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

*A Project report submitted in partial fulfillment of the requirement for the*

*award of Degree of*

**BACHELOR OF TECHNOLOGY**

In

ELECTRONICS AND COMMUNICATION ENGINEERING

*Submitted by*

**KADARA MADHURI (206L1A0424)**

**KANTAMREDDYVINAY (21L65A0428)**

**BOURUBILLI BHANU TEJA (21L65A0448)**

**PADALA SAI KUMAR (20L61A0421)**

*Under the esteemed guidance of*

**Mr. K. Suresh Kumar, M.tech,(PhD)**

**Assistant professor**

**DEPARTMENT OF ECE**

****

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CHAITANYA ENGINEERING COLLEGE

(Approved by AICTE)

(Affiliated to jntugv,vizianagaram, AP)

Madhurawada,Vishakhapatnam-41)

**( 2020-2024)**

## DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

## CHAITANYA ENGINEERING COLLEGE



Approved by AICTE,New Delhi,Affiliated to JNTU,Gurujada Vizinagaram,Chaitanya valley.



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**CERTIFICATE**

This is to certify that the project work entitled “**DETECTION OF CYBER ATTACK IN NETWORK USING MACHINE LEARNING TECHNIQUE** is a

Work done by Kadara Madhuri(206L1A0424), Kantam Reddy Vinay (21l65A0428), Bourubilli Bhanu Teja(21L65A0448), Padala Sai Kumar(20L61A0421)During the year 2017-2021 under the our guidance. This work was submitted to Department of” ELECTRONIC & COMMUNICATIONS ENGINEERING” in partial fulfillment for the award of a Bachelor of Technology “degree

**Project Guide by**

**Mr. K. Suresh Kumar, M.tech,(PhD)**

**Assistant professor**

ECE Departament,EceDepartament

Chaitanya engineering college

EXTERNAL EXAMINER

ENGINEERING

**DECLARATION**

I hereby declare that this project entitled **“DETECTION OF CYBER ATTACK IN NETWORK USING MACHINE LEARNING TECHNIQUE”** was carried out and written by me under theguidance of **Mr. K. Suresh Kumar,** Asst.Professor, Dept of ECE, CHAITANYA ENGINEERING COLLEGE KOMMADI. This work has not been previously formed the basis for the award of any degree or diploma or certificate nor has been submitted elsewhere for the award of any degree or diploma.

Place:

Date: Student Signature

**ACKNOWLEDGEMENT**

In our opinion, writing acknowledgement is a difficult but most beautiful part of the seminar report ending. It is difficult because it is hard to express the heartfelt accomplishments and the gratitude for those who are really or virtually supported for making the work successful and memorable.

We would like to convey our sincere gratitude and admiration to our Guide,

**Mr. K. Suresh Kumar,** Asst.Professor, Dept of ECE, CHAITANYA ENGINEERING COLLEGE

KOMMADI(CEVP) for his excellent support and valuable guidance throughout our study. His marvellous advice motivated us to workon this and make it successful. His supportive nature and thought-provoking meetings made this work possible.

We are very grateful to **Ms V.PAVANI**, HOD of ECE, CHAITANYA ENGINEERING COLLEGE KOMMADI(CEVP) for her keen interest, counselling, constant encouragement, personal affection and valuable suggestions in completion of this research work.

We would like to express our thanks to **Dr.K.Suresh,M.Tech,Ph.D.** Principal of CHAITANYA ENGINEERING COLLEGE KOMMADI(CEVP).

We are grateful to the **Authorities** of CHAITANYA ENGINEERING COLLEGE KOMMADI(CEVP) enabling me to pursue our work.

We are very thankful to all **Teaching** and **Non-teaching** staff of Department of Electronics & Communication Engineering, CHAITANYA ENGINEERING COLLEGE KOMMADI(CEVP), Guntur, for their encouragement, co-operation, and assistance during the work period.

We would like to express our grateful thanks to our parents, our whole beloved families for their love, unwavering support, and encouragement throughout the time of my study.

Finally, we thank one and all, who helped us to complete this work.

KADARA MADHURI

KANTAMREDDYVINAY

BOURUBILLI BHANU TEJA

PADALA SAI KUMAR

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **FIGURE TITLE** | **PAGE No.** |
| 1.6 |  | 4 |
| 1.6 | Structure of project | 9 |
| 3.2 | System Design | 10 |
| 3.3 | Use case diagram | 11 |
| 3.4 | Class diagram | 13 |
| 3.5 | Sequence diagram | 13 |
| 5.1 | Implementation | 15 |

|  |  |  |
| --- | --- | --- |
| **S. No.** | **TABLE NAME** | **PAGE No.** |
| 4.1 | Black box texting |  |
|  |  |  |

|  |  |
| --- | --- |
| **ABSTRACT** | **V** |
| **LIST OF FIGURES** | **VI** |
| **LIST OF TABLES** | **VIII** |

|  |  |  |
| --- | --- | --- |
| **S. No** | **TITLE** | **Page No.** |
| **CHAPTER 1** | **INTRODUCTION** | **1-6** |
| **1.1** | Motivation | 1 |
| **1.2** | Existing system | 2 |
| **1.3** | Objective | 2 |
| **1.4** | Out comes | 2 |
| **1.5** | Application | 3 |
| **1.6**  **1.7**  **1.8** | Structure of project  Functional requirements  Non-functional requirements | 4  5  6 |
| **CHAPTER 2** | **LITERATURE SURVEY** | **7-9** |
| **CHAPTER 3** | **PROBLEM ANALYSIS** | **1O-12** |
| **3.1** | Existing Approach | 10 |
| **3.2** | Proposed System | 11 |
| **3.3** | Software and Hardware | 11-12 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 4** | **SYSTEM DESIGN** | **13-33** |
| **4.1** | UML Diagram | 13-15 |
| **4.2** | Flow of events | 15-16 |
| **4.3** | Sequence Diagram | 16-17 |
| **4.4** | Class Diagram | 17-20 |
| **4.5** | Use case digram | 21-23 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 5** | **IMPLEMENTATION** | **24-51** |
| **5.1** | Flow chart | 25-31 |
| **5.2** | History of python | 31 |
| **5.3** | Categories of machine learning | 32-34 |
| 1.5.1 | Application of machine learning | 34 |
| **5.4** | How to start learing machine learning | 34-39 |
| **5.5** | Python development steps | 39-43 |
| **5.6** | How to install python | 43-47 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 6** | **TESTING** | **52-59** |
| **6.1** | Software Testing | 52 |
| 6.1.1 | Types of Testing | 52-59 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 7** | **RESULTS And DISCUSSIONS** | **60-61** |

|  |  |  |
| --- | --- | --- |
| **CHAPTER 8** | **CONCLUSION AND FUTURE SCOPE** | 62-63 |
|  | **REFERENCES** | **64** |

# ABSTRACT

Contrasted with the past, improvements in PC and correspondence innovations have given broad and propelled changes. The use of innovations gives 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 caused a lot of issues for people and establishments, has arrived at a level that could undermine open and national security by different gatherings, for example, criminal associations, proficient people, and digital activists. Along these lines, Intrusion Detection Systems (IDS) have been created to maintain a strategic distance from digital assaults.

**1. INTRODUCTION**

Contrasted with the past, improvements in PC and correspondence innovations have given broad and propelled changes. The use of new innovations gives 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 and decision tree .

**1.1** **MOTIVATION**

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.

* 1. **Existing 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.

**1.2.1 Limitations of existing system**

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

**1.3 Objectives**

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

* ANN
* CNN
* Random forest

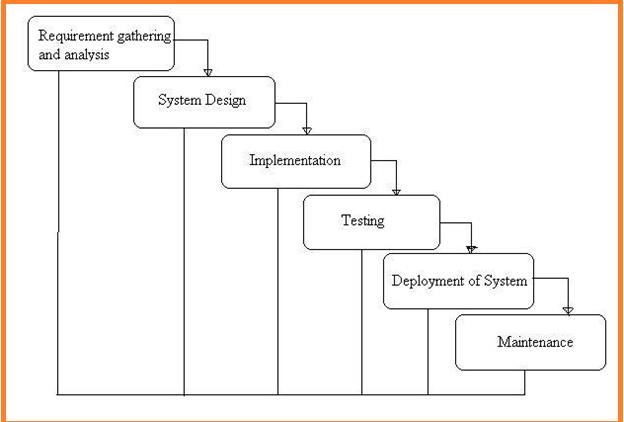
**1.4 Outcomes**

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.

**1.5Applications**

This strategy used in Detection of Cyber Attack in Network using Machine Learning Techniques

### STRUCTURE OF PROJECT (SYSTEM ANALYSIS)



**Fig: 1 Project SDLC**

* Project Requisites Accumulating and Analysis
* Application System Design
* Practical Implementation
* Manual Testing of My Application
* Application Deployment of System
* Maintenance of the Project

### 

### 1.6.1REQUISITES ACCUMULATING A ANALYSIS

It’s the first and foremost stage of the any project as our is a an academic leave for requisites amassing we followed of IEEE Journals and Amassed so many IEEE Relegated papers and final culled a Paper designated “Individual web revisitation by setting and substance importance input and for analysis stage we took referees from the paper and did literature survey of some papers and amassed all the Requisites of the project in this stage

### SYSTEM DESIGN

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 our project

### 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 UNIT TESTING

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

### 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 with having Windows OS .

The Maintenance of our Project is one-time process only

### FUNCTIONAL REQUIREMENTS

1. Data Collection

2. Data Preprocessing

3. Training And Testing

4.Modeling

5. Predicting

### 1.8NON 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

### EXAMPLES OF NON-FUNCTIONAL REQUIREMENTS

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 create any problem.
      3. Privacy of information, the export of restricted technologies, intellectual property rights, etc. should be audited.
    1. **ADVANTAGES OF NON-FUNCTIONAL REQUIREMENT**

Benefits/pros of Non-functional testing are:

* The nonfunctional requirements ensure the software system follow legal and compliance rules.
* They ensure the reliability, availability, and performance of the software system
* They ensure good user experience and ease of operating the software.
* They help in formulating security policy of the software system.
  + 1. **DISADVANTAGES OF NON-FUNCTIONAL REQUIREMENT**

Cons/drawbacks of Non-function requirement are:

* None functional requirement may affect the various high-level software subsystem
* They require special consideration during the software architecture/high-level design phase which increases costs.
* Their implementation does not usually map to the specific software sub-system,
* It is tough to modify non-functional once you pass the architecture phase.
  + 1. **KEY LEARNING**

# The character of the time period, the length of road, the weather, the bus speed and the rate of road usage are adopted as input vectors in Support

# 2.LITERATURE SURVEY

**2.1 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.

**2.2 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.

**2.3 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.**

Compared to the past security of networked systems has become a critical universal issue that influences individuals, enterprises and governments. The rate of attacks against networked systems has increased melodramatically, and the strategies used by the attackers are continuing to evolve. For example, the privacy of important information, security of stored data platforms, availability of knowledge etc. Depending on these problems, cyber terrorism is one of the most important issues in today’s world. Cyber terror, which caused a lot of problems to individuals and institutions, has reached a level that could threaten public and country security by various groups such as criminal organizations, professional persons and cyber activists. Intrusion detection is one of the solutions against these attacks. A free and effective approach for designing Intrusion Detection Systems (IDS) is Machine Learning. In this study, deep learning and support vector machine (SVM) algorithms were used to detect port scan attempts based on the new CICIDS2017 dataset Introduction Network Intrusion Detection System (IDS) is a software-based application or a hardware device that is used to identify malicious behavior in the network [1,2]. Based on the detection technique, intrusion detection is classified into anomaly-based and signature-based.

**2.4 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.**

n network security, intrusion detection plays an important role. Feature subsets obtained by different feature selection methods will lead to different accuracy of intrusion detection. Using individual feature selection method can be unstable in different intrusion detection scenarios. In this paper, the idea of ensemble is applied to feature selection to adjust feature subsets. Feature selection is converted into a two-category problem, and odd number of feature selection methods is used for voting method to decide whether a feature is required or discarded. In actual operation, mean decrease impurity, random forest classifier, stability selection, recursive feature elimination and chi-square test are used. Feature subsets obtained from them will be adjusted by our proposed method to get ensemble feature subsets. To test the performance, support vector machine, decision tree, knn and multi-layer perception are used to observe and compare the classification accuracy with ensemble feature subsets. Three intrusion detection data sets, including kddcup99, cidds-001 and unsw\_nb15 are used in our experiments. The best result is reflected on cidds-001 with a 99.40% classification accuracy.

# 

# 3. PROBLEM ANALYSIS

### EXISTING APPROACH:

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.

* 1. **Drawbacks**

1. Strict Regulations
2. Difficult to work with for non-technical users
3. Restrictive to resources
4. Constantly needs Patching
5. Constantly being attacked

### 3.2 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

.

**3.2.1 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

# 3.3 Software And Hardware 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**

# 4. SYSTEM DESIGN

### 4.1 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 external interfaces.

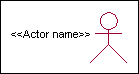
### Global Use Case Diagrams:

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:

<<Actor name>>





Actor

An actor is someone or something that:

Interacts with or uses the system.

Provides input to and receives information from the system.

Is external to the system and has no control over the use cases. Actors are discovered by examining:

* + - * Who directly uses the system?
      * Who is responsible for maintaining the system?
      * External hardware used by the system.
      * Other systems that need to interact with the system. Questions to identify actors:
        + Who is using the system? Or, who is affected by the system? Or, which groups need help from the system to perform a task?
        + Who affects the system? Or, which user groups are needed by the system to perform its functions? These functions can be both main functions and secondary functions such as administration.
        + Which external hardware or systems (if any) use the system to perform tasks?
        + What problems does this application solve (that is, for whom)?
        + And, finally, how do users use the system (use case)? What are they doing with the system?

The actors identified in this system are:

### System Administrator

1. **Customer**
2. **Customer Care**

Identification of usecases:

**Usecase:** A use case can be described as a specific way of using the system from a user’s (actor’s) perspective.

### Graphical representation:



A more detailed description might characterize a use case as:

* + Pattern of behavior the system exhibits
  + A sequence of related transactions performed by an actor and the system
  + Delivering something of value to the actor Use cases provide a means to:
  + capture system requirements
  + communicate with the end users and domain experts
  + test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

Guide lines for identifying use cases:

* For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform. The use case should represent a course of events that leads to clear goal
* Name the use cases.
* Describe the use cases briefly by applying terms with which the user is familiar. This makes the description less ambiguous

Questions to identify use cases:

* What are the tasks of each actor?
* Will any actor create, store, change, remove or read information in the system?
* What use case will store, change, remove or read this information?
* Will any actor need to inform the system about sudden external changes?
* Does any actor need to inform about certain occurrences in the system?
* What usecases will support and maintains the system?

### 4.2 Flow of Events

A flow of events is a sequence of transactions (or events) performed by the system. They typically contain very detailed information, written in terms of what the system should do, not how the system accomplishes the task. Flow of events are created as separate files or documents in your favorite text editor and then attached or linked to a use case using the Files tab of a model element.

A flow of events should include:

* When and how the use case starts and ends
* Use case/actor interactions
* Data needed by the use case
* Normal sequence of events for the use case
* Alternate or exceptional flows Construction of Usecase diagrams:

Use-case diagrams graphically depict system behavior (use cases). These diagrams present a high level view of how the system is used as viewed from an outsider’s (actor’s) perspective. A use-case diagram may depict all or some of the use cases of a system.

A use-case diagram can contain:

* + actors ("things" outside the system)
  + use cases (system boundaries identifying what the system should do)
  + Interactions or relationships between actors and use cases in the system including the associations, dependencies, and generalizations.

Relationships in use cases:

### Communication:

The communication relationship of an actor in a usecase is shown by connecting the actor symbol to the usecase symbol with a solid path. The actor is said to communicate with the usecase.

### Uses:

A Uses relationship between the usecases is shown by generalization arrow from the usecase.

### Extends:

The extend relationship is used when we have one usecase that is similar to another usecase but does a bit more. In essence it is like subclass.

### 4.3 SEQUENCE DIAGRAMS

A sequence diagram is a graphical view of a scenario that shows object interaction in a time- based sequence what happens first, what happens next. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces. There are two main differences between sequence and collaboration diagrams: sequence diagrams show time-based object interaction while collaboration diagrams show how objects associate with each other. A sequence diagram has two dimensions: typically, vertical placement represents time and horizontal placement represents different objects.

### Object:

An object has state, behavior, and identity. The structure and behavior of similar objects are defined in their common class. Each object in a diagram indicates some instance of a class. An object that is not named is referred to as a class instance.

The object icon is similar to a class icon except that the name is underlined: An object's concurrency is defined by the concurrency of its class.

### Message:

A message is the communication carried between two objects that trigger an event. A message carries information from the source focus of control to the destination focus of control. The synchronization of a message can be modified through the message specification. Synchronization means a message where the sending object pauses to wait for results.

### Link:

A link should exist between two objects, including class utilities, only if there is a relationship between their corresponding classes. The existence of a relationship between two classes symbolizes a path of communication between instances of the classes: one object may send messages to another. The link is depicted as a straight line between objects or objects and class instances in a collaboration diagram. If an object links to itself, use the loop version of the icon.

**4.4 CLASS DIAGRAM:**

Identification of analysis classes:

A class is a set of objects that share a common structure and common behavior (the same attributes, operations, relationships and semantics). A class is an abstraction of real-world items. There are 4 approaches for identifying classes:

* 1. Noun phrase approach:
  2. Common class pattern approach.
  3. Use case Driven Sequence or Collaboration approach.
  4. Classes , Responsibilities and collaborators Approach

### Noun Phrase Approach:

The guidelines for identifying the classes:

* + Look for nouns and noun phrases in the usecases.
  + Some classes are implicit or taken from general knowledge.
  + All classes must make sense in the application domain; Avoid computer implementation classes – defer them to the design stage.
  + Carefully choose and define the class names After identifying the classes we have to eliminate the following types of classes:
  + Adjective classes.

### Common class pattern approach:

The following are the patterns for finding the candidate classes:

* + Concept class.
  + Events class.
  + Organization class
  + Peoples class
  + Places class
  + Tangible things and devices class.

### Use case driven approach:

We have to draw the sequence diagram or collaboration diagram. If there is need for some classes to represent some functionality then add new classes which perform those functionalities.

### CRC approach:

The process consists of the following steps:

* + Identify classes’ responsibilities ( and identify the classes )
  + Assign the responsibilities
  + Identify the collaborators. Identification of responsibilities of each class:

The questions that should be answered to identify the attributes and methods of a class respectively are:

1. What information about an object should we keep track of?
2. What services must a class provide? Identification of relationships among the classes:

Three types of relationships among the objects are:

Association: How objects are associated?

Super-sub structure: How are objects organized into super classes and sub classes? Aggregation: What is the composition of the complex classes?

Association:

The **questions** that will help us to identify the associations are:

* 1. Is the class capable of fulfilling the required task by itself?
  2. If not, what does it need?
  3. From what other classes can it acquire what it needs? Guidelines for identifying the tentative associations:
* A dependency between two or more classes may be an association. Association often corresponds to a verb or prepositional phrase.
* A reference from one class to another is an association. Some associations are implicit or taken from general knowledge.

Some common association patterns are:

Location association like part of, next to, contained in….. Communication association like talk to, order to ……

We have to eliminate the unnecessary association like implementation associations, ternary or n- ary associations and derived associations.

Super-sub class relationships:

Super-sub class hierarchy is a relationship between classes where one class is the parent class of another class (derived class).This is based on inheritance.

Guidelines for identifying the super-sub relationship, a generalization are

### Top-down*:*

Look for noun phrases composed of various adjectives in a class name. Avoid excessive refinement. Specialize only when the sub classes have significant behavior.

### Bottom-up*:*

Look for classes with similar attributes or methods. Group them by moving the common attributes and methods to an abstract class. You may have to alter the definitions a bit.

### Reusability*:*

Move the attributes and methods as high as possible in the hierarchy.

### Multiple inheritances*:*

Avoid excessive use of multiple inheritances. One way of getting benefits of multiple inheritances is to inherit from the most appropriate class and add an object of another class as an attribute.

### Aggregation or a-part-of relationship:

It represents the situation where a class consists of several component classes. A class that is composed of other classes doesn’t behave like its parts. It behaves very difficultly. The major properties of this relationship are transitivity and anti symmetry.

The **questions** whose answers will determine the distinction between the part and whole relationships are:

* + Does the part class belong to the problem domain?
  + Is the part class within the system’s responsibilities?
  + Does the part class capture more than a single value?( If not then simply include it as an attribute of the whole class)
  + Does it provide a useful abstraction in dealing with the problem domain? There are three types of aggregation relationships. They are:

### Assembly:

It is constructed from its parts and an assembly-part situation physically exists.

### Container:

A physical whole encompasses but is not constructed from physical parts.

### Collection member:

A conceptual whole encompasses parts that may be physical or conceptual. The container and collection are represented by hollow diamonds but composition is represented by solid diamond.

**4.5 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.

****

### 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.

### 

### Fig 2:Class Diagram

**SEQUENCE DIAGRAM**

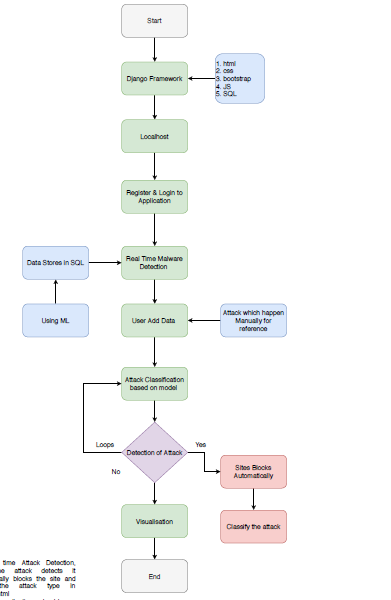
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

**Fig 3: Sequence Diagram**

**5.IMPLEMENTATION**

**5.1 FLOW CHART:**

****

# 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](https://www.geeksforgeeks.org/kivy-tutorial/), Tkinter, PyQt etc. )
  + Web frameworks like [Django](https://www.geeksforgeeks.org/django-tutorial/) (used by YouTube, Instagram, Dropbox)
  + Image processing (like [OpenCV](https://www.geeksforgeeks.org/opencv-python-tutorial/), 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.

**5.2 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.

**5.3 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.

## 5.4 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

# 5.5 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.

**5.6 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.

## 5.7 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”



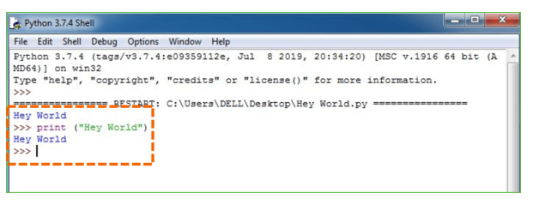
**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**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 (“Hey World”)** and Press Enter.



You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

***Note:*** Unlike Java, Python doesn’t need semicolons at the end of the statements otherwise it won’t work.

This stack that includes:

* world.

**Django – Design Philosophies**

Django comes with the following design philosophies −

* **Loosely Coupled** − Django aims to make each element of its stack independent of the others.
* **Less Coding** − Less code so in turn a quick development.
* **Don't Repeat Yourself (DRY)** − Everything should be developed only in exactly one place instead of repeating it again and again.
* **Fast Development** − Django's philosophy is to do all it can to facilitate hyper-fast development.
* **Clean Design** − Django strictly maintains a clean design throughout its own code and makes it easy to follow best web-development practices.

**Advantages of Django**

Here are few advantages of using Django which can be listed out here −

* **Object-Relational Mapping (ORM) Support** − Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django-nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.
* **Multilingual Support** − Django supports multilingual websites through its built-in internationalization system. So you can develop your website, which would support multiple languages.
* **Framework Support** − Django has built-in support for Ajax, RSS, Caching and various other frameworks.
* **Administration GUI** − Django provides a nice ready-to-use user interface for administrative activities.
* **Development Environment** − Django comes with a lightweight web server to facilitate end-to-end application development and testing.

As you already know, Django is a Python web framework. And like most modern framework, Django supports the MVC pattern. First let's see what is the Model-View-Controller (MVC) pattern, and then we will look at Django’s specificity for the Model-View-Template (MVT) pattern.

**MVC Pattern**

When talking about applications that provides UI (web or desktop), we usually talk about MVC architecture. And as the name suggests, MVC pattern is based on three components: Model, View, and Controller. [Check our MVC tutorial here](http://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm) to know more.

**Django MVC – MVT Pattern**

The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), leaving us with the template. The template is a HTML file mixed with Django Template Language (DTL).

The following diagram illustrates how each of the components of the MVT pattern interacts with each other to serve a user request −



**Fig 2.2: Django MVC – MVT Pattern**

The developer provides the Model, the view and the template then just maps it to a URL and Django does the magic to serve it to the user.

**Jupyter Notebook**

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at [Project Jupyter](http://jupyter.org/).

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

**Anaconda :-**

## What is Anaconda Python?

Together with a list of Python packages, tools like editors, Python distributions include the Python interpreter. Anaconda is one of several Python distributions. Anaconda is a new distribution of the Python and R data science package. It was formerly known as Continuum Analytics. Anaconda has more than 100 new packages.

This work environment, Anaconda is used for scientific computing, [data science](https://www.springpeople.com/data-science-training-certification), statistical analysis, and machine learning. The latest version of Anaconda 5.0.1 is released in October 2017.

The released version 5.0.1 addresses some minor bugs and adds useful features, such as updated R language support. All of these features weren’t available in the original 5.0.0 release.

This package manager is also an environment manager, a Python distribution, and a collection of open source packages and contains more than 1000 R and [Python Data Science](https://www.springpeople.com/blog/25-python-questions-answers-for-data-science-interviews/) Packages.

## Why Anaconda for Python?

There’s no big reason to switch to Anaconda if you are completely happy with you regular python. But some people like data scientists who are not full-time developers, find anaconda much useful as it simplifies a lot of common problems a beginner runs into.

Anaconda can help with –

* [Installing Python](https://www.ics.uci.edu/~pattis/common/handouts/pythoneclipsejava/python.html) on multiple platforms
* Separating out different environments
* Dealing with not having correct privileges and
* Getting up and running with specific packages and libraries

## How to Download Anaconda 5.0.1?

**6.TESTING**

### SOFTWARE TESTING

## Testing

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

#### 6.1.1 Types of Testing

* + - 1. White Box Testing
      2. Black Box Testing
      3. Unit testing
      4. Integration Testing
      5. Alpha Testing
      6. Beta Testing
      7. Performance Testing and so on

#### White Box Testing

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 performed by software developers

**Black Box Testing**

A method of software testing that verifies the functionality of an application without having specific knowledge of the application's code/internal structure. Tests are based on requirements and functionality.

**Unit Testing**

Software verification and validation method in which a programmer tests if individual units of source code are fit for use. It is usually conducted by the development team.

**Integration Testing**

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.

**Beta Testing**

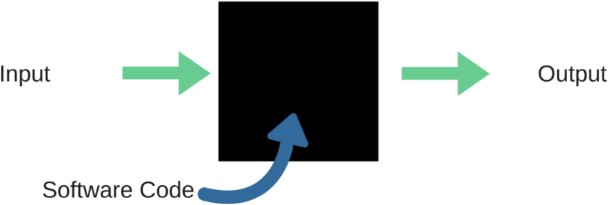
Final testing before releasing application for commercial purpose. It is typically done by end- users or others.

**Performance Testing**

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 Box Testing

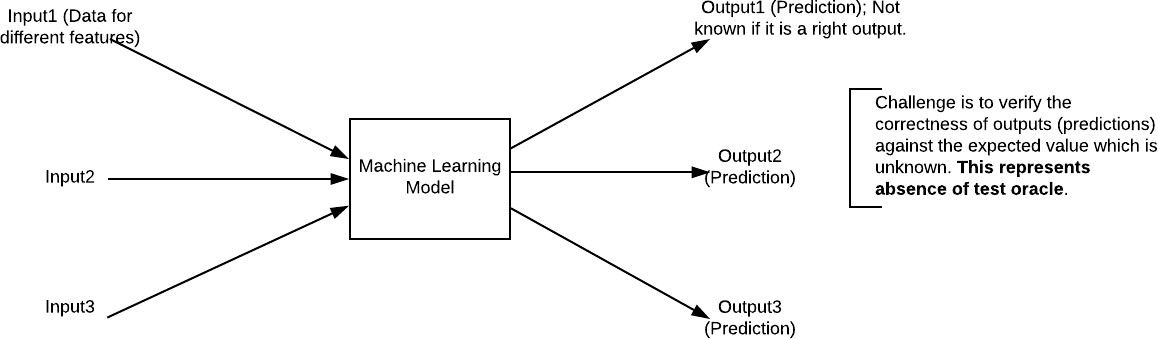
Blackbox testing is testing the functionality of an application without knowing the details of its implementation including internal program structure, data structures etc. Test cases for black box testing are created based on the requirement specifications. Therefore, it is also called as specification-based 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

## Testing

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

#### 7.2.2 Types of Testing

* + - 1. White Box Testing
      2. Black Box Testing
      3. Unit testing
      4. Integration Testing
      5. Alpha Testing
      6. Beta Testing
      7. Performance Testing and so on

#### White Box Testing

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 performed by software developers

**Black Box Testing**

A method of software testing that verifies the functionality of an application without having specific knowledge of the application's code/internal structure. Tests are based on requirements and functionality.

**Unit Testing**

Software verification and validation method in which a programmer tests if individual units of source code are fit for use. It is usually conducted by the development team.

**Integration Testing**

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.

**Beta Testing**

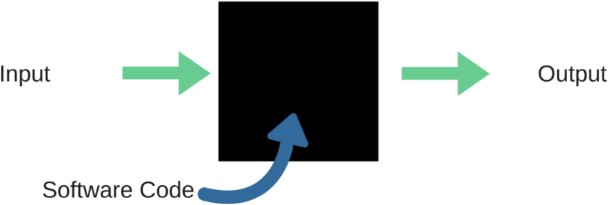
Final testing before releasing application for commercial purpose. It is typically done by end- users or others.

**Performance Testing**

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 Box Testing

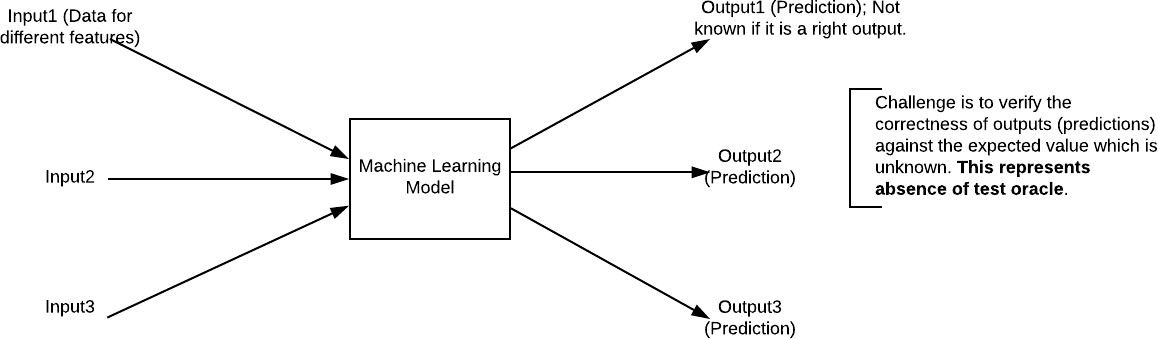
Blackbox testing is testing the functionality of an application without knowing the details of its implementation including internal program structure, data structures etc. Test cases for black box testing are created based on the requirement specifications. Therefore, it is also called as specification-based 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.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | Test Case | Test Case | Test Steps | | | Test | Test |
| Cas | Name | Description | Step | Expected | Actual | Case | Priorit |
| e Id |  |  |  |  |  | Statu | Y |
|  |  |  |  |  |  | s |  |
| 01 | Start the | Host the | If it | We | The | High | High |
|  | Applicatio | application | doesn't | cannot | application |  |  |
|  | N | and test if it | Start | run the | hosts |  |  |
|  |  | starts |  | applicati | success. |  |  |
|  |  | making sure |  | on. |  |  |  |
|  |  | the required |  |  |  |  |  |
|  |  | software is |  |  |  |  |  |
|  |  | available |  |  |  |  |  |
| 02 | Home Page | Check the | If it | We | The | High | High |
|  |  | deployment | doesn’t | cannot | application |  |  |
|  |  | environmen | load. | access | is running |  |  |
|  |  | t for |  | the | successfully |  |  |
|  |  | properly |  | applicati | . |  |  |
|  |  | loading the |  | on. |  |  |  |
|  |  | application. |  |  |  |  |  |
| 03 | User | Verify the | If it | We | The | High | High |
|  | Mode | working of | doesn’t | cannot | application |  |  |
|  |  | the | Respond | use the | displays the |  |  |
|  |  | application |  | Freestyle | Freestyle |  |  |
|  |  | in freestyle |  | mode. | Page |  |  |
|  |  | mode |  |  |  |  |  |
| 04 | Data Input | Verify if the | If it fails | We | The | High | High |
|  |  | application | to take the | cannot | application |  |  |
|  |  | takes input | input or | proceed | updates the |  |  |
|  |  | and updates | store in | further | input to application |  |  |
|  |  |  | The |  |  |  |  |
|  |  |  | Database |  |  |  |  |

**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

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

|  |  |  |
| --- | --- | --- |
| **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 |

# 7. RESULTS AND DISCUSSIONS

# 

# Fig 2:Data set Loaded

# 

# Fig 3:Accuracy Comparison Graph

# 

# Fig 4:Predict Attacks In Graph

**8. CONCLUSION**

# Right now, estimations of help vector machine, decision tree,SVM, Random Forest and profound learning calculations dependent on modern dataset were introduced relatively. Results show that the profound learning calculation performed fundamentally preferable outcomes over 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 three algorithms 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.

**FUTURE SCOPE**

In enhancement we will add some ML Algorithms to increase accuracy n this article, the method for application layer attack detection based on machine learning was proposed. The model consists of patterns that are obtained using graph-based segmentation technique and dynamic programming. The regular expressions are used for modelling the genuine behaviour of the applications and detecting cyber attacks. We also presented the results that prove the efficiency of the proposed algorithm that can be effectively used for application layer attack detection.

# 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 ˘gi sistemlerinde kullanilan arac¸larin incelenmesi,” in 1st International Symposium on Digital Forensics and Security (ISDFS13), 2013, pp. 231–239.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.