diagrams:

Identification of actors:

**Actor:** Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

Graphical representation

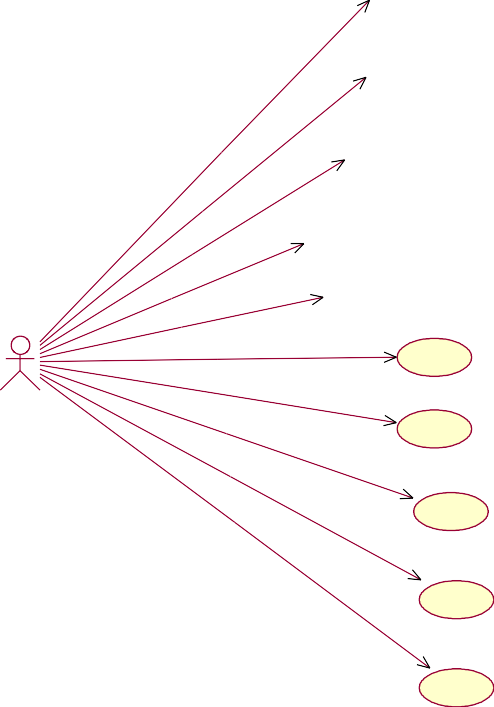
#### USE CASE DIAGRAM

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.

#### 

Start

Localhost



Register & Login to Application

Real Time Malware Detection

Data Stores in SQL

User

User Add Data

Attack Classification based on model

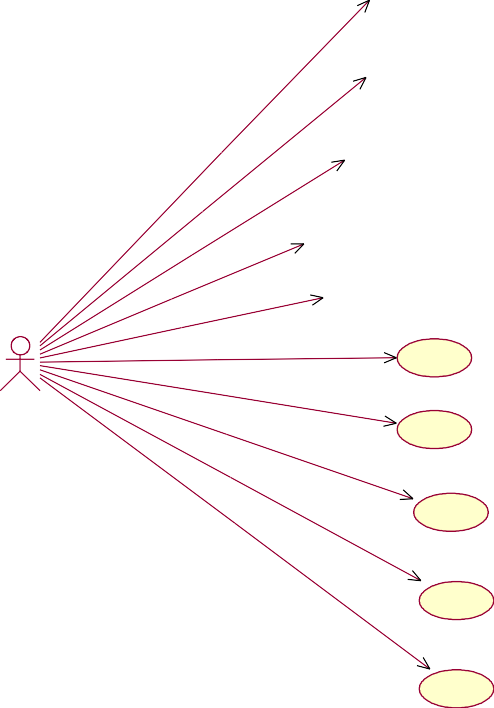
Detection of Attack

Visualisation

End

Start

Localhost



Register & Login to Application

Real Time Malware Detection

Data Stores in SQL

User

User Add Data

Attack Classification based on model

Detection of Attack

Visualisation

End

**Fig 1: Use Case Diagram**

#### CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

#### SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

User

System

Start

Localhost

Register & Login to Application

Real Time Malware Detection

Data Stores in SQL

User Add Data

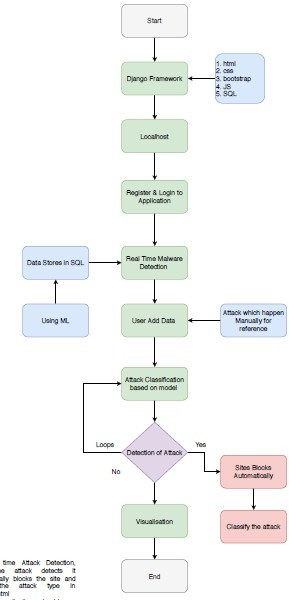
Attack Classification based on model Detection of Attack

Visualisation

**Fig 3: Sequence Diagram**

**IMPLEMENTATION**

**5.1 Flow Chart**



**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
* GUI Applications (like Kivy, Tkinter, PyQtetc. )
* 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.

##### 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.

##### 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.

##### 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.

##### 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.

##### 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.

##### 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.

##### 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.

##### 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.

##### Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** 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.

##### 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.

##### 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**

##### 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.

##### 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.

1. **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**, 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.

##### Speed Limitations

We have seen that Python code is executed line by line. But since Python 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.

##### 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.

##### 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**.

##### 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.

##### 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 voorWiskundeen 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, 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.

##### 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.

##### 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.

##### 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! 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, TensorFlow, Scikit- learn, 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** 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 :-**

##### 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.

##### 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.

1. **Continuous Improvement**

As **ML 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.

1. **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.

1. **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 :-**

##### 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.

* 1. **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.

* 1. **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.

* 1. **High error-susceptibility**

**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 and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 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 shells, the Jupyter 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 and thumbnail gallery.

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 Cheatsheethere.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](http://www.python.org/)**

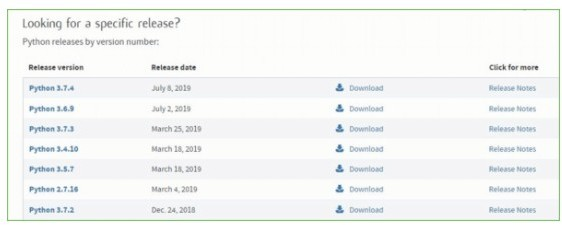


Now, check for the latest and the correct version for your operating system.

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

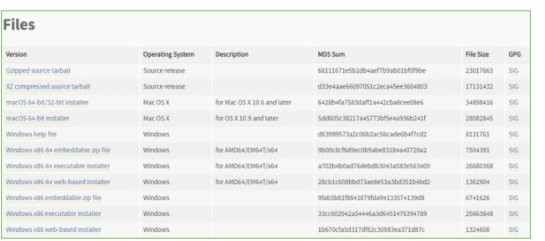


**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



**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.



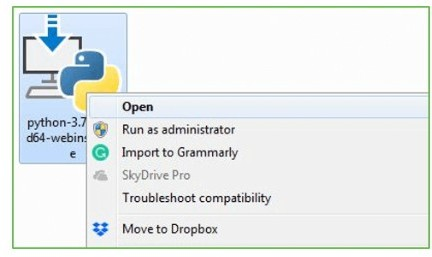
* To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
* To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, 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

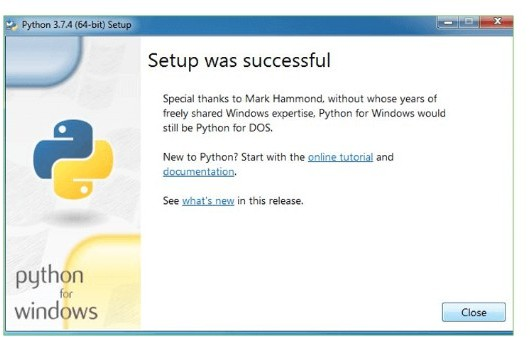
**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**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.

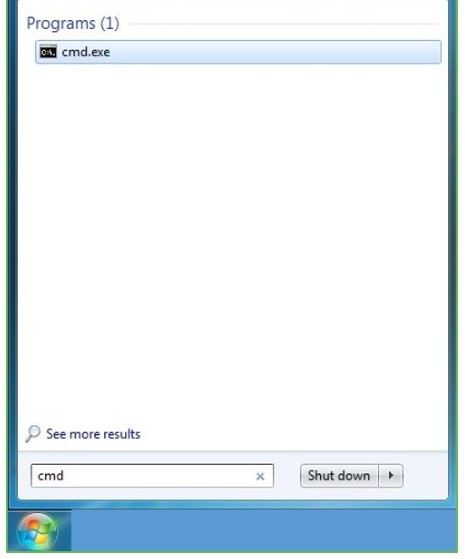


With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

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

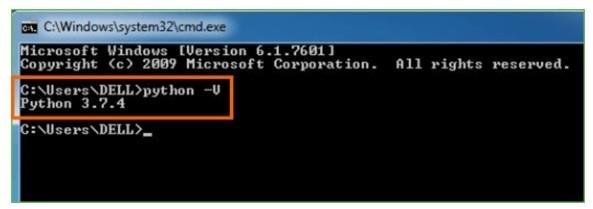
**Verify the Python Installation Step 1:** Click on Start

**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.

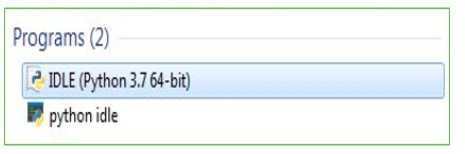


**Step 5:** You will get the answer as 3.7.4

**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.

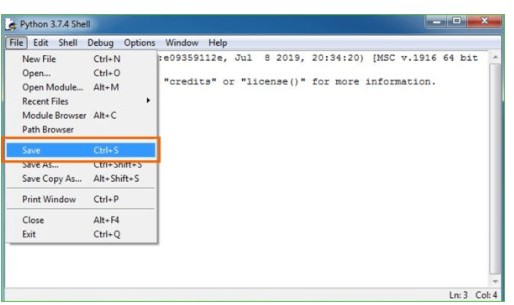
**Check how the Python IDLE works Step 1:** Click on Start

**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**

* 1. **Code**

**fromtkinter import messagebox fromtkinter import \***

**fromtkinter import simpledialog importtkinter**

**fromtkinter import filedialog importmatplotlib.pyplot as plt fromtkinter.filedialog import askopenfilename**

**fromsklearn.model\_selection import train\_test\_split fromsklearn.metrics import accuracy\_score importnumpy as np**

**import pandas as pd**

**fromgenetic\_selection import GeneticSelectionCV fromsklearn.metrics import classification\_report fromsklearn.metrics import confusion\_matrix fromsklearn import svm**

**fromkeras.models import Sequential fromkeras.layers import Dense import time**

**main = tkinter.Tk()**

**main.title("Android Malware Detection") main.geometry("1300x1200")**

**global filename global train**

**globalsvm\_acc, nn\_acc, svmga\_acc, annga\_acc globalX\_train, X\_test, y\_train, y\_test globalsvmga\_classifier**

**globalnnga\_classifier globalsvm\_time,svmga\_time,nn\_time,nnga\_time**

**def upload():**

**global filename**

**filename = filedialog.askopenfilename(initialdir="dataset") pathlabel.config(text=filename)**

**text.delete('1.0', END) text.insert(END,filename+" loaded\n");**

**defgenerateModel():**

**globalX\_train, X\_test, y\_train, y\_test text.delete('1.0', END)**

**train = pd.read\_csv(filename)**

**rows = train.shape[0] # gives number of row count cols = train.shape[1] # gives number of col count features = cols - 1**

**print(features)**

**X = train.values[:, 0:features] Y = train.values[:, features]**

**print(Y)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 0)**

**text.insert(END,"Dataset Length : "+str(len(X))+"\n"); text.insert(END,"Splitted Training Length : "+str(len(X\_train))**

**+"\n");**

**text.insert(END,"Splitted Test Length : "+str(len(X\_test))+"\n\ n");**

**def prediction(X\_test, cls): #prediction done here**

**y\_pred = cls.predict(X\_test) fori in range(len(X\_test)):**

**print("X=%s, Predicted=%s" % (X\_test[i], y\_pred[i])) returny\_pred**

**# Function to calculate accuracy defcal\_accuracy(y\_test, y\_pred, details):**

**cm = confusion\_matrix(y\_test, y\_pred) accuracy = accuracy\_score(y\_test,y\_pred)\*100 text.insert(END,details+"\n\n")**

**text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")**

**text.insert(END,"Report : "+str(classification\_report(y\_test, y\_pred))+"\n")**

**text.insert(END,"Confusion Matrix : "+str(cm)+"\n\n\n\n\n") return accuracy**

**defrunSVM(): globalsvm\_acc globalsvm\_time start\_time = time.time()**

**text.delete('1.0', END)**

**cls = svm.SVC(C=2.0,gamma='scale',kernel = 'rbf', random\_state = 2)**

**cls.fit(X\_train, y\_train)**

**prediction\_data = prediction(X\_test, cls)**

**svm\_acc = cal\_accuracy(y\_test, prediction\_data,'SVM Accuracy')**

**svm\_time = (time.time() - start\_time)**

**defrunSVMGenetic():**

**text.delete('1.0', END) globalsvmga\_acc globalsvmga\_classifier globalsvmga\_time**

**estimator = svm.SVC(C=2.0,gamma='scale',kernel = 'rbf', random\_state = 2)**

**svmga\_classifier = GeneticSelectionCV(estimator, cv=5,**

**verbose=1, scoring="accuracy",**

**max\_features=5, n\_population=50, crossover\_proba=0.5, mutation\_proba=0.2, n\_generations=40,**

**crossover\_independent\_proba=0.5, mutation\_independent\_proba=0.05, tournament\_size=3, n\_gen\_no\_change=10, caching=True,**

**n\_jobs=-1)**

**start\_time = time.time()**

**svmga\_classifier = svmga\_classifier.fit(X\_train, y\_train) svmga\_time = svm\_time/2**

**prediction\_data = prediction(X\_test, svmga\_classifier)**

**svmga\_acc = cal\_accuracy(y\_test, prediction\_data,'SVM with GA Algorithm Accuracy, Classification Report & Confusion Matrix')**

**defrunNN():**

**globalnn\_acc globalnn\_time text.delete('1.0', END) start\_time = time.time() model = Sequential()**

**model.add(Dense(4, input\_dim=215, activation='relu')) model.add(Dense(215, activation='relu')) model.add(Dense(1, activation='sigmoid'))**

**model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])**

**model.fit(X\_train, y\_train, epochs=50, batch\_size=64)**

**\_, ann\_acc = model.evaluate(X\_test, y\_test) nn\_acc = ann\_acc\*100**

**text.insert(END,"ANN Accuracy : "+str(nn\_acc)+"\n\n") nn\_time = (time.time() - start\_time)**

**defrunNNGenetic(): globalannga\_acc globalnnga\_time**

**text.delete('1.0', END)**

**train = pd.read\_csv(filename)**

**rows = train.shape[0] # gives number of row count cols = train.shape[1] # gives number of col count features = cols - 1**

**print(features)**

**X = train.values[:, 0:100]**

**Y = train.values[:, features] print(Y)**

**X\_train1, X\_test1, y\_train1, y\_test1 = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 0)**

**model = Sequential()**

**model.add(Dense(4, input\_dim=100, activation='relu')) model.add(Dense(100, activation='relu')) model.add(Dense(1, activation='sigmoid'))**

**model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])**

**start\_time = time.time() model.fit(X\_train1, y\_train1) nnga\_time = (time.time() - start\_time)**

**\_, ann\_acc = model.evaluate(X\_test1, y\_test1) annga\_acc = ann\_acc\*100**

**text.insert(END,"ANN with Genetic Algorithm Accuracy : "+str(annga\_acc)+"\n\n")**

**def graph():**

**height = [svm\_acc, nn\_acc, svmga\_acc, annga\_acc]**

**bars = ('SVM Accuracy','NNAccuracy','SVM Genetic Acc','NN Genetic Acc')**

**y\_pos = np.arange(len(bars)) plt.bar(y\_pos, height) plt.xticks(y\_pos, bars) plt.show()**

**deftimeGraph():**

**height = [svm\_time,svmga\_time,nn\_time,nnga\_time] bars = ('SVM Time','SVM Genetic Time','NNTime','NN**

**Genetic Time')**

**y\_pos = np.arange(len(bars))**

**plt.bar(y\_pos, height) plt.xticks(y\_pos, bars) plt.show()**

**font = ('times', 16, 'bold')**

**title = Label(main, text='Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning')**

**#title.config(bg='brown', fg='white') title.config(font=font) title.config(height=3, width=120) title.place(x=0,y=5)**

**font1 = ('times', 14, 'bold')**

**uploadButton = Button(main, text="Upload Android Malware Dataset", command=upload)**

**uploadButton.place(x=50,y=100) uploadButton.config(font=font1)**

**pathlabel = Label(main) pathlabel.config(bg='brown', fg='white') pathlabel.config(font=font1) pathlabel.place(x=460,y=100)**

**generateButton = Button(main, text="Generate Train & Test Model", command=generateModel)**

**generateButton.place(x=50,y=150) generateButton.config(font=font1)**

**svmButton = Button(main, text="Run SVM Algorithm", command=runSVM)**

**svmButton.place(x=330,y=150) svmButton.config(font=font1)**

**svmgaButton = Button(main, text="Run SVM with Genetic Algorithm", command=runSVMGenetic)**

**svmgaButton.place(x=540,y=150) svmgaButton.config(font=font1)**

**nnButton = Button(main, text="Run Neural Network Algorithm", command=runNN)**

**nnButton.place(x=870,y=150) nnButton.config(font=font1)**

**nngaButton = Button(main, text="Run Neural Network with Genetic Algorithm", command=runNNGenetic)**

**nngaButton.place(x=50,y=200) nngaButton.config(font=font1)**

**graphButton = Button(main, text="Accuracy Graph", command=graph)**

**graphButton.place(x=460,y=200) graphButton.config(font=font1)**

**exitButton = Button(main, text="Execution Time Graph", command=timeGraph)**

**exitButton.place(x=650,y=200) exitButton.config(font=font1)**

**font1 = ('times', 12, 'bold') text=Text(main,height=20,width=150) scroll=Scrollbar(text) text.configure(yscrollcommand=scroll.set) text.place(x=10,y=250) text.config(font=font1)**

**#main.config() main.mainloop()**

## 5.TESTING

* 1. **SOFTWARETESTING**

### Testing

Testingisaprocessofexecutingaprogramwiththeaimoffindingerror.Tomakeoursoftware perform well it should be error free. If testing is done successfully it will remove all the errors from thesoftware.

##### Types ofTesting

* + - 1. White BoxTesting
      2. Black BoxTesting
      3. Unit testing
      4. IntegrationTesting
      5. AlphaTesting
      6. BetaTesting
      7. Performance Testing and so on

**White BoxTesting**

Testing technique based on knowledge of the internal logic of an application's code and includes tests like coverage of code statements, branches, paths, conditions. It is performedby softwaredevelopers

**Black BoxTesting**

A method of software testing that verifies the functionality of an application without having specificknowledgeoftheapplication'scode/internalstructure.Testsarebasedonrequireme nts andfunctionality.

**Unit Testing**

Software verification and validation method in which a programmer tests if individual unitsof source code are fit for use. It is usually conducted by the developmentteam **IntegrationTesting**

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

**Alpha Testing**

Type of testing a software product or system conducted at the developer's site. Usually it is performed by the end users

**BetaTesting**

Final testing before releasing application for commercial purpose. It is typically done by end- users or others.

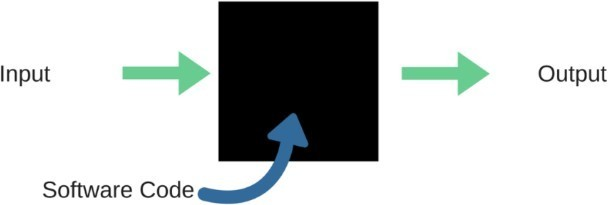
**PerformanceTesting**

Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements. It is usually conducted by the performance engineer.

##### Black BoxTesting

Blackbox testing is testing the functionality of an application without knowing the details of

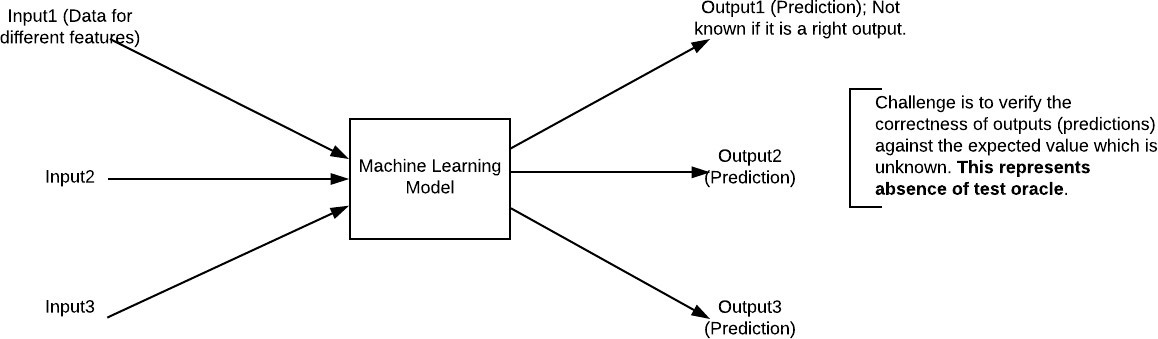
itsimplementationincludinginternalprogramstructure,datastructuresetc.Testcasesforbla ck box testing are created based on the requirement specifications. Therefore, it is also called as specification-based testing. Fig.4.1 represents the black box testing:



**Fig.:**Black Box Testing

When applied to machine learning models, black box testing would mean testing machine learning models without knowing the internal details such as features of the machine learning

model, the algorithm used to create the model etc. The challenge, however, is to verify the test outcome against the expected values that are known beforehand.



**Fig.:**Black Box Testing for Machine Learning algorithms

The above Fig.4.2 represents the black box testing procedure for machine learning algorithms.

**Table.4.1:**Black box Testing

|  |  |  |
| --- | --- | --- |
| **Input** | **Actual Output** | **Predicted Output** |
| [16,6,324,0,0,0,22,0,0,0,0,0,0] | 0 | 0 |
| [16,7,263,7,0,2,700,9,10,1153,832,  9,2] | 1 | 1 |

The model gives out the correct output when different inputs are given which are mentioned in Table 4.1. Therefore the program is said to be executed as expected or correct program

### Testing

Testingisaprocessofexecutingaprogramwiththeaimoffindingerror.Tomakeoursoftware perform well it should be error free. If testing is done successfully it will remove all the errors from thesoftware.

**7.2.2 Types ofTesting**

1. White BoxTesting
2. Black BoxTesting
3. Unit testing
4. IntegrationTesting
5. AlphaTesting
6. BetaTesting
7. Performance Testing and so on

**White BoxTesting**

Testing technique based on knowledge of the internal logic of an application's code and includes tests like coverage of code statements, branches, paths, conditions. It is performedby softwaredevelopers

**Black BoxTesting**

A method of software testing that verifies the functionality of an application without having

specificknowledgeoftheapplication'scode/internalstructure.Testsarebasedonrequire ments andfunctionality.

**Unit Testing**

Software verification and validation method in which a programmer tests if individual unitsof source code are fit for use. It is usually conducted by the developmentteam. **IntegrationTesting**

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

**Alpha Testing**

Type of testing a software product or system conducted at the developer's site. Usually it is performed by the end users.

**BetaTesting**

Final testing before releasing application for commercial purpose. It is typically done by end- users or others.

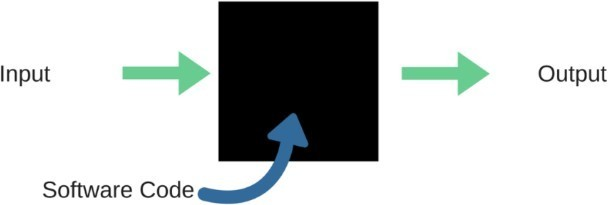
**PerformanceTesting**

Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements. It is usually conducted by the performance engineer.

##### Black BoxTesting

Blackbox testing is testing the functionality of an application without knowing the details of

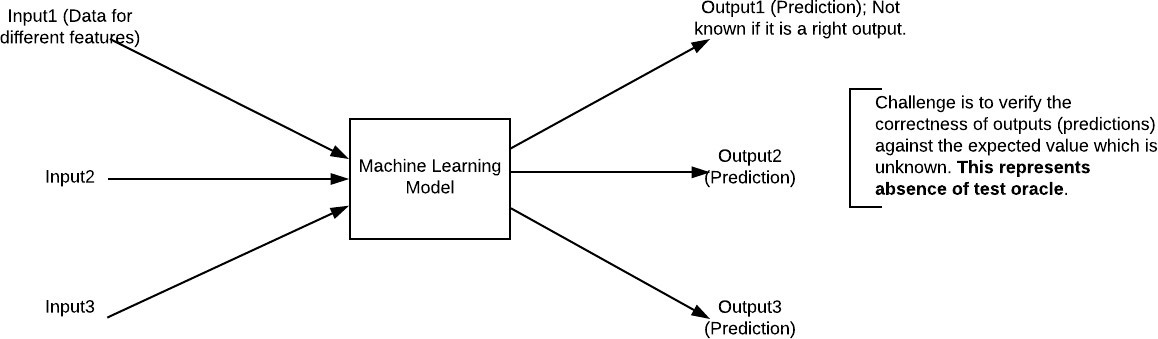
itsimplementationincludinginternalprogramstructure,datastructuresetc.Testcasesfor black box testing are created based on the requirement specifications. Therefore, it is also called as specification-based testing. Fig.4.1 represents the black box testing:



**Fig.:**Black Box Testing

When applied to machine learning models, black box testing would mean testing machine learning models without knowing the internal details such as features of the machine learning

model, the algorithm used to create the model etc. The challenge, however, is to verify the test outcome against the expected values that are known beforehand.



**Fig.:**Black Box Testing for Machine Learning algorithms

The above Fig.4.2 represents the black box testing procedure for machine learning algorithms.

**Table.4.1:**Black box Testing

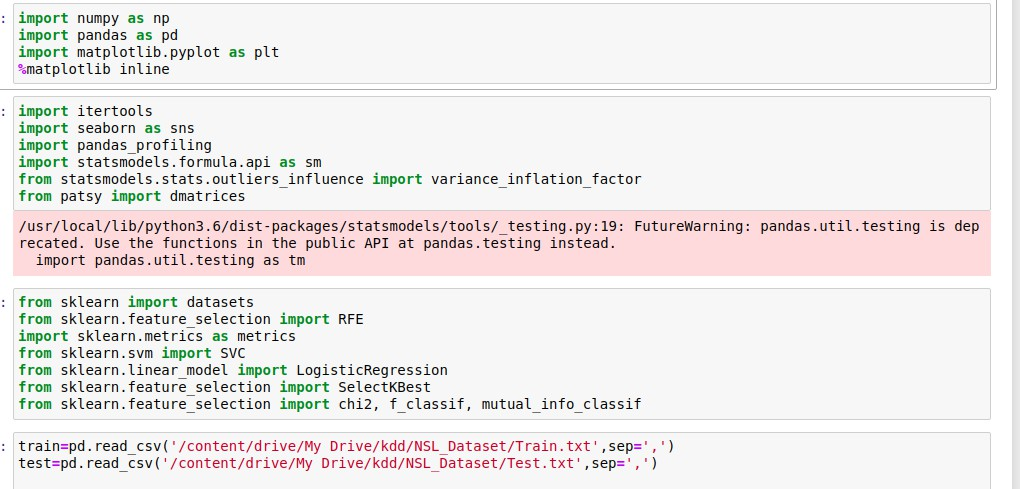
|  |  |  |
| --- | --- | --- |
| **Input** | **Actual Output** | **Predicted Output** |
| [16,6,324,0,0,0,22,0,0,0,0,0,0] | 0 | 0 |
| [16,7,263,7,0,2,700,9,10,1153,832,9,2] | 1 | 1 |

The model gives out the correct output when different inputs are given which are mentioned in Table 4.1. Therefore the program is said to be executed as expected or correct program

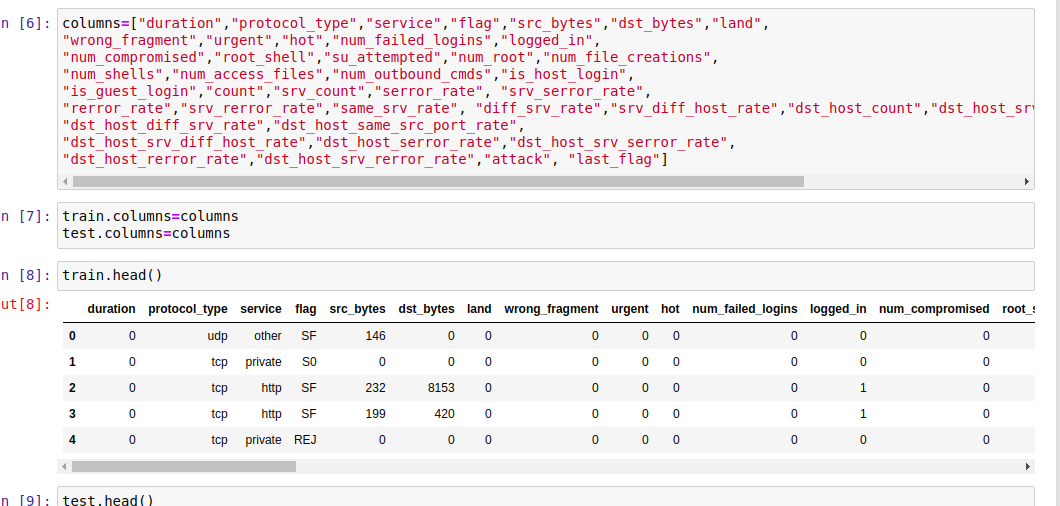
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Cas e Id | Test Case Name | Test Case Description | Test Steps | | | Test Case Statu  s | Test Priorit Y |
| Step | Expected | Actual |
| 01 | Start the Applicatio  N | Host the application  and test if it starts  making sure the required  software is available | If it doesn't  Start | We cannot  run the applicati  on. | The application  hosts success. | High | High |
| 02 | Home Page | Check the deployment  environmen t for  properly loading the  application. | If it doesn’t  load. | We cannot  access the  applicati on. | The application  is running successfully  . | High | High |
| 03 | User  Mode | Verify the  working of | If it  doesn’t | We  cannot | The  application | High | High |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | the  application in freestyle  mode | Respond | use the  Freestyle mode. | displays the  Freestyle Page |  |  |
| 04 | Data Input | Verify if the  application takes input  and updates | If it fails  to take the input or  store in  The Database | We  cannot proceed  further | The  application updates the  input to application | High | High |

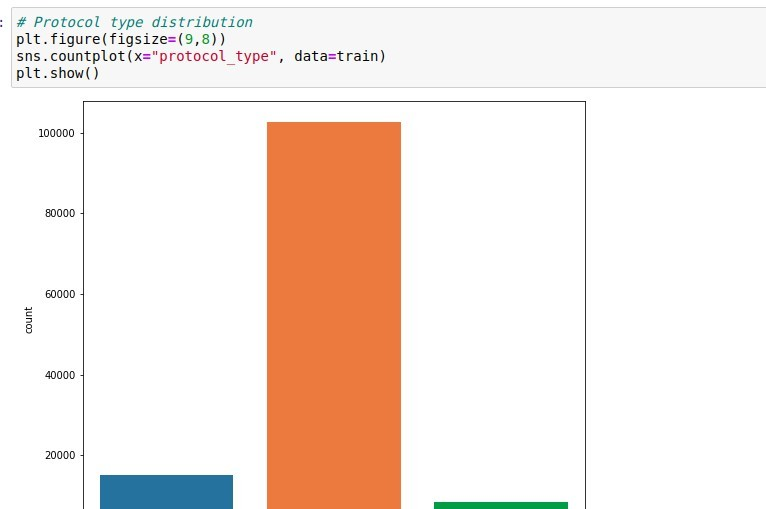
## RESULTS AND DISCUSSIONS

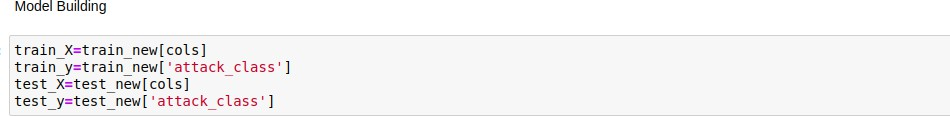


**Data preprocessing**



**Data EDA**

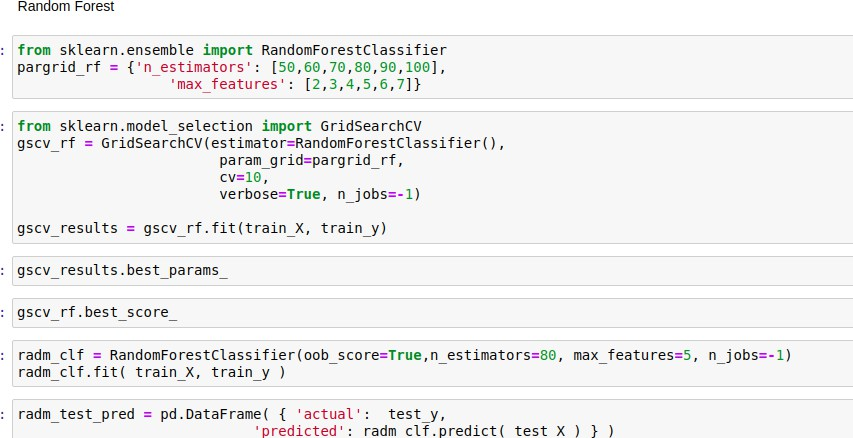




**ML Deploy**



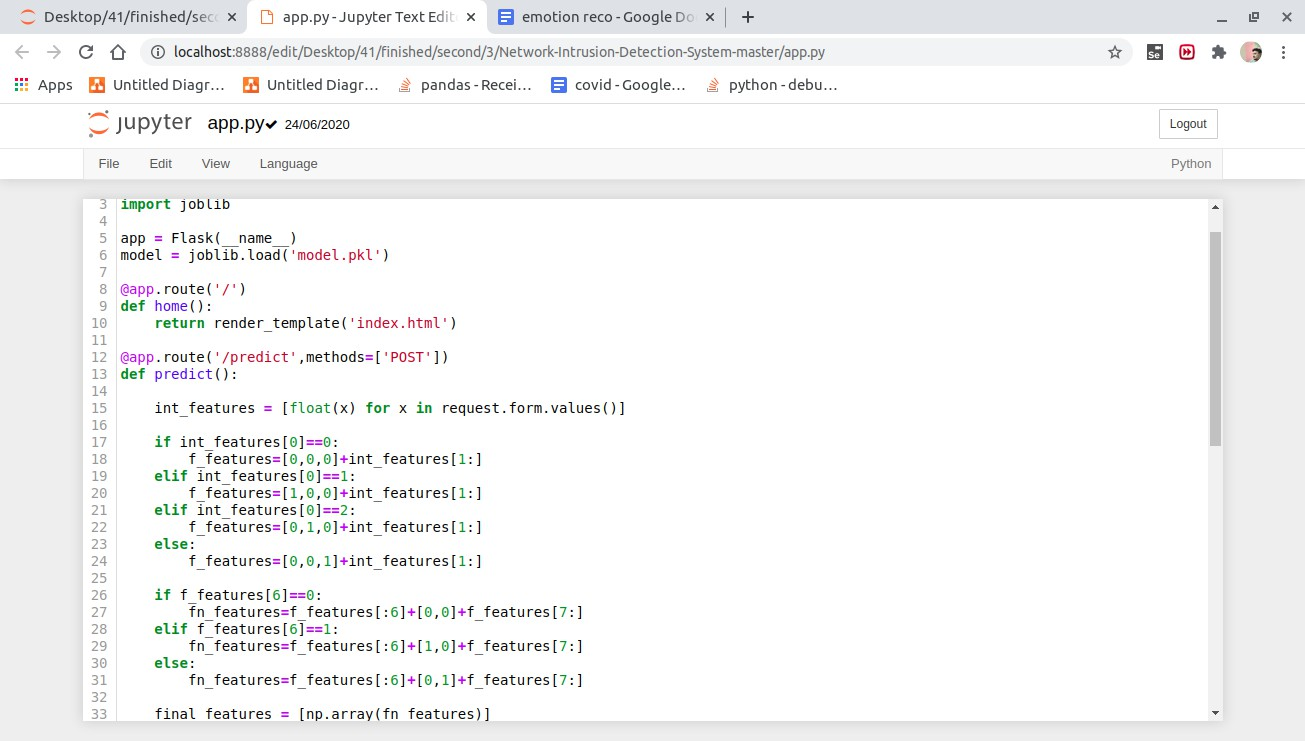




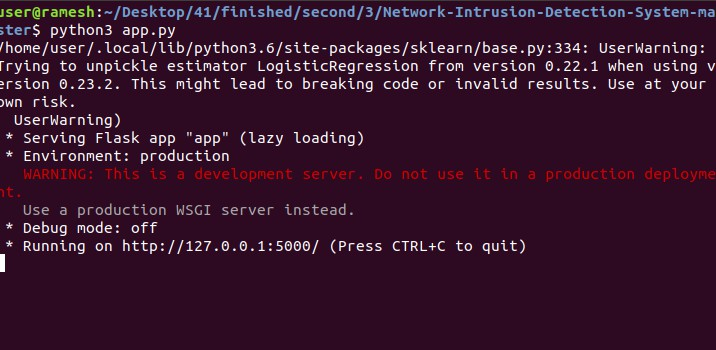


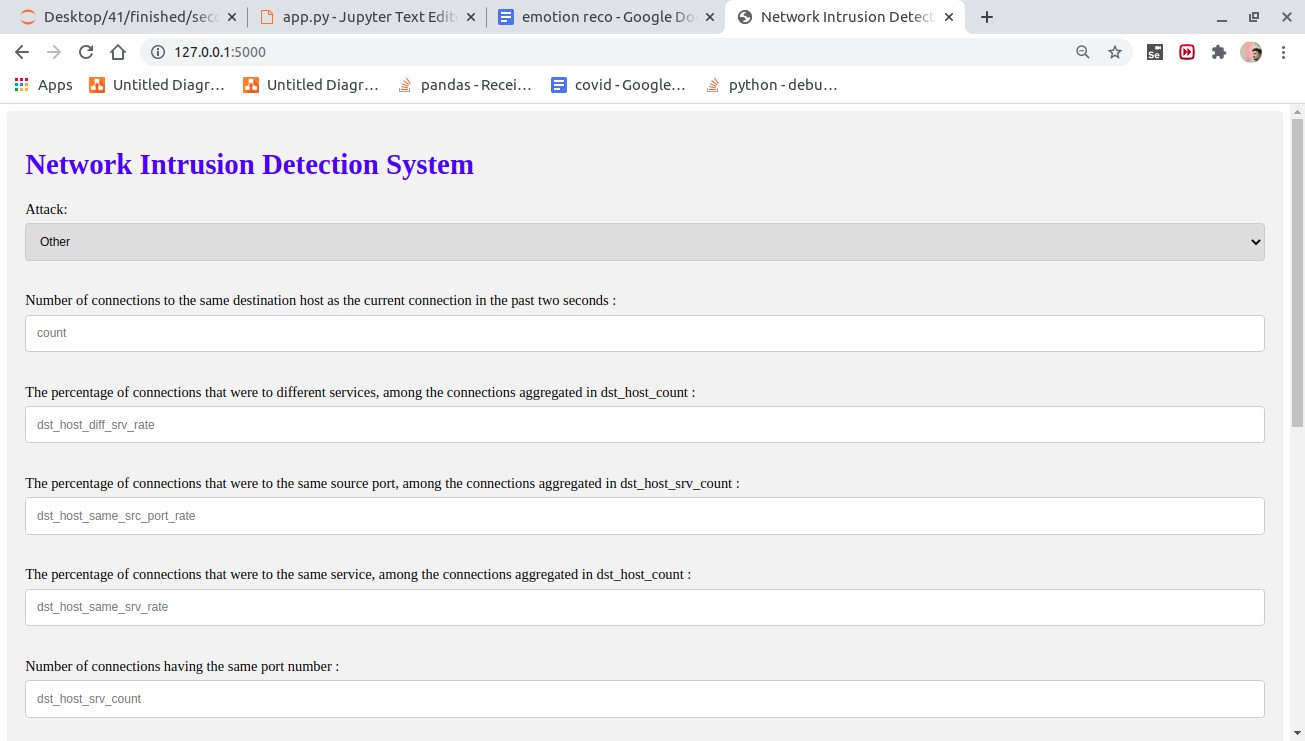
**From the score accuracy we concluding the DT & RF give better accuracy and building pickle file for predicting the user input**

**Application**

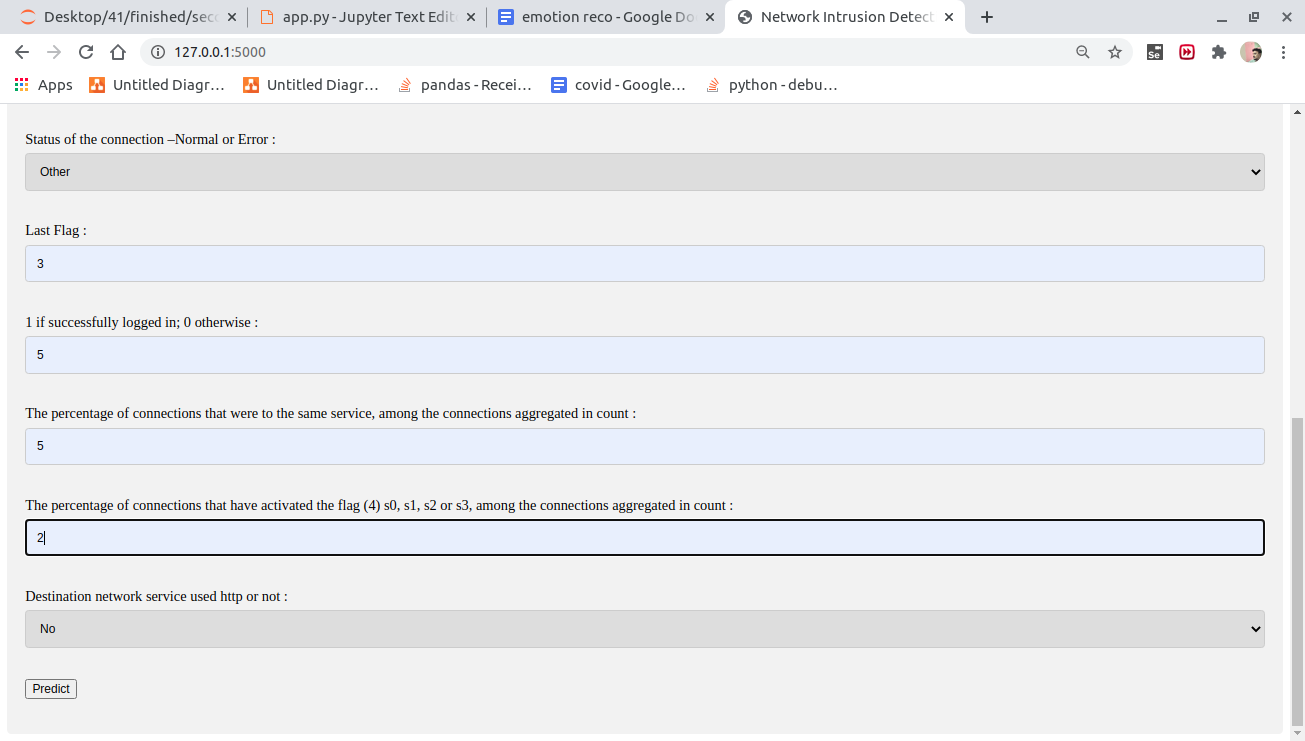


**Localhost - in cmd python app.py**

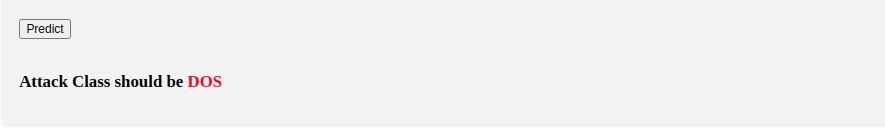




**Enter the input**



**Predict attack -**



## CONCLUSION

Right now, estimations of help vector machine, ANN, CNN, Random Forest and profound learning calculations dependent on modern CICIDS2017 dataset were introduced relatively. Results show that the profound learning calculation performed fundamentally preferable outcomes over SVM, ANN, RF and CNN. We are going to utilize port sweep endeavors as well as other assault types with AI and profound learning calculations, apache Hadoop and sparkle innovations together dependent on this dataset later on. All these calculation helps us to detect the cyber attack in network. It happens in the way that when we consider long back years there may be so many attacks happened so when these attacks are recognized then the features at which values these attacks are happening will be stored in some datasets. So by using these datasets we are going to predict whether cyber attack is done or not. These predictions can be done by four algorithms like SVM, ANN, RF, CNN this paper helps to identify which algorithm predicts the best accuracy rates which helps to predict best results to identify the cyber attacks happened or not.

**URL listing**

|  |  |
| --- | --- |
| **Websites** | **Data collected** |
| **[https://wikipedia.org](https://wikipedia.org/)** | **Searching of any information that will be used in documentation.** |
| **<https://dev.sqlserver.com/doc>** | **SQL server it performing in mainly depending on the one of the database using.** |
| **[https://www.answers.com](https://www.answers.com/)** | **Answers.com, online dictionary, encyclopedia and much more.** |
| **[https://google.co.in](https://google.co.in/)** | **Any information searching and downloading.** |
| **[https://training-classes.com](https://training-classes.com/)** | **Designing part information as gathered** |

**8.REFERENCES**

1. K. Graves, Ceh: Official certified ethical hacker review guide: Exam 312-50. John Wiley & Sons, 2007.
2. R. Christopher, “Port scanning techniques and the defense against them,” SANS Institute, 2001.
3. M. Baykara, R. Das¸, and I. Karado ˘gan, “Bilgi g ¨uvenli

˘gisistemlerindekullanilanarac¸larinincelenmesi,” in 1st International Symposium on Digital Forensics and Security (ISDFS13), 2013, pp. 231–239.

1. S. Staniford, J. A. Hoagland, and J. M. McAlerney, “Practical automated detection of stealthy portscans,” Journal of Computer Security, vol. 10, no. 1-2,

pp. 105–136, 2002.

1. S. Robertson, E. V. Siegel, M. Miller, and S. J. Stolfo, “Surveillance detection in high bandwidth environments,” in DARPA Information Survivability Conference and Exposition, 2003.Proceedings, vol. 1. IEEE, 2003, pp. 130–138.
2. K. Ibrahimi and M. Ouaddane, “Management of intrusion detection systems based-kdd99: Analysis with lda and pca,” in Wireless Networks and Mobile Communications (WINCOM), 2017 International Conference on. IEEE, 2017, pp. 1–6.
3. N. Moustafa and J. Slay, “The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems,” in Building Analysis Datasets and Gathering

Experience Returns for Security (BADGERS), 2015 4th International Workshop on. IEEE, 2015, pp. 25–31.

1. L. Sun, T. Anthony, H. Z. Xia, J. Chen, X. Huang, and Y. Zhang, “Detection and classification of malicious patterns in network traffic using benford’s law,” in Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017. IEEE, 2017, pp. 864–872.
2. S. M. Almansob and S. S. Lomte, “Addressing challenges for intrusion detection system using naive bayes and pca algorithm,” in Convergence in

Technology (I2CT), 2017 2nd International Conference for. IEEE, 2017, pp. 565– 568.

1. M. C. Raja and M. M. A. Rabbani, “Combined analysis of support vector machine and principle component analysis for ids,” in IEEE International Conference on Communication and Electronics Systems, 2016, pp. 1–5.
2. S. Aljawarneh, M. Aldwairi, and M. B. Yassein, “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” Journal of Computational Science, vol. 25, pp. 152–160, 2018.
3. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization.” in ICISSP, 2018, pp. 108–116.
4. D. Aksu, S. Ustebay, M. A. Aydin, and T. Atmaca, “Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm,” in International Symposium on Computer and Information Sciences. Springer, 2018, pp. 141–149.
5. N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, “Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark,” IEEE Access, 2018.
6. P. A. A. Resende and A. C. Drummond, “Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling,” Security and Privacy, vol. 1, no. 4, p. e36, 2018.
7. C. Cortes and V. Vapnik, “Support-vector networks,” Machine learning, vol. 20, no. 3, pp. 273–297, 1995.
8. R. Shouval, O. Bondi, H. Mishan, A. Shimoni, R. Unger, and A. Nagler, “Application of machine learning algorithms for clinical predictive modeling: a data-mining approach in sct,” Bone marrow transplantation, vol. 49, no. 3, p. 332, 2014**.**