A PROJECT REPORT

ON

Cyber Attack Detection Model Based on Machine Learning

Submitted to JNTUA for the partial fulfillment of the requirements for the award of the degree

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

By

D. LAKSHMI NARASIMHULU	**20L21A0510**
C. GURU TEJA	**20L21A0508**
D. SHANATH RESHON	**20L21A0511**
R. PRADEEP	**20L21A0540**
S. NOOR BASHA	**20L21A0543**

Under the esteemed guidance of

Mr. T. VENKATA KRISHNA REDDY, MTech

Asst. Prof, Dept. Of CSE

VITS, PRODDATUR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VAAGDEVI INSTITUTE OF TECHNOLOGY AND SCIENCE

(Affiliated to JNTU, Anantapur, College Code: L2)

(Approved by A.I.C.T.E, NEW DELHI)

Peddasettypally, Proddatur-516360.

2020-2024

VAAGDEVI INSTITUTE OF TECHNOLOGY & SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project work entitled "CYBER ATTACK DETECTION MODEL BASED ON MACHINE LEARNING" is a Bonafide work of **D. LAKSHMI NARASIMHULU, C. GURU TEJA, D. SHANATH RESHON, R. PRADEEP, S. NOOR BASHA** submitted to **VAAGDEVI INSTITUTE OF TECHNOLOGY &SCIENCE, PRODDATUR** in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in **COMPUTER SCIENCE & ENGINEERING.** The work reported here in does not from part of any otherthesis on which a degree has been awarded earlier.

This is to further certify that they have worked for a period of one semester for preparing their work under our supervision and guidance.

Guide:	**Head of the department:**
Mr. T. Venkata Krishna Reddy, MTech	**Mr. V. Narasimhaswamy, MTech**
Asst, Prof. Of CSE Department,	**Asst, Prof. Of CSE Department**
Vaagdevi Institute of Technology & Science	**Vaagdevi Institute of Technology & Science**
Proddatur-516360.	**Proddatur-516360.**

External Project viva held on: ____________________

INTERNALEXAMINER **EXTERNALEXAMINER**

ACKNOWLEDGEMENT

Developing a project is not a single person's effort but it is a combined achievement of group people. So many contributions from different category of fulfilled. During the development of text mining project, right from the initiation stage to implementation stage, so many people have helped me in studying the system, researching, developing and at last preparing and at last preparing documentation.

I would like to thank and express my sincere gratitude to the correspondent Sri whole heartedly. **G. Hussain Reddy, B.E., VITS for** his constant encouragement in the development of this project. I also sincerely thanks to principal **Dr. B. Siddeswara Rao, PhD for** the constant encouragement and stimulating atmosphere provide to me.

I wish to thank **Mr. V. NarasimhaSwamy, MTech** and **HOD** of **C.S.E.** for his valuable advice and support. Most of all, I extend my sincere thanks to my project guide **Mr. T. Venkata Krishna Reddy, Asst. Prof. of C.S.E VITS** for their continuous encouragement, guidance and support throughout the development of this project.

I would like to thank to the project committee members and faculty, teaching, and non-teaching staff of CSE department, **Vaagdevi Institute of Technology and Science, Proddatur.**

Last but far from least, I also thank my parents, family members and my friends for their moral support and constant encouragement. I am very much thankful to one and all those helped me for the successful completion of this project.

PROJECT ASSOCIATES

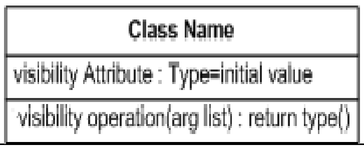
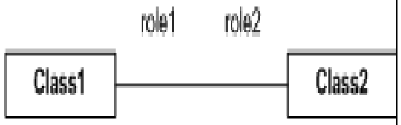
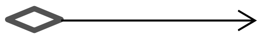
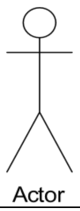
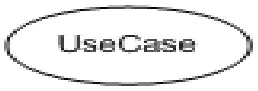
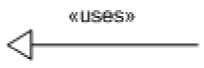
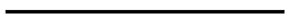
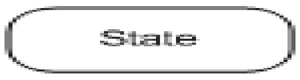
D. LAKSHMI NARASIMHULU	**20L21A0510**
C. GURU TEJA	**20L21A0508**
D. SHANATH RESHON	**20L21A0511**
R. PRADEEP	**20L21A0540**
S. NOOR BASHA	**20L21A0543**

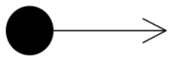
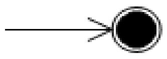
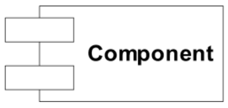
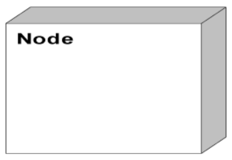
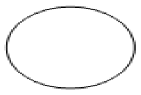
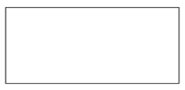
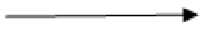
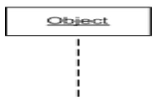
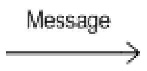
INDEX

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF SYMBOLS

S.NO	SYMBOL NAME	SYMBOL	DESCRIPTION
1.	Class	 Class Name visibility Attribute : Type=initial value visibility operation(arg list) : return type()	Classes represent a collection of similar entities grouped together.
2.	Association		Association represents a static relationship between classes.
3.	Aggregation		Aggregation is a form of association. It aggregates several classes into single class.
4.	Actor		Actors are the users of the system and other external entity that react with the system.
5.	Use case		A use case is an interaction between the system and the external environment.
6.	Relation (Uses)		It is used for additional process communication.
7.	Communication		It is the communication between various use cases.
8.	State		It represents the state of a process. Each state goes Through various flow.

9.	Initial State		It represents the initialstate of the object.
10.	Final State		It represents the final stateof the object.
11.	Component		Components represent the physical components used in the system.
12.	Node		Deployment diagrams use thenodes for representing physical modules, which is a collection of components.
13.	Data Process/State		A circle in DF Represents a state or process which has been triggered due some event or action.
14.	External Entity		It represents any external entity such as keyboard, sensors whichare used in the system.
15.	Transition		It represents any communication that occursbetween.
16.	Object Lifeline		Object life lines represents the vertical dimension that objects communicate.
17.	Message		It represents the messages exchanged.

ABSTRACT

Stood out from the past, enhancements in PC and correspondence advancements have given expansive and moved changes. The utilization of new developments gives inconceivable benefits to individuals, associations, and governments, nevertheless, some against them. For example, the assurance of critical information, security of set aside data stages, availability of data, etc. Dependent upon these issues, advanced anxiety-based abuse is perhaps the main issues nowadays. Computerized fear, which made a lot of issues individuals and foundations, has shown up at a level that could subvert open and country security by various social occasions, for instance, criminal affiliation, capable individuals and advanced activists. Thusly, Intrusion Detection Systems (IDS) has been made to keep an essential separation fromadvanced attacks.

CHAPTER-1

INTRODUCTION

Lately, the world has seen a critical evolution in the various spaces of associated innovations like brilliant matrices, the Internet of vehicles, long haul advancement, and 5G correspondence. By 2022, it is normal that the quantity of IP-associated gadgets will be multiple times bigger than the worldwide populace, delivering 4.8 ZB of IP traffic yearly, as revealed by Cisco [1]. This sped up development raises overpowering security worries because of the trading of enormous measures of sensitive data through asset compelled gadgets and over the untrusted "Internet" utilizing heterogeneous advances and correspondence conventions. To keep up feasible and secure the internet, progressed security controls and flexibility investigation ought to be applied in the prior stages before sending.

The applied security controls are answerable for forestalling, identifying, and reacting to assaults. For location purposes an interruption recognition framework (IDS) is a generally utilized procedure for identifying interior and outer interruptions that objective a system, just as irregularities that show likely interruptions and dubious exercises. An IDS includes a bunch of instruments and mech anises for observing the PC framework and the organization traffic, as well as breaking down exercises with the point of detecting potential interruptions focusing on the framework. An IDS can be executed as signature-based, inconsistency based, or mixture IDS. In signature-based IDS, interruptions are identified by contrasting observed practices and pre-characterized interruption designs, while oddity put together IDS centers with respect to knowing typical conduct in or der to distinguish any deviation [2]. Various strategies are utilized to recognize oddities, for example, factual based, information based, and AI procedures; as of late, profound learning techniques have been researched.

Presentation PC wrong doings continue growing consistently. They are not simply bound to irrelevant demonstrations, for instance, evaluating the login accreditations of a structure yet what's more they are essentially riskier. Information security is the route toward protecting information from unapproved will, use, openness, destruction, change or damage. used correspondingly.

1.2 LITERATURE REVIEW

This segment presents different late achievements around here. It ought to be noticed that we just examine the work that have utilized the NSL-KDD dataset for their performance benchmarking. Subsequently, any dataset alluded from here on out ought to be considered as NSL-KDD. This methodology permits a more exact examination of work with other found in the writing. Another restriction is the utilization of preparing information for both preparing and testing by most work. At long last, we examine a couple of profound learning-based methodologies that have been attempted so far for comparable sort of work.

One of the most punctual works found in writing utilized ANN with improved strong back- spread for the plan of such an IDS [6]. This work utilized just the preparation dataset for preparing (70%), approval (15%) and testing (15%).

As expected, utilization of unlabeled information for testing brought about a reduction of execution. A later work utilized J48 choicetree classifier with 10-overlay cross-approval for testing on the preparation dataset [4]. This work utilized a decreased list of capabilities of 22 highlights rather than the full arrangement of 41 highlights. A comparable work assessed different well known regulated tree-based classifiers and tracked down that Random Tree model performed best with the most extensivelevel of exactness alongside a decreased bogus alert rate [5].

Numerous 2-level characterization approaches have likewise been mastering presented. One such work utilized Discriminative Multinomial Naive Bayes (DMNB) as a base classifier and Nominal-to Binary directed separating at the second level alongside 10-crease cross approval for testing [9]. This work was hiding the reached out to utilize Ensembles of Balanced Nested Dichotomies (END) at the main level and Random Forest at the second level [10]. True to form, this upgrade resulted in an improved location rate and a lower bogus positive rate. Another 2-level execution utilized head segment examination (PCA) for the list of capabilities decrease and afterward SVM (utilizing Radial Basis Function) for last classification, brought about a high recognition precision with just the preparation dataset and full 41 highlights set. A decrease in features set to 23 came about in far better location exactness in a portion of the assault classes, however the general execution was diminished [11].

CHAPTER 2

SYSTEM ANALYSIS

2.1 Existing system:

Within the ever-growing and quickly increasing field of cyber security, it is nearly impossible to quantify or justify the explanations why cyber security has such an outsized impact. Permitting malicious threats to run any place, at any time or in any context is a long way from being acceptable, and may cause forceful injury. It particularly applies to the Byzantine web ofconsumers and using the net and company information that cyber security groups are findingit hard to shield and contain. Cyber security may be a necessary thought for people and families alike, also for businesses, governments, and academic establishments that operate inside the compass of the world network or net. With the facility of Machine Learning, we will advance the cyber security landscape. Today's high-tech infrastructure, that has network and cyber security systems, is gathering tremendous amounts of data and analytics on almost all the key aspects of mission-critical systems. Whereas people still give the key operational oversight and intelligent insights into today's infrastructure. Most intrusion detection systems are focused on the perimeter attack surface threats, starting with your firewall. That offers protection of your network's north south traffic, but what it doesn't take into account is the lateral spread (east- west) that many network threats today take advantage of as they infiltrate your organization's network and remain their unseen. We know this is true because research has shown that only 20% of discovered threats come from north south monitoring. When an IDS (Intrusion Detection System) detects suspicious activity, the violation is typically reported to a security information and event management (SIEM) system where real threats are ultimately determined amid benign traffic abnormalities or other false alarms. However, the longer it takes to distinguish a threat, the more damage can be done. An IDS is immensely helpful for monitoring the network, but their usefulness all depends on what you do with the information that they give you. Because detection tools don't block or resolve potential issues, they are ineffective at adding a layer of security unless you have the right personnel and policy to administer them and act on any threats.

a. An IDS cannot see into encrypted packets, so intruders can use them to slip into the network.

b. An IDS will not register these intrusions until they are deeper into the network, which leaves your systems vulnerable until the intrusion is discovered. This is a huge concern as encryption is becoming more prevalent to keep our data secure.
c. Significant issue with an IDS is that they regularly alert you to false positives. In many cases false positives are more frequent than actual threats.

2.2 PROPOSED SYSTEM:

At the present time, assessments of help vector machine, ANN, CNN, Random Forest and significant learning estimations reliant upon current CICIDS2017 dataset were presented moderately. Results show that the significant learning estimation performed generally best results over SVM, ANN, RF and CNN. We will use port scope attempts just as other attack types with AI and significant learning computations, Apache Hadoop and shimmer advancements together ward on this dataset later on. So, by utilizing these datasets we will anticipate if digital assault is finished. These forecasts should be possible by four calculations like SVM, ANN, RF, CNN this paper assists with distinguishing which calculation predicts the best precision rates which assists withforeseeing best outcomes to recognize the digital assaults occurred or not.

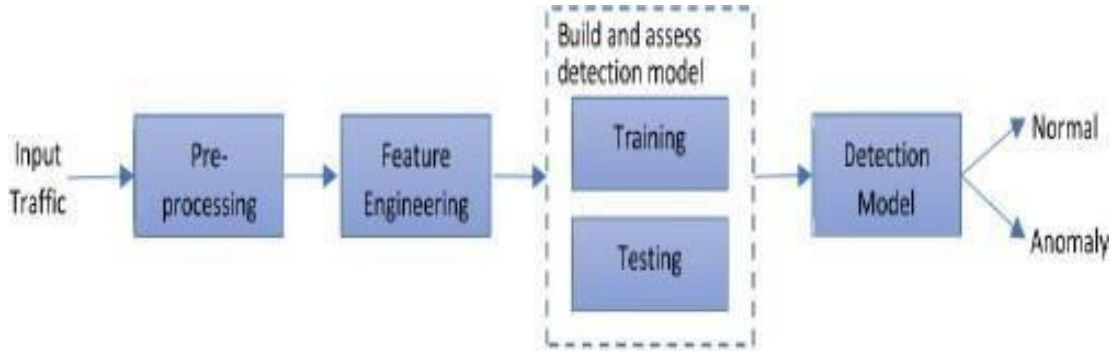
Advantages:

Advantages of the proposed systems are follows:

- a. Protection from malicious attacks on your network.
- b. Deletion and/or guaranteeing malicious elements within a preexisting network.
- c. Prevents users from unauthorized access to the network.
- d. Deny's programs from certain resources that could be infected.
- e. Securing confidential information.

So, by utilizing these datasets we will anticipate if digital assault is finished. These forecasts should be possible by four calculations this paper assists with distinguishing which calculation predicts the best precision rates which assists with foreseeing best outcomes to recognize the digital assaults occurred or not.

a. SYSTEM ARCHITECTURE



2.3 FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are,

2.3.1 ECONOMICAL FEASIBILITY
2.3.2 TECHNICAL FEASIBILITY
2.3.3 SOCIAL FEASIBILITY

2.3.1 ECONOMICAL FEASIBILITY

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

2.3.2 TECHNICAL FEASIBILITY

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

2.3.3 SOCIAL FEASIBILITY

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

CHAPTER-3

SYSTEM REQUIREMENTS SPECIFICATION

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screento the other well-ordered and at the same time reducing the amount of typing the user needs to-do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

3.1 FUNCTIONAL REQUIREMENTS

Functional requirement should include function performed by a specific screen outline work- flows performed by the system and other business or compliance requirement the system must meet. Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified. The functional specification describes what the system mustdo, how the system does it is described in the design specification. If a user requirement specification was written, all requirements outlined in the user requirements specifications should be addressed in the functional requirements.

3.2 NON-FUNCTIONAL REQUIREMENTS

Describe user-visible aspects of the system that are not directly related with the functional behave the input is designed in such a way so that it provides security and ease of use retaining the privacy or of the system. Input Design is the process of converting a user-oriented description of the input into a computer-based system. When the data is entered it will check for its validity. Data can be entered with the help of screens. On-Functional requirements include quantitative constraints, such as response time (i.e. how fast the system reacts to user commands.) or accuracy numerical answers).

3.3 INPUT & OUTPUT

3.3.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of usewith retaining the privacy. Input Design considered the following things:

a. What data should be given as input?
b. How the data should be arranged or coded?
c. The dialog to guide the operating personnel in providing input.
d. Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities. Appropriate messages are provided as when needed so that the user will not be in maize of instant.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant.

3.3.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

OBJECTIVES

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

a. Convey information about past activities, current status or projections of the

b. Future.

c. Signal important events, opportunities, problems, or warnings.

d. Trigger an action.

e. Confirm an action.

3.4 SYSTEMS REQUIREMENT AND SPECIFICATION

3.4.1 HARDWARE REQUIREMENT SPECIFICATION:

Processor	**: intel core i5**
HDD	**:** 80 GB
Ram	**: 2 GB**

3.4.2 SOFTWARE REQUIREMENT SPECIFICATION:

Operating system	**:** windows 10/11
Programming language	**:** python 3.6.0
Front end	**:** HTML/CSS/XHTML
Dataset	**:** cyber-attack dataset
Tools	**:** python

CHAPTER- 4

SYSTEM DESIGN

4.1INTRODUCTION

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement has been specified and analyzed, system design is the first of the three technical activities - design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage. The purpose of the design phase is to plan a solution of the problem specified by the requirement document.

This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affection the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design.

System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided. During, Detailed Design, the internal logic of each of the modules specified in system design is decided. During this phase, the details of the data of a module is usually specified in a high- level design description language, which is independent.

4.2 ARCHITECTURE

Design is concerned with identifying software components specifying relationships among components. Specifying software structure and providing blue print for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is minimal clearly specified.
During the system design activities, Developers bridge the gap between the requirements specification, produced during requirements elicitation and analysis, and the system that is delivered to the user.

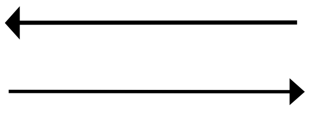
Design is the place where the quality is fostered in development. Software design is a process through which requirements are translated into a representation of software.

1. What is Data flow diagrams:

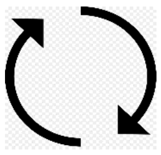
A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart.

DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts. The Basic Notation used to create a DFD's are as follows:

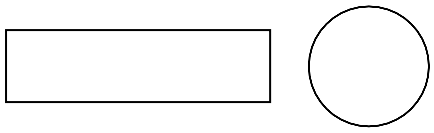
1. Dataflow: Data move in a specific direction from an origin to a destination.



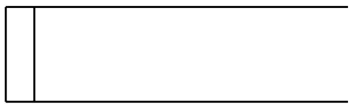
2. Process: People, procedures, or devices that use or produce (Transform) Data.



3. Source: External sources or destination of data, which may be people, programs, organizations or other entities.



4. Data Store: Here data are stored or referenced by a process in the System.



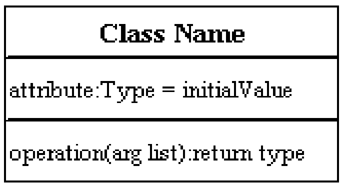
2. What is a UML Class Diagram?

Class diagrams are the backbone of almost every object-oriented method including UML. They describe the static structure of a system.

Basic Class Diagram Symbols and Notations

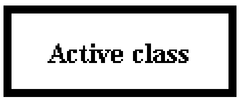
Classes represent an abstraction of entities with common characteristics. Associations represent the relationships between classes.

Illustrate classes with rectangles divided into compartments. Place the name of the class in the first partition (centered, bolded, and capitalized), list the attributes in the second partition third.


Class Name
attribute:Type = initialValue
operation(arg list):return type

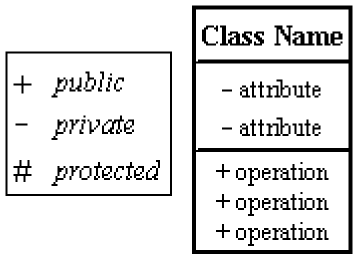
a. Active Class

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.



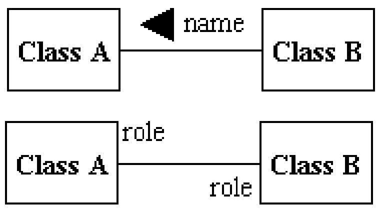
b. Visibility

Use visibility markers to signify who can access the information contained within a class. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class. .



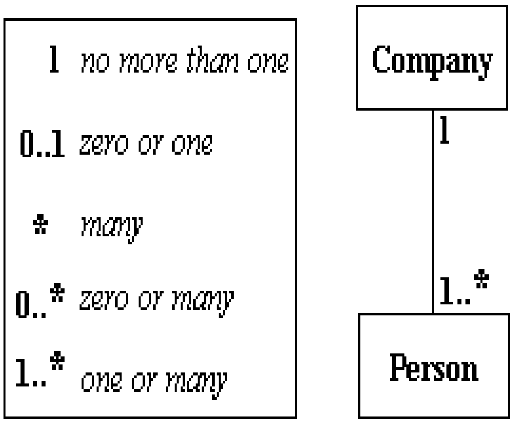
c. Associations

Associations represent static relationships between classes. Place association names above, on, or below the association line. Use a filled arrow to indicate the direction of the relationship. Place roles near the end of an association. Roles represent the way the two classes see each other. Use a filled arrow to indicate the direction of the relationship. Place roles near the end of an association.



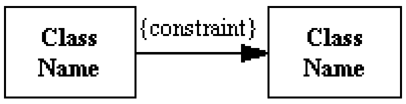
d. Multiplicity (Cardinality)

Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one class linked to one instance of the other class. For example, one company will have one or more employees, but each employee works for one company only. In this example, "1" near the company class indicates that one company instance can be linked to multiple employee instances. The "*" near the employee class indicates that each employee instance can work for one and only one company instance.

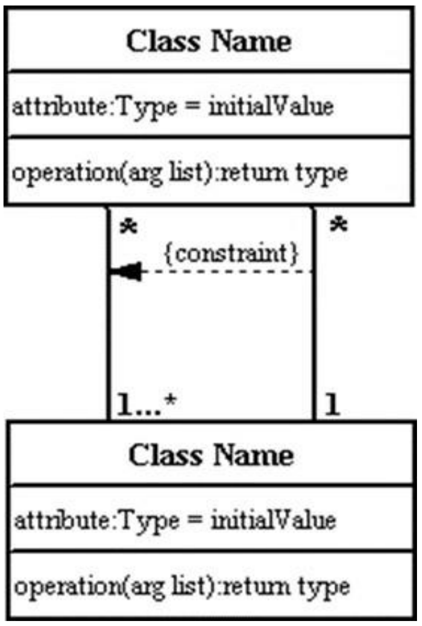


Constraint

Place constraints inside curly braces {}.



Simple Constraint

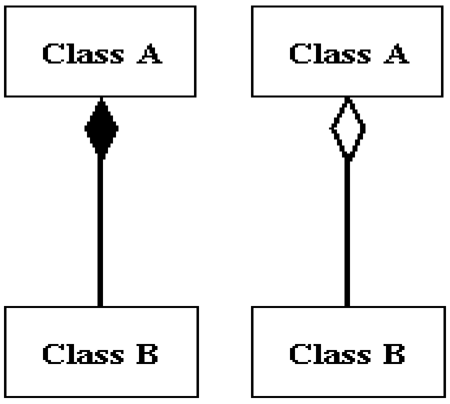


Class Name
attribute:Type = initialValue
operation(arg list):return type
*
*
{constraint}
1...*
1
Class Name
attribute:Type = initialValue
operation(arg list):return type

Fig: Class name

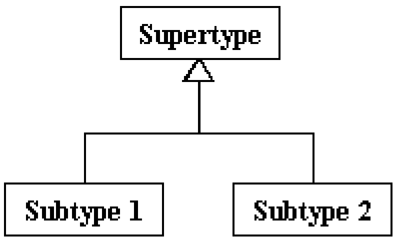
e. Composition and Aggregation

Composition is a special type of aggregation that denotes a strong ownership between Class A, the whole, and Class B, its part. Illustrate composition with a filled diamond. Use a hollow diamond to represent a simple aggregation relationship, in which the "whole" class plays a more important role than the "part" class, but the two classes are not dependent on each other. The diamond end in both a composition and aggregation relationship points toward the "whole" class or the aggregate



f. Generalization

Generalization is another name for inheritance or an "is a" relationship. It refers to a relationship between two classes where one class is a specialized version of another. For example, Honda is a type of car. So, the class Honda would have a generalization relationship with the class car.



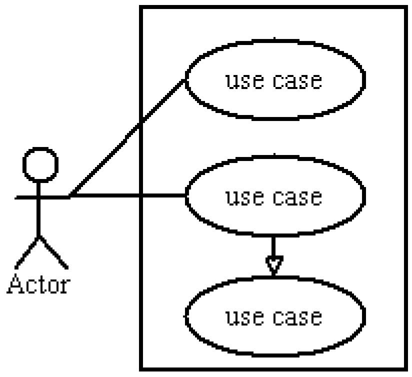
In real life coding examples, the difference between inheritance and aggregation can be confusing. If you have an aggregation relationship, the aggregate (the whole) can access only the PUBLIC functions of the part class. On the other hand, inheritance allows the inheriting class to access both the PUBLIC and PROTECTED functions of super class.

3. What is a UML Use Case Diagram?

Use case diagrams model the functionality of a system using actors and use cases. Use cases are services or functions provided by the system to its users.

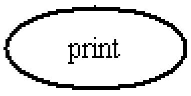
Basic Use Case Diagram Symbols and Notations System

Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.



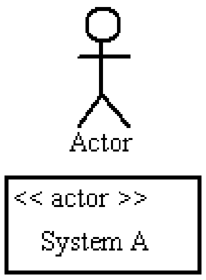
Use Case

Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.



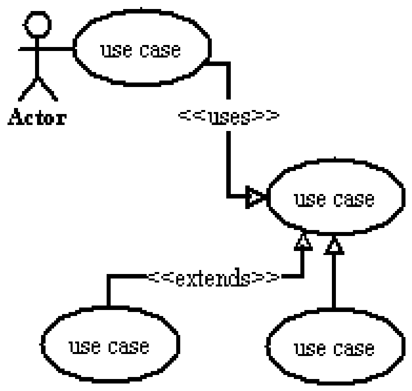
a. Actors

Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.



b. Relation ships

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



c. Class roles

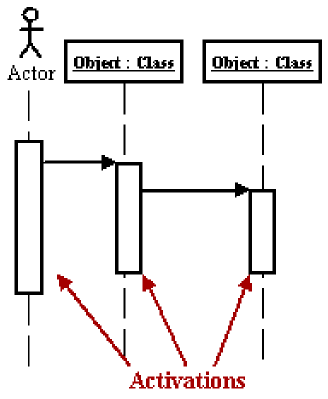
Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes. Class roles clarify the responsibilities and interactions of each class within a system.

They help stakeholders understand the purpose and behavior of classes, facilitating better system design and communication.

Object : Class

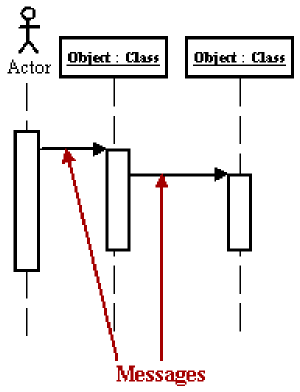
d. Activation

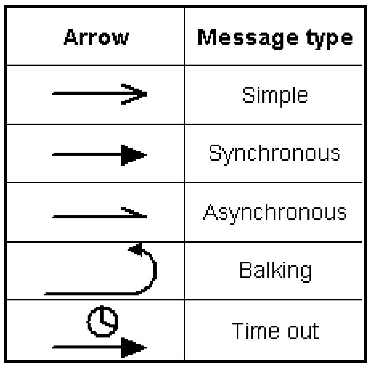
Activation boxes represent the time an object needs to complete a task. Activation visually illustrates the lifespan of an object's activity or behavior during runtime. It helps in understanding the sequence and duration of interactions between objects in a system.



e. Messages

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.





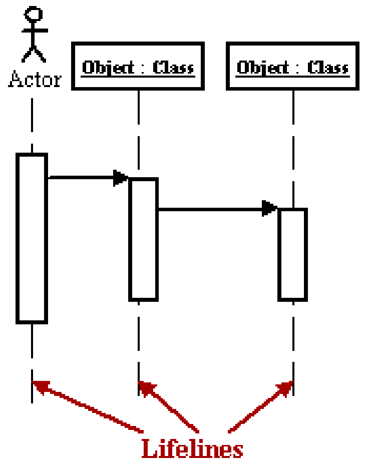
Arrow	Message type
→	Simple
▶	Synchronous
⇀	Asynchronous
↩	Balking
◷ ▶	Time out

Various message types for Sequence and Collaboration diagrams

f. Lifelines

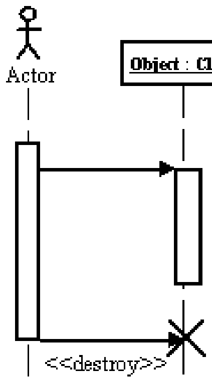
Lifelines depict the existence and lifespan of an object during the execution of a sequence or interaction diagram. They provide a visual representation of how objects interact and communicate with each other over time, aiding in understanding system behavior and sequence of events.

Lifelines are vertical dashed lines that indicate the object's presence over time.



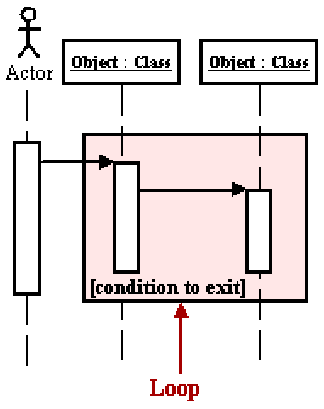
g. Destroying Objects

Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X.



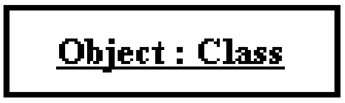
h. Loops

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [ ].



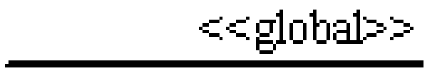
i. Class roles

Class roles describe how objects behave. Use the UML object symbol to illustrate class roles, but don't list object attributes.



j. Association roles

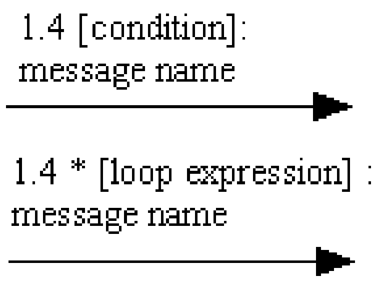
Association roles describe how an association will behave given a particular situation. You can draw association roles using simple lines labeled with stereotypes.



k. Messages

Unlike sequence diagrams, collaboration diagrams do not have an explicit way to denote time and instead number messages in order of execution. Sequence numbering can become nested using the Dewey decimal system. For example, nested messages under the first message are labeled 1.1, 1.2, 1.3, and so on. A condition for a message is usually placed in square brackets immediately following the sequence number. Use a * after the sequence number to indicate
a loop.

Learn how to add arrows to your lines.



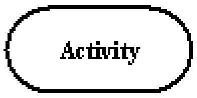
4. What is Activity Diagram?

An activity diagram illustrates the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation. Because an activity diagram is a special kind of state chart diagram, it uses some of the same modeling conventions.

Basic Activity Diagram Symbols and Notations

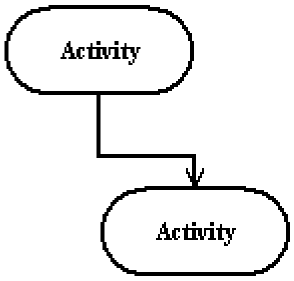
a. Action states

Action states represent the non-interruptible actions of objects. You can draw an action state in Smart Draw using a rectangle with rounded corners.



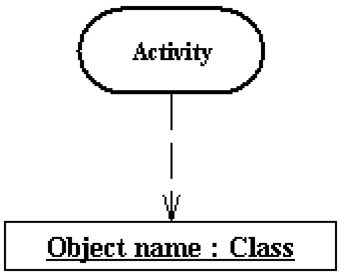
b. Action Flow

Action flow arrows illustrate the relationships among action states.



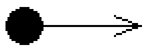
c. Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



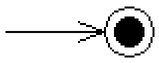
d. Initial State

A filled circle followed by an arrow represents the initial action state.



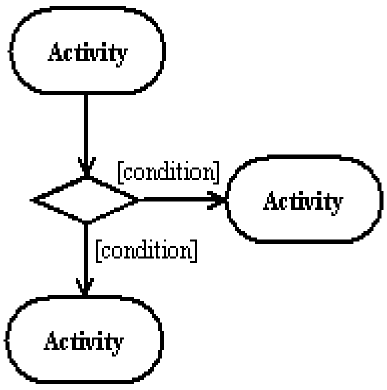
e. Final State

An arrow pointing to a filled circle nested inside another circle represents the final action state.



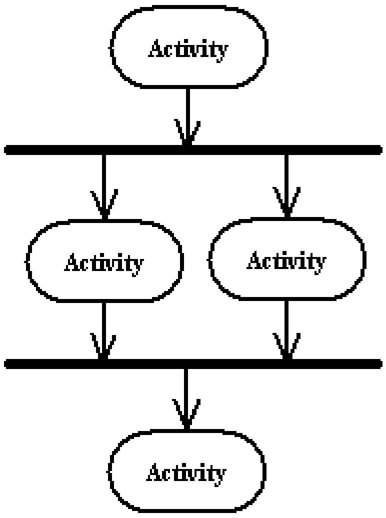
f. Branching

A diamond represents a decision with alternate paths. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."



g. Synchronization

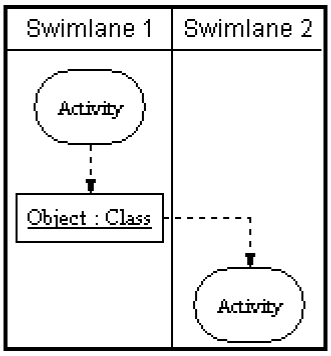
A synchronization bar helps illustrate parallel transitions. Synchronization is also called forking and joining.



States represent situations during the life of an object. You can easily illustrate a state in draw by using a rectangle with rounded corners.

h. Swim lanes

Swim lanes group related activities into one column.

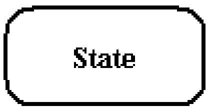


State chart Diagram

A state chart diagram shows the behavior of classes in response to external stimuli. This diagram models the dynamic flow of control from state to state within a system.

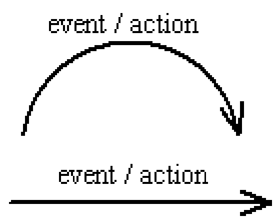
Basic State chart Diagram Symbols and Notations

i. States



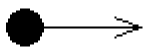
j. Transition

A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it.



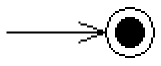
k. Initial State

A filled circle followed by an arrow represents the object's initial state.



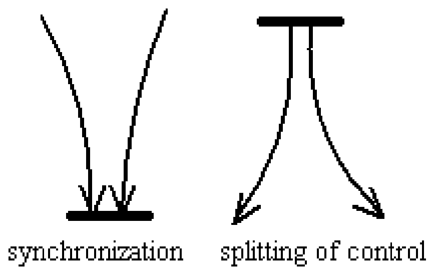
l. Final State

An arrow pointing to a filled circle nested inside another circle represents the object's final state.



m. Synchronization and Splitting of Control

A short heavy bar with two transitions entering it represents a synchronization of control. A short heavy bar with two transitions leaving it represents a splitting of control that creates multiple states.



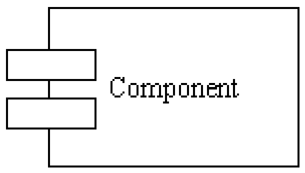
5. What is a UML Component Diagram?

A component diagram describes the organization of the physical components in a system. Basic Component Diagram Symbols and Notations

a. Component

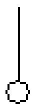
A component is a physical building block of the system. It is represented as a rectangle with tabs.

Learn how to resize grouped objects like components.



b. Interface

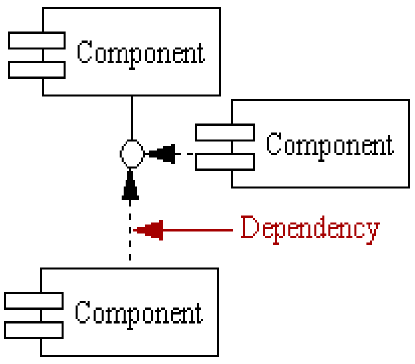
An interface describes a group of operations used or created by components.



c. Dependencies

Draw dependencies among components using dashed arrows.

Learn about line styles in Smart Draw.



6. What is a UML Deployment Diagram?

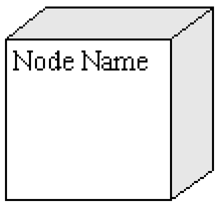
Deployment diagrams depict the physical resources in a system including nodes, components, and connections.

Basic Deployment Diagram Symbols and

Notations

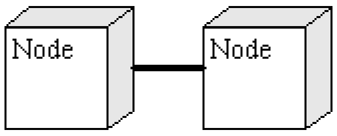
a. Component

A node is a physical resource that executes code components. Learn how to resize grouped objects like nodes.

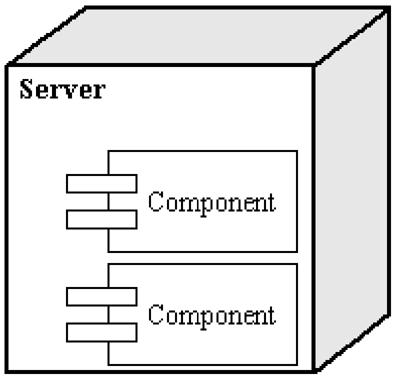


b. Association

Association refers to a physical connection between nodes, such as Ethernet. Learn how to connect two nodes.



c. Components and Nodes



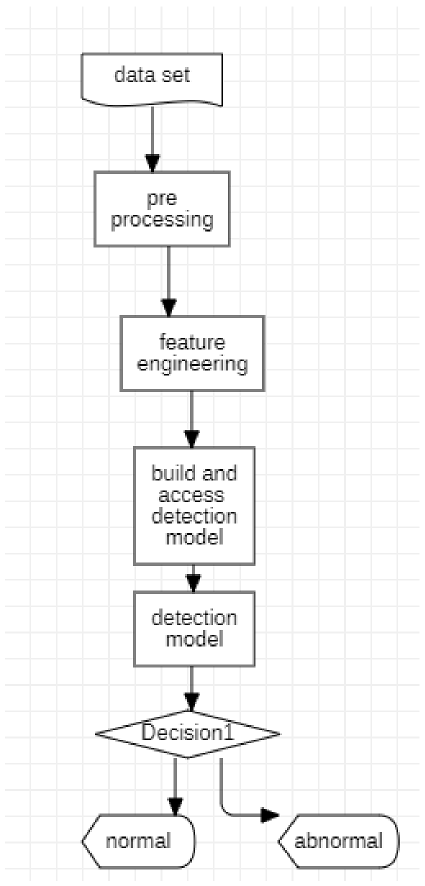
4.3 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



4.4 UML DIAGRAMS

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

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

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

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

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

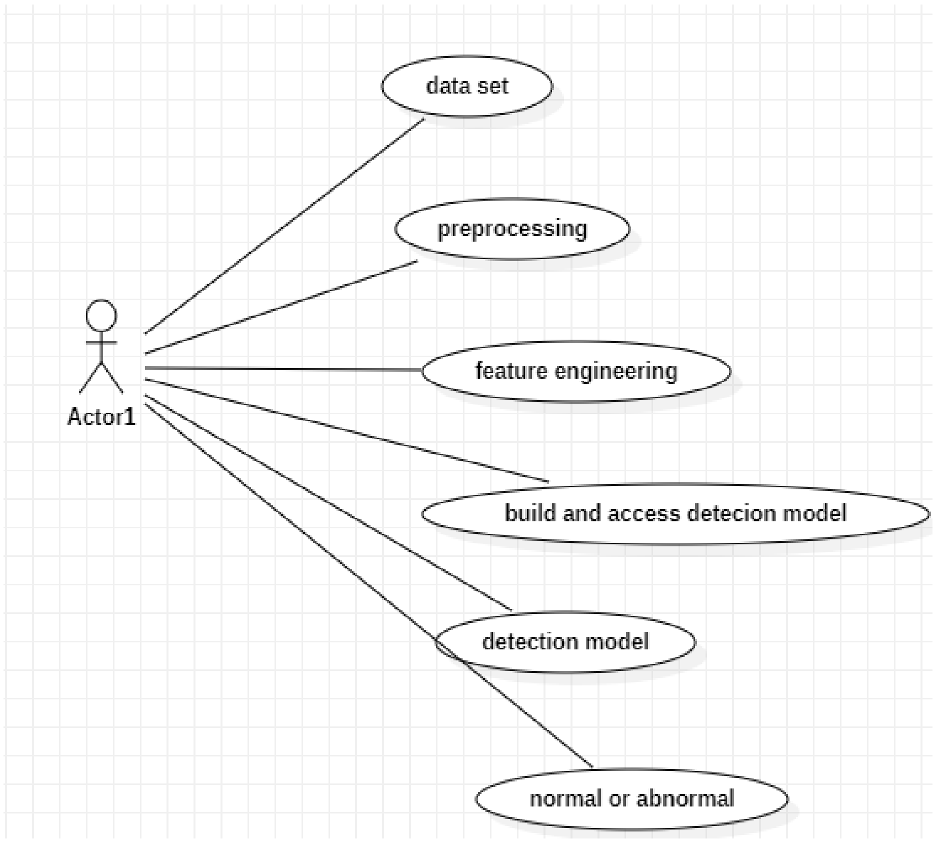
a. GOALS:

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

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

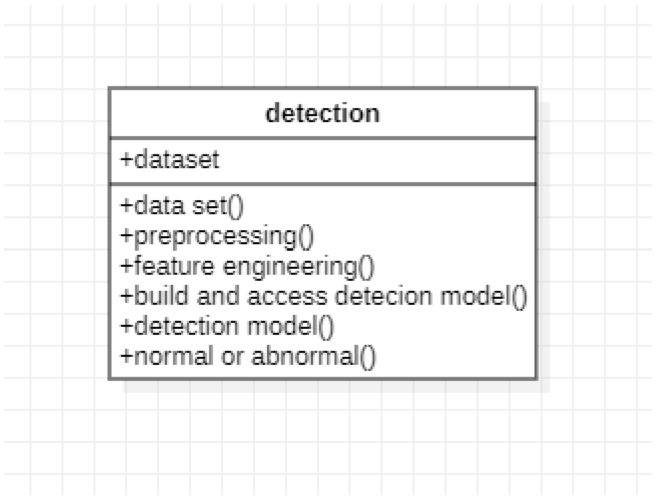
4.5 USE CASE DIAGRAM:

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 purposeis 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 systemcan be depicted.



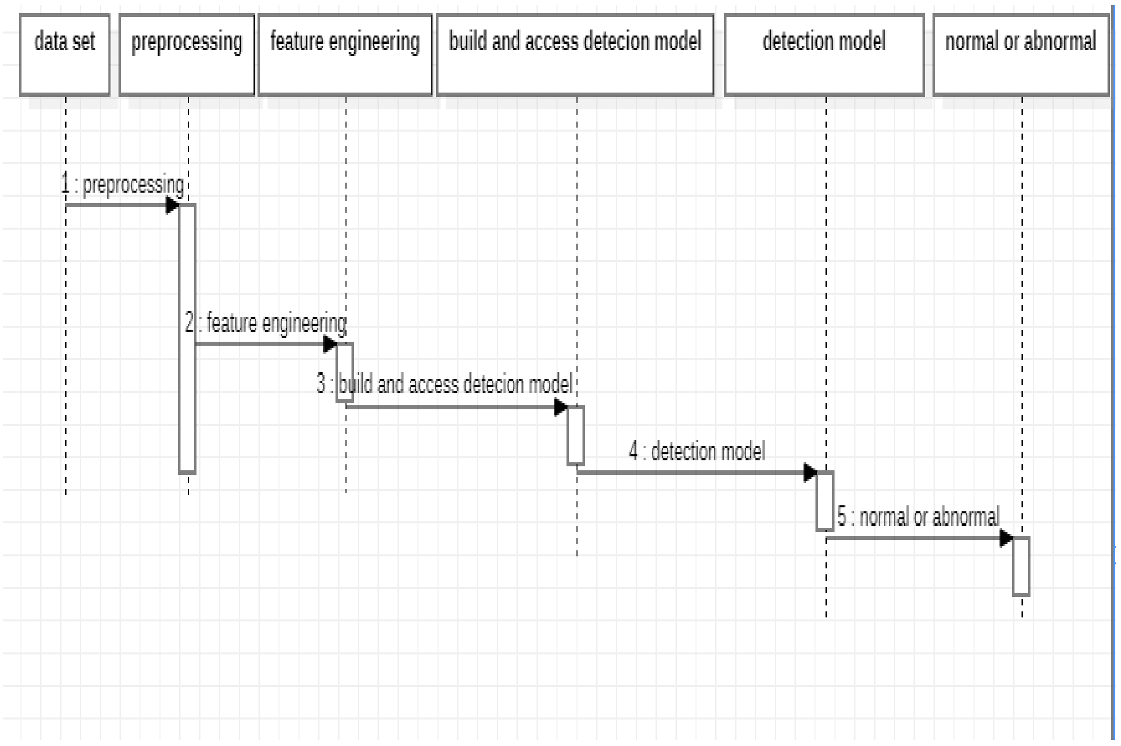
4.6 CLASS DIAGRAM:

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.



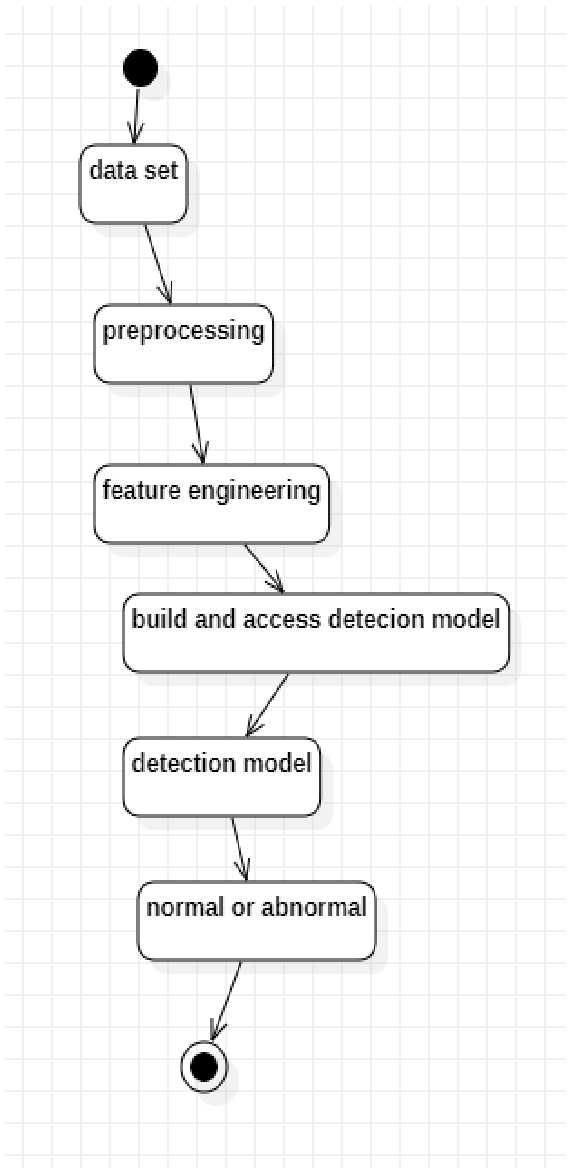
4.7 SEQUENCE DIAGRAM:

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



4.8 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



Diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency

CHAPTER-5

IMPLEMENTATION

5.1 MODULE & DESCRIPTION

a. Modules

1. Data collection
2. Data pre-processing
3. Training and testing
4. Attack detection

b. Modules description

1. Data Collection:

Collect sufficient data samples and legitimate software samples.

2. Data Preprocessing:

Data Augmented techniques will be used for better performance

3. Train and Test Modeling:

Split the data into train and test data Train will be used for training the model and Test data to check the performance.

4. Attack Detection Model:

Based on the model trained algorithm will detect whether the given transaction is anomalous or not. Feedback from the response mechanisms and any new data collected during operation are used to continuously improve the model over time.

These are the modules we had used in this implementation. C Python is managed by the non- profit Python Software Foundation. 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.

5.2 TECHNOLOGY DESCRIPTION

a. Introduction of Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non- profit Python Software Foundation. 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.

b. What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

1. System web development (server-side),
2. Software development,
3. Mathematics,
4 . Scripting.

c. What can Python do

1. Python can be used on a server to create web applications.

2.Python can be used alongside software to create workflows.

3. Python can connect to database systems. It can also read and modify files.
4. Python can be used to handle big data and perform complex mathematics.
5. Python can be used for rapid prototyping, or for production-ready software.

d. Why Python

1. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).

2. Python has a simple syntax similar to the English language.

3. Python has syntax that allows developers to write programs with fewer lines than some programming languages.
4. Python runs on an interpreter system, meaning that code can be executed as soon as itis written. This means that prototyping can be very quick.
5. Python can be treated in a procedural way, an object-orientated way or a functional way.

e. Good to know

1. The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

2. In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, PyCharm, NetBeans or Eclipse which are particularly useful when managing larger collections of Python files.

f. Python Syntax compared to other programming languages

1. Python was designed for readability, and has some similarities to the English language with influence from mathematics.
2. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
3. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

g. Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on Linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download itfor free from the following website: https://www.python.org/

h. Python QuickStart

Python is an interpreted programming language; this means that as a developer you Python (.py). Python is an interpreted programming language; this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed. Practice coding regularly to reinforce your learning. Write small scripts, solve coding challenges, or contribute to open-source projects to gain hands-on experience with Python. By focusing on these key points, you can quickly start your journey with Python and gradually build your proficiency in the language. Choose a learning resource that suits your learning style and goals, and start exploring the language at your own pace.

By following these three points, you can quickly get started with Python and begin your journey to becoming proficient in the language. Write small scripts, solve coding challenges, or contribute to open-source projects to gain hands-on experience with python.

i. The way to run a python file is like this on the command line:

C:\Users\Your Name>python helloworld.py

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.helloworld.py

print ("Hello, World!")

Simple as that. Save your file. Open your command line, navigate to the directory where yousaved your file, and run:

C:\Users\Your Name>python helloworld.pyThe output should read:

Hello, World!

Congratulations, you have written and executed your first Python program.

j. The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write thecode in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line: C:\Users\Your Name>python

Or, if the "python" command did not work, you can try "py":C:\Users\Your Name>py

The "Python command line" typically refers to the command-line interface (CLI) provided by Python, which allows users to interactively execute Python code and run scripts by following these three points, you can quickly get started with Python and begin your journey to becoming proficient in the language. By following these three points, you can quickly get started with Python and begin your journey.

k. From there you can write any python, including our hello world example from earlier inthe tutorial:

```
C:\Users\Your Name>python

Python 3.6.4 (v3.6.4: d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]

onwin32Type "help", "copyright", "credits" or "license" for more information.

>>>print ("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python

Python 3.6.4 (v3.6.4: d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]

onwin32Type "help", "copyright", "credits" or "license" for more information.

>>>print ("Hello, World!")Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quitthe python command line interface:

```
exit ()
```

Virtual Environments and Packages

a. Introduction

Python applications will often use packages and modules that don’t come as part of the standardlibrary. Applications will sometimes need a specific version of a library, because the application may require that a particular bug has been fixed or the application may be written using an obsolete version of the library’s interface.

This means it may not be possible for one Python installation to meet the requirements of everyapplication. If application A needs version 1.0 of a particular module but application B needs version 2.0, then the requirements are in conflict and installing either version 1.0 or 2.0 will leave one application unable to run.

The solution for this problem is to create a virtual environment, a self-contained directory treethat contains a Python installation for a particular version of Python, plus a number of additional packages.
Different applications can then use different virtual environments. To resolve the earlier example of conflicting requirements, application A can have its own virtual environment with version 1.0 installed while application B has another virtual environment with version 2.0. If application B requires a library be upgraded to version 3.0, this will not affect application A'senvironment.

b. Creating Virtual Environments

The module used to create and manage virtual environments is called venv. venv will usually install the most recent version of Python that you have available. If you have multiple versionsof Python on your system, you can select a specific Python version by running python3 or whichever version you want.
To create a virtual environment, decide upon a directory where you want to place it, and run the venv module as a script with the directory path:

```
python3 -m venv tutorial-env
```

This will create the tutorial-env directory if it doesn't exist, and also create directories inside it containing a copy of the Python interpreter, the standard library, and various supporting files. A common directory location for a virtual environment is. venv. This name keeps the directory typically hidden in your shell and thus out of the way while giving it a name that explains why the directory exists. It also prevents clashing with. env environment variable definition files that some tooling supports. This name keeps the directorytypically hidden in your shell and thus out of the way while giving it a name that explains whythe directory exists. It also prevents clashing with. env environment variable definition files that some tooling supports. If you have multiple versions of Python on your system, you can select a specific Python version by running python3 or whichever version you want. This name keeps the directory typically hidden in your shell and thus out of the way while giving it a name that explains why the directory exists. Once you've created a virtual environment, you may activate it.

Once you've created a virtual environment, you may

activate it.On Windows, run:

```
tutorial-

env\Scripts\activate.bat
```

On Unix or MacOS, run:

```
source tutorial-env/bin/activate
```

(This script is written for the bash shell. If you use the csh or fish shells, there are alternate activates and activate. Fish scripts you should use instead.)
Activating the virtual environment will change your shell's prompt to show what virtual environment you're using, and modify the environment so that running python will get you that particular version and installation of Python. For example:

```
$ source ~/envs/tutorial-

env/bin/activate(tutorial-env) $

python

Python 3.5.1 (default, May  6 2016, 10:59:36)

...

>>> import sys

>>>sys.path

['', '/usr/local/lib/python35.zip', ...,

'~/envs/tutorial-env/lib/python3.5/site-

packages']

>>>
```

Managing Packages with pip

You can install, upgrade, and remove packages using a program called pip. By default, pip willinstall packages from the Python Package Index, <https://pypi.org>.

```
(tutorial-env) $ pip search astronomy

Skyfield    - Elegant astronomy for Python

gary        - Galactic astronomy and gravitational dynamics.

novas       - The United States Naval Observatory NOVAS astronomy libraryastroobs

            - Provides astronomy ephemeris to plan telescope observations PyAstronomy

            - A collection of astronomy related tools for Python.

...
```

pip has a number of subcommands: “search”, “install”, “uninstall”, “freeze”, etc. (Consult theInstalling Python Modules guide for complete documentation for pip.)

You can install the latest version of a package by specifying a package’s name:

```
(tutorial-env) $ pip install novas

Collecting novas

Downloading novas-3.1.1.3.tar.gz (136kB)Installing collected packages: novas Running

setup.py install for novas Successfully installed novas-3.1.1.3
```

You can also install a specific version of a package by giving the package name followed by
== and the version number:

```
(tutorial-env) $ pip install requests==2.6.0Collecting requests==2.6.0

Using cached requests-2.6.0-py2.py3-none-any.whlInstalling collected packages:

requests
```

pip has a number of subcommands: “search”, “install”, “uninstall”, “freeze”, etc. (Consult theInstalling Python Modules guide for complete documentation for pip.
If you re-run this command, pip will notice that the requested version is already installed and do nothing. You can supply a different version number to get that version, or you can run pip install --upgrade to upgrade the package to the latest.

```
(tutorial-env) $ pip install --upgrade

requestsCollecting requests

Installing collected packages:

requests Found existing installation:

requests 2.6.0

Uninstalling requests-2.6.0:

Successfully uninstalled requests-

2.6.0Successfully installed requests-

2.7.0
```

pip uninstall followed by one or more package names will remove the packages from the virtualenvironment.

pip show will display information about a particular package:

```
(tutorial-env) $ pip show requests

---

Metadata-

Version: 2.0

Name: requests

Version: 2.7.0

Summary: Python HTTP for

Humans.              Home-page:

http://python-requests.org

Author: Kenneth Reitz
```

pip uninstall followed by one or more package names will remove the packages from the virtualenvironment.

Author-email: me@kennethreitz.com

License: Apache 2.0

Location: /Users/akuchling/envs/tutorial-env/lib/python3.4/site-

packagesRequires:

pip list will display all of the packages installed in the virtual environment:

(tutorial-env) $

pip listnovas

(3.1.1.3)

numpy (1.9.2)

pip (7.0.3)

requests (2.7.0)

setuptools (16.0)

pip freeze will produce a similar list of the installed packages, but the output uses the formatthat pip install expects. A common convention is to put this list in a requirements.txt file:
(tutorial-env) $ pip freeze >

requirements.txt(tutorial-env) $ cat

requirements.txt novas==3.1.1.3

numpy==

1.9.2

requests=

=2.7.0

The requirements.txt can then be committed to version control and shipped as part of anapplication. Users can then install all the necessary packages with install -r:
(tutorial-env) $ pip install -r requirements.txt

Collecting novas==3.1.1.3 (from -r requirements.txt (line 1))

...

Collecting numpy==1.9.2 (from -r requirements.txt (line 2))

...

Collecting requests==2.7.0 (from -r requirements.txt (line 3))

...

Installing collected packages: novas, numpy, requests

Running setup.py install for novas

Successfully installed novas-3.1.1.3 numpy-1.9.2 requests-2.7.0

pip has many more options. Consult the Installing Python Modules guide for complete documentation for pip. When you've written a package and want to make it available on the Python Package Index, consult the Distributing Python Modules guide.

Cross Platform

Platform. Architecture (executable=sys.executable, bits='', linkage='')

Queries the given executable (defaults to the Python interpreter binary) for various architectureinformation.

Returns a tuple (bits, linkage) which contain information about the bit architecture and the linkage format used for the executable. Both values are returned as strings.

Values that cannot be determined are returned as given by the parameter presets. If bits are given'', the size of(pointer) (or size of (long) on Python version < 1.5.2) is used as indicator for thesupported pointer size.

The function relies on the system's file command to do the actual work. This is available on most if not all Unix platforms and some non-Unix platforms and then only if the executable points to the Python interpreter.

points to the Python interpreter. Reasonable defaults are used when the above needs are notmet.

Note On Mac OS X (and perhaps other platforms), executable files may be universal filescontaining multiple architectures.

To get at the “64-bitness” of the current interpreter, it is more reliable to query the sys.maxsizeattribute:

is_64bits = sys.maxsize>

2**32platform.machine

()

Returns the machine type, e.g. 'i386'. An empty string is returned if the value cannot bedetermined.

platform.node ()

Returns the computer’s network name (may not be fully qualified!). An empty string is returnedif the value cannot be determined.

platform. Platform (aliased=0, terse=0)

Returns a single string identifying the underlying platform with as much useful information aspossible.

The output is intended to be human readable rather than machine parseable. It may lookdifferent on different platforms and this is intended.
If aliased is true, the function will use aliases for various platforms that report system names which differ from their common names, for example SunOS will be reported as Solaris. The system_alias() function is used to implement this.

Setting terse to true causes the function to return only the absolute minimum information needed to identify the platform.

platform.processor()

Returns the (real) processor name, e.g. 'amdk6'.

An empty string is returned if the value cannot be determined. Note that many platforms do notprovide this information or simply return the same value as for machine(). NetBSD does this.

platform.python_build()

Returns a tuple (buildno, builddate) stating the Python build number and date as

strings.platform.python_compiler()

Returns a string identifying the compiler used for compiling

Python.platform.python_branch()

Returns a string identifying the Python implementation SCM

branch.New in version 2.6.

platform.python_implementation()

Returns a string identifying the Python implementation. Possible return values are: ‘CPython’,‘IronPython’, ‘Jython’, ‘PyPy’.

New in version 2.6.

platform.python_rev

ision()

Returns a string identifying the Python implementation SCM

revision.New in version 2.6.

platform.python_version()

Returns the Python version as string 'major.minor.patchlevel'.

Note that unlike the Python sys.version, the returned value will always include the patchlevel(it defaults to 0).

platform.python_version_tuple()

Returns the Python version as tuple (major, minor, patchlevel) of strings.
platform.release()

Returns the system's release, e.g. '2.2.0' or 'NT' An empty string is returned if the value cannotbe determined.

platform.system()

Returns the system/OS name, e.g. 'Linux', 'Windows', or 'Java'. An empty string is returned ifthe value cannot be determined.

platform.system_alias(system, release, version)

Returns (system, release, version) aliased to common marketing names used for some systems.It also does some reordering of the information in some cases where it would otherwise causeconfusion.

platform.version()

Returns the system's release version, e.g. '#3 on degas'. An empty string is returned if the valuecannot be determined.

platform.uname()

Fairly portable uname interface. Returns a tuple of strings (system, node, release, version,machine, processor) identifying the underlying platform.

Note that unlike the os.uname() function this also returns possible processor information asadditional tuple entry.

Entries which cannot be determined are set to ".

Java Platform

platform.java_ver(release=", vendor=", vminfo=(", ", "),

osinfo=(", ", "))Version interface for Jython.

Returns a tuple (release, vendor, vminfo, osinfo) with vminfo being a tuple (vm_name, vm_release, vm_vendor) and osinfo being a tuple (os_name, os_version, os_arch). Values which cannot be determined are set to the defaults given as parameters (which all default to ").

platform.win32_ver(release='', version='', csd='', ptype='')

Get additional version information from the Windows Registry and return a tuple (release, version, csd, ptype) referring to OS release, version number, CSD level (service pack) and OStype (multi/single processor).

As a hint: ptype is 'Uniprocessor Free' on single processor NT machines and 'Multiprocessor Free' on multi processor machines. The ‘Free’ refers to the OS version being free of debuggingcode. It could also state ‘Checked’ which means the OS version uses debugging code, i.e. codethat checks arguments, ranges, etc.
Note : This function works best with Mark Hammond’s win32all package installed, but also onPython 2.3 and later (support for this was added in Python 2.6). It obviously only runs on Win32compatible platforms.

Win95/98 specific

platform.popen(cmd, mode='r', bufsize=None)

Portable popen() interface. Find a working popen implementation preferring win32pipe.popen(). On Windows NT, win32pipe.popen() should work; on Windows 9x it hangs due to bugs in the MS C library.

Mac OS Platform

platform.mac_ver(release='', versioninfo=('', '', ''), machine='')

Get Mac OS version information and return it as tuple (release, versioninfo, machine) with versioninfo being a tuple (version, dev_stage, non_release_version).

Entries which cannot be determined are set to ''. All tuple entries are strings.

Unix Platforms

platform.dist(distname='', version='', id='', supported_dists=('SuSE', 'debian', 'redhat', 'mandrake', ...))

This is an old version of the functionality now provided by linux_distribution(). For new code,please use the linux_distribution().

The Python interpreter is usually installed as /usr/local/bin/python3.8 on those machines where it is available; putting /usr/local/bin in your Unix shell's search path makes it possible to start it by typing the command:

python3.8

to the shell. 1 Since the choice of the directory where the interpreter lives is an installation option, other places are possible; check with your local Python guru or system administrator. (E.g., /usr/local/python is a popular alternative location.)
On Windows machines where you have installed Python from the Microsoft Store, the python3.8 command will be available. If you have the py.exe launcher installed, you can use the py command. See Excursus: Setting environment variables for other ways to launch Python.

Typing an end-of-file character (Control-D on Unix, Control-Z on Windows) at the primary prompt causes the interpreter to exit with a zero exit status. If that doesn't work, you can exit the interpreter by typing the following command: quit().

The interpreter's line-editing features include interactive editing, history substitution and code completion on systems that support the GNU Readline library. Perhaps the quickest check to see whether command line editing is supported is typing Control-P to the first Python prompt you get. If it beeps, you have command line editing; see Appendix Interactive Input Editing and History Substitution for an introduction to the keys. If nothing appears to happen, or if ^P is echoed, command line editing isn't available; you'll only be able to use backspace to remove characters from the current line.

The interpreter operates somewhat like the Unix shell: when called with standard input connected to a tty device, it reads and executes commands interactively; when called with a file name argument or with a file as standard input, it reads and executes a script from that file.

A second way of starting the interpreter is python -c command [arg] ..., which executes the statement(s) in command, analogous to the shell's -c option. Since Python statements often contain spaces or other characters that are special to the shell, it is usually advised to quote command in its entirety with single quotes.

Some Python modules are also useful as scripts. These can be invoked using python -m module[arg] ..., which executes the source file for module as if you had spelled out its full name on thecommand line.

When a script file is used, it is sometimes useful to be able to run the script and enter interactivemode afterwards. This can be done by passing -i before the script.

All command line options are described in Command line and environment.

Argument Passing

When known to the interpreter, the script name and additional arguments thereafter are turnedinto a list of strings and assigned to the argv variable in the sys module. You can access this listby executing import sys. The length of the list is at least one; when no script and no argumentsare given, sys.argv[0] is an empty string. When the script name is given as '-' (meaning standardinput), sys.argv[0] is set to '-'. When -c command is used, sys.argv[0] is set to '-c'. When -m module is used, sys.argv[0] is set to the full name of the located module. Options found after - c command or -m module are not consumed by the Python interpreter’s option processing but left in sys.argv for the command or module to handle.

Interactive Mode

When commands are read from a tty, the interpreter is said to be in interactive mode. In this mode it prompts for the next command with the primary prompt, usually three greater-than signs (>>>); for continuation lines it prompts with the secondary prompt, by default three dots (...). The interpreter prints a welcome message stating its version number and a copyright noticebefore printing the first prompt:

```
$ python3.8

Python 3.8 (default, Sep 16 2015, 09:25:04)[GCC 4.8.2] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>
```

Continuation lines are needed when entering a multi-line construct. As an example, take a lookat this if statement:

```
>>>

>>>the_world_is_flat = True

>>>ifthe_world_is_flat:

...     print("Be careful not to fall off!")

...

Be careful not to fall off!
```

For more on interactive mode, see Interactive Mode.

The Interpreter and Its Environment

Source Code Encoding

By default, Python source files are treated as encoded in UTF-8. In that encoding, characters of most languages in the world can be used simultaneously in string literals, identifiers and

comments — although the standard library only uses ASCII characters for identifiers, a convention that any portable code should follow. To display all these characters properly, youreditor must recognize that the file is UTF-8, and it must use a font that supports all the characters in the file.

To declare an encoding other than the default one, a special comment line should be added as the first line of the file. The syntax is as follows:

```
# -*- coding: encoding -*-
```

where encoding is one of the valid codecs supported by Python.
comments — although the standard library only uses ASCII characters for identifiers, a convention that any portable code should follow. To display all these characters properly, youreditor must recognize that the file is UTF-8, and it must use a font that supports all the characters in the file a special comment line should be added as the first line of the file. To display all these characters properly, youreditor must recognize.

To declare an encoding other than the default one, a special comment line should be added as the first line of the file. The syntax is as follows:

# -*- coding: encoding -*-

where encoding is one of the valid codecs supported by Python.

For example, to declare that Windows-1252 encoding is to be used, the first line of your sourcecode file should be:

# -*- coding: cp1252 -*-

One exception to the first line rule is when the source code starts with a UNIX “shebang” line.In this case, the encoding declaration should be added as the second line of the file. For example:
#!/usr/bin/env python3 # -*- coding: cp1252 -*-

Introduction to Artificial Intelligence

“The science and engineering of making intelligent machines, especially intelligent computer programs”. -John McCarthy-
Artificial Intelligence is an approach to make a computer, a robot, or a product to think how smart human think. AI is a study of how human brain think, learn, decide and work, when it tries to solve problems. The aim ofAI is to improve computer functions which are related to human knowledge, for example, reasoning, learning, and problem-solving. And finally, this study outputs intelligent software systems. The aim ofAI is to improve computer functions which are related to human knowledge, for example, reasoning, learning, and problem-solving.

The intelligence is intangible. It is composed of

1. Reasoning
2. Learning
3. Problem Solving
4. Perception
5.Linguistic Intelligence

The objectives of AI research are reasoning, knowledge representation, planning, learning, natural language processing, realization, and ability to move and manipulate objects. There arelong-term goals in the general intelligence sector.

Approaches include statistical methods, computational intelligence, and traditional coding AI. During the AI research related to search and mathematical optimization, artificial neural networks and methods based on statistics, probability, and economics, we use many tools. Computer science attracts AI in the field of science, mathematics, psychology, linguistics, philosophy and so on.

Trending AI Articles:

1. Cheat Sheets for AI, Neural Networks, Machine Learning, Deep Learning & Big Data2. Data Science Simplified Part 1: Principles and Process

3. Getting Started with Building Realtime API Infrastructure4. AI & NLP Workshop

Applications of AI

1. Gaming − AI plays important role for machine to think of large number of possible positions based on deep knowledge in strategic games. for example, chess, river crossing, N-queens' problems and etc.

2. Natural Language Processing − Interact with the computer that understands natural language spoken by humans.

3. Expert Systems − Machine or software provide explanation and advice to the users.

4. Vision Systems − Systems understand, explain, and describe visual input on the computer.

5. Speech Recognition − There are some AI based speech recognition systems have ability to hear and express as sentences and understand their meanings while a person talks to it. For example, Siri and Google assistant.

6. Handwriting Recognition − The handwriting recognition software reads the text written on paper and recognize the shapes of the letters and convert it into editable text.

7. Intelligent Robots − Robots are able to perform the instructions given by a human.

Major goals

1. Knowledge reasoning
2. Planning
3. Machine Learning
4. Natural Language Processing
5. Computer Vision
6. Robotics

IBM Watson

“Watson” is an IBM supercomputer that combines Artificial Intelligence (AI) and complex inquisitive programming for ideal execution as a “question answering” machine. The supercomputer is named for IBM’s founder, Thomas J. Watson.

IBM Watson is at the forefront of the new era of computing. At the point when IBM Watson made, IBM communicated that “more than 100 particular techniques are used to inspect perceive sources, find and make theories, find and score affirm, and combination and rank speculations.” recently, the Watson limits have been expanded and the way by which Watson works has been changed to abuse new sending models (Watson on IBM Cloud) and propelled machine learning capacities and upgraded hardware open to architects and authorities. It isn’t any longer completely a request answering figuring system arranged from Q&A joins yet can now ‘see’, ‘hear’, ‘read’, ‘talk’, ‘taste’, ‘translate’, ‘learn’ and ‘endorse’.

5.2.1 Machine Learning

Introduction

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision- making processes based on data inputs. Any technology user today has benefitted from machine learning. Facial recognition technology allows social media platforms to help users tag and share photos of friends. Optical character recognition (OCR) technology converts images of text into movable type. Recommendation engines, powered by machine learning, suggest what movies or television shows to watch next based on user preferences. Self-driving cars that rely on machine learningto navigate may soon be available to consumers. Machine learning is a continuously developing field. Because of this, there are some considerations to keep in mind as you work with machine learning methodologies, or analyze the impact of machine learning processes. In this tutorial, we'll look into the common machine learning methods of supervised and unsupervised learning, and common algorithmic approaches in machine learning, including thek-nearest neighbor algorithm, decision tree learning, and deep learning. We'll explore which programming languages are most used in machine learning, providing you with some of the positive and negative attributes of each. Additionally, we'll discuss biases that are perpetuated by machine learning algorithms, and consider what can be kept in mind to prevent these biases when building algorithms.

a. Machine Learning Methods

In machine learning, tasks are generally classified into broad categories. These categories are based on how learning is received or how feedback on the learning is given to the system developed.

Two of the most widely adopted machine learning methods are **supervised learning** which trains algorithms based on example input and output data that is labeled by humans, and **unsupervised learning** which provides the algorithm with no labeled data in order to allow it to find structure within its input data. Let’s explore these methods in more detail.

1. Supervised Learning

In supervised learning, the computer is provided with example inputs that are labeled with their desired outputs. The purpose of this method is for the algorithm to be able to “learn” by comparing its actual output with the “taught” outputs to find errors, and modify the model accordingly. Supervised learning therefore uses patterns to predict label values on additional unlabeled data. For example, with supervised learning, an algorithm may be fed data with images of sharks labeled as fish and images of oceans labeled as water. By being trained on this data, the supervised learning algorithm should be able to later identify unlabeled shark images as fish and unlabeled ocean images as water. A common use case of supervised learning is to use historical data to predict statistically likely future events. It may use historical stock market information to anticipate upcoming fluctuations, or be employed to filter out spam emails. In supervised learning, tagged photos ofdogs can be used as input data to classify untagged photos of dogs.

2. Unsupervised Learning

In unsupervised learning, data is unlabeled, so the learning algorithm is left to find commonalities among its input data. As unlabeled data are more abundant than labeled data, machine learning methods that facilitate unsupervised learning are particularly valuable. The goal of unsupervised learning may be as straightforward as discovering hidden patterns within a dataset, but it may also have a goal of feature learning, which allows the computational machine to automatically discover the representations that are needed to classify raw data. Unsupervised learning is commonly used for transactional data. You may have a large dataset of customers and their purchases, but as a human you will likely not be able to make sense of what similar attributes can be drawn from customer profiles and their types of purchases. With this data fed into an unsupervised learning algorithm, it may be determined that women of a certain age range who buy unscented soaps are likely to be pregnant, and therefore a marketing campaign related.

Without being told a “correct” answer, unsupervised learning methods can look at complex data that is more expansive and seemingly unrelated in order to organize it in potentially meaningful ways. Unsupervised learning is often used for anomaly detection including for fraudulent credit card purchases, and recommender systems that recommend what products to buy next. In unsupervised learning, untagged photos of dogs can be used as input data for the algorithm to find likenesses and classify dog photos together.

b. Approaches

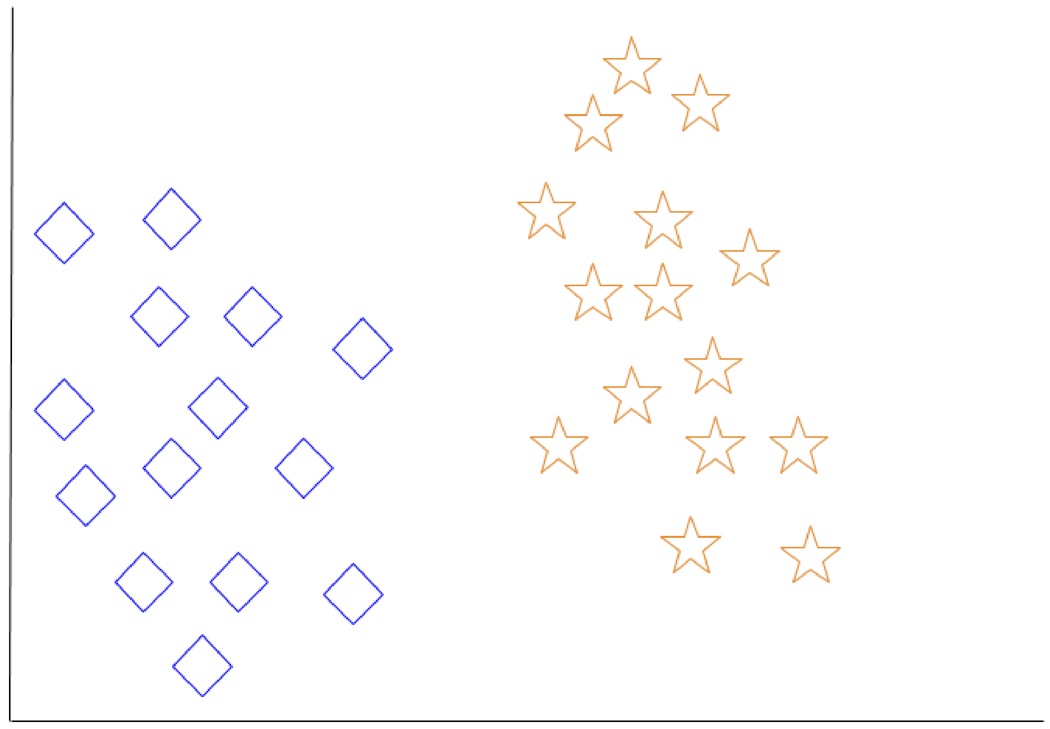
As a field, machine learning is closely related to computational statistics, so having a background knowledge in statistics is useful for understanding and leveraging machine learning algorithms. For those who may not have studied statistics, it can be helpful to first define correlation and regression, as they are commonly used techniques for investigating the relationship among quantitative variables. Correlation is a measure of association between two variables that are not designated as either dependent or independent. Regression at a basic level is used to examine the relationship between one dependent and one independent variable. Because regression statistics can be used to anticipate the dependent variable when the independent variable is known, regression enables prediction capabilities. Approaches to machine learning are continuously being developed. For our purposes, we’ll go through a few of the popular approaches that are being used in machine learning at the time of writing.

c. k-nearest neighbor

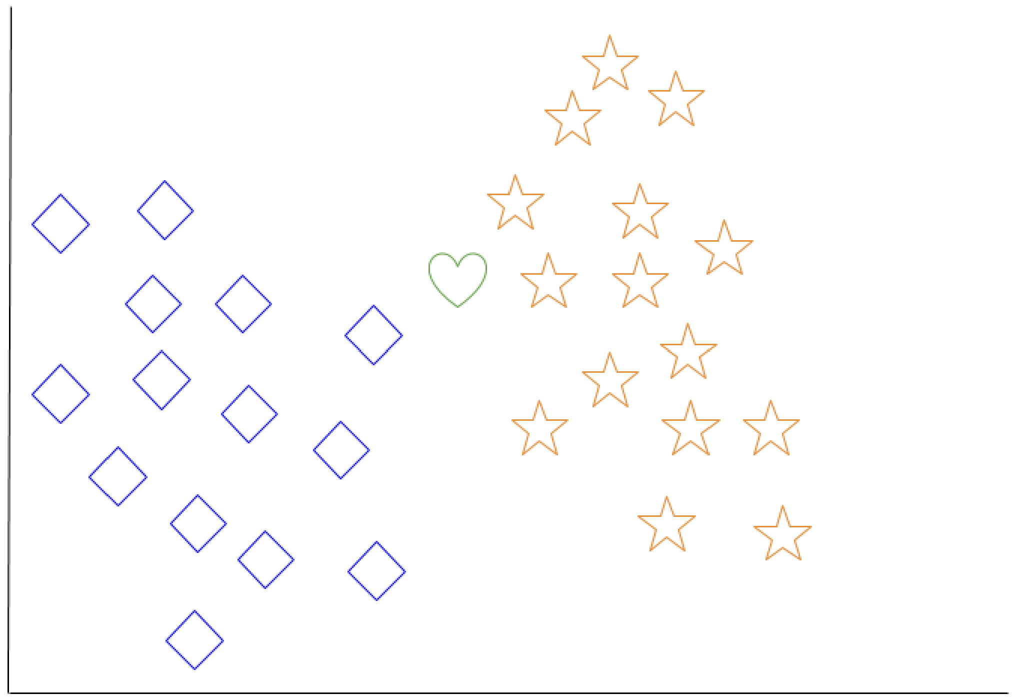
The k-nearest neighbor algorithm is a pattern recognition model that can be used for classification as well as regression. Often abbreviated as k-NN, the **k** in k-nearest neighbor is a positive integer, which is typically small. In either classification or regression, the input will consist of the k closest training examples within a space.

We will focus on k-NN classification. In this method, the output is class membership. This will assign a new object to the class most common among its k nearest neighbors. In the case of k= 1, the object is assigned to the class of the single nearest neighbor.

Let’s look at an example of k-nearest neighbor. In the diagram below, there are blue diamond objects and orange star objects. These belong to two separate classes: the diamond class and the star class.

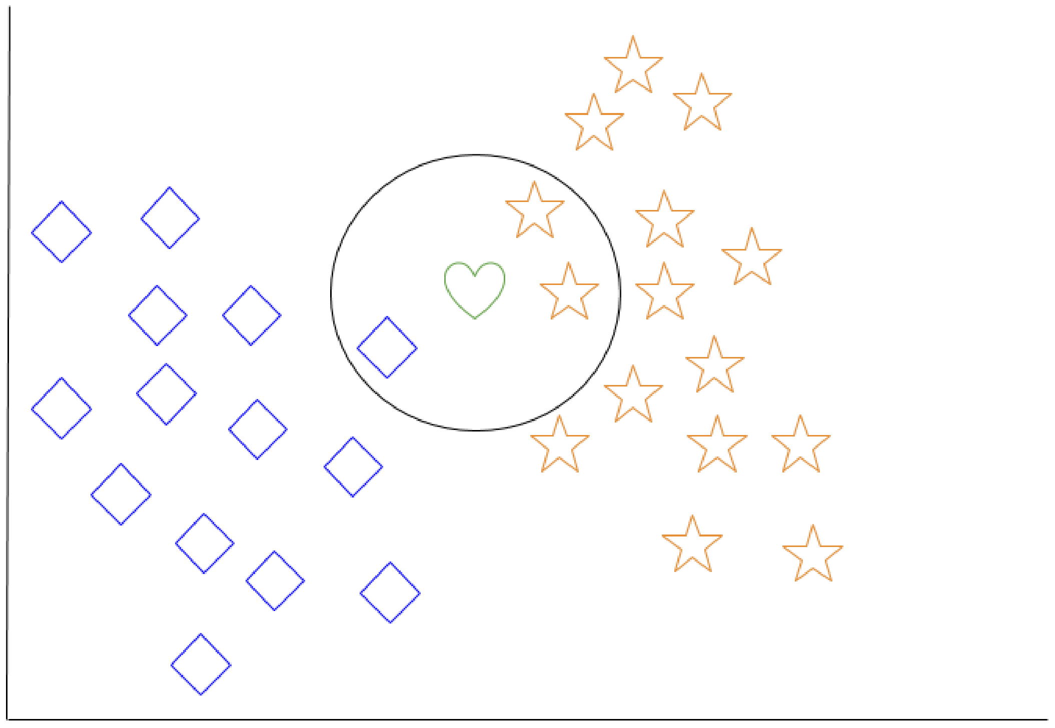


When a new object is added to the space — in this case a green heart — we will want the machine learning algorithm to classify the heart to a certain class.



When we choose k = 3, the algorithm will find the three nearest neighbors of the green heart in order to classify it to either the diamond class or the star class.

In our diagram, the three nearest neighbors of the green heart are one diamond and two stars. Therefore, the algorithm will classify the heart with the star class.



Among the most basic of machine learning algorithms, k-nearest neighbor is considered to be a type of “lazy learning” as generalization beyond the training data does not occur until a query is made to the system.

d. Decision Tree Learning

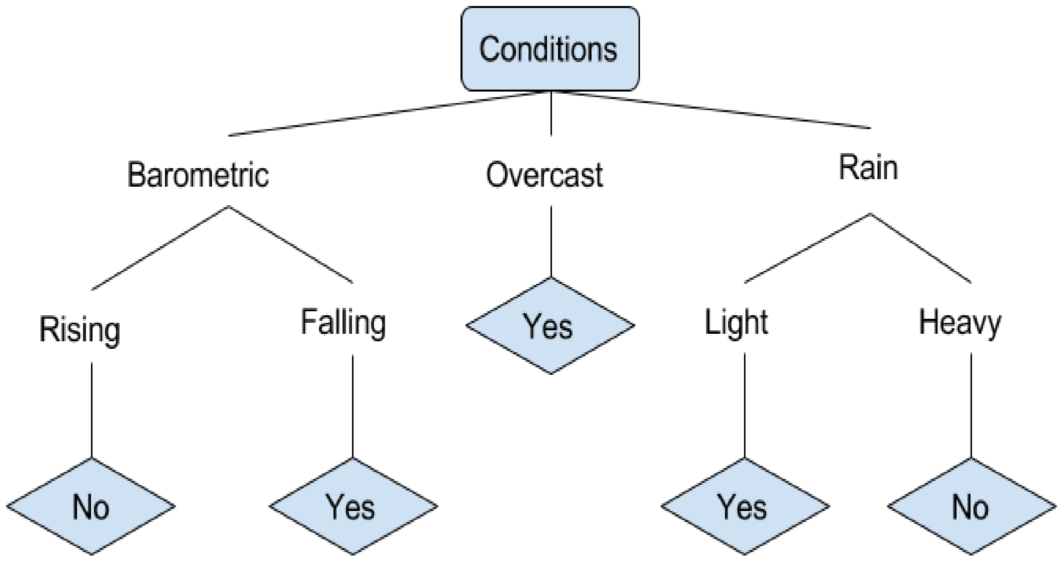
For general use, decision trees are employed to visually represent decisions and show or inform decision making. When working with machine learning and data mining, decision trees are used as a predictive model. These models map observations about data to conclusions about the data’s target value.

The goal of decision tree learning is to create a model that will predict the value of a target based on input variables.

In the predictive model, the data’s attributes that are determined through observation are represented by the branches, while the conclusions about the data’s target value are represented in the leaves.

When “learning” a tree, the source data is divided into subsets based on an attribute value test, which is repeated on each of the derived subsets recursively.

Let's look at an example of various conditions that can determine whether or not someone should go fishing. This includes weather conditions as well as barometric pressure conditions.



In the simplified decision tree above, an example is classified by sorting it through the tree to the appropriate leaf node. This then returns the classification associated with the particular leaf, which in this case is either a Yes or a No. The tree classifies a day's conditions based on whether or not it is suitable for going fishing.

A true classification tree data set would have a lot more features than what is outlined above, but relationships should be straightforward to determine. When working with decision tree learning, several determinations need to be made, including what features to choose, what conditions to use for splitting, and understanding when the decision tree has reached a clear ending.

5.2.3 Deep Learning

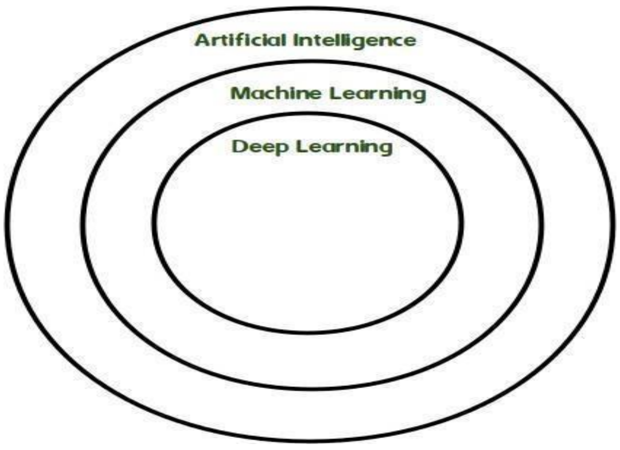
What is deep learning

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new.

As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture.

A formal definition of deep learning is- neurons

Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept definedin relation to simpler concepts, and more abstract representations computed in terms of less abstract ones. In human brain approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousands of their neighbors. The question here is how we recreate these neurons in a computer. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neuronsfor input value and some for-output value and in between, there may be lots of neurons interconnected in the hidden layer.



5.2.4 PYTHON

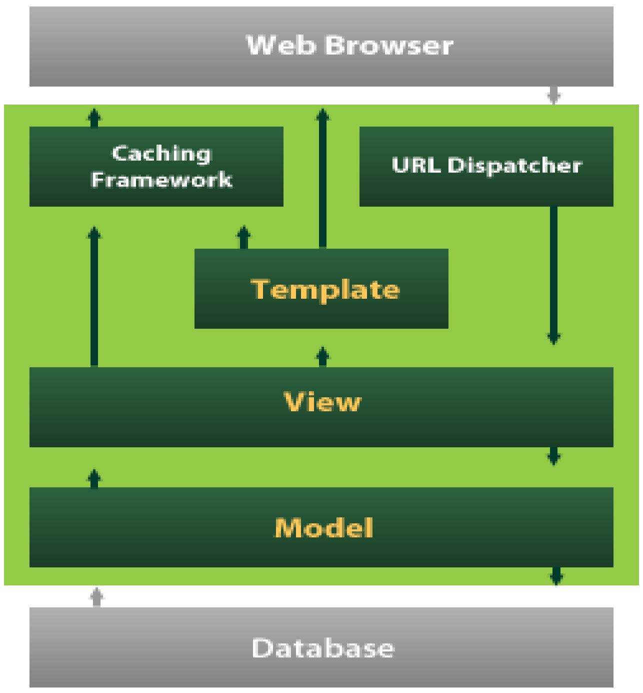
Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java.

It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. Python is managed by the non- profit Python Software Foundation. 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.

5.2.5 DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.



5.3 CODING

Settings.py

```
"""
Django settings for QuickShorts_NewsAggregator project. Generated by
'django-admin startproject' using Django 3.0.8. For more information on this
file, see https://docs.djangoproject.com/en/3.0/topics/settings/
For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/ """
import os
# Build paths inside the project like this: os.path.join(BASE_DIR, ...) BASE_DIR =
os.path.dirname(os.path.dirname(os.path.abspath( file ))) # Quick-start development
settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/ #
SECURITY
WARNING: keep the secret key used in production secret!
SECRET_KEY = 'uqc64g_-d=cnk_24w0g06is6a*_aj$vrqqfpi(tcjpyo7l8gff' #
SECURITY
WARNING: don't run with debug turned on in production! DEBUG = True
ALLOWED_HOSTS = []
# Application definition
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
Coding
Detection of Cyber Attack in Network using Machine Learning Techniques 43
'django.contrib.staticfiles', 'news',
]
MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'QuickShorts_NewsAggregator.urls' TEMPLATES
= [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates', 'DIRS': [],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [ 'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
]
WSGI_APPLICATION = 'QuickShorts_NewsAggregator.wsgi.application’
# Database

# https://docs.djangoproject.com/en/3.0/ref/settings/#databases DATABASES
= {
'default': {
'ENGINE': 'django.db.backends.sqlite3', 'NAME':
os.path.join(BASE_DIR, 'db.sqlite3'),
}
}
# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators
AUTH_PASSWORD_VALIDATORS = [
{
'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]
# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/ LANGUAGE_CODE =
'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True USE_L10N =
True USE_TZ = True
# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.0/howto/static-files/
STATIC_URL = '/static/'
views.py
from django.shortcuts import render import requests
from bs4 import BeautifulSoup
# Getting news from Times of India
toi_r = requests.get("https://timesofindia.indiatimes.com/briefs") toi_soup =
BeautifulSoup(toi_r.content, 'html5lib')
toi_headings = toi_soup.find_all('h2')
toi_headings = toi_headings[0:-13] # removing footers toi_news = []
for th in toi_headings:
toi_news.append(th.text)
#Getting news from Indian Express
ie_r = requests.get("https://indianexpress.com/article") ie_soup =
BeautifulSoup(ie_r.content, 'html5lib') ie_headings =
ie_soup.findAll('h3')
ie_headings = ie_headings[0:-2] ie_news = []
for ieh in ie_headings:
ie_news.append(ieh.text)
def index(req):
```

Coding

Detection of Cyber Attack in Network using Machine Learning Techniques 46

```
return render(req, 'news/index.html', {'toi_news':toi_news, 'ie_news': ie_news})
```

urls.py

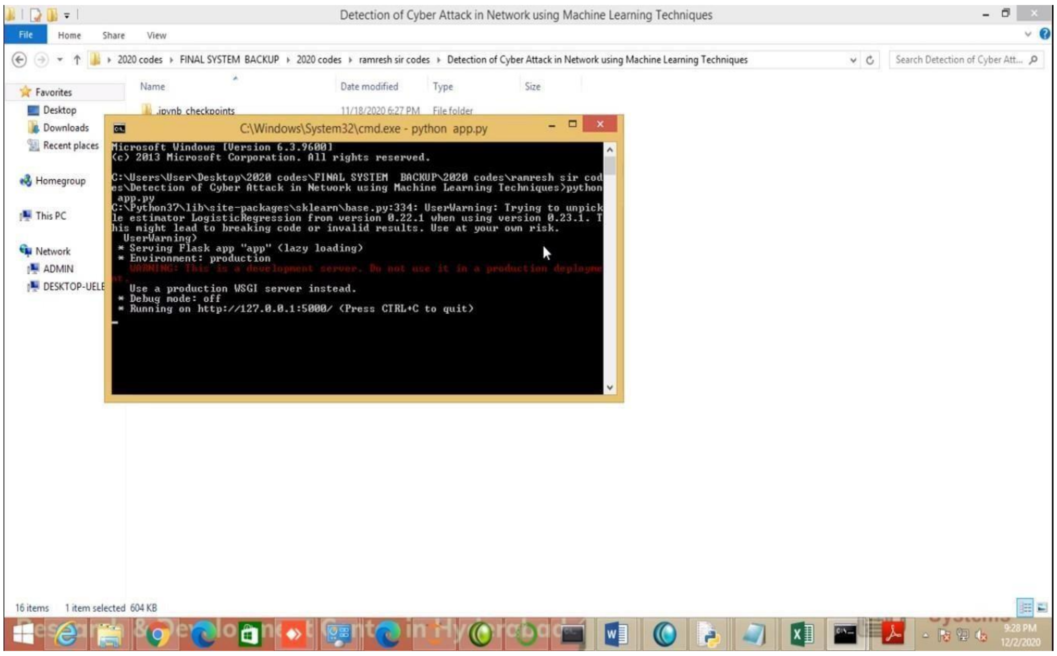
```
"""QuickShorts_NewsAggregator URL Configuration
The urlpatterns list routes URLs to views. For more information please see:
https://docs.djangoproject.com/en/3.0/topics/http/urls/
Examples Function views
* Add an import: from my_app import views
* Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
* Add an import: from other_app.views import Home

* Add a URL to urlpatterns: path('', Home.as_view(), name='home')
* Including another URLconf
* Import the include() function: from django.urls import include, path
* Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""
from django.contrib import admin from
django.urls import path from news import views
urlpatterns = [
path('admin/', admin.site.urls),
path('', views.index, name = "home"),
]
index.html
<!DOCTYPE html>
<html>
<head>
<title>QuickShorts News Aggregator</title><linkrel="stylesheet"
Coding
Detection of Cyber Attack in Network using Machine Learning Techniques 47
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAi
S6 JXm"
crossorigin="anonymous">
</head>
<body>
<div class="jumbotron">
<center><h1>QuickShorts News Aggregator</h1>
<a href="/" class="btn btn-danger">Refresh News</a>
</form>
</center>
</div>
<div class="container">

<div class="row">
<div class="col-6">
<h3 class="text-centre">News from Times of india</h3>
{% for n in toi_news %}
<h5> - {{n}} </h5>
<hr>
{% endfor %}
<br>
</div>
```

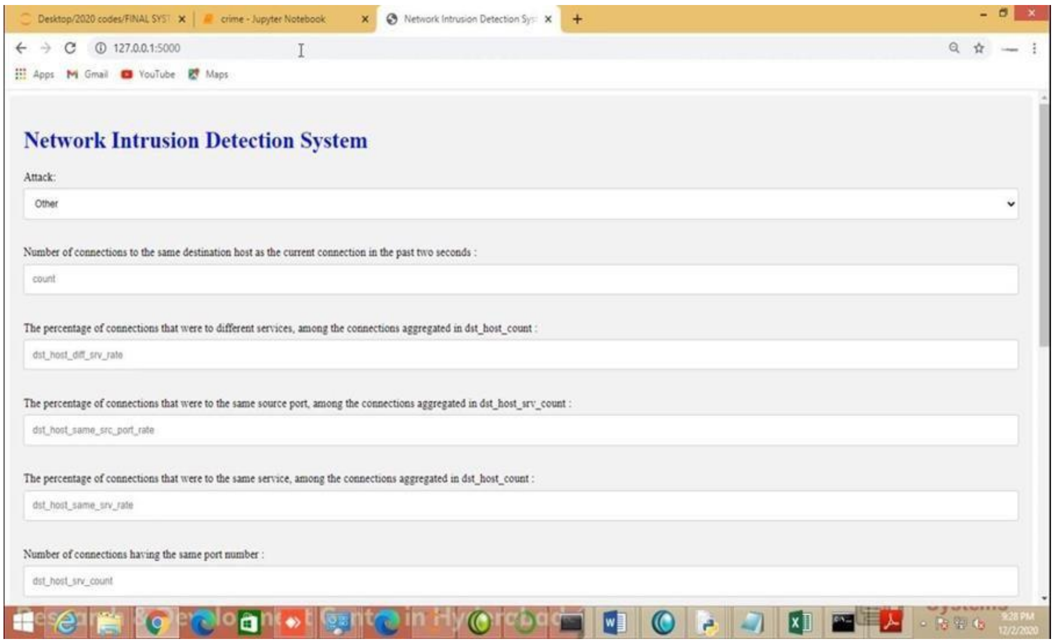
```
<div class="col-6">
<h3 class="text-centre">News from Indian Express</h3>
{% for ien in ie_news %}
<h5> - {{ien}} </h5>
<hr>
{% endfor %}<br>
</div>
</div>
</div>
<script
src="http://code.jquery.com/jquery-3.3.1.min.js"
integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0
b4 Q"
crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVC
mY l"
```

5.4 OUTPUT SCREENS



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\User\Desktop\2020 codes\FINAL SYSTEM  BACKUP\2020 codes\ramresh sir cod
es\Detection of Cyber Attack in Network using Machine Learning Techniques>python
 app.py
C:\Python37\lib\site-packages\sklearn\base.py:334: UserWarning: Trying to unpick
le estimator LogisticRegression from version 0.22.1 when using version 0.23.1. T
his might lead to breaking code or invalid results. Use at your own risk.
  UserWarning)
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



Network Intrusion Detection System

Attack:

Other

Number of connections to the same destination host as the current connection in the past two seconds :

count

The percentage of connections that were to different services, among the connections aggregated in dst_host_count :

dst_host_diff_srv_rate

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count :

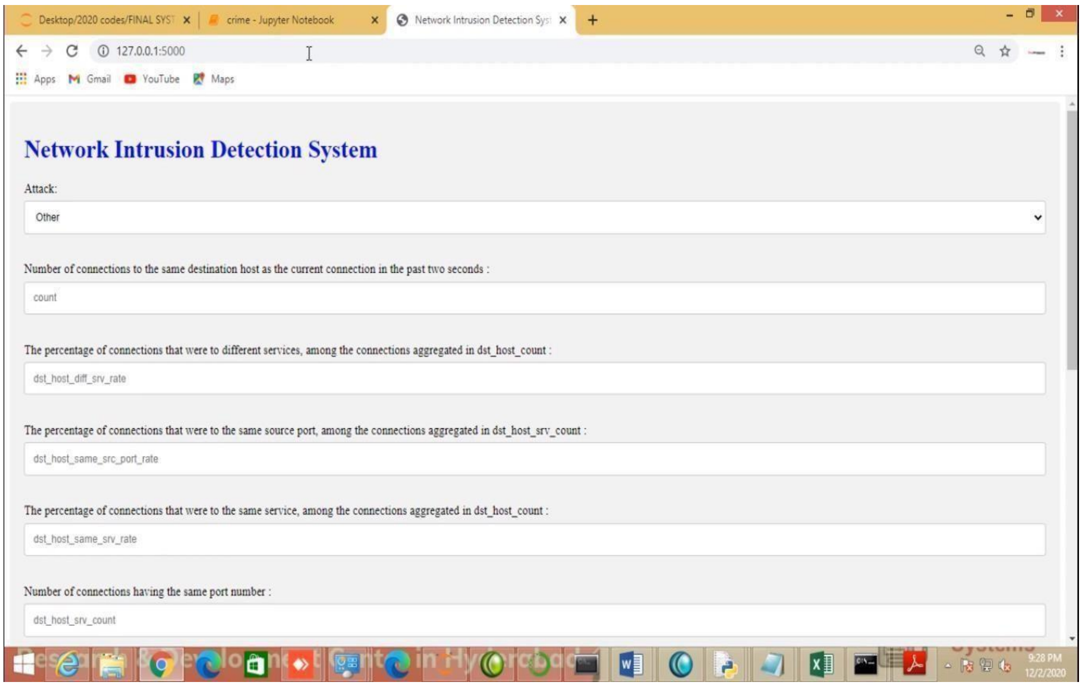
dst_host_same_src_port_rate

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

dst_host_same_srv_rate

Number of connections having the same port number :

dst_host_srv_count


Desktop/2020 codes/FINAL SYST
crime - Jupyter Notebook
Network Intrusion Detection Sys
127.0.0.1:5000
Apps
Gmail
YouTube
Maps

Network Intrusion Detection System

Attack:

Other

Number of connections to the same destination host as the current connection in the past two seconds :

count

The percentage of connections that were to different services, among the connections aggregated in dst_host_count :

dst_host_diff_srv_rate

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count :

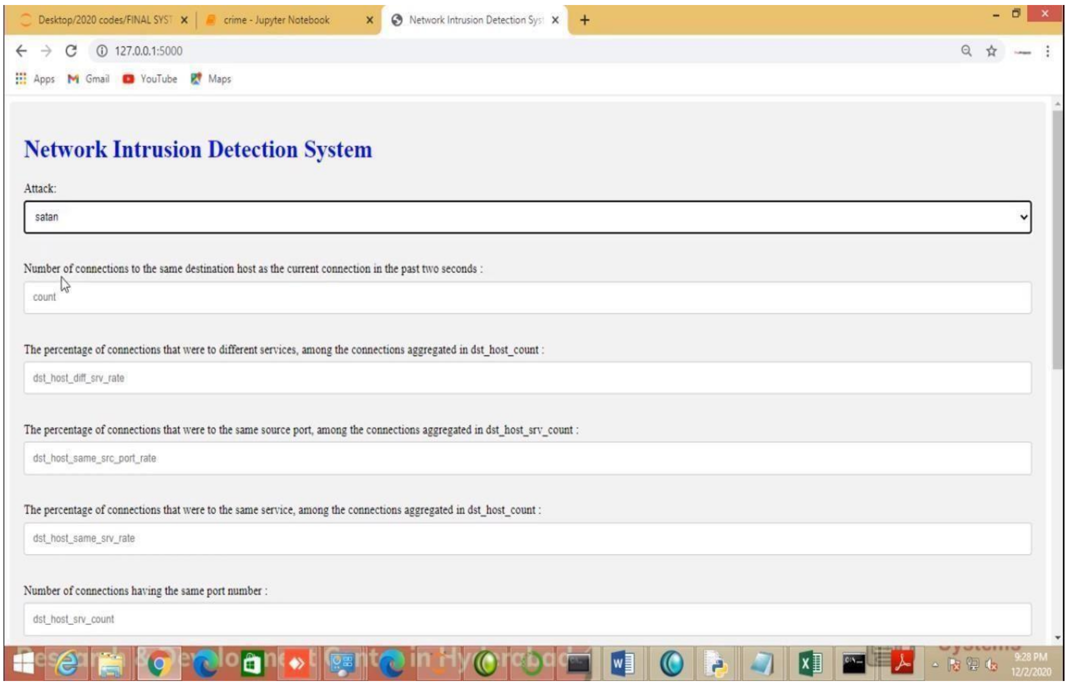
dst_host_same_src_port_rate

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

dst_host_same_srv_rate

Number of connections having the same port number :

dst_host_srv_count


Desktop/2020 codes/FINAL SYST
crime - Jupyter Notebook
Network Intrusion Detection Sys
127.0.0.1:5000
Apps
Gmail
YouTube
Maps

Network Intrusion Detection System

Attack:

satan

Number of connections to the same destination host as the current connection in the past two seconds :

count

The percentage of connections that were to different services, among the connections aggregated in dst_host_count :

dst_host_diff_srv_rate

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count :

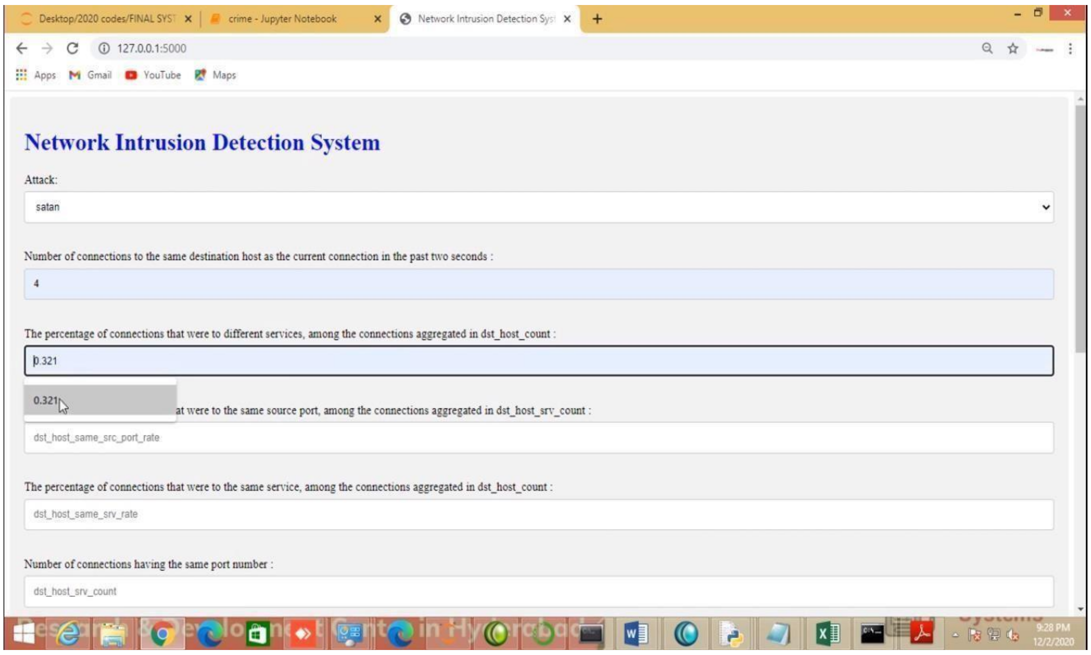
dst_host_same_src_port_rate

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

dst_host_same_srv_rate

Number of connections having the same port number :

dst_host_srv_count



Network Intrusion Detection System

Attack:

satan

Number of connections to the same destination host as the current connection in the past two seconds :

4

The percentage of connections that were to different services, among the connections aggregated in dst_host_count :

0.321

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count :

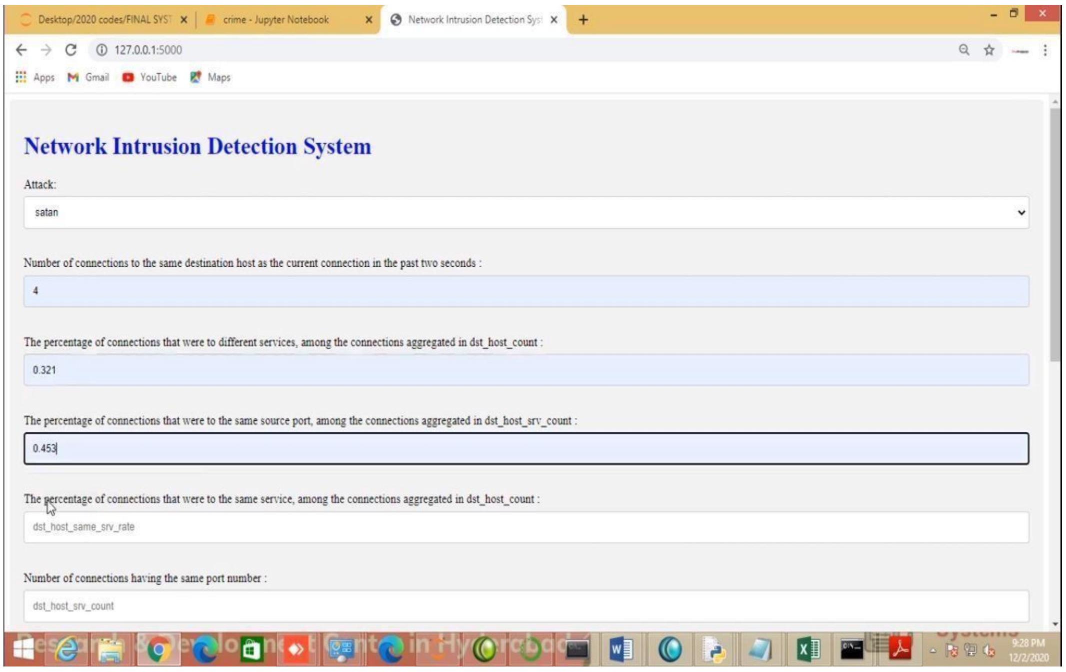
dst_host_same_src_port_rate

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

dst_host_same_srv_rate

Number of connections having the same port number :

dst_host_srv_count



Network Intrusion Detection System

Attack:

satan

Number of connections to the same destination host as the current connection in the past two seconds :

4

The percentage of connections that were to different services, among the connections aggregated in dst_host_count :

0.321

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count :

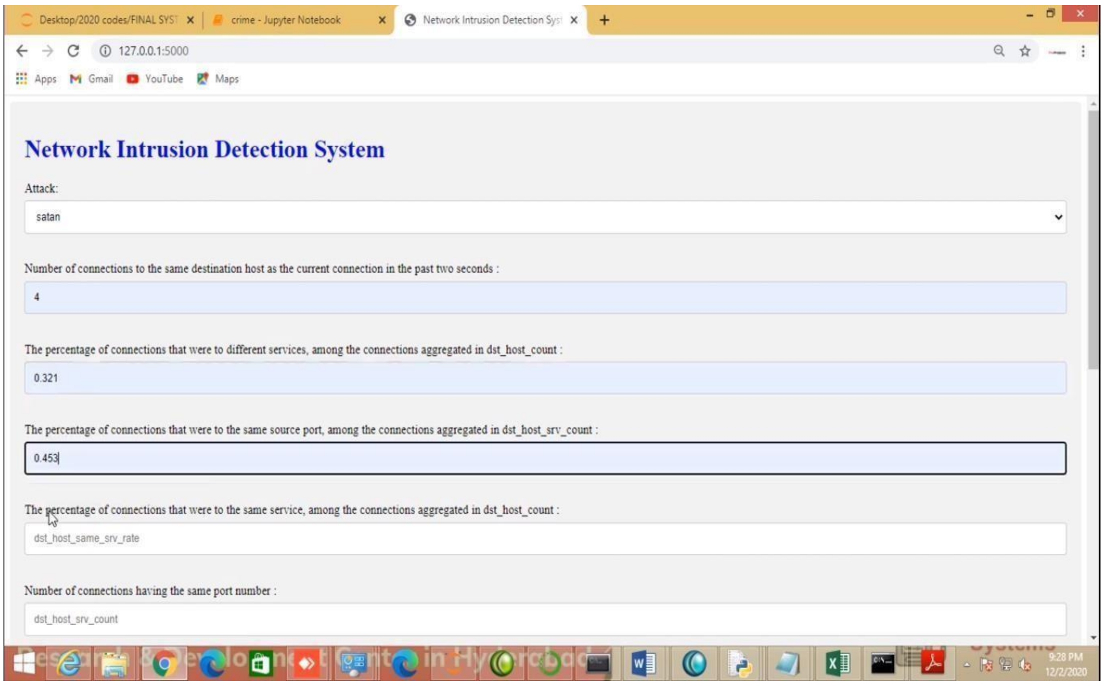
0.453

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

dst_host_same_srv_rate

Number of connections having the same port number :

dst_host_srv_count



Network Intrusion Detection System

Attack: satan

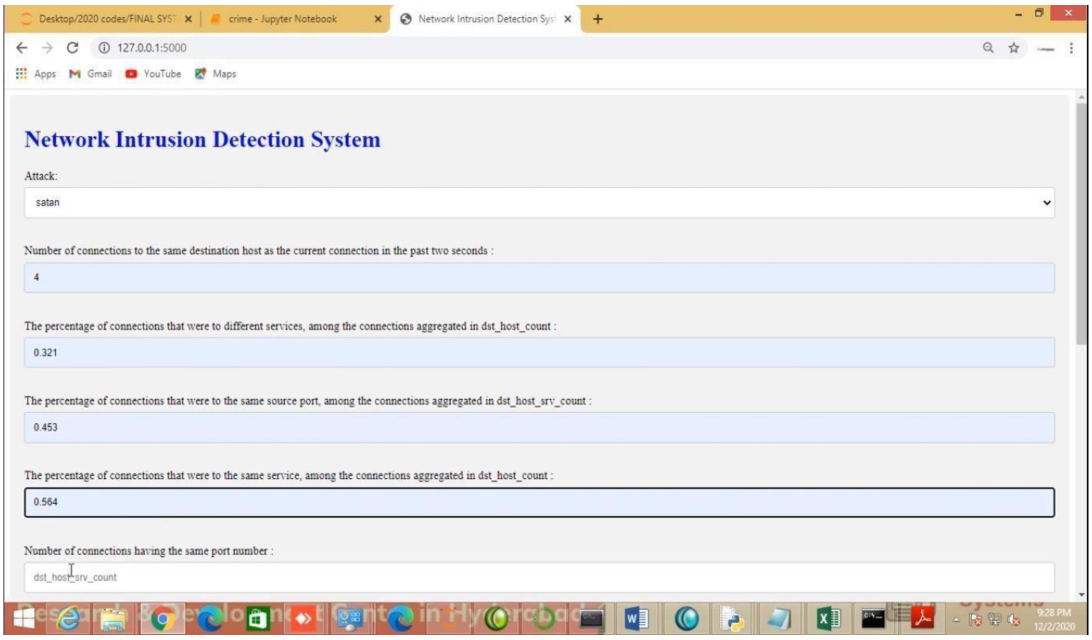
Number of connections to the same destination host as the current connection in the past two seconds : 4

The percentage of connections that were to different services, among the connections aggregated in dst_host_count : 0.321

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count : 0.453

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count : dst_host_same_srv_rate

Number of connections having the same port number : dst_host_srv_count



Network Intrusion Detection System

Attack: satan

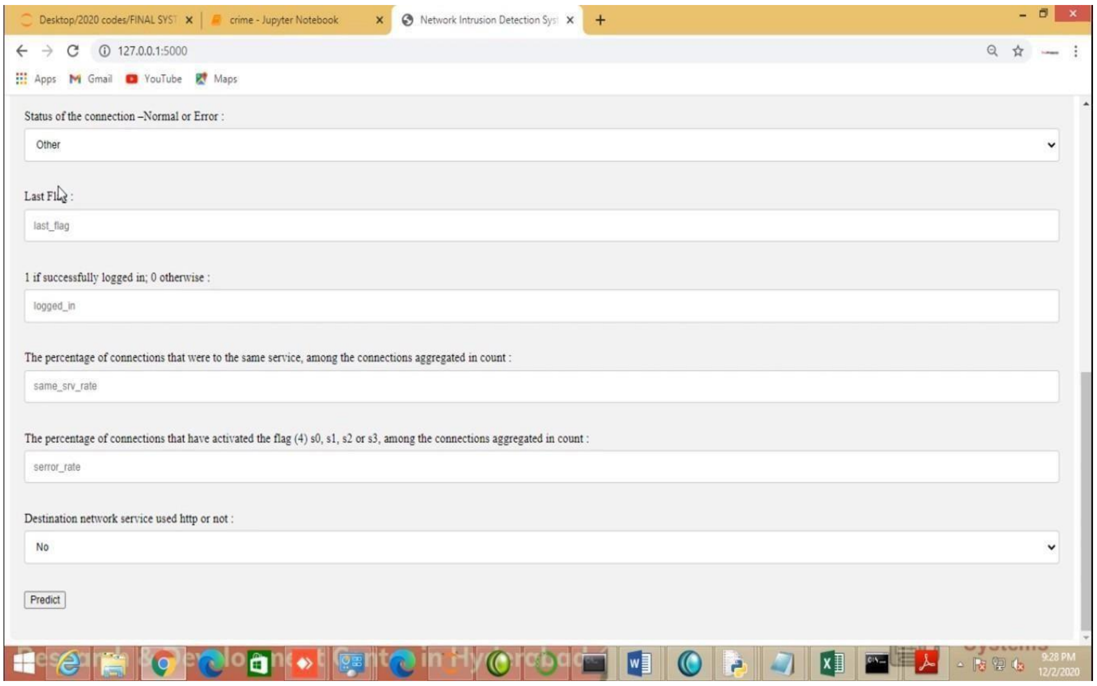
Number of connections to the same destination host as the current connection in the past two seconds : 4

The percentage of connections that were to different services, among the connections aggregated in dst_host_count : 0.321

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count : 0.453

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count : 0.564

Number of connections having the same port number : dst_host_srv_count


Desktop/2020 codes/FINAL SYST
crime - Jupyter Notebook
Network Intrusion Detection Syst
127.0.0.1:5000
Apps
Gmail
YouTube
Maps

Status of the connection –Normal or Error :

Other

Last Flag :

last_flag

1 if successfully logged in; 0 otherwise :

logged_in

The percentage of connections that were to the same service, among the connections aggregated in count :

same_srv_rate

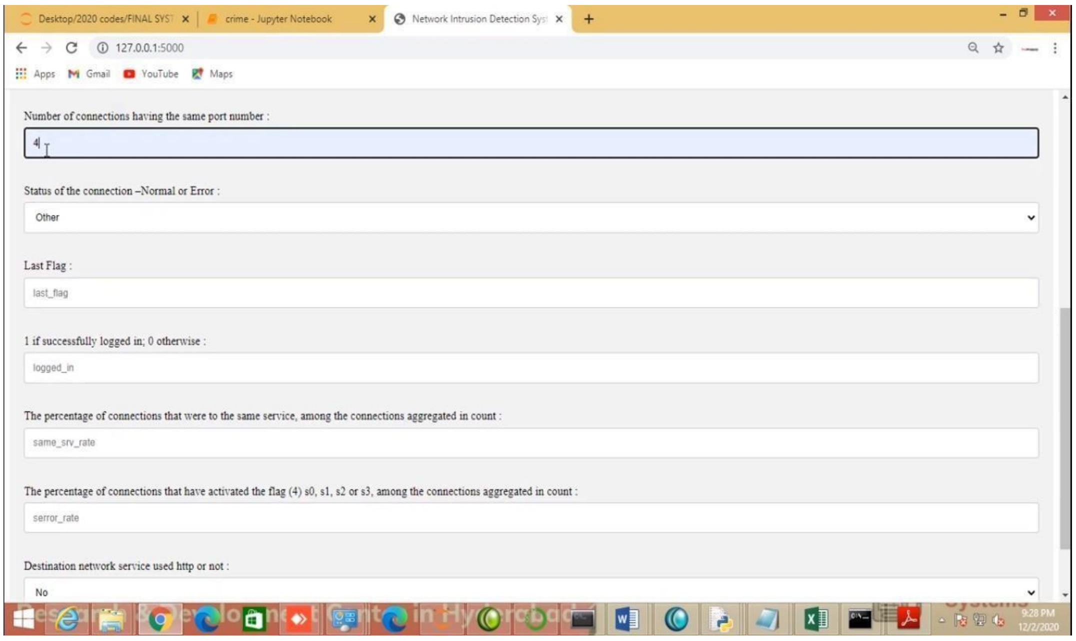
The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

serror_rate

Destination network service used http or not :

No

Predict


Desktop/2020 codes/FINAL SYST
crime - Jupyter Notebook
Network Intrusion Detection Syst
127.0.0.1:5000
Apps
Gmail
YouTube
Maps

Number of connections having the same port number :

4

Status of the connection –Normal or Error :

Other

Last Flag :

last_flag

1 if successfully logged in; 0 otherwise :

logged_in

The percentage of connections that were to the same service, among the connections aggregated in count :

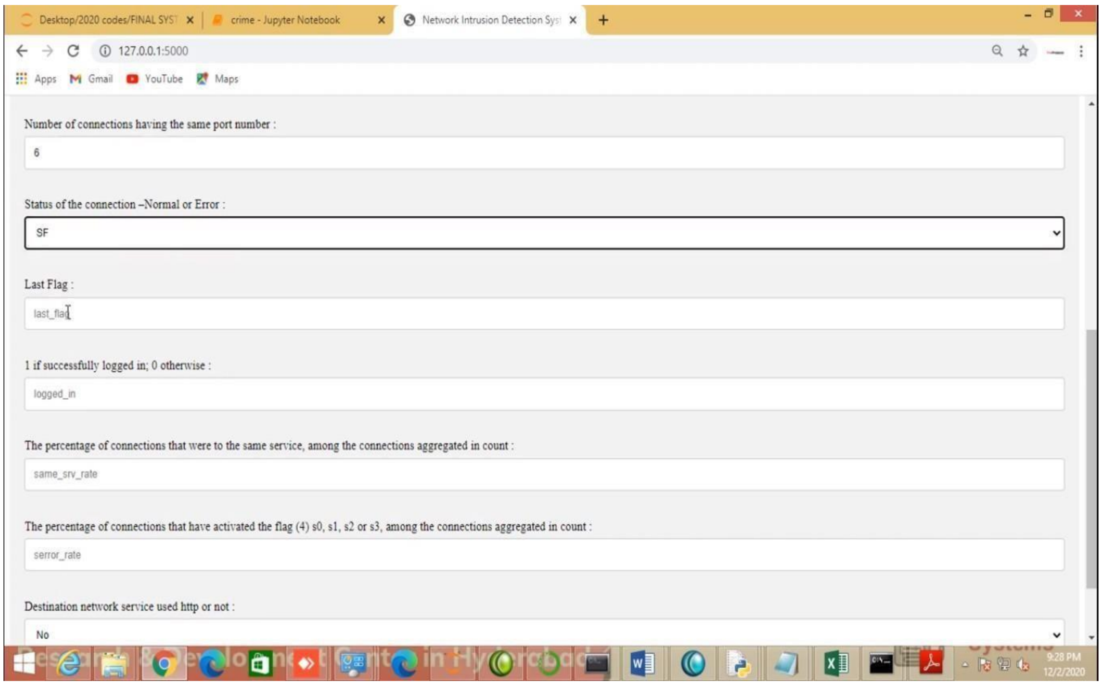
same_srv_rate

The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

serror_rate

Destination network service used http or not :

No



Number of connections having the same port number :

6

Status of the connection –Normal or Error :

SF

Last Flag :

last_flag

1 if successfully logged in; 0 otherwise :

logged_in

The percentage of connections that were to the same service, among the connections aggregated in count :

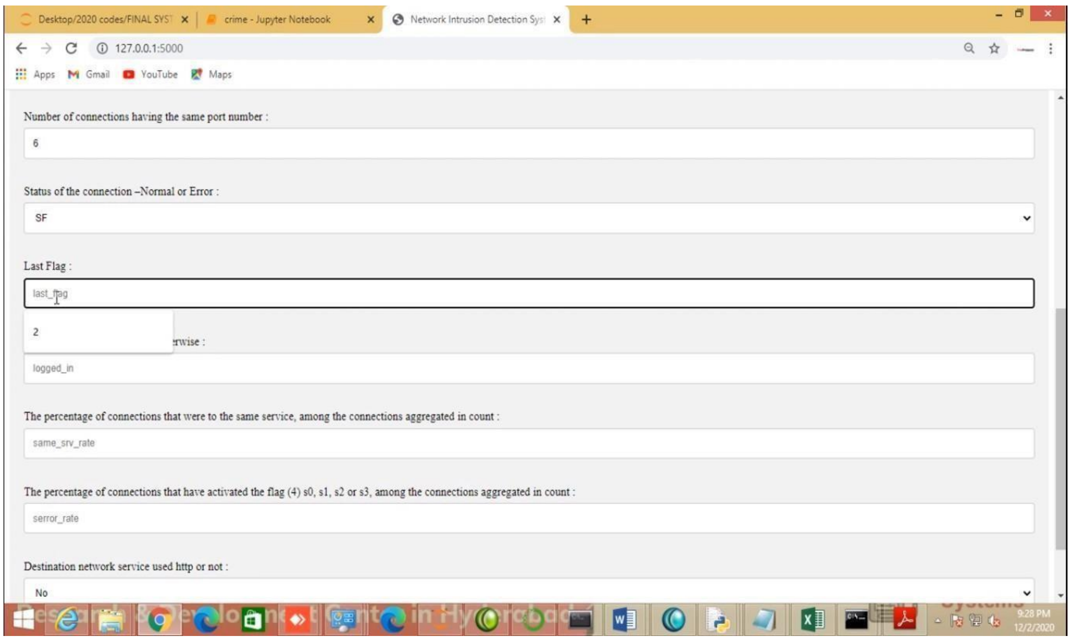
same_srv_rate

The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

serror_rate

Destination network service used http or not :

No



Number of connections having the same port number :

6

Status of the connection –Normal or Error :

SF

Last Flag :

last_flag

2

erwise :

logged_in

The percentage of connections that were to the same service, among the connections aggregated in count :

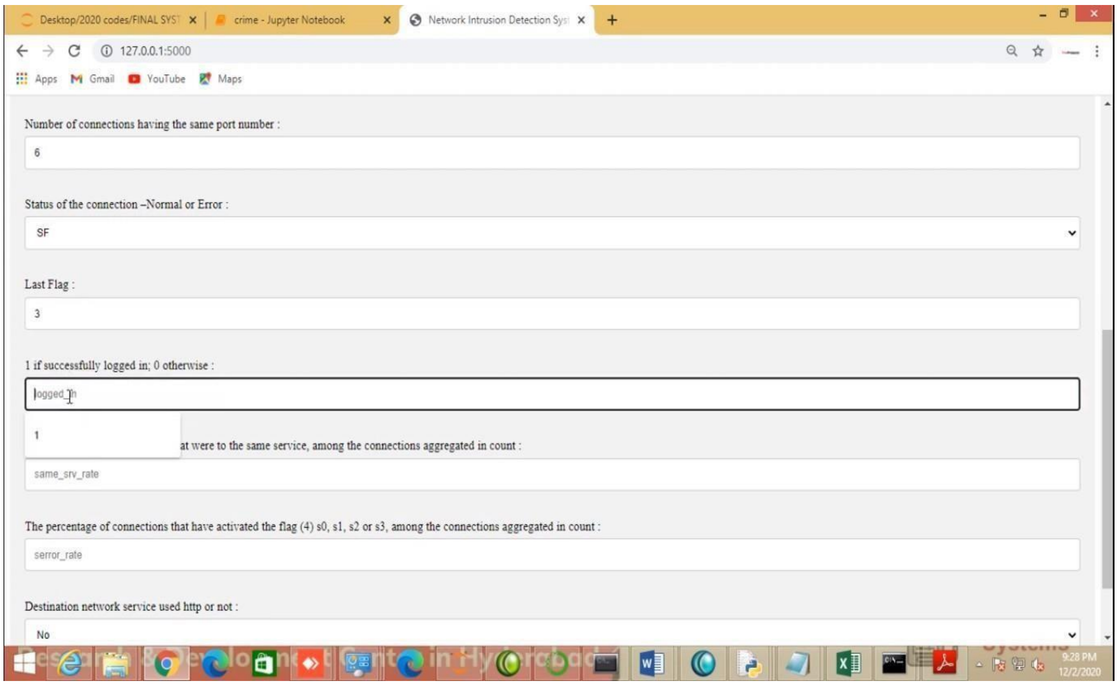
same_srv_rate

The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

serror_rate

Destination network service used http or not :

No



Number of connections having the same port number :

6

Status of the connection –Normal or Error :

SF

Last Flag :

3

1 if successfully logged in; 0 otherwise :

logged_in

1

at were to the same service, among the connections aggregated in count :

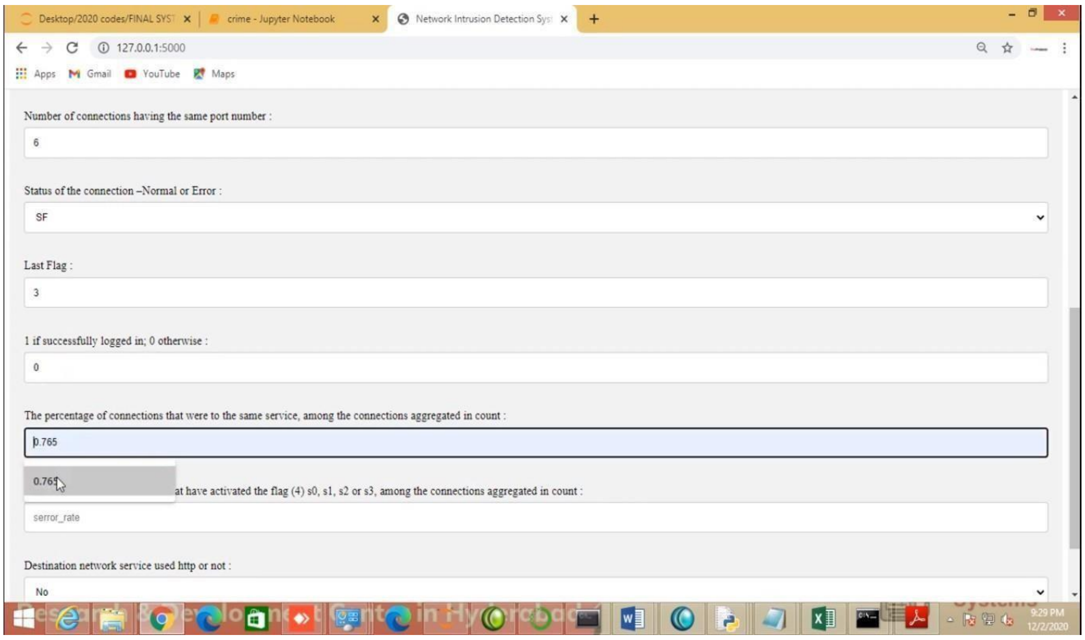
same_srv_rate

The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

serror_rate

Destination network service used http or not :

No



Number of connections having the same port number :

6

Status of the connection –Normal or Error :

SF

Last Flag :

3

1 if successfully logged in; 0 otherwise :

0

The percentage of connections that were to the same service, among the connections aggregated in count :

0.765

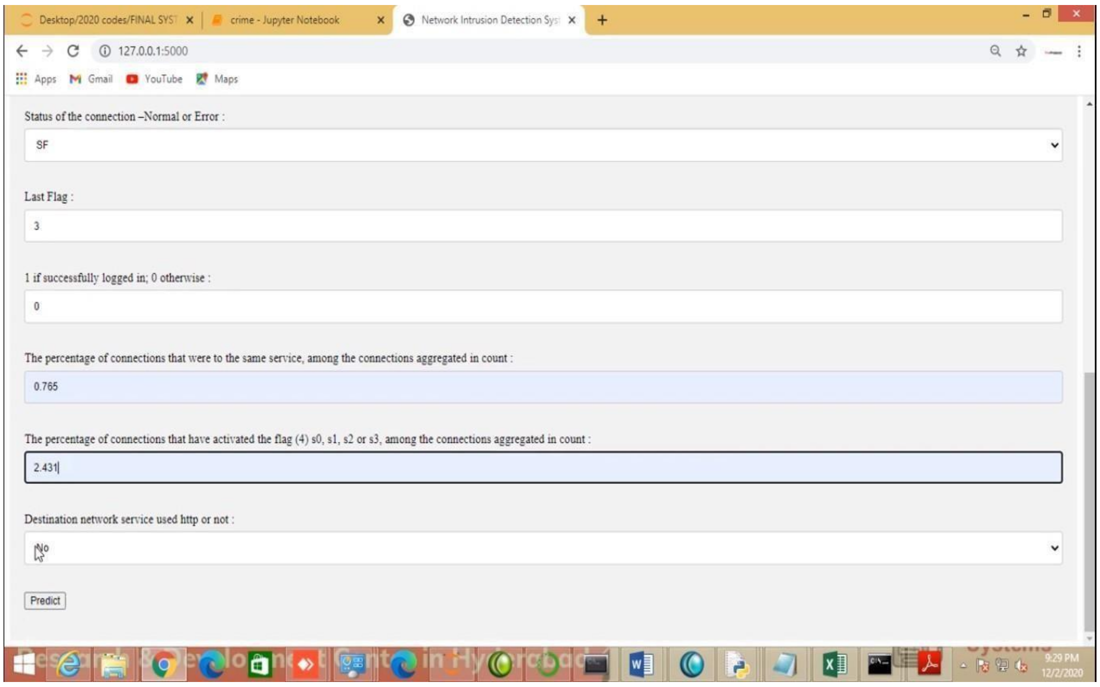
0.765

at have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

serror_rate

Destination network service used http or not :

No



127.0.0.1:5000

Status of the connection –Normal or Error :

SF

Last Flag :

3

1 if successfully logged in; 0 otherwise :

0

The percentage of connections that were to the same service, among the connections aggregated in count :

0.765

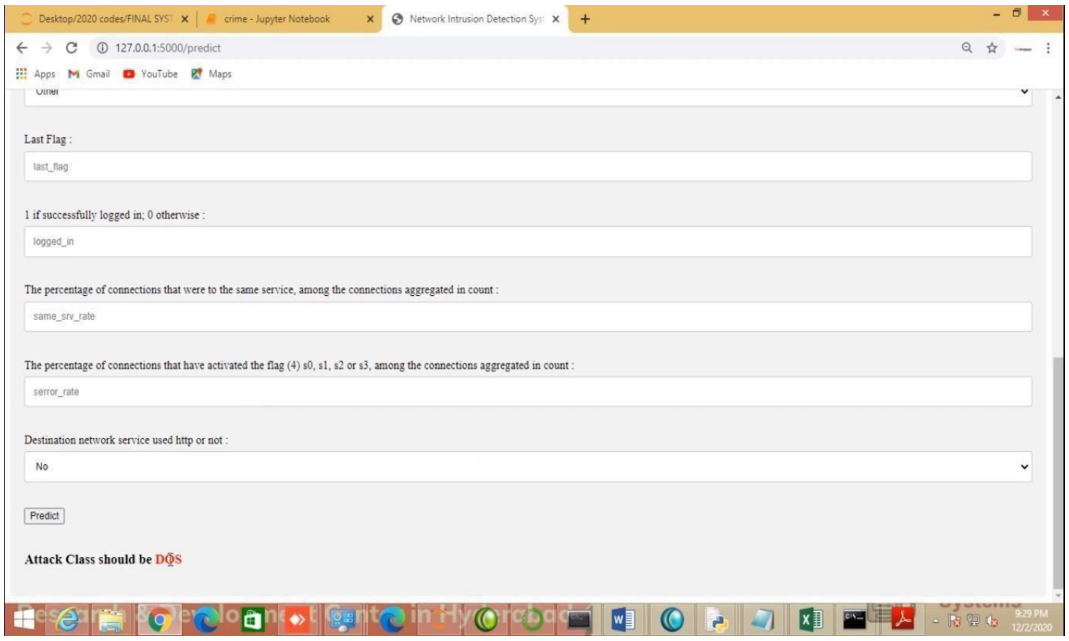
The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

2.431

Destination network service used http or not :

No

Predict



127.0.0.1:5000/predict

Last Flag :

last_flag

1 if successfully logged in; 0 otherwise :

logged_in

The percentage of connections that were to the same service, among the connections aggregated in count :

same_srv_rate

The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

serror_rate

Destination network service used http or not :

No

Predict

Attack Class should be DOS

CHAPTER-6

SYSTEM TESTING

6.1 SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.1.1 TYPES OF TESTS

a. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests performbasic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

b. INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

c. FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and usermanuals. Functional testing is centered on the following items:

a. Valid Input : identified classes of valid input must be accepted.

b. Invalid Input : identified classes of invalid input must be rejected.

c. Functions : identified functions must be exercised.

d. Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

d. SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

e. WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the innerworkings, structure and language of the software, or at least its purpose. It is purpose. It is usedto test areas that cannot be reached from a black box level.

f. BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

g. UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

h. TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

1. All field entries must work properly.
2. Pages must be activated from the identified link.
3. The entry screen, messages and responses must not be delayed.

Features to be tested

1. Verify that the entries are of the correct format
2. No duplicate entries should be allowed
3. All links should take the user to the correct page.

i. INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

j. ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functionalrequirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7. CONCLUSION

At the present time, assessments of help vector machine, ANN, CNN, Random Forest and significant learning estimations reliant upon current CICIDS2017 dataset were presented moderately. Results show that the significant learning estimation performed generally best results over SVM, ANN, RF and CNN. We will use port scope attempts just as other attack types with AI and significant learning computations, Apache Hadoop and shimmer advancements together ward on this dataset later on. Every one of these estimation assists us with recognizing the digital assault in network. It occurs in the manner that when we think about long back a long time there might be such countless assaults occurred so when these assaults are perceived then the highlights at which esteems these assaults are going on will be put away in some datasets. So, by utilizing these datasets we will anticipate if digital assault is finished. These forecasts should be possible by four calculations like SVM, ANN, RF, CNN this paper assists with distinguishing which calculation predicts the best precision rates which assists with foreseeing best outcomes to recognize the digital assaults occurred or not.

8. FUTURE SCOPE

Advanced Threat Detection: Evolving models to detect new and sophisticated cyber threats by analyzing patterns and anomalies. Real-time Detection and Response: Developing models capable of quickly identifying and responding to threats as they occur, minimizing damage. Adaptive and Self-learning Systems: Creating models that continuously learn from new data, improving detection accuracy over time. Integration with Security Ecosystems: Seamless integration with existing security tools like SIEM systems for comprehensive threat detection.

IoT Security: Extending detection capabilities to analyses data from Internet of Things (IoT) devices and prevent attacks in real-time. Automated Threat Response: Developing automated response systems that can mitigate threats without human intervention. Privacy-preserving Techniques: Incorporating methods like federated learning to protect sensitive data while enhancing threat detection capabilities.

9.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. Rashmi T V. “Predicting the System Failures Using Machine Learning Algorithms”. International Journal of Advanced Scientific Innovation, vol. 1, no. 1, Dec. 2020, doi:10.5281/zenodo.4641686.

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. Girish L, Rao SKN (2020) “Quantifying sensitivity and performance degradation of virtual machines using machine learning.”, Journal of Computational and Theoretical Nanoscience, Volume 17, Numbers 9-10, September/October 2020, pp. 4055-4060(6) https://doi.org/10.1166/jctn.2020.9019

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.

10. M. C. Raja and M. M. A. Rabbani, “Combined analysis of support vector machine and principal component analysis for ids,” in IEEE International Conference on Communication and Electronics Systems, 2016, pp. 1–5.

11. Nayana, Y., Justin Gopinath, and L. Girish. "DDoS Mitigation using Software Defined Network." International Journal of Engineering Trends and Technology (IJETT) 24.5 (2015): 258-264.

12. Shambulingappa H S. “Crude Oil Price Forecasting Using Machine Learning”. International Journal of Advanced Scientific Innovation, vol. 1, no. 1, Mar. 2021, doi:10.5281/zenodo.4641697.

13. D. Aksu, S. Ustebay, M. A. Aydin, and T. Atmaca, “Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm,” in International Symposium on Computer and Information Sciences. Springer, 2018, pp. 141– 149.