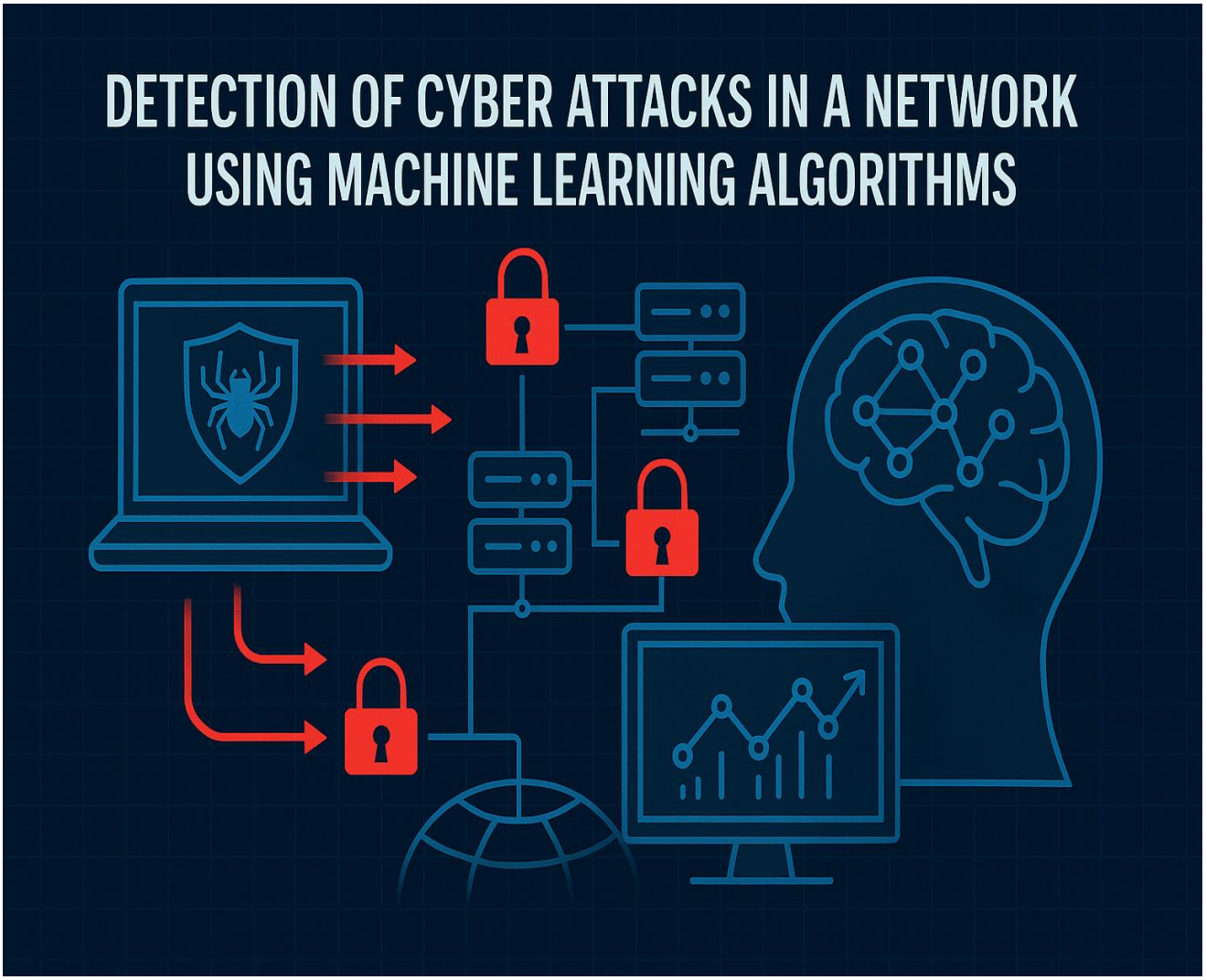
DETECTION OF CYBER ATTACKS IN A NETWORK USING MACHINE LEARNING ALGORITHMS



ABSTRACT

The world spending more time on the internet then ever before .As a result , the risk of cyberattacks and cybercrimes are increasing.The term ‘cyber threats’ is referred to as the illegal activity performed using the internet.Cybercriminals are changing their techniques with time to pass through the wall of protection.Conventional techniques are not capable of zero-day attacks and sophisticated attacks. Then , the machine learning techniques have been developed to detect cybercrimes and battle against the cyber threats.Three primary machine learning techniques are mainly investigated , including deep belief network, decision tree and support vector machine.

LIST OF CONTENTS

VII

LIST OF FIAGURES

CHAPTER -1

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 .

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 mechanisms 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 order 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 more risky. Information security is the route toward protecting information from unapproved will, use, openness, destruction, change or damage.

These domains are related to each other and have shared destinations to give availability, mystery, and genuineness of information.Studies show that the underlying advance of an attack is divulgence. Observation is made in order to get information about the structure at this moment. Finding a quick overview of open ports in a design gives unbelievably fundamental data to an assailant.

Therefore, there are loads of devices to perceive open ports [3], for example, subterranean insect infections and IDS. As of now, learning and SVM AI calculations were been applied to make IDS models to see port yield attempts the models were given the clarification of utilized material and strategies.

CHAPTER- 2

2. LITERATURE SURVEY

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 work found in writing utilized ANN with improved strong back spread for the plan of such an IDS . This work utilized just the preparation dataset for preparing (70%), approval (15%) and testing (15%). As expected, utilization of unlabelled information for testing brought about a reduction of execution. A later work utilized J48 choice tree classifier with 10 - overlay cross approval for testing on the preparation dataset .

This work utilized a decreased list of capabilities of 22 highlights rather than the full arrangement of 41 highlights. A comparable work assessed different well known regulated tree- based classifiers and tracked down that Random Tree model performed best with the most extensive level of exactness alongside a decreased bogus alert rate .

Numerous 2- level characterization approaches have likewise

been master presented. One such work utilized Discriminative Multinomial Naive Bayes (DMNB) as a base classifier and Nominal - to Binary directed separating at the second level alongside 10 - crease cross approval for testing .

This work was hide the reached out to utilize Ensembles of Balanced Nested Dichotomies (END) at the main level and Random Forest at the second level . 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.

The creators improved their work by utilizing data gain to rank the highlights and afterward a conduct based element determination to lessen the list of
capabilities to 20. This brought about an improvement in detailed precision utilizing the preparation dataset .

The subsequent class to take a gander at, utilized both the preparation and test dataset. An underlying endeavour in this classification utilized fluffy characterization with hereditary calculation and came about in a detection precision of 80%+ with a low bogus positive rate .

Another significant work utilized unaided grouping algorithms and tracked down that the exhibition utilizing just the preparation information was diminished radically when test information was likewise utilized . A comparative execution utilizing the k-point calculation brought about a marginally better recognition exactness and lower bogus positive rate, utilizing both preparing and test datasets .

Another less well known strategy, OPF (ideal way woods) which uses chart apportioning for include clas

sification, was found to show a high identification accuracy inside 33% of the time contrasted with SVM RBF technique.

CHAPTER -3

3. SYSTEM ANALYSIS

3.1 Existing System

Many challenges arise since malicious attacks are continually changing and are occurring in very large volumes requiring a scalable solution. There are different malware datasets available publicly for further research by cyber security community. However, no existing study has shown the detailed analysis of the performance of various machine learning algorithms on various publicly available datasets. Due to the dynamic nature of malware with continuously changing attacking methods, the malware datasets available publicly are to be updated systematically and benchmarked.

DISADVANTAGES OF EXISTING SYSTEM:

1. Malicious cyber-attacks pose serious security issues.
2. Data theft is the serious issue faced by cyber-attack.

3.2 PROPOSED SYSTEM

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

SOFTWARE REQUIREMENT SPECIFICATION

H/W System Configuration:-

Processor	- P–IV
RAM	- 2 GB (min)
Hard Disk	- 40 GB

Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA 21"

SOFTWARE REQUIREMENTS:

- **Operating system :** Windows 7 Ultimate or above.
- **Coding Language** **:** Python.
- **Front-End** **:** Python.
- **Back-End** **:** **Flask**

CHAPTER -4

4.SOFTWARE ENVIRONMENT

INTRODUCTION TO DJANGO

Django is a Web Application Framework which is used to develop web applications. Our Django Tutorial includes all topics of Django such as introduction, features, installation, environment setup, admin interface, cookie, form validation, Model, Template Engine, Migration, MVT etc. All the topics are explained in detail so that reader can get enough knowledge of Django. Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

This framework uses a famous tag line: **The web framework for perfectionists with deadlines.**

By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.

Django was design and developed by Lawrence journal world in 2003 and publicly released under BSD license in July 2005. Currently, DSF (Django Software Foundation) maintains its development and release cycle.

Django was released on 21, July 2005. Its current stable version is 2.0.3 which was released on 6 March, 2018.

Django is widely accepted and used by various well-known sites such as:

- Instagram
- Mozilla
- Disqus

Features of Django

- Rapid Development
- Secure
- Scalable
- Fully loaded
- Versatile
- Open Source
- Vast and Supported Community

Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.

Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.

Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.

Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.

Django is versatile in nature which allows it to build applications for different-different domains. Now a days, Companies are using Django to build various types of applications like: content

management systems, social networks sites or scientific computing platforms etc.

The model layer

Django provides an abstraction layer (the “models”) for structuring and manipulating the data of your web application. Learn more about it below:

- **Models:** Introduction to models | Field types | Indexes | Meta options | Model class
- **QuerySets:** Making queries | QuerySet method reference | Lookup expressions
- **Model instances:** Instance methods | Accessing related objects
- **Migrations:** Introduction to Migrations | Op erations reference | SchemaEditor | Writing migrations
- **Advanced:** Managers | Raw SQL | Transactions | Aggregation | Search | Custom fields | Multiple databases | Custom lookups | Query Expressions | Conditional Expressions | Database Functions
- **Other:** Supported databases | Legacy databases | Providing initial data | Optimize database access | PostgreSQL specific features

The view layer

Django has the concept of “views” to encapsulate the logic responsible for processing a user’s request and for returning the response. Find all you need to know about views via the links below:

- **The basics:** URLconfs | View functions | Shortcuts | Decorators | Asynchronous Support

- **Reference:** Built-in Views | Request/response objects | TemplateResponse objects
- **File uploads:** Overview | File objects | Storage API | Managing files | Custom storage
- **Class-based views:** Overview | Built-in display views | Built-in editing views | Using mixins | API reference | Flattened index

- **Advanced:** Generating CSV | Generating PDF

- **Middleware:** Overview | Built-in middleware classes The

template layer

The template layer provides a designer-friendly syntax for rendering the information to be

presented to the user. Learn how this syntax can be used by designers and how it can be extended by programmers:

- **The basics:** Overview
- **For designers:** Language overview | Built-in tags and filters | Humanization
- **For programmers:** Template API | Custom tags and filters | Custom template backend

Forms

Django provides a rich framework to facilitate the creation of forms and the manipulation of form data.

- **The basics:** Overview | Form API | Built-in fields | Built-in widgets
- **Advanced:** Forms for models | Integrating media | Formsets | Customizing validation

The development process

Learn about the various components and tools to help you in the development and testing of Django applications:

- **Settings:** Overview | Full list of settings
- **Applications:** Overview

- **Exceptions:** Overview
- **django-admin and manage.py:** Overview | Adding custom commands

- **Deployment:** Overview | WSGI servers | ASGI servers | Deploying static files | Tracking code errors by email | Deployment checklist

The admin

Find all you need to know about the automated admin interface, one of Django's most popular features:

- Admin site
- Admin actions
- Admin documentation generator

Security:

Security is a topic of paramount importance in the development of web applications and Django provides multiple protection tools and mechanisms:

- Security overview
- Disclosed security issues in Django
- Clickjacking protection
- Cross Site Request Forgery protection
- Cryptographic signing
- Security Middleware Performance and optimization

There are a variety of techniques and tools that can help get your code running more
efficiently - faster, and using fewer system resources.

Other core functionalities

Learn about some other core functionalities of the Django framework:

- Flatpages
- Redirects
- Signals
- System check framework
- The sites framework
- Unicode in Django

VIEW:

Django Views are one of the vital participants of MVT Structure of Django. As per Django Documentation, A view function is a Python function that takes a Web request and returns a Web response. This **response** can be the HTML contents of a Web page, or a redirect, or a 404 error, or an XML document, or an image, anything that a web browser can display.

Django views are part of the user interface — they usually render the HTML/CSS/Javascript in your Template files into what you see in your browser when you render a web page. (Note that if you've used other frameworks based on the MVC (Model-View-Controller), do not get confused between Django views and views in the MVC paradigm. Django views roughly correspond to controllers in MVC, and Django templates to views in MVC.)

USER

The Django authentication system handles both authentication and authorization. Briefly, authentication verifies a user is who they claim to be, and authorization determines what an authenticated user is allowed to do. Here the term authentication is used to refer to both tasks.

- Users
- Permissions: Binary (yes/no) flags designating whether a user may perform a certain task.
- Groups: A generic way of applying labels and permissions to more than one user.
- A configurable password hashing system
- Forms and view tools for logging in users, or restricting content
- A pluggable backend system

The authentication system in Django aims to be very generic and doesn't provide some features commonly found in web authentication systems.

Solutions for some of these common problems have been implemented in third-party packages:

- Password strength checking
- Throttling of login attempts
- Authentication against third-parties (OAuth, for example)
- Object-level permissions

Installation¶

Authentication support is bundled as a Django contrib module in **django.contrib.auth**. By default, the required configuration is already included in the **settings.py** generated by **django-admin startproject**, these consist of two items listed in your **INSTALLED_APPS** setting:

1. **'django.contrib.auth'** contains the core of the authentication framework, and its default models.
2. **'django.contrib.contenttypes'** is the Django content type system, which allows permissions to be associated with models you create.

and these items in your **MIDDLEWARE** setting:

1. **SessionMiddleware** manages sessions across requests.
2. **AuthenticationMiddleware** associates users with requests using sessions.

With these settings in place, running the command **manage.py migrate** creates the necessary database tables for auth related models and permissions for any models defined in your installed apps.

URL:

The **route** argument should be a string or **gettext_lazy()** (see Translating URL patterns) that contains a URL pattern. The string may contain angle brackets (like **<username>** above) to capture part of the URL and send it as a keyword argument to the view. The angle brackets may include a converter specification (like the **int** part of **<int:section>**) which limits the characters matched and may also change the type of the variable passed to the view. For example, **<int:section>** matches a string of decimal digits and converts the value to an **int**. See How Django processes a request for more details.

The **view** argument is a view function or the result of **as_view()** for class-based views. It can also be an **django.urls.include()**.

Machine learning Introduction:

The objective of this briefing is to present an overview of the machine learning techniques currently in use or in consideration at statistical agencies worldwide. Section I, outlines the main reason why statistical agencies should start exploring the use of machine learning techniques. Section II outlines what machine learning is, by comparing a well- known statistical technique (logistic regression)

with a (non-statistical) machine learning counterpart (support vector machine).

Sections III, IV, and V discuss current research or applications of machine learning techniques within the field of official statistics in the areas of automatic coding, editing and imputation, and record linkage, respectively. The material presented in this project is the result of a literature review, of direct contacts with authors during conferences. Statistical Data Editing, and members of the Modernization Committee on Production and Methods.

Section VI contains a list of machine learning applications in official statistics outside of the three areas mentioned above

Machine Learning means In the statistical context, Machine Learning is defined as an application of artificial intelligence where available information is used through algorithms to process or assist the processing of statistical data. While Machine Learning involves concepts of automation, it requires human guidance. Machine Learning involves a high level of generalization in order to get a system that performs well on yet unseen data instances. Why should statistical agencies consider machine learning?

Machine learning is a relatively new discipline within Computer Science that provides a collection of data analysis techniques. Some of these techniques are based on well established statistical methods (e.g. logistic regression and principal component analysis) while many others are not. Most statistical techniques follow the paradigm of determining a particular probabilistic model that best describes observed data among a class of related models. Similarly, most machine learning techniques are designed to find models that best fit data (i.e. they solve certain optimization problems), except that these machine learning models are no longer restricted to probabilistic ones. Therefore, an advantage of machine learning techniques over statistical ones is that the latter require underlying probabilistic models while the former do not.

Even though some machine learning techniques use probabilistic models, the classical statistical techniques are most often too stringent for the oncoming Big Data era, because data sources are increasingly complex and multi-faceted. Prescribing probabilistic models relating variables from disparate data sources that are plausible and amenable to statistical analysis might be extremely difficult if not impossible. Machine learning might be able to provide a broader class of more flexible alternative analysis methods better

suited to modern sources of data. It is imperative for statistical agencies to explore the possible use of machine learning techniques to determine whether their future needs might be better met with such techniques than with traditional ones.

In a poster session at the Statistics Canada's 2014 International Methodology Symposium, Bethmann et al. (of Intitut für Arbeitsmarkt-und Berufsforschung) have reported on research on applying two types of probabilistic supervised machine learning algorithms -- Naïve Bayes (NB) and conjugate Bayesian analysis based on multinomial distributions (BMN) -- for automatic occupation coding for German panel surveys. The authors used a large volume (approximately 300,000) of manually coded occupation text strings from recent surveys as training data. The rate of agreement between automatic coding and manual

coding was used as a metric to evaluate the algorithms. Although both methods exhibited good agreement rates by common machine learning standards, the authors cautioned that they might not be sufficiently satisfactory given the considerably higher accuracy requirements of occupation coding in production settings (the authors suggested a minimum agreement rate of 95%). On the other hand, the authors pointed out that when the target variable was changed to "social-economic status" or "occupational prestige" (more precisely, ISEI-08 and SIOPS-08 scores, both derived from occupation codes), both methods yielded dramatically improved results. The authors concluded that the current versions of their methods may be sufficient for production of socio-economic status or occupational prestige predictions, but further improvements are required for production of reliable occupation coding.

Possibilities for improvement include the addition of a preprocessing step (to "clean up" input text strings, thereby reduce noise in training data), incorporation of a certain distance measure in the existing models, as well as different machine learning methods altogether (such as random forests or support vector machines). The authors project that their methods will be ready for release as an open-source R package in several years.

2. Automatic occupation coding via CASCOT (United Kingdom).

Computed-assisted Structured Coding Tool is an automatic occupation coding software tool developed by the Institute for Employment Research at the University of Warwick, a partner in the EurOccupations project. The objective of the project is to construct a

publicly available database of the most frequent occupations to facilitate multi-country data collection. Since 2009, CASCOT has been able to perform automated coding into the ISCO'08 classification of occupational texts in any of the seven languages of the eight EurOccupations partner countries.

CASCOT is available for online use for free and a desktop version is available for purchase should high-volume processing be required. However, CASCOT's underlying methodology has not been published.

3. Automatic coding via open-source indexing utility (Ireland)

The Central Statistics Office of Ireland has reported they are developing an automatic coding system for Classification of Individual Consumption by Purpose (COICOP)

assignment for their Household Budget Survey, using previously coded records as training data. Their method is based on the open-source indexing and searching tool Apache Lucene (http://lucene.apache.org).

4. Automatic coding of census variables via Support Vector Machines (New Zealand) Statistics New Zealand investigated the potential of using Support Vector Machines (SVM) to improve coding of item responses in their Census. They applied SVM to code the variables Occupation and Post-school Qualification, using two disjoint sets of observations, each of size 10,000, from Census 2013 data for training and testing. They reported 50% correctness rate on testing data for both variables, and concluded that further investigations would be necessary to further evaluate SVM as an automatic coding methodology.

About python:

The Python language has a substantial body of documentation, much of it contributed by various authors. The markup used for the Python documentation is restructured text, developed by the docutils project, amended by custom directives and using a toolset named sphinx to post-process the HTML output.

This document describes the style guide for our documentation as well as the custom restructured text markup introduced by Sphinx to support Python documentation and how it should be used.

Introduction

Python’s documentation has long been considered to be good for a free programming language. There are a number of reasons for this, the most important being the early commitment of Python’s creator, Guido van Rossum, to providing documentation on the language and its libraries, and the continuing involvement of the user community in providing assistance for creating and maintaining documentation.

The involvement of the community takes many forms, from authoring to bug reports to just plain complaining when the documentation could be more complete or easier to use.

This document is aimed at authors and potential authors of documentation for Python. More specifically, it is for people contributing to the standard documentation and developing additional documents using the same tools as the standard documents. This guide will be less useful for authors using the Python documentation tools for topics other than Python, and less useful still for authors not using the tools at all.

If your interest is in contributing to the Python documentation, but you don't have the time or inclination to learn restructured Text and the markup structures documented here, there's a welcoming place for you among the Python contributors as well. Any time you feel that you can clarify existing documentation or provide documentation that's missing, the existing documentation team will gladly work with you to integrate your text, dealing with the markup for you. Please don't let the material in this document stand between the documentation and your desire to help out!

Style guide:

Use of whitespace

All reST files use an indentation of 3 spaces; no tabs are allowed. The maximum line length is 80 characters for normal text, but tables, deeply indented code samples and long links may extend beyond that. Code example bodies should use normal Python 4-space indentation.

A sentence-ending period may be followed by one or two

spaces; while reST ignores the second space, it is customarily put in by some users, for example to aid Emacs' auto-fill mode.

Footnotes

Footnotes are generally discouraged, though they may be used when they are the best way to present specific information. When a footnote reference is added at the end of the sentence, it should follow the sentence-ending punctuation. The reST markup should appear something like this:

```
This sentence has a footnote reference. [#]_ This is the next sentence.
```

Footnotes should be gathered at the end of a file, or if the file is very long, at the end of a section. The docutils will automatically create backlinks to the footnote reference.

Footnotes may appear in the middle of sentences where appropriate.
Capitalization

Sentence case

Sentence case is a set of capitalization rules used in English sentences: the first word is always capitalized and other words are only capitalized if there is a specific rule requiring it.

In the Python documentation, the use of sentence case in section titles is preferable, but consistency within a unit is more important than following this rule. If you add a section to a chapter where most sections are in title case, you can either convert all titles to sentence case or use the dominant style in the new section title.

Sentences that start with a word for which specific rules require starting it with a lower case letter should be avoided.

Many special names are used in the Python documentation, including the names of operating systems, programming languages, standards bodies, and the like. Most of these entities are not assigned any special markup.

Good example (establishing confident knowledge in the effective use of the language):
A best practice for using files is use a try/finally pair to explicitly close a file after it is used. Alternatively, using a with-statement can

achieve the same effect. This assures that files are flushed and file descriptor resources are released in a timely manner.

Economy of Expression

More documentation is not necessarily better documentation. Err on the side of being succinct.It is an unfortunate fact that making documentation longer can be an impediment to understanding and can result in even more ways to misread or misinterpret the text. Long descriptions full of corner cases and caveats can create the impression that a function is more complex or harder to use than it actually is.

Machine learning (ML)

Machine learningis the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

The name machine learning was coined in 1959 by Arthur Samuel. Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E. This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's

proposal in his Sproject "Computing Machinery and Intelligence, in which the question Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can doIn Turing's proposal the various characteristics that could be possessed by a thinking machine and the various implications in constructing one are exposed.

Machine learning uses data to detect various patterns in a given dataset.

1. It can learn from past data and improve automatically.
2. It is a data-driven technology.

3.Machine learning is much similar to data mining as it also deals with the huge amount of the data.

How does Machine Learning Work?

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedbackSemi-supervised learning algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

Classification algorithms and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails.

the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and

false. Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object.

In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of features, or inputs, in a set of data.

Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning to play a game against a human opponent Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

Relation to data mining

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of (previously) unknown properties in the data (this is the analysis step of knowledge discovery in databases). Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as unsupervised

learning or as a preprocessing step to improve learner accuracy. Much of the confusion between these two research communities (which do often have separate conferences and separate journals, ECML PKDD being a major exception) comes from the basic assumptions they work with: in machine learning, performance is usually evaluated with respect to the ability to reproduce known knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously unknown knowledge. Evaluated with respect to known knowledge, an uninformed (unsupervised) method will easily be outperformed by other supervised methods, while in a typical KDD task supervised methods cannot be used due to the unavailability of training data.

Relation to statistics

Machine learning and statistics are closely related fields in terms of methods, but distinct in their principal goal: statistics draws population inferences from a sample, while machine learning finds generalizable predictive patterns. According to Michael I. Jordan, the ideas of machine learning, from methodological principles to theoretical tools, have had a long pre- history in statistics. He also suggested the term data science as a placeholder to call the overall field.

Leo-Breiman distinguished two statistical modeling paradigms: data model and algorithmic model, wherein "algorithmic model" means more or less the machine learning algorithms like Random forest.

Some statisticians have adopted methods from machine learning, leading to a combined field that they call statistical learning.

Types of learning algorithms

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

Supervised learning Unsupervised learning Reinforcement learning

Supervised learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is

represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

Supervised learningalgorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression

algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

Supervised learning can be grouped further in two categories of algorithms: 1.Classification

2.Regression

Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predestinated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and

separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

It can be further classifieds into two categories of algorithms:

3.Clustering 4.Association

Reinforcement learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based

optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov Decision Process (MDP). Many reinforcement learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

Prerequisites

Before learning machine learning, you must have the basic knowledge of followings so that you can easily understand the concepts of machine learning:

1. Fundamental knowledge of probability and linear algebra.
2. The ability to code in any computer language, especially in Python language. 3.Knowledge of Calculus, especially derivatives of single variable and multivariate functions.

Linear Regression in Machine Learning

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc. Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of

the dependent variable is changing according to the value of the independent variable. The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

Linear regression can be further divided into two types of the algorithm:

Simple Linear Regression:

If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

Multiple Linear regression:

If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

What is the Classification Algorithm?

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data.

In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc. Classes can be called as targets/labels or categories. Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output

Types of ML Classification Algorithms:

Classification Algorithms can be further divided into the mainly two categories:

Linear Models

Logistic Regression Support Vector Machines

Non-linear Models

K-Nearest Neighbours Kernel SVM

Naïve Bayes

Decision Tree Classification Random Forest Classification

Logistic Regression in Machine Learning

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

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value.

It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

Assumptions for Logistic Regression:

The dependent variable must be categorical in nature.

The independent variable should not have multi-collinearity.

Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

K-Nearest Neighbor(KNN) Algorithm for Machine Learning

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases.

put the new case into the category that is most similar to the available categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular

Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyper plane.

SVM chooses the extreme points/vectors that help in creating the hyper plane. These extreme cases are called as support vectors, and hence algorithm is termed as Support

Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyper plane:

Naïve Bayes Classifier Algorithm

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.

It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Decision Tree Classification Algorithm

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into sub trees.

Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.

It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

PACKAGES

NumPy:

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin.

Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

The best way to enable NumPy is to use an installable binary package specific to your operating system. These binaries contain full SciPy stack (inclusive of NumPy, SciPy, matplotlib, IPython, SymPy and nose packages along with core Python).

Buildingfrom Source

Core Python (2.6.x, 2.7.x and 3.2.x onwards) must be installed with distutils and zlib module should be enabled.

GNU gcc (4.2 and above) C compiler must be available. To install NumPy, run the following command.

```
Python setup.py install
```

To test whether NumPy module is properly installed, try to import it from Python prompt.

```
import numpy
```

If it is not installed, the following error message will be displayed.

Traceback (most recent call last):

```
File "<pyshell#0>", line 1, in <module>
import numpy
ImportError: No module named 'numpy'
```

Alternatively, NumPy package is imported using the following syntax −

```
import numpy as np
```

Pandas:

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use

them in practice.

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can

accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

Standard Python distribution doesn't come bundled with Pandas module. A lightweight alternative is to install NumPy using popular Python package installer, **pip.**

```
pip install pandas
```

If you install Anaconda Python package, Pandas will be installed by default with the following −

Windows

Anaconda (from https://www.continuum.io) is a free Python distribution for SciPy stack. It is also available for Linux and Mac.

Canopy (https://www.enthought.com/products/canopy/) is available

as free as well as commercial distribution with full SciPy stack for Windows, Linux and Mac.

Python (x,y) is a free Python distribution with SciPy stack and Spyder IDE for Windows OS. (Downloadable from http://python-xy.github.io/)

By now, we learnt about the three Pandas DataStructures and how to create them. We will majorly focus on the DataFrame objects because of its importance in the real time data processing and also discuss a few other DataStructures.

Series Basic Functionality

S. No.	Attribute or Method & Description
1	**Axes** Returns a list of the row axis labels
2	**Dtype**Returns the dtype of the object.
3	**Empty** Returns True if series is empty.
4	**Ndim**Returns the number of dimensions of the underlying data, by definition 1.
5	**Size** Returns the number of elements in the underlying data.
6	**Values** Returns the Series as ndarray.
7	**head()** Returns the first n rows.
8	**tail()** Returns the last n rows.

Tensor flow:

Deep learning is a subfield of machine learning that is a set of algorithms that is inspired by the structure and function of the brain.

TensorFlow is the second machine learning framework that Google created and used to design, build, and train deep learning models. You can use the TensorFlow library do to numerical computations, which in itself doesn't seem all too special, but these computations are done with data flow graphs. In these graphs, nodes represent mathematical operations, while the edges represent the data, which usually are multidimensional data arrays or tensors, that are

communicated between these edges.

You see? The name “TensorFlow” is derived from the operations which neural networks perform on multidimensional data arrays or tensors! It’s literally a flow of tensors. For now, this is all you need to know about tensors, but you’ll go deeper into this in the next sections!

Today's TensorFlow tutorial for beginners will introduce you to performing deep learning in an interactive way:

- You'll first learn more about tensors;
- Then, the tutorial you'll briefly go over some of the ways that you can install TensorFlow on your system so that you're able to get started and load data in your workspace;
- After this, you'll go over some of the TensorFlow basics: you'll see how you can easily get started with simple computations.
- After this, you get started on the real work: you'll load in data on Belgian traffic signs and exploring it with simple statistics and plotting.
- In your exploration, you'll see that there is a need to manipulate your data in such a way that you can feed it to your model. That's why you'll take the time to rescale your images and convert them to grayscale.
- Next, you can finally get started on your neural network model! You'll build up your model layer per layer;
- Once the architecture is set up, you can use it to train your model interactively and to eventually also evaluate it by feeding some test data to it.
- Lastly, you'll get some pointers for further improvements that you can do to the model you just constructed and how you can continue your learning with TensorFlow.

Download the notebook of this tutorial here.

Also, you could be interested in a course on Deep Learning in Python, DataCamp's Keras tutorial or the keras with R tutorial.

Introducing Tensors

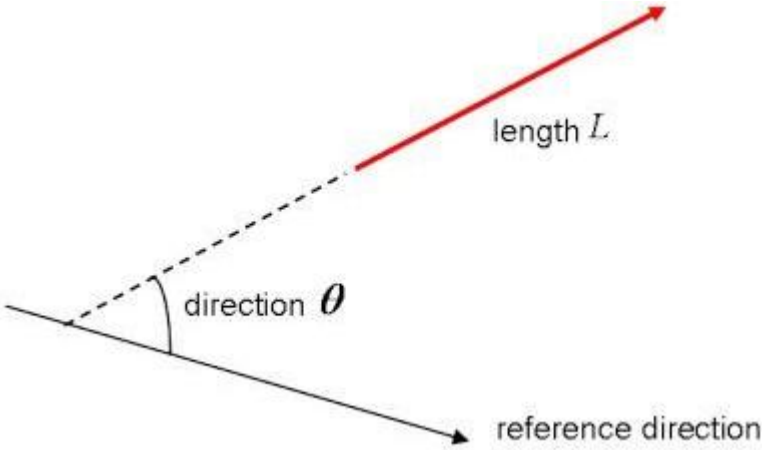
To understand tensors well, it's good to have some working knowledge of linear algebra and vector calculus. You already read in the introduction that tensors are

implemented in TensorFlow as multidimensional data arrays, but some more introduction is maybe needed in order to completely grasp tensors and their use in machine learning.

Plane Vectors

Before you go into plane vectors, it's a good idea to shortly revise the concept of "vectors"; Vectors are special types of matrices, which are rectangular arrays of numbers. Because vectors are ordered collections of numbers, they are often seen as column matrices: they have just one column and a certain number of rows. In other terms, you could also consider vectors as scalar magnitudes that have been given a direction.

Remember: an example of a scalar is "5 meters" or "60 m/sec", while a vector is, for example, "5 meters north" or "60 m/sec East". The difference between these two is obviously that the vector has a direction. Nevertheless, these examples that you have seen up until now might seem far off from the vectors that you might encounter when you're working with machine learning problems. This is normal; The length of a mathematical vector is a pure number: it is absolute. The direction, on the other hand, is relative: it is measured relative to some reference direction and has units of radians or degrees. You usually assume that the direction is positive and in counterclockwise rotation from the reference direction.



Visually, of course, you represent vectors as arrows, as you can see in the picture above. This means that you can consider vectors also as arrows that have direction and

length. The direction is indicated by the arrow's head, while the length is indicated by the length of the arrow.

So what about plane vectors then?

Plane vectors are the most straightforward setup of tensors. They are much like regular vectors as you have seen above, with the sole difference that they find themselves in a vector space. To understand this better, let's start with an example: you have a vector that is *2 X 1*. This means that the vector belongs to the set of real numbers that come paired two at a time. Or, stated differently, they are part of two-space. In such cases, you can represent vectors on the coordinate *(x,y)* plane with arrows or rays.

Working from this coordinate plane in a standard position where vectors have their endpoint at the origin *(0,0)*, you can derive the *x* coordinate by looking at the first row of the vector, while you'll find the *y* coordinate in the second row. Of course, this standard position doesn't always need to be maintained: vectors can move parallel to themselves in the plane without experiencing changes.

Note that similarly, for vectors that are of size *3 X 1*, you talk about the three- space. You can represent the vector as a three-dimensional figure with arrows pointing to positions in the vectors pace: they are drawn on the standard *x*, *y* and *z* axes.

It's nice to have these vectors and to represent them on the coordinate plane, but in essence, you have these vectors so that you can perform operations on them and one thing that can help you in doing this is by expressing your vectors as bases or unit vectors.

Unit vectors are vectors with a magnitude of one. You'll often recognize the unit vector by a lowercase letter with a circumflex, or "hat". Unit vectors will come in convenient if you want to express a 2-D or 3-D vector as a sum of two or three orthogonal components, such as the x− and y−axes, or the z−axis.

And when you are talking about expressing one vector, for example, as sums of components, you'll see that you're talking about component vectors, which are two or more vectors whose sum is that given vector.

Tip: watch this video, which explains what tensors are with the help of simple household objects!

Tensors

Next to plane vectors, also covectors and linear operators are two other cases that all three together have one thing in common: they are specific cases of tensors. You still remember how a vector was characterized in the previous section as scalar magnitudes that have been given a direction. A tensor, then, is the mathematical representation of a physical entity that may be characterized by magnitude and *multiple* directions.

And, just like you represent a scalar with a single number and a vector with a sequence of three numbers in a 3-dimensional space, for example, a tensor can be represented by an array of 3R numbers in a 3-dimensional space.

The "R" in this notation represents the rank of the tensor: this means that in a 3- dimensional space, a second-rank tensor can be represented by 3 to the power of 2 or 9 numbers. In an N-dimensional space, scalars will still require only one number, while vectors will require N numbers, and tensors will require N^R numbers. This explains why you often hear that scalars are tensors of rank 0: since they have no direction, you can represent them with one number.

With this in mind, it's relatively easy to recognize scalars, vectors, and tensors and to set them apart: scalars can be represented by a single number, vectors by an ordered set of numbers, and tensors by an array of numbers.

What makes tensors so unique is the combination of components and basis vectors: basis vectors transform one way between reference frames and the components transform in just such a way as to keep the combination between components and basis vectors the same.

Installing TensorFlow

Now that you know more about TensorFlow, it's time to get started and install the library. Here, it's good to know that TensorFlow provides APIs for Python, C++, Haskell, Java, Go, Rust, and there's also a third-party package for R called tensorflow.

Tip:

if you want to know more about deep learning packages in R, consider checking out DataCamp's keras: Deep Learning in R Tutorial.

In this tutorial, you will download a version of TensorFlow that will enable you to write the code for your deep learning project in Python. On the TensorFlow installation webpage, you'll see some of the most common ways and latest instructions to install TensorFlow using virtualenv, pip, Docker and lastly, there are also some of the other ways of installing TensorFlow on your personal computer.

Note:

You can also install TensorFlow with Conda if you're working on Windows. However, since the installation of TensorFlow is community supported, it's best to check the official installation instructions.

Now that you have gone through the installation process, it's time to double check that you have installed TensorFlow correctly by importing it into your workspace under the alias tf:

Note:

that the alias that you used in the line of code above is sort of a convention - It's used to ensure that you remain consistent with other developers that are using TensorFlow in data science projects on the one hand, and with open-source TensorFlow projects on the

other hand.

Keras:

Two of the top numerical platforms in Python that provide the basis for Deep

Learning research and development are Theano and TensorFlow. Both are very powerful libraries, but both can be difficult to use directly for creating deep learning models.

In this post, you will discover the Keras Python library that provides a clean and convenient way to create a range of deep learning models on top of Theano or TensorFlow.

Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow.

It was developed to make implementing deep learning models as fast and easy as possible for research and development.

It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license.

Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles:

Modularity:

A model can be understood as a sequence or a graph alone. All the concerns of a deep learning model are discrete components that can be combined in arbitrary ways.

Minimalism:

The library provides just enough to achieve an outcome, no frills and maximizing readability.

Extensibility:

New components are intentionally easy to add and use within the framework, intended for researchers to trial and explore new ideas.

Python:

No separate model files with custom file formats. Everything is native Python Keras is relatively straightforward to install if you already have a working Python and SciPy environment. You must also have an installation of Theano or TensorFlow on your system already. You can see installation instructions for both platforms here:

Installation instructions for Theano Installation instructions for

TensorFlow

Keras can be installed easily using PyPI, as follows:

At the time of writing, the most recent version of Keras is version 1.1.0. You can check your version of Keras on the command line using the following snippet:

You can check your version of Keras on the command line using the following snippet.

Sklearn:

In general, a learning problem considers a set of n samples of data and then tries to predict properties of unknown data. If each sample is more than a single number and, for instance, a multi-dimensional entry (aka multivariate data), it is said to have several attributes or **features**.

Learning problems fall into a few categories:

supervised learning, in which the data comes with additional attributes that we want to predict (Click here to go to the scikit-learn supervised learning page).This problem can be either:

classification:

samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data. An example of a classification problem would be handwritten digit recognition, in which the aim is to assign each input vector to one of a finite number of discrete categories. Another way to think of classification is as a discrete (as opposed to continuous) form of supervised learning where one has a limited number of categories and for each of the n samples provided, one is to try to label them with the correct category or class.

regression:

if the desired output consists of one or more continuous variables, then the task is called *regression*. An example of a regression problem would be the prediction of the length of a salmon

as a function of its age and weight.

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

scikit-learn comes with a few standard datasets, for instance the iris and digits datasets for classification and the boston house prices dataset for regression.

In the following, we start a Python interpreter from our shell and then load the iris and digits datasets. Our notational convention is that $ denotes the shell prompt while >>> denotes the Python interpreter prompt:

```
$ python
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
```

A dataset is a dictionary-like object that holds all the data and some metadata about the data. This data is stored in the .data member, which is a n_samples, n_features array. In the case of supervised problem, one or more response variables are stored in the .target member. More details on the different datasets can be found in the dedicated section.

For instance, in the case of the digits dataset, digits.data gives access to the features that can be used to classify the digits samples:

>>>

```
>>>print(digits.data)

[      0    5     0    0    0
[      .    .     .    .    .
0           .                ]
.           .
            .
[      0    0     1    0    0
 [ 0.   0.   2 ..... 12.   0.   0.]
 [ 0.   0.  10. ...  12.   1.   0.]]
```

```
0   .   .   0   .   .
.       .   .       ]
        .
        .
[   0   0   1   9   0
0   .   .   6   .   .
.       .   .       ]
        .
        .
.
.
.
[   0   1   6   0   0
0   .   .   .   .   .
.       .           ]
        .
        .
```

and digits. Target gives the ground truth for the digit dataset, that is the number corresponding to each digit image that we are trying to do.

Python:

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural,) object- oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end-of-life, only Python 3.5.xand later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open sourcereference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Python is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Python do?:

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?:

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Python compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- 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.

Python installation procedure Windows Based

It is highly unlikely that your Windows system shipped with Python already installed. Windows systems typically do not. Fortunately, installing does not involve much

more than downloading the Python installer from the python.org website and running it. Let's take a look at how to install Python 3 on Windows:

Step 1: Download the Python 3 Installer

1. Open a browser window and navigate to the Download page for Windows at python.org.
2. Underneath the heading at the top that says **Python Releases for Windows**, click on the link for the **Latest Python 3 Release - Python 3.x.x**. (As of this writing, the latest is Python 3.6.5.)
3. Scroll to the bottom and select either **Windows x86-64 executable installer** for 64- bit or **Windows x86 executable installer** for 32-bit. (See below.)

Sidebar: 32-bit or 64-bit Python?
For Windows, you can choose either the 32-bit or 64-bit installer. Here's what the difference between the two comes down to:

- If your system has a 32-bit processor, then you should choose the 32-bit installer.
- On a 64-bit system, either installer will actually work for most purposes. The 32-bit version will generally use less memory, but the 64-bit version performs better for applications with intensive computation.
- If you're unsure which version to pick, go with the 64-bit version.

Note: Remember that if you get this choice "wrong" and would like to switch to another version of Python, you can just uninstall Python and then re-install it by downloading another installer from

python.org.

Step 2: Run the Installer

Once you have chosen and downloaded an installer, simply run it by double-clicking on the downloaded file. A dialog should appear that looks something like this:

Important: You want to be sure to check the box that says **Add Python 3.x to PATH** as shown to ensure that the interpreter will be placed in your execution path.

Then just click **Install Now**. That should be all there is to it. A few minutes later you should have a working Python 3 installation on your system.

Mac OS based

While current versions of macOS (previously known as "Mac OS X") include a version of Python 2, it is likely out of date by a few months. Also, this tutorial series uses Python 3, so let's get you upgraded to that.

The best way we found to install Python 3 on macOS is through the Homebrew package manager. This approach is also recommended by community guides like The Hitchhiker's Guide to Python.

Step 1: Install Homebrew (Part 1)

To get started, you first want to install Homebrew:

1. Open a browser and navigate to http://brew.sh/. After the page has finished loading, **select the Homebrew bootstrap code under "Install Homebrew"**. Then hit cmd+c to copy it to the clipboard. Make sure you've captured the text of the complete command because otherwise the installation will fail.

2. Now you need to **open a Terminal app window, paste the Homebrew bootstrap code, and then hit** Enter. This will begin the Homebrew installation.
3. If you're doing this on a fresh install of macOS, you may get a pop up alert **asking you to install Apple's "command line developer tools"**. You'll need those to continue with the installation, so please **confirm the dialog box by clicking on "Install"**.

At this point, you're likely waiting for the command line developer tools to finish installing, and that's going to take a few minutes. Time to grab a coffee or tea!

Step 2: Install Homebrew (Part 2)

You can continue installing Homebrew and then Python after the command line developer tools installation is complete:

1. Confirm the "The software was installed" dialog from the developer tools installer.
2. Back in the terminal, hit Enter to continue with the Homebrew installation.
3. Homebrew asks you to enter your password so it can finalize the installation. **Enter your user account password and hit** Enter to continue.
4. Depending on your internet connection, Homebrew will take a few minutes to download its required files. Once the installation is complete, you'll end up back at the command prompt in your terminal window.

Whew! Now that the Homebrew package manager is set up, let's continue on with installing Python 3 on your system.

Step 3:

Install Python

Once Homebrew has finished installing, **return to your terminal and run the following command**:

$ brew install python3

Note: When you copy this command, be sure you don't include the $ character at the beginning. That's just an indicator that this is a console command.

This will download and install the latest version of Python. After the Homebrew brew install command finishes, Python 3 should be installed on your system.

You can make sure everything went correctly by testing if Python can be accessed from the terminal:

1. Open the terminal by launching **Terminalapp**.
2. Type pip3 and hit Enter.
3. You should see the help text from Python's "Pip" package manager. If you get an error message running pip3, go through the Python install steps again to make sure you have a working Python installation.

Assuming everything went well and you saw the output from Pip in your command prompt window…congratulations! You just installed Python on your system, and you're all set to continue with the next section in this tutorial.

Packages need for python based programming:

- **Numpy**

  NumPy is a Python package which stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc.

- **Pandas**

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

of variables.

- **Keras**

  Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. Use Keras if you need a deep learning library that: Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).

- **Sklearn**

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

- **Scipy**

  SciPy is an open-source Python library which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands.

- **Tensorflow**

  TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

- **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.

- **Pyodbc**

  pyodbc is an open source Python module that makes accessing ODBC databases simple. It implements the DB API 2.0 specification but is packed with even more Pythonic convenience. Precompiled binary wheels are provided for most Python versions on Windows and macOS. On other operating systems this will build from source.

- **Matplotlib**

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

- **Opencv**

  OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Python is a general purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability.

- **Nltk**

  Natural Language Processing with Python NLTK is one of the leading platforms

  for working with human language data and Python, the module NLTK is used for natural language processing. NLTK is literally an acronym for Natural Language Toolkit. In this article you will learn how to tokenize data (by words and sentences).

- **SQLAIchemy**

  SQLAlchemy is a library that facilitates the communication between Python programs and databases. Most of the times, this library is used as an Object Relational Mapper (ORM) tool that translates Python classes to tables on relational databases and automatically converts function calls to SQL statements.

- **Urllib**

  urllib is a Python module that can be used for opening URLs. It defines functions and classes to help in URL actions. With Python you can also access and retrieve data from the internet like XML, HTML, JSON, etc. You can also use Python to work with this data directly.

Installation of packages:

Syntax for installation of packages via cmd terminal using the basic

Step:1- First check pip cmd

First check pip cmd If ok then

Step:2- pip list

Check the list of packages installed and then install required by following cmds

Step:3- pip install package name

The package name should as requirement.

CHAPTER -5

5. SYSTEM DESIGN

Architecture Diagram:-

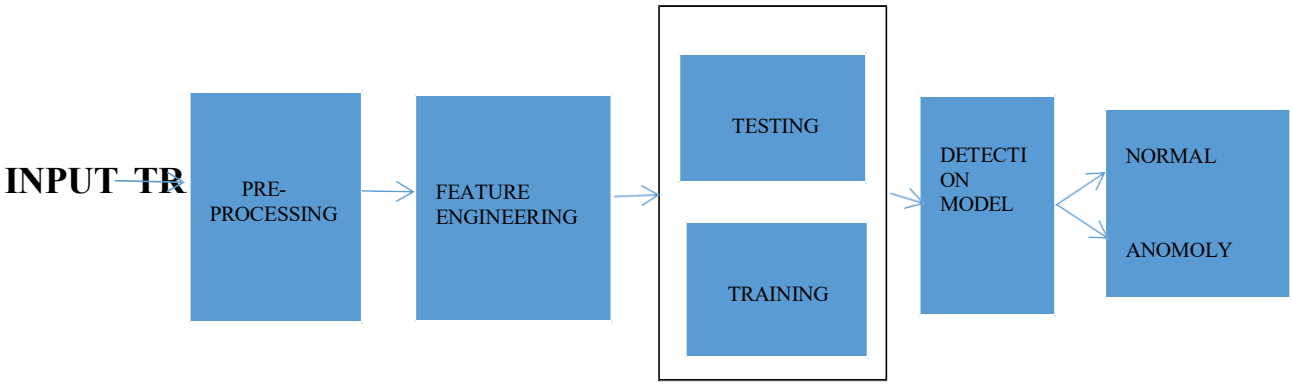


FIGURE 5.1 System architecture

UML DIAGRAMS

A UML diagram is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence. UML diagram contains graphical elements (symbols) - UML nodes connected with edges (also known as paths or flows) - that represent elements in the UML model of the designed system. The UML model of the system might also contain other documentation such as use cases written as templated texts. The kind of the diagram is defined by the primary graphical symbols shown on the diagram. For example, a diagram where the primary symbols in

the contents area are classes is class diagram.

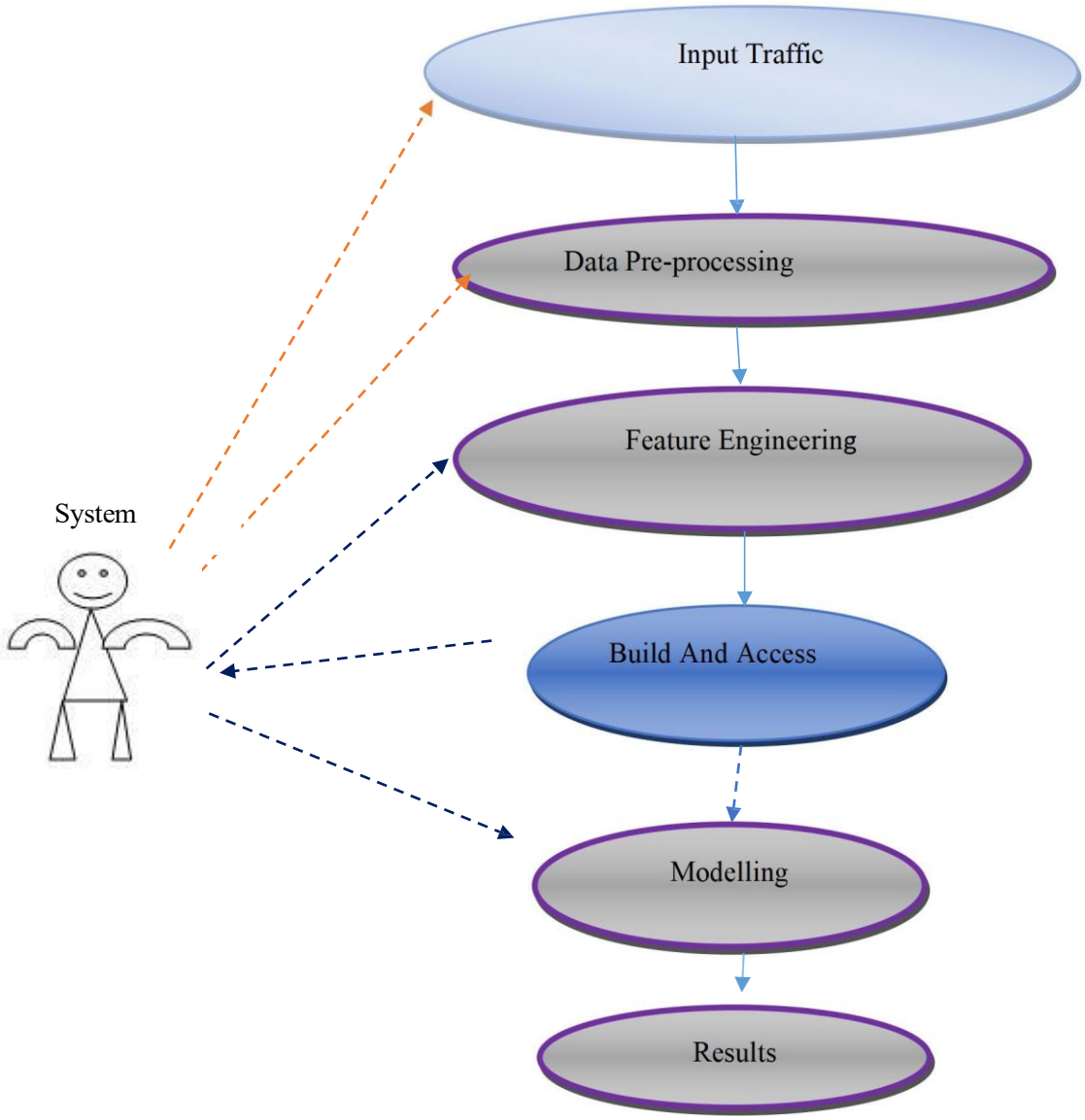
A diagram which shows use cases and actors is use case diagram. A sequence diagram shows sequence of message exchanges between lifelines. UML specification does not preclude mixing of different kinds of diagrams, e.g. to combine structural and behavioral elements to show a state machine nested inside a use case. Consequently, the boundaries between the various kinds of diagrams are not strictly enforced.

At the same time, some UML Tools do restrict set of available graphical elements which could be used when working on specific type of diagram. UML specification defines two major kinds of UML diagram: structure diagrams and behavior diagrams. Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts. Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

5.1 USE CASE DIAGRAM

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent: Scenarios in which your system or application interacts with people, organizations, or external systems. Goals that your system or application helps those entities (known as actors) achieve.

➢ **Use case**



FIAGURE 5.2 : use case diagram

5.2 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

- **Sequence Diagram**

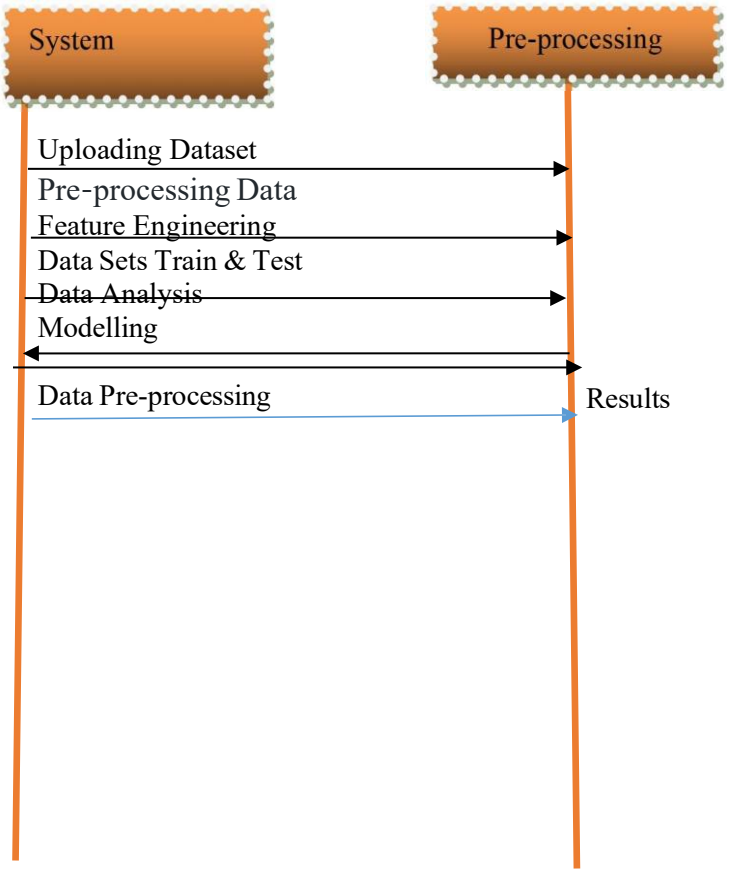
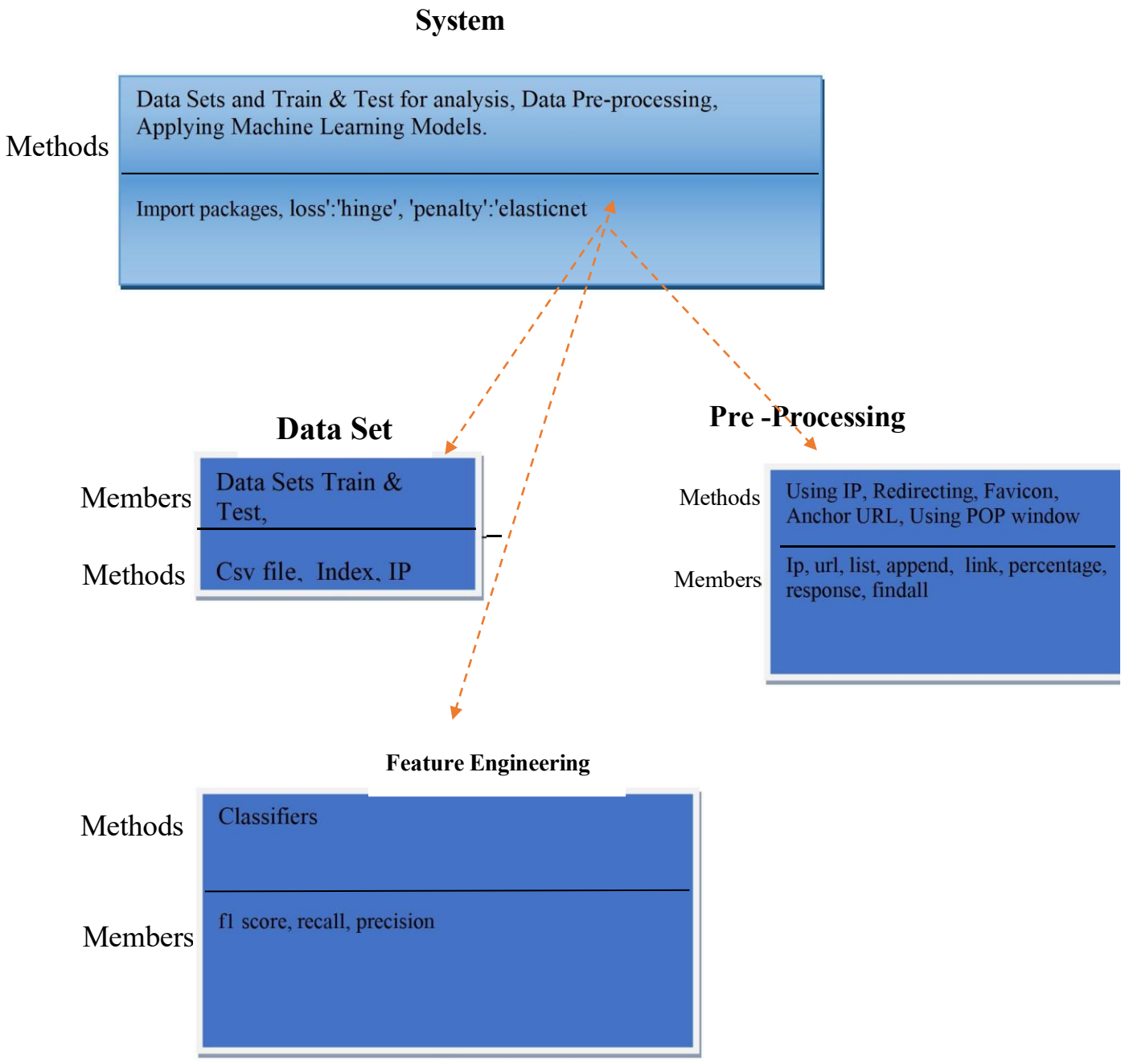


FIGURE 5.3 : sequence diagram

5.3 Class Diagram :



FIAGURE 5.4 : Class diagram

CHAPTER -6

6. source code

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import joblib

app = Flask(__name__)
model =
joblib.load(r'C:\Users\Reddy\Pycharmprojects\CAML\model.pkl
')

@app.route('/') def home():
  return render_template('index.html')

@app.route('/predict',methods=['POST']) def predict():

  int_features = [float(x) for x in request.form.values()] if

  int_features[0]==0:
    f_features=[0,0,0]+int_features[1:] elif int_features[0]==1:
    f_features=[1,0,0]+int_features[1:] elif int_features[0]==2:
    f_features=[0,1,0]+int_features[1:] else:
    f_features=[0,0,1]+int_features[1:]

  if f_features[6]==0:
    fn_features=f_features[:6]+[0,0]+f_features[7:]
  elif f_features[6]==1:
    fn_features=f_features[:6]+[1,0]+f_features[7:]
  else:
    fn_features=f_features[:6]+[0,1]+f_features[7:]

  final_features = [np.array(fn_features)] predict =
  model.predict(final_features)

  if predict==0: output='Normal'
  elif predict==1: output='DOS'
  elif predict==2: output='PROBE'
  elif predict==3: output='R2L'
```

```
 else:
      output='U2R’

   return render_template('index.html', output=output)

@app.route('/results',methods=['POST'])
def results():

   data = request.get_json(force=True)
   predict = model.predict([np.array(list(data.values()))])


 if predict==0: output='Normal'
   elif predict==1: output='DOS'
   elif predict==2: output='PROBE'
   elif predict==3: output='R2L'
   else:
      output='U2R' return jsonify(output)

if __name__== "__main__": app.run()



<!DOCTYPE html>
<html >
<head>
 <meta charset="UTF-8">
 <title>Cyber Attack Detection System</title>
 <link rel="stylesheet" href="static/style.css">
</head>

<body>
 <div class="login">
  <h1 style="color:blue;">Cyber Attack Detection System</h1>

    <!-- Main Input For Receiving Query to our ML -->
   <form action="{{ url_for('predict')}}"method="post">
      <label for="attack">Attack:</label>
```

```
      <select id="attack" name="attack">
        <option value="0">Other</option>
        <option value="1">neptune</option>
        <option value="2">normal</option>
        <option value="3">satan</option>
      </select><br><br>

      <label for="count">Number of connections to the same
destination host as the current connection in the past two
seconds :</label>
     <input type="text" name="count" placeholder="count"
     required="required"
/><br><br>

     <label for="dst_host_diff_srv_rate">The percentage of
connections that were to different services, among the
connections aggregated in dst_host_count :</label>
      <input type="text" name="dst_host_diff_srv_rate"
placeholder="dst_host_diff_srv_rate" required="required"
/><br><br>

      <label for="dst_host_same_src_port_rate">The percentage
of connections that were to the same source port, among the
connections aggregated in dst_host_srv_count :</label>
    <input type="text" name="dst_host_same_src_port_rate"
placeholder="dst_host_same_src_port_rate"
required="required" /><br><br>

    <label for="dst_host_same_srv_rate">The percentage of
connections that were to the same service, among the
connections aggregated in dst_host_count :</label>
    <input type="text" name="dst_host_same_srv_rate"
placeholder="dst_host_same_srv_rate" required="required"
/><br><br>

    <label for="dst_host_srv_count">Number of connections
having the same port number :</label>
    <input type="text" name="dst_host_srv_count"
placeholder="dst_host_srv_count" required="required"
/><br><br>

    <label for="flag">Status of the connection –Normal or Error
    :</label>
```

```
    <select id="flag" name="flag">
      <option value="0">Other</option>
      <option value="1">S0</option>
      <option value="2">SF</option>
     </select><br><br>

     <label for="last_flag">Last Flag :</label>
    <input type="text" name="last_flag"
placeholder="last_flag" required="required" /><br><br>

    <label for="logged_in">1 if successfully logged in; 0
    otherwise :</label>
     <input type="text" name="logged_in"
placeholder="logged_in" required="required" /><br><br>
```

```
    <label for="same_srv_rate">The percentage of connections
that were to the same service, among the connections aggregated
in count :</label>
     <input type="text" name="same_srv_rate"
placeholder="same_srv_rate" required="required"
/><br><br>

    <label for="serror_rate">The percentage of connections
that have activated the flag (4) s0, s1, s2 or s3, among the
connections aggregated in count :</label>
    <input type="text" name="serror_rate"
placeholder="serror_rate" required="required" /><br><br>

   <label for="service_http">Destination network service used
   http or not :</label>
   <select id="service_http" name="service_http">
     <option value="0">No</option>
     <option value="1">Yes</option>
    </select><br><br>

    <button type="submit" class="btn btn-primary btn-block
btn- large">Predict</button>
  </form>
 <br>
  {% if output %}
    <h3> Attack Class should be <b style="color:red;">{{
    output }}</b>  </h3>
  {% endif %}
 </div>
</body>
</html>
```

CHAPTER -7

7. Screenshots

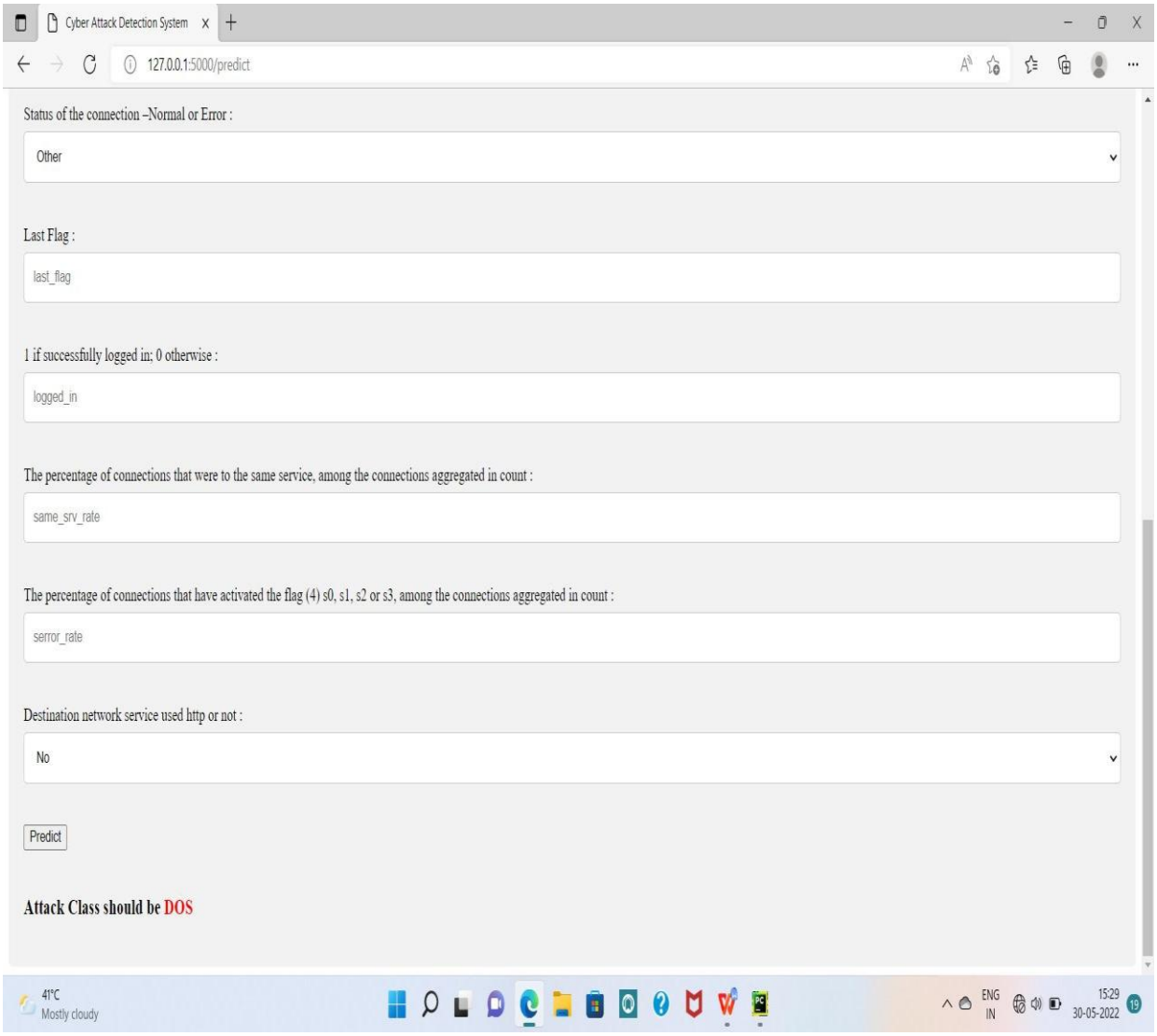


Figure 7.1:**OUTPUT1**

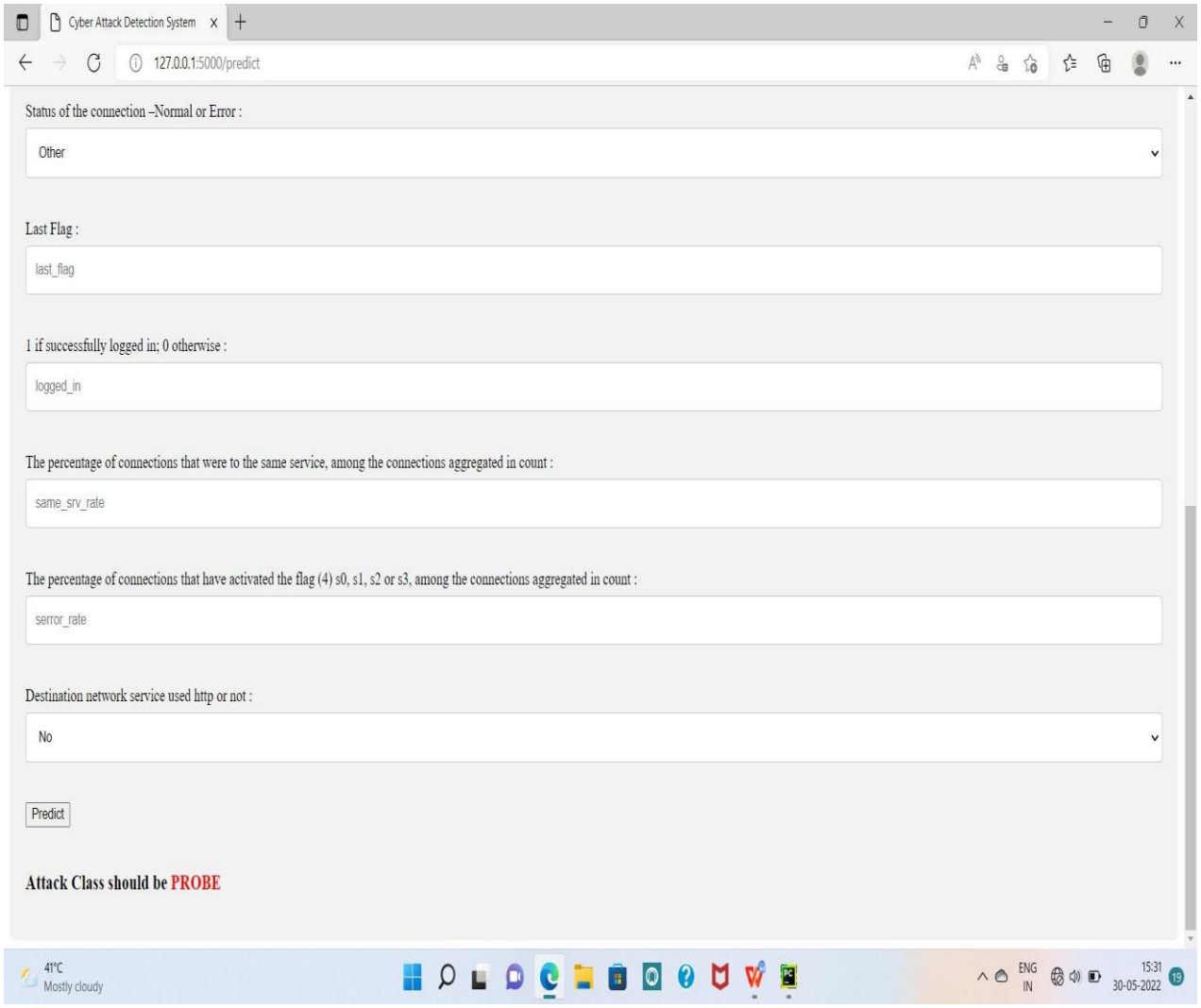


Figure 7.2: OUTPUT2

CHAPTER -8

8. SYSTEM STUDY

8.1 FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

8.2 ECONOMICAL FEASIBILITY

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

- TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical

resources.

This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

8.3 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, as he is the final user of the system.

CHAPTER -9

9.IMPLEMENTATION

MODULES

1. Data Collection
2. Data Pre-processing
3. Train and Test Modelling
4. Attack Detection Model

MODULE DESCRIPTION

1. Data Collection:

Collect sufficient data samples and legitimate software samples.

2. Data Pre-processing:

Data Augmented techniques will be used for better performance.

3. Train and Test Modelling:

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 no.

CHAPTER -10

10. SYSTEM TESTING

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 test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

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 perform basic 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.

10.1 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.

10.2 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

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.

10.3 System Testing

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.

10.4 White Box Testing

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

10.5 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. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

10.6 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.

Test strategy and approach

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

Test objectives

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

Features to be tested

Verify that the entries are of the correct format

- No duplicate entries should be allowed
- All links should take the user to the correct page.

10.7 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.

10.7 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 functional requirements.

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

10.5 SYSTEM TESTING

TESTING METHODOLOGIES

The following are the Testing Methodologies:

- **Unit Testing.**
- **Integration Testing.**
- **User Acceptance Testing.**
- **Output Testing.**
- **Validation Testing.**

Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1) Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downward.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

OTHER TESTING METHODOLOGIES

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with

the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements

are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the

requirements specified as per software requirement specification and was accepted.

USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's

requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

TESTING STRATEGY :

A strategy for system testing integrates system test cases and design techniques into a well- planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation.

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module. In Due Course, latest technology advancements will be taken into consideration.

CHAPTER -11

11. 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 project assists with distinguishing which calculation predicts the best precision rates which assists with foreseeing best outcomes to recognize the digital assaults occurred or not.

CHAPTER -12

12. 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 –

[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] Girish, L., & Deepthi ,T. K.(2018). Efficient Monitoring Of Time Series Data Using Dynamic Alerting. i-manager’s Journal on Computer Science, 6(2), 1-6. https://doi.org/10.26634/jcom.6.2.14870

Girish. "DDoS Mitigation using Software Defined Network."International Journal of Engineering Trends and Technology (IJETT) 24.5 (2015): 258 -264.

[12] ShambulingappaH 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.

Chapter 13

FUTURE SCOPE

In future, this model can be used to compare various machine learning algorithms generated prediction models and the model which will give higher accuracy will be chosen as the prediction model.This project work can be extended to higher level by working with multiple datasets
at a time with multiple attributes.