Cover / Title Page

A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks

A Project report submitted for the Award of Degree of

## Master of Business Administration (MBA)

**BY**

**XXXXXX**

**XXXXX**

**Under the guidance of**

**--------NAME WITH DESIGNATION------**

## XXXXXXX

**SAMPLE COPY**

**CERTIFICATE**

This is to certify that the project entitled (A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks) submitted in partial fulfillment for the award of the degree of MBA. in XXXX, Visakhapatnam was carried out by **Mr./Ms.\_ (ID NO- )** under my guidance. This work is not submitted to any other University or Institution for the award of any Degree / Diploma Certificate.

## GUIDE

**SAMPLE COPY**

**DECLARATION**

I hereby declare that this Project Report titled “A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks”submitted by me to the XXXX, is a bonafied work under taken by me and it is not submitted to any other University or Institution for the award of any Degree

/Diploma Certificate.

Name & Address of the Student Signature of the Student

**ACKNOWLEDGEMENTS**

I would like to express my gratitude to **XXXX**, Director , XXXXX, and to all the people, who extended unending support to all stages of project. This report is a product of not only my sincere effort but also guidance and support given by **XXXX**.

I wish to express my sincere thanks ---------------------for project

Guidance and also the management and staff of ----------------------providing the guidance and support.

**ABSTRACT**

The botnet attack is a multi-stage and the most prevalent cyber-attack in the Internet of Things (IoT) environment that initiates with scanning activity and ends at the distributed denial of service (DDoS) attack. The existing studies mostly focus on detecting botnet attacks after the IoT devices get compromised, and start performing the DDoS attack. Similarly, the performance of most of the existing machine learning based botnet detection models is limited to a specific dataset on which they are trained. As a consequence, these solutions do not perform well on other datasets due to the diversity of attack patterns. Therefore, in this work, we first produce a generic scanning and DDoS attack dataset by generating 33 types of scan and 60 types of DDoS attacks. In addition, we partially integrated the scan and DDoS attack samples from three publicly-available datasets for maximum attack coverage to better train the machine learning algorithms. Afterwards, we propose a two-fold machine learning approach to prevent and detect IoT botnet attacks. In the first fold, we trained a state-of-the-art deep learning model, i.e., ResNet-18 to detect the scanning activity in the premature attack stage to prevent IoT botnet attacks. While, in the second fold, we trained another ResNet-18 model for DDoS attack identification to detect IoT botnet attacks. Overall, the proposed two-fold approach manifests 98.89% accuracy, 99.01% precision, 98.74% recall, and 98.87% f1-score to prevent and detect IoT botnet attacks. To demonstrate the effectiveness of the proposed two-fold approach, we trained three other ResNet-18 models over three different datasets for detecting scan and DDoS attacks and compared their performance with the proposed two-fold approach. The experimental results prove that the proposed two-fold approach can efficiently prevent and detect botnet attacks as compared to other trained models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **INDEX** | |  | |
| **LIST OF CONTENTS** | | | | **PAGE NO** | |
| Chapter 1: Introduction | | | | 01 | |
| Chapter 2: Literature Survey | | | | 03 | |
| Chapter 3: System Analysis | | | | 04 | |
| 3.1 | Existing System | | | 04 | |
| 3.2 | Disadvantages | | | 04 | |
| 3.2 | Proposed System | | | 04 | |
| 3.4 | Advantages | | | 05 | |
| 3.5 | Modules | | | 05 | |
|  | | | |  | |
| Chapter 4: Software Requirement Specification | | | | 06 | |
|  | | | |  | |
| Chapter 5: System Design &UML Design | | | | 11 | |
| Chapter 6: Code | | | 20 | |
| Chapter 7 : Output Screens | | | 33 | |
| Chapter 8: Conclusion | | | 40 | |
| Chapter 9: Bibliography | | | 41 | |
|  | | |  | |

**CHAPTER 1**

**INTRODUCTION**

**1. INTRODUCTION**

Internet of Things (IoT) devices are increasingly integrated in cyber-physical systems (CPS), including in critical infrastructure sectors such as dams and utility plants. In these settings, IoT devices (also referred to as Industrial IoT or IIoT) are often part of an Industrial Control System (ICS), tasked with the reliable operation of the infrastructure. ICS can be broadly defined to include supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and systems that comprise programmable logic controllers (PLC) and Modbus protocols. The connection between ICS or IIoT-based systems with public networks, however, increases their attack surfaces and risks of being targeted by cyber criminals. One high-profile example is the Stuxnet campaign, which reportedly targeted Iranian centrifuges for nuclear enrichment in 2010, causing severe damage to the equipment [1], [2]. Another example is that of the incident targeting a pump that resulted in the failure of an Illinois water plant in 2011 [3]. BlackEnergy3 was another campaign that targeted Ukraine power grids in 2015, resulting in power outage that affected approximately 230,000 people [4]. In April 2018, there were also reports of successful cyber-attacks affecting three U.S. gas pipeline firms, and resulted in the shutdown of electronic customer communication systems for several days [1]. Although security solutions developed for information technology (IT) and operational technology (OT) systems are relatively mature, they may not be directly applicable to ICSs. For example, this could be the case due to the tight integration between the controlled physical environment and the cyber systems. Therefore, system-level security methods are necessary to analyze physical behaviour and maintain system operation availability [1]. ICS security goals are prioritized in the order of availability, integrity, and confidentiality, unlike most IT/OT systems (generally prioritized in the order of confidentiality, integrity, and availability) [5]. Due to close coupling between variables of the feedback control loop and physical processes, (successful) cyber-attacks on ICS can result in severe and potentially fatal consequences for the society and our environment. This reinforces the importance of designing extremely robust safety and security measurements to detect and prevent intrusions targeting ICS [1]. Popular attack detection and attribution approaches include those based on signatures and anomalies. To mitigate the known limitations in both signature-based and anomaly-based detection and attribution approaches, there have been attempts to introduce hybrid-based approaches [6]. Although hybridbased approaches are effective at detecting unusual activates, they are not reliable due to frequent network upgrades, resulting in different Intrusion Detection System (IDS) typologies [7]. Beyond this, conventional attack detection and attribution techniques mainly rely on network metadata analysis (e.g. IP addresses, transmission ports, traffic duration, and packet intervals). Therefore, there has been renewed interest in utilizing attack detection and attribution solutions based on Machine Learning (ML) or Deep Neural Networks (DNN) in recent times. In addition, attack detection approaches can be categorized into network-based or host-based approaches. Supervised clustering, single-class or multi-class Support Vector Machine (SVM), fuzzy logic, Artificial Neural Network (ANN), and DNN are commonly used techniques for attack detection in network traffic. These techniques analyze real-time traffic data to detect malicious attacks in a timely manner. However, attack detection that considers only network and host data may fail to detect sophisticated attacks or insider attacks.

Unsupervised models that incorporate process/physical data can complement a system’s monitoring since they do not rely on detailed knowledge of the cyber-threats. In general, a sophisticated attacker with sufficient knowledge and time, such as a nation state advanced persistent threat actor, can potentially circumvent robust security solutions. Furthermore, most of the existing approaches ignore the imbalanced property of ICS data by modeling only a system’s normal behavior and reporting deviations from normal behavior as anomalies. This is, perhaps, due to limited attack samples in existing datasets and real-world scenarios. Although using majority class samples is a good solution to avoid issues due to imbalanced datasets, the trained model will have no view of the attack samples’ patterns. In other words, such an approach fails to detect unseen attacks and suffers from a high falsepositive rate [8]. Thus, there have been attempts to utilize DL approaches, for example, to facilitate automated feature (representation) learning to model complex concepts from simpler ones [9] without depending on human-crafted features [10]. Motivated by the above observations, this paper presents our proposed novel two-stage ensemble deep learning-based attack detection and attack attribution framework for imbalanced ICS datasets. In the first stage, an ensemble representation learning model combined with a Decision Tree (DT) is designed to detect attacks in an imbalanced environment. Once the attack is detected, several one-vs-all classifiers will ensemble together to form a larger DNN to classify the attack attributes with a confidence interval during the second stage. Moreover, the proposed framework is capable of detecting unseen attack samples. A summary of our approach in this study is as follows:

1) We develop a novel two-phase ensemble ICS attack detection method capable of detecting both previously seen and unseen attacks. We will also demonstrate that the proposed method outperforms other competing approaches in terms of accuracy and f-measure. The proposed deep representation learning results in this method being robust to imbalanced data.

2) We propose a novel self-tuning two-phase attack attribution method that ensembles several deep one-vsall classifiers using a DNN architecture for reducing false alarm rates. The proposed method can accurately attribute attacks with high similarity. This is the first ML-based attack attribution method in ICS/IIoT at the time of this research.

3) We analyze the computational complexity of the proposed attack detection and attack attribution framework, demonstrating that despite its superior performance, its computational complexity is similar to that of other DNN-based methods in the literature.

**CHAPTER 2**

**LITERATURE SURVEY**

**2. LITERATURE SURVEY**

ML-based attack detection techniques are generally designed to detect moving targets that constantly evolve by learning new vulnerabilities and not relying on known attack signatures or normal network patterns [6]. We will now discuss the related literature as follows.

Conventional Machine Learning In [11], ML algorithms, such as K-Nearest Neighbor (KNN), Random Forest (RF), DT, Logistic Regression (LR), ANN, Na¨ıve Bayes (NB), and SVM were compared in terms of their effectiveness in detecting backdoor, command, and SQL injection attacks in water storage systems. The comparative summary suggested that the RF algorithm has the best attack detection, with a recall of 0.9744; the ANN is the fifth-best algorithm, with a recall of 0.8718; and the LR is the worstperforming algorithm, with a recall of 0.4744. The authors also reported that the ANN could not detect 12.82% of the attacks and considered 0.03% of the normal samples to be attacks. In addition, LR, SVM, and KNN considered many attack samples as normal samples, and these ML algorithms are sensitive to imbalanced data. In other words, they are not suitable for attack detection in ICS. In [12], the authors presented a KNN algorithm to detect cyber-attacks on gas pipelines. To minimize the effect of using an imbalanced dataset in the algorithm, they performed oversampling on the dataset to achieve balance. Using the KNN on the balanced dataset, they reported an accuracy of 97%, a precision of 0.98, a recall of 0.92, and an f-measure of 0.95. In [13], the authors presented a Logical Analysis of Data (LAD) method to extract patterns/rules from the sensor data and use these patterns/rules to design a two-step anomaly detection system. In the first step, a system is classified as stable or unstable, and in the second one, the presence of an attack is determined. They compared the performance of the proposed LAD method with the DNN, SVM, and CNN methods. Based on these experiments, the DNN outperformed the LAD method in the precision metric; however, the LAD performed better in recall and f-measure.

Deep Learning In [14], the authors used the DNN algorithm to detect false data injection attacks in power systems. Findings of their evaluation using two datasets suggested 91.80% accuracy. In [15], the authors proposed an autoencoder-based method to detect false data injection attacks and clean them using denoising autoencoders. Their experiments showed that these methods outperformed the SVM-based method. To handle the effect of imbalanced data on the algorithm, they ignored attack data in training the autoencoder. In [16], the authors presented a technique based on Extreme Learning Machine (ELM) for attack detection in CPS. To address the imbalanced challenge of neural networks, training was conducted using only normal data. Based on these experiments, the proposed ELM-based method outperformed the SVM attack detection method. Despite promising results in both conventional ML and deep learning-based techniques, most existing ML algorithms suffer from the curse of dimensionality due to the large data volume generated in real-world ICS. Therefore, feature engineering must reduce the number of features or generate a new representation of the features to reduce computational overhead. Moreover, an imbalanced dataset of the ICS is another challenge that should be considered. Researchers have attempted to resolve this issue using oversampling/undersampling, as well as ignoring attack samples and building algorithms using normal samples.

Attack attribution seeks to answer the question of “What kind of attack was it?” and this is generally more challenging to answer in ICS than in typical IT/OT systems due to the different network structures, industry-specific protocols, and so forth [17], [18]. While there have been a small number of ML-based malware attack attributions [19], [20], designing robust and effective ML-based attack attribution for ICS and IIoT systems appears to be understudied. Thus, this paper proposes a two-stage ensemble deep learning-based attack detection and attack attribution framework for ICS. Our approach incorporates both process and physical data to solve the imbalanced data problem without subsampling or oversampling. The proposed framework utilizes an unsupervised ensemble of learned representations from normal and attack instances for attack detection. Next, using an ensemble of several one-vs-all classifiers trained on each attack attribute, it forms a two-part DNN to attribute the samples into their corresponding attack attributes.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3. SYSTEM ANALYSIS**

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering), [information systems](http://en.wikipedia.org/wiki/Information_systems) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering), is the process of creating or altering systems, and the models and [methodologies](http://en.wikipedia.org/wiki/Methodologies) that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of [software development methodologies](http://en.wikipedia.org/wiki/Software_development_methodologies).

## 3.1 EXISTING SYSTEM

The existing system for detection and attribution of cyber-attacks in IoT-enabled cyber-physical systems (CPS) involves a combination of techniques and technologies to monitor and analyze network traffic, system logs, and other sources of data to identify anomalies and suspicious activities.

One of the primary techniques used in the existing system is intrusion detection, which involves analyzing network traffic to identify patterns and behaviors that may indicate an attack. This can be done using signature-based detection, which matches traffic patterns against known attack signatures, or using behavior-based detection, which analyzes traffic for unusual behavior that may indicate an attack.

In addition to intrusion detection, the existing system may also use other techniques such as honeypots, which are decoy systems that are designed to attract attackers and provide a platform for studying their behavior. The system may also use network segmentation, which involves dividing the network into smaller segments to limit the impact of an attack and make it easier to detect.

Once an attack has been detected, the existing system may use various techniques to attribute the attack to a specific actor or group. This can involve analyzing the attack's tactics, techniques, and procedures (TTPs), as well as any other available information such as IP addresses and metadata.

Overall, the existing system for detection and attribution of cyber-attacks in IoT-enabled CPS is a complex and multifaceted process that requires the use of various techniques and technologies. As cyber-attacks become increasingly sophisticated, it will be important to continue to develop and refine these techniques to stay ahead of the threat.

## 3.2 PROPOSED SYSTEM

The proposed attack detection and attack attribution methods form a framework that can keep ICS/IIoT systems secure. This framework is proposed to address the challenge of ICS imbalanced data without ignoring the minority class or balancing the dataset. The proposed framework should be deployed on the physical layer to passively monitor the sensor data and give an alert when an attack happens. In such a case, the data is sent to the attribution model to detect the attack attribute. Finally, security experts and incident response teams can handle attacks and prevent potential damages using the proposed framework’s efficient, accurate information.

**CHAPTER 4**

**IMPLEMENTATION**

**4. IMPLEMENTATION**

# What is Python :-

Below are some facts about Python.

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

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

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

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

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

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

### Advantages of Python :-

Let’s see how Python dominates over other languages.

#### 1. Extensive Libraries

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

#### 2. Extensible

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

#### 3. Embeddable

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

#### 4. Improved Productivity

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

#### 5. IOT Opportunities

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

#### 6. Simple and Easy

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

#### 7. Readable

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

#### 8. Object-Oriented

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

#### 9. Free and Open-Source

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

#### 10. Portable

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

#### 11. Interpreted

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

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

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

#### 1. Less Coding

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

#### 2. Affordable

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

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

#### 3. Python is for Everyone

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

### **Disadvantages of Python**

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

#### 1. Speed Limitations

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

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

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

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

#### 3. Design Restrictions

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

#### 4. Underdeveloped Database Access Layers

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

#### 5. Simple

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

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

**History of Python : -**

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

**What is Machine Learning : -**

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

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

**Categories Of Machine Leaning :-**

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

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

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

## Need for Machine Learning

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

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

## Challenges in Machines Learning :-

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

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

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

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

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

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

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

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

## Applications of Machines Learning :-

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

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

# How to Start Learning Machine Learning?

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

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

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

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

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

### Step 1 – Understand the Prerequisites

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

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

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

#### (b) Learn Statistics

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

#### (c) Learn Python

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

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

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

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

#### (a) Terminologies of Machine Learning

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

#### (b) Types of Machine Learning

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

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

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

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

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

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

#### 3. Continuous Improvement

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

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

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

#### 5. Wide Applications

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

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

#### 1. Data Acquisition

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

#### 2. Time and Resources

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

#### 3. Interpretation of Results

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

#### 4. High error-susceptibility

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

**Python Development Steps : -**

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

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

**Purpose :-**

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

**Python**

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

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

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

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

**Modules Used in Project :-**

**Tensorflow**

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

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

**Numpy**

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

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

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

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

**Pandas**

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

**Matplotlib**

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

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

**Scikit – learn**

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

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

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

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

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

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

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

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

## How to Install Python on Windows and Mac :

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

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

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

### Download the Correct version into the system

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



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

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

****

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

****

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

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



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

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

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

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

### Installation of Python

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



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



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



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

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

### Verify the Python Installation

**Step 1:** Click on Start

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



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

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



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

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

### Check how the Python IDLE works

**Step 1:** Click on Start

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



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

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



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

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

**CHAPTER 5**

**SOFTWARE REQUIREMENT SPECIFICATION**

#### 5. SOFTWARE REQUIREMENT SPECIFICATION

#### 5.1 Requirements Specification:

#### Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a serve which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

#### 5.1.1 HARDWARE REQUIREMENTS:

#### The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

#### System : Pentium IV 2.4 GHz.

#### Hard Disk : 100 GB.

#### Monitor : 15 VGA Color.

#### Mouse : Logitech.

#### RAM : 1 GB.

#### 5.1.2 SOFTWARE REQUIREMENTS:

#### The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

#### Operating system : Windows XP/7/10

* Coding Language: Python 3.7

#### 5.2 FUNCTIONAL REQUIREMENTS:

The functional requirement refers to the system needs in an exceedingly computer code engineering method.

The key goal of determinant “functional requirements” in an exceedingly product style and implementation is to capture the desired behavior of a software package in terms of practicality and also the technology implementation of the business processes.

#### 5.3 NON FUNCTIONAL REQUIREMENTS

All the other requirements which do not form a part of the above specification are categorized as Non-Functional needs. A system perhaps needed to gift the user with a show of the quantity of records during info. If the quantity must be updated in real time, the system architects should make sure that the system is capable of change the displayed record count at intervals associate tolerably short interval of the quantity of records dynamic. Comfortable network information measure may additionally be a non-functional requirement of a system.

The following are the features:

* Accessibility
* Availability
* Backup
* Certification
* Compliance
* Configuration Management
* Documentation
* Disaster Recovery
* Efficiency(resource consumption for given load)
* Interoperability

**5.4 PERFORMANCE** **REQUIREMENTS**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

**5.5 Feasibility Study:**

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operation Feasibility

Economical Feasibility

**5.5.1 Technical Feasibility**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

**5.5.2 Operational Feasibility**

**User-friendly**

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

**Reliability**

The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

**Security**

The web server and database server should be protected from hacking, virus etc

**Portability**

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

**Availability**

This software will be available always.

**Maintainability**

The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

**5.5.3 Economic Feasibility**

The computerized system takes care of the present existing system’s data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

**CHAPTER 6**

**METHODOLOGY**

**6. Methodology**

**SDLC (Software Development Life Cycle) – Umbrella Model**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

**Fig no. 6.1 Umbrella model**

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Requirements Gathering Stage**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



**Fig no. 6.2 Requirements Gathering stage**

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

Feasibility study is all about identification of problems in a project, number of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.

Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



**Fig no. 6.3 Analysis stage**

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.



**Fig no. 6.4 Designing stage**

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



**Fig no. 6.5 Coding stage**

**Integration & Test Stage**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

**Fig no. 6.6 Integration and Testing Stage**

**Installation & Acceptance Test**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



**Fig no. 6.7 Installation**

**Maintenance**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category.

**CHAPTER 7**

**SYSTEM DESIGN & UML DESIGN**

## 7. System Design

## 7.1 SYSTEM ARCHITECTURE

## 

## The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It’s a serious impact on the later part, notably testing and maintenance.

## The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.

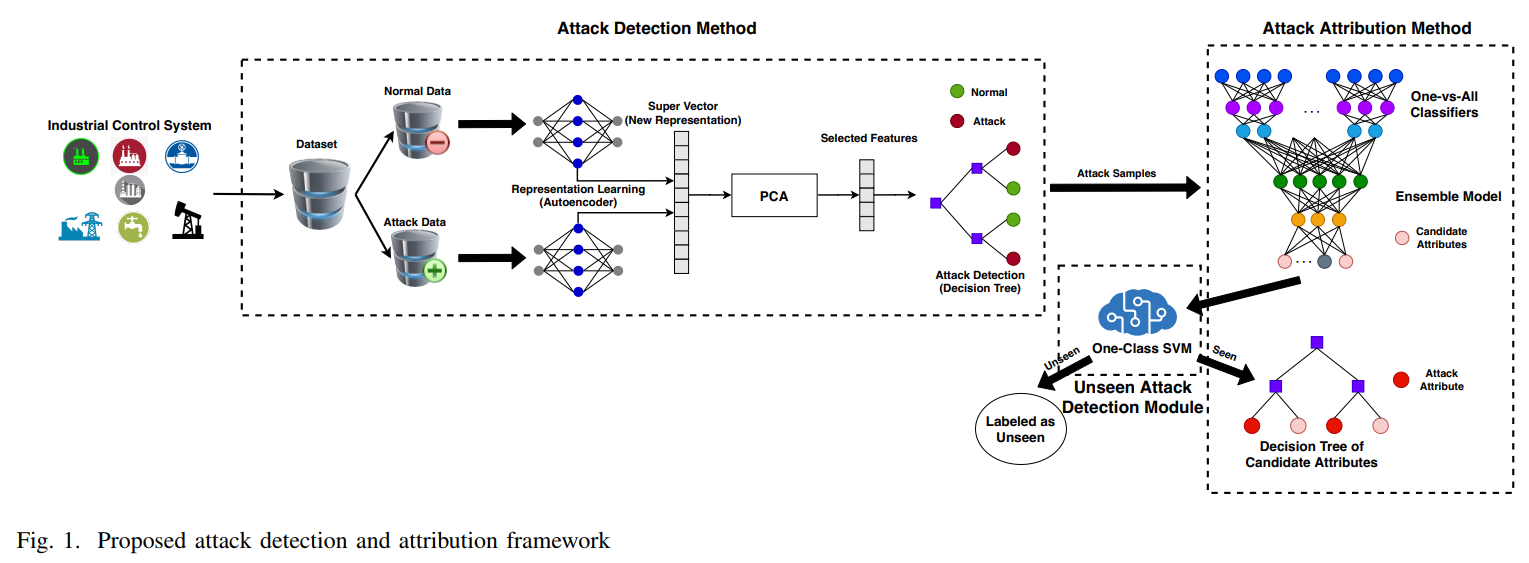
## System Design conjointly referred to as top-ranking style aims to spot the modules that ought to be within the system, the specifications of those modules, and the way them move with one another to supply the specified results.

## At the top of the system style all the main knowledge structures, file formats, output formats, and also the major modules within the system and their specifications square measure set. System design is that the method or art of process the design, components, modules, interfaces, and knowledge for a system to satisfy such as needs. Users will read it because the application of systems theory to development.

## Detailed Design, the inner logic of every of the modules laid out in system design is determined. Throughout this part, the small print of the info of a module square measure sometimes laid out in a high-level style description language that is freelance of the target language within which the software package can eventually be enforced.

## In system design the main target is on distinguishing the modules, whereas throughout careful style the main target is on planning the logic for every of the modules.

## 



## Figure 7.1: Architecture diagram

**7.3 UML DIAGRAMS**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

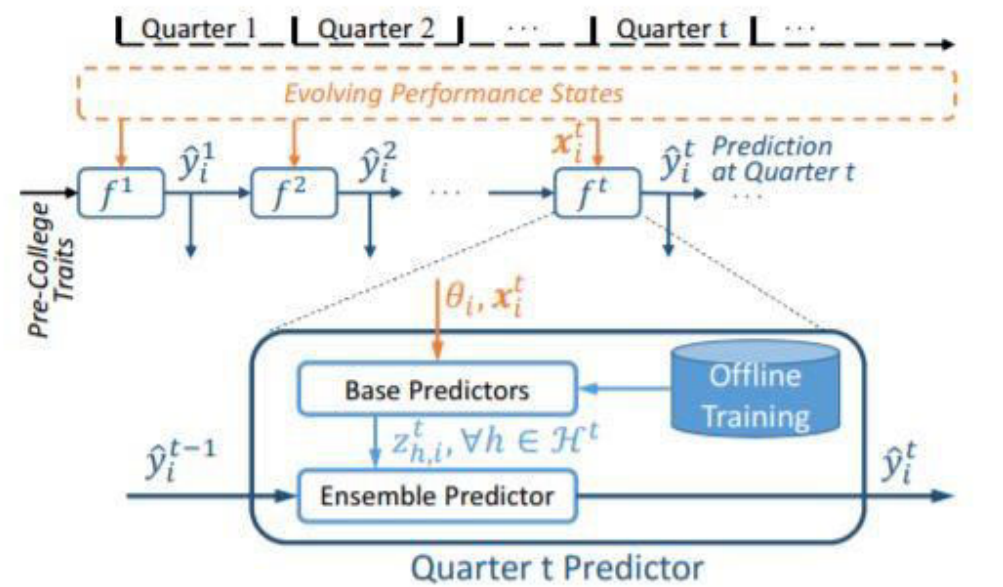
**User Model View**

This view represents the system from the user’s perspective. The analysis representation describes a usage scenario from the end-users perspective.

**Structural Model view**

In this model the data and functionality are arrived from inside the system. This model view models the static structures.

**Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

**Implementation Model View**

In this the structural and behavioral as parts of the system are represented as they are to be built.

**5.3.1 USE CASE DIAGRAM**

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

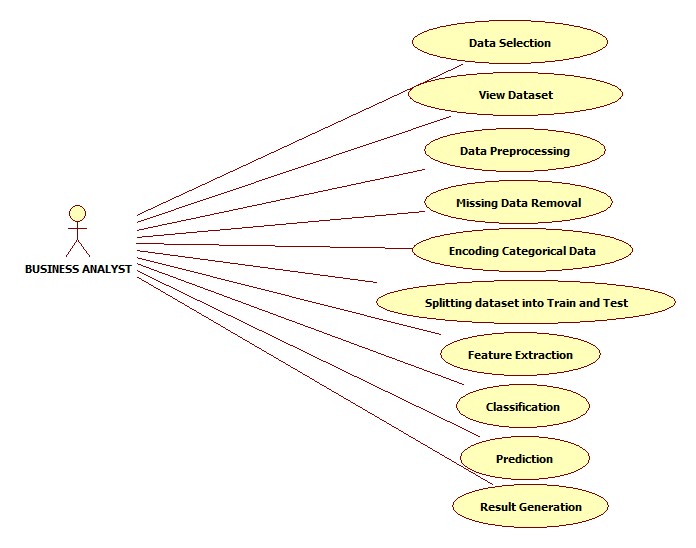
****

Figure 7.3.1 Use Case Diagram

## 5.3.2 SEQUENCEDIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

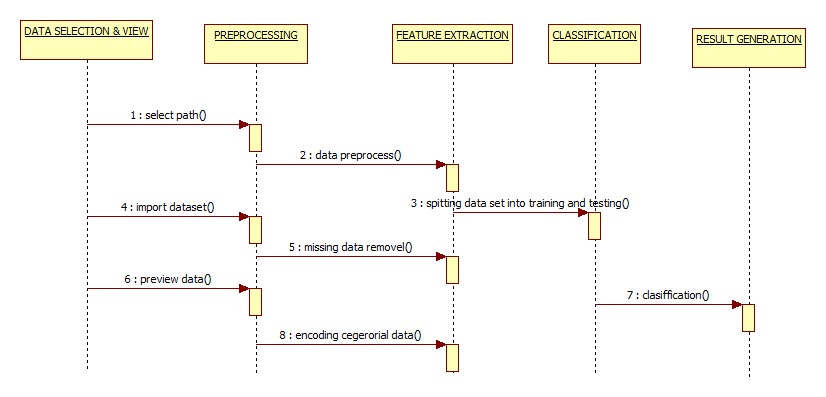


Figure 7.3.2: Sequence diagram

## 5.3.3 ACTIVITY DIAGRAM

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

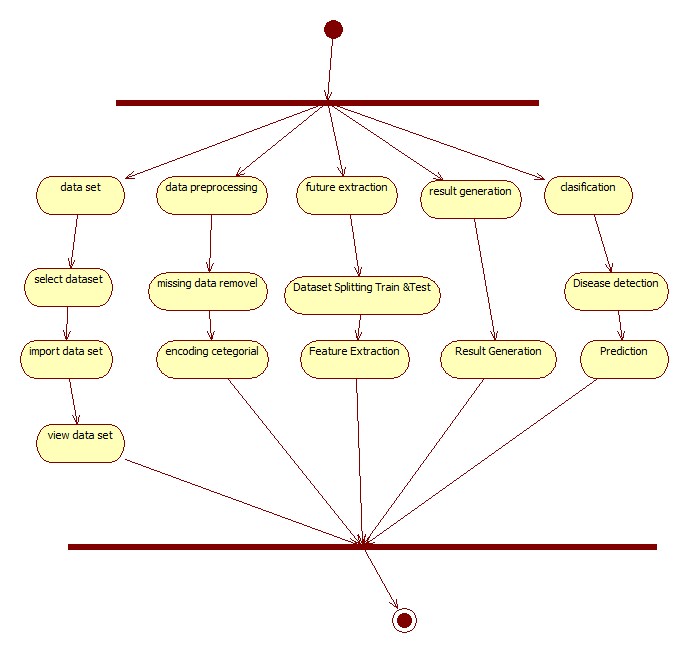


Figure 7.3.3: Activity Diagram

**CLASS DIAGRAM:**

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

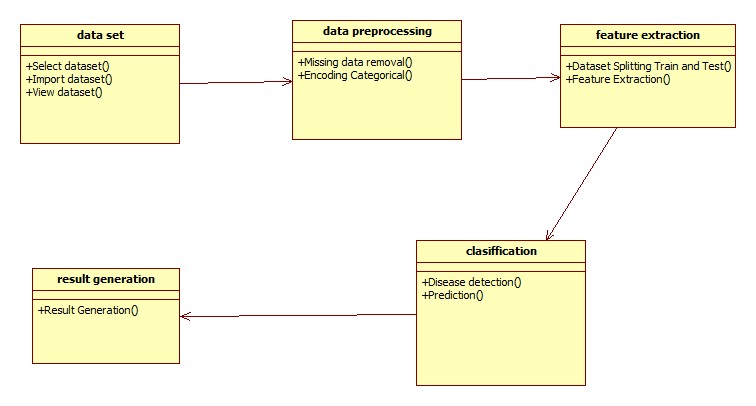


Figure 7.3.4: Class Diagram

**CHAPTER 8**

**SYSTEM TESTING**

**8. TESTING**

## Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

## 8.1 SYSTEM TESTING

## Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

## 8.2 MODULE TESTING

## To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

## 8.3 INTEGRATION TESTING

## After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system

## 8.4 ACCEPTANCE TESTING

## When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

**8.5 TEST CASES:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Upload the tasks dataset | Verify either file is loaded or not | If dataset is not uploaded | It cannot display the file loaded message | File is loaded which displays task waiting time | High | High |
| 02 | Upload patients dataset | Verify either dataset loaded or not | If dataset is not uploaded | It cannot display dataset reading process completed | It can display dataset reading process completed | low | High |
| 03 | Preprocessing | Whether preprocessing on the dataset applied or not | If not applied | It cannot  display the necessary data for further process | It can display the necessary data for further process | Medium | High |
| 04 | Prediction Random Forest | Whether  Prediction algorithm applied on the data or not | If not applied | Random tree is not generated | Random tree is generated | High | High |
| 05 | Recommendation | Whether predicted data is displayed or not | If not displayed | It cannot view prediction containing patient data | It can view prediction containing patient data | High | High |
| 06 | Noisy Records Chart | Whether the graph is displayed or not | If graph is not displayed | It does not show the variations in between clean and noisy records | It shows the variations in between clean and noisy records | Low | Medium |

TABLE 8.5.1 TESTCASES

**CHAPTER 9**

**SCREENS**

**9. SCREEN SHOTS**

Internet of Things enabled cyber physical systems such as Industrial equipment’s and operational IT to send and receive data over internet. This equipment’s will have sensors to sense equipment condition and report to centralized server using internet connection. Sometime some malicious users may attack or hack such sensors and then alter their data and this false data will be report to centralized server and false action will be taken. Due to false data many countries equipment got failed and many algorithms was developed to detect attack but all this algorithms suffers from data imbalance (one class my contains huge records (for example NORMAL records and other class like attack may contains few records which lead to imbalance problem and detection algorithms may failed to predict accurately). To deal with data imbalance existing algorithms were using OVER and UNDER sampling which will generate new records for FEWER class but this technique improve accuracy but not up to the mark.

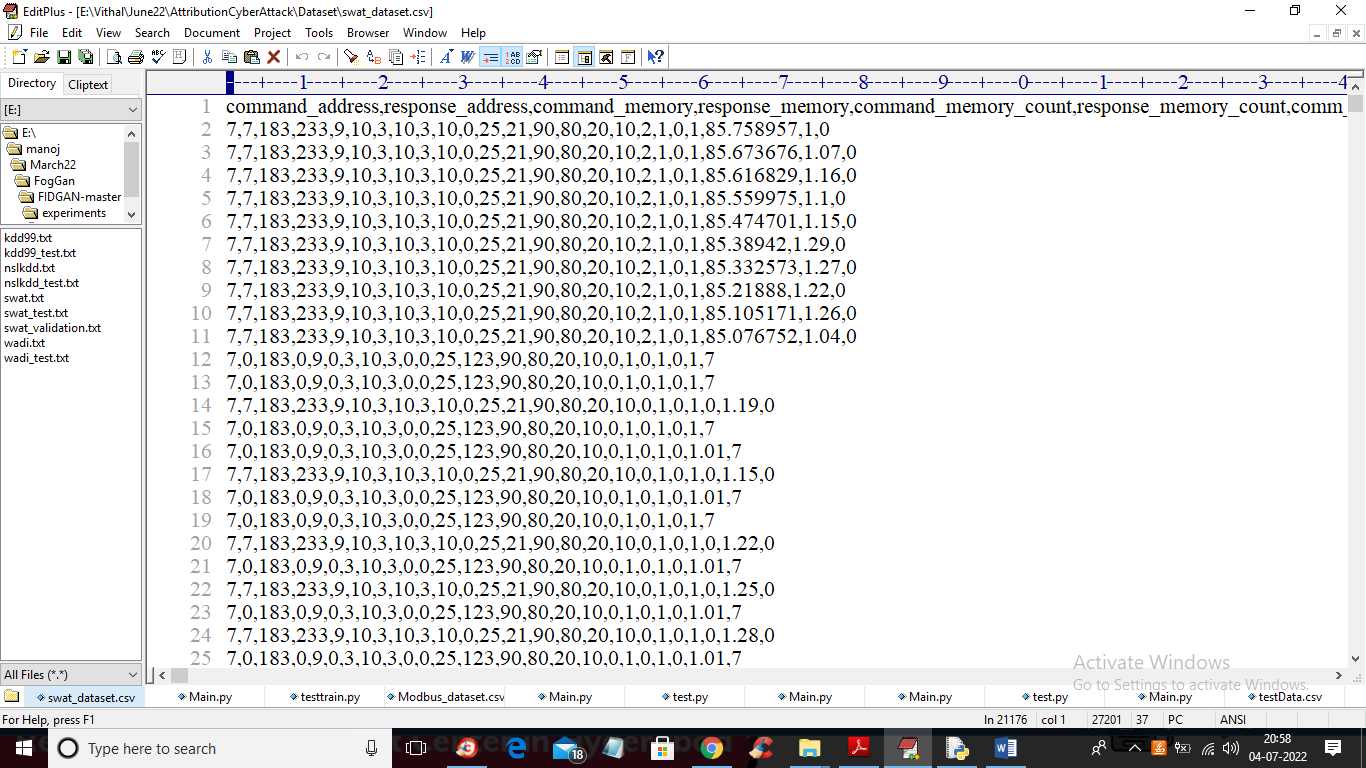
To overcome from this issue author is introducing novel technique without using any under or oversampling algorithms and this technique consists of two parts

1. Auto Encoder: auto encoder deep learning will get trained on imbalanced dataset and then extract features from it and this extracted featured will get trained with DECISION TREE algorithm to predict label for known or unknown attacks. Decision tree get trained on reduced number of features obtained from PCA (principal component analysis) algorithm.
2. Deep Neural Network (DNN): in this level DNN algorithm get trained on known and unknown attacks. If any records contains attack signature then DNN will identify attack label or class and attribute them.

To implement this project author has used SWAT (secure water treatment) and this dataset contains IOT request and response signature and associate each dataset with unique attack label and dataset contains below cyber-attack labels

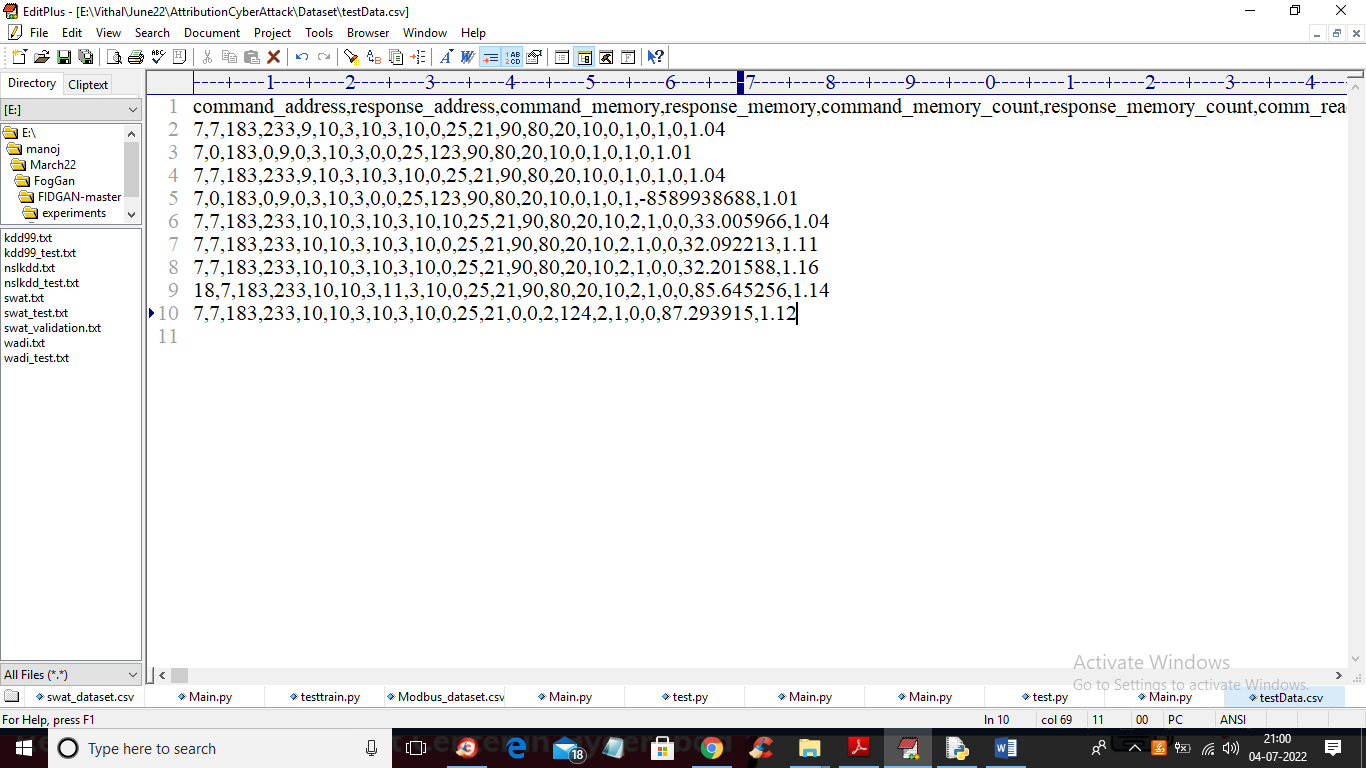
**'Normal', 'Naive Malicious Response Injection (NMRI)', 'Complex Malicious', 'Response Injection (CMRI)', 'Malicious State Command Injection (MSCI)', 'Malicious Parameter Command Injection (MPCI)', 'Malicious Function Code Injection (MFCI)', 'Denial of Service (DoS)'**

Above are the attacks found in dataset and dataset contains above labels as integer value of its index for example NORMAL label index will be 0 and continues up to 8 class labels. Below screen showing dataset details



In above dataset screen first row contains dataset column names and remaining rows contains dataset values and in last column we have attack type from label 0 to 7. We will used above dataset to train propose Auto Encoder, decision tree and DNN algorithms.

In below screen we are using NEW test data which contains only signature and there is no class label and propose algorithm will detect and attribute class labels.

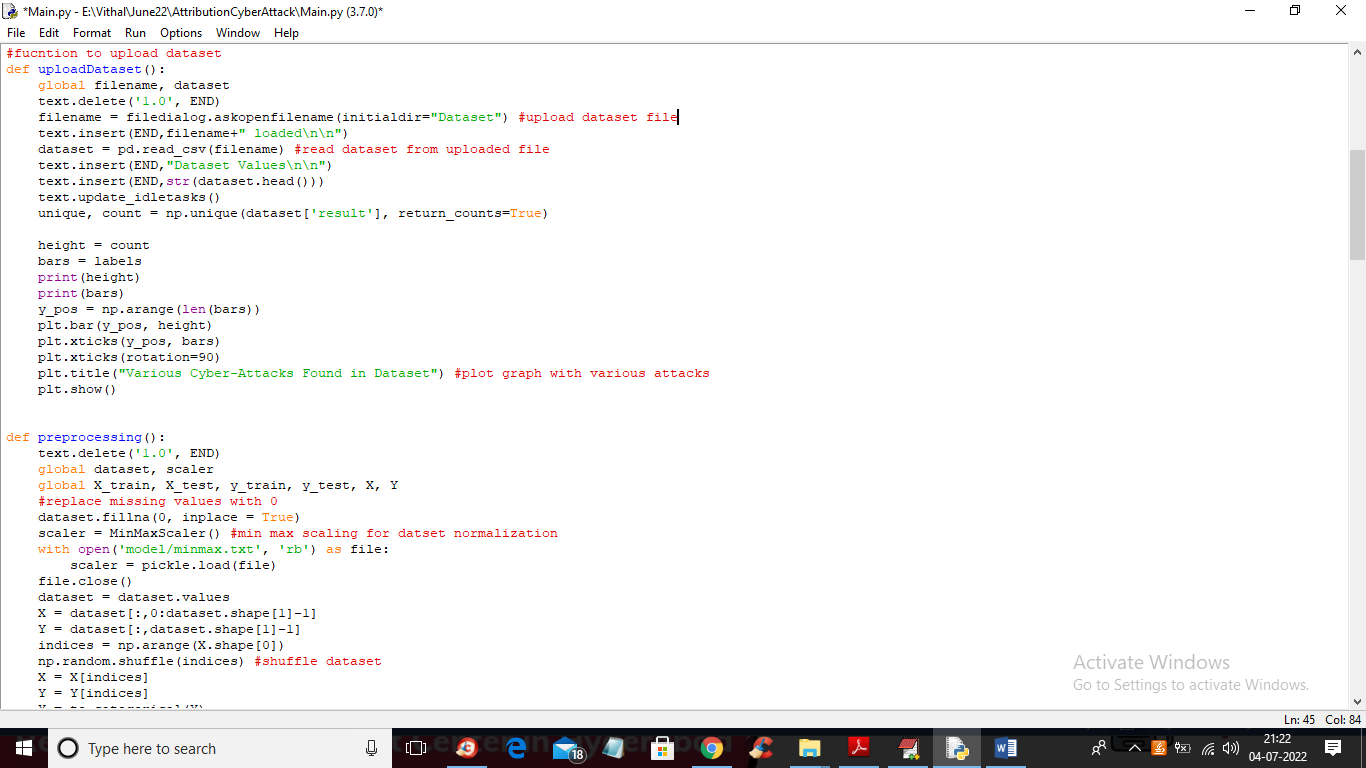


In above test data we have IOT request signature without class labels.

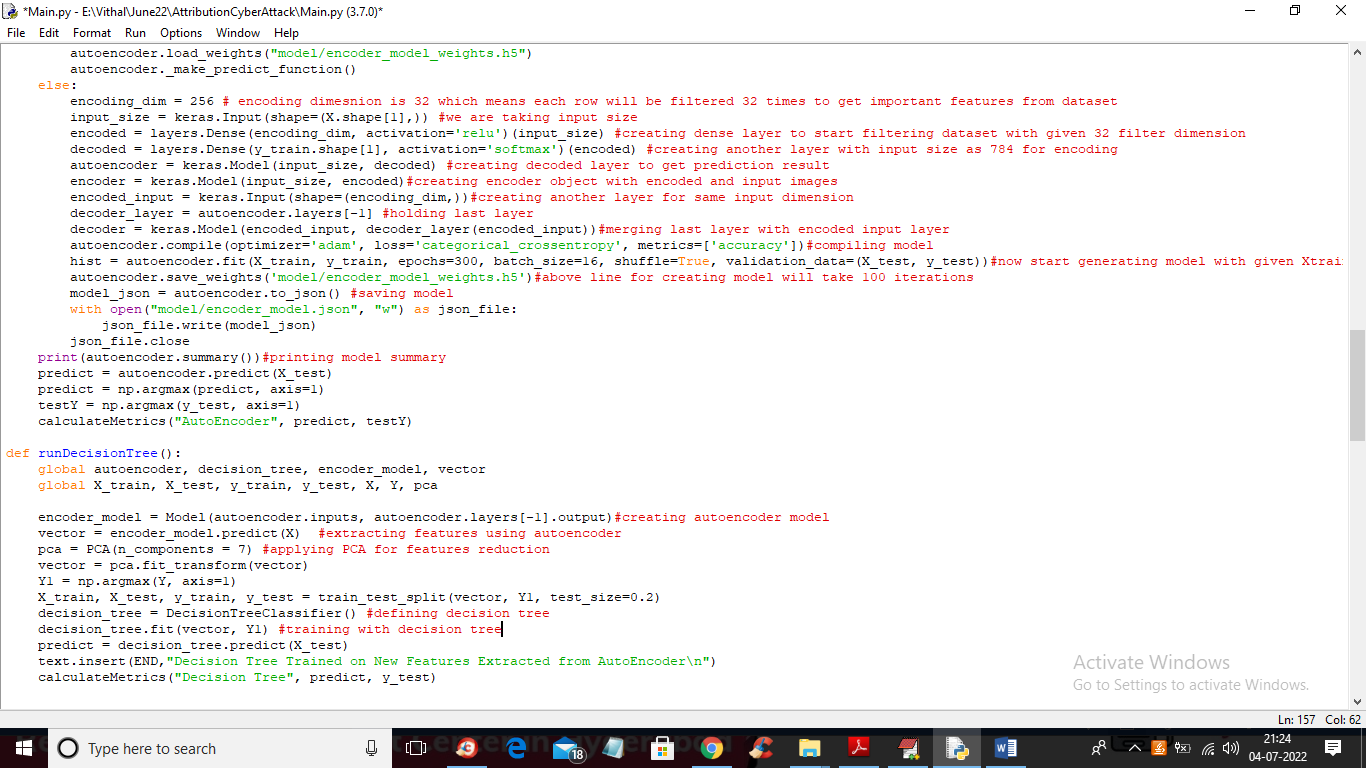
To implement this project we have designed following modules

1. Upload SWAT Water Dataset: using this module we will upload dataset to application and then read dataset and then find different attacks found in dataset
2. Preprocess Dataset: using this module we will replace all missing values with 0 and then apply MIN-MAX scaling algorithm to normalized features values and then split dataset into train and test where application used 80% dataset for training and 20% for testing
3. Run AutoEncoder Algorithm: using this module we will trained AutoEncoder deep learning algorithm and then extract features from that model.
4. Run Decision Tree with PCA: extracted features from AutoEncoder will get transform using PCA to reduce features size and then retrain with Decision tree. Decision tree will predict label for each record based on dataset signatures
5. Run DNN Algorithm: predicted decision tree label will further train with DNN (deep neural network) algorithm to detect and attribute attacks
6. Detection & Attribute Attack Type: using this module we will upload unknown or un-label TEST DATA and then DNN will predict attack type
7. Comparison Graph: using this module we will plot comparison graph between all algorithms
8. Comparison Table: using this module we will display comparison table of all algorithms which contains metrics like accuracy, precision, recall and FSCORE.

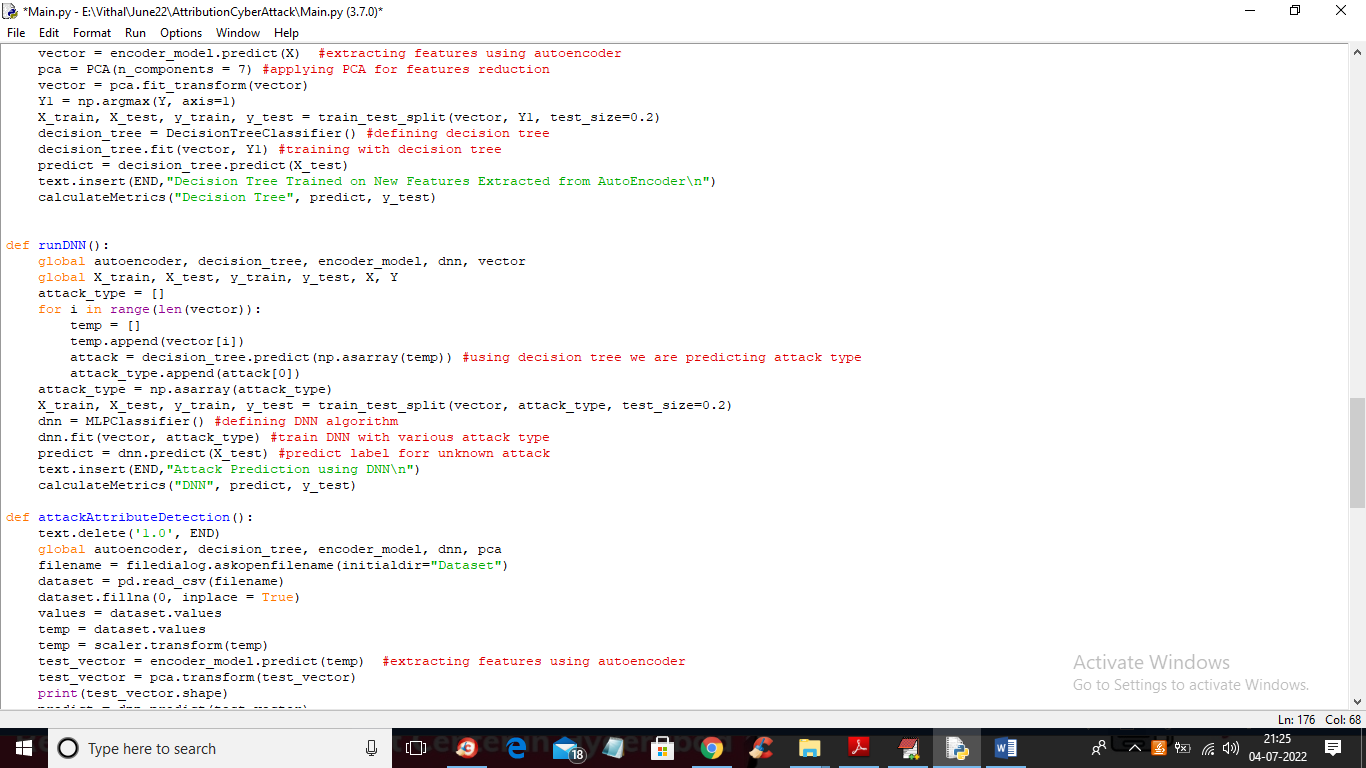
In below screen you can read red colour comments to know about algorithms implementation



In above screen read red colour comments to know about dataset loading and min-max normalization



In above screen you can see we are using AutoEncoder, PCA and decision tree to train dataset and in below screen we are using DNN algorithms to train dataset with predicted labels from Decision Tree.



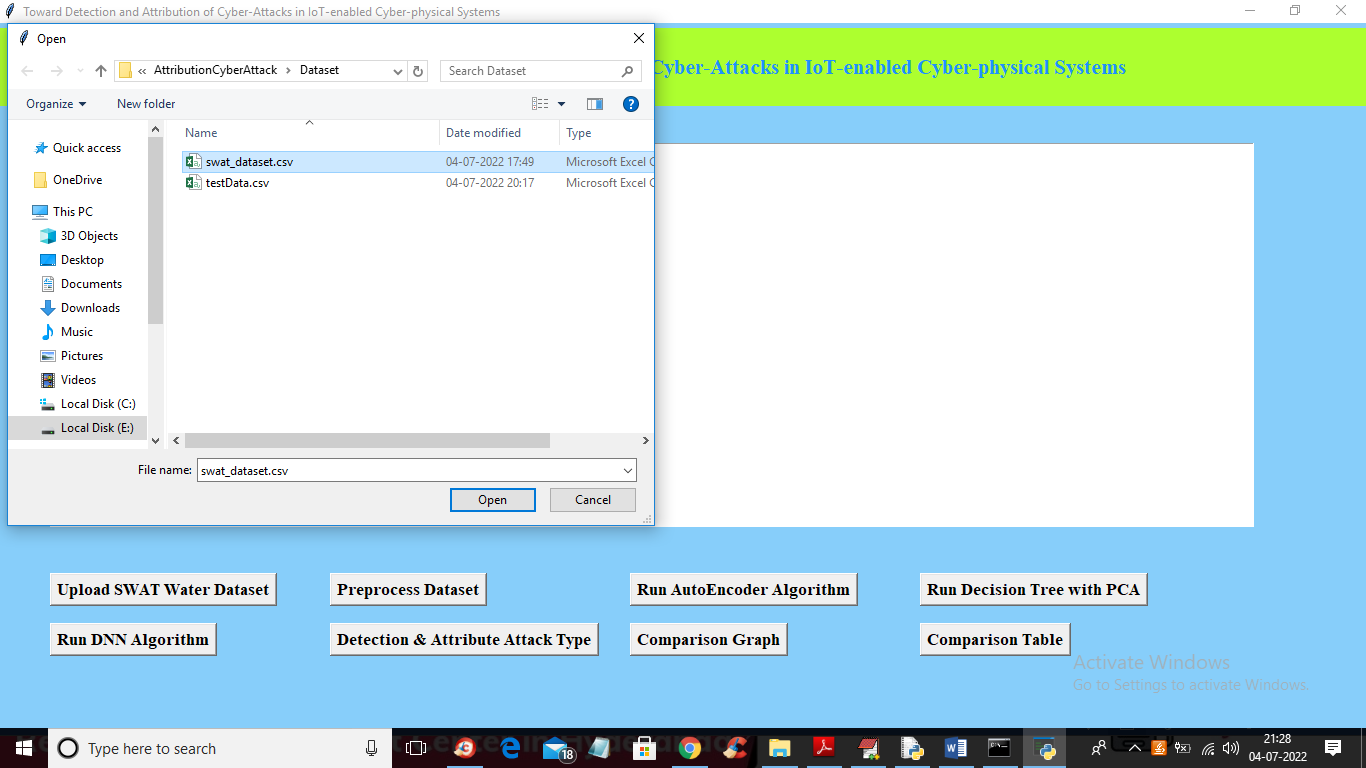
In above screen we are training dataset with DNN algorithms

SCREEN SHOTS

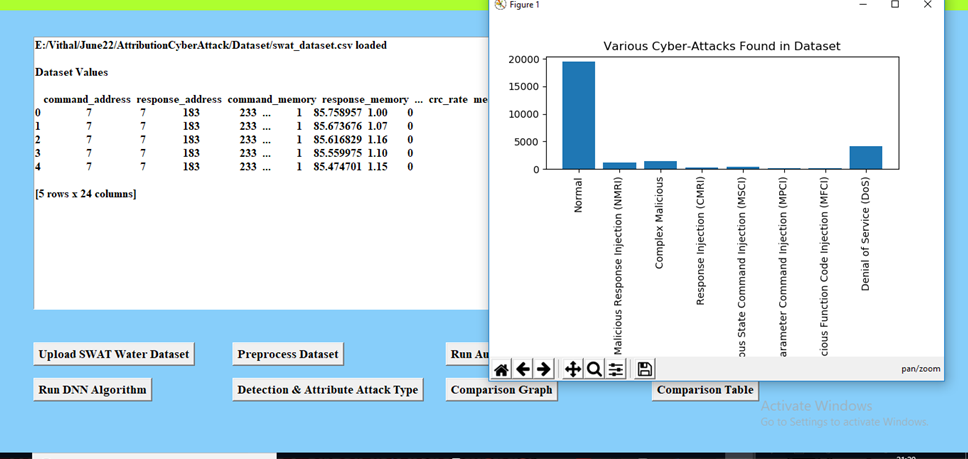
To run project double click on ‘run.bat’ file to get below screen



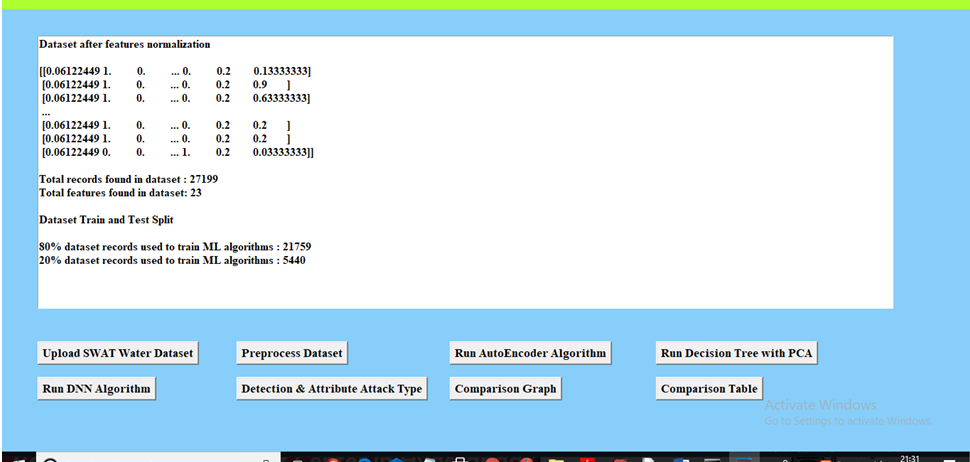
In above screen click on ‘Upload SWAT Water Dataset’ button to upload dataset to application and get below output



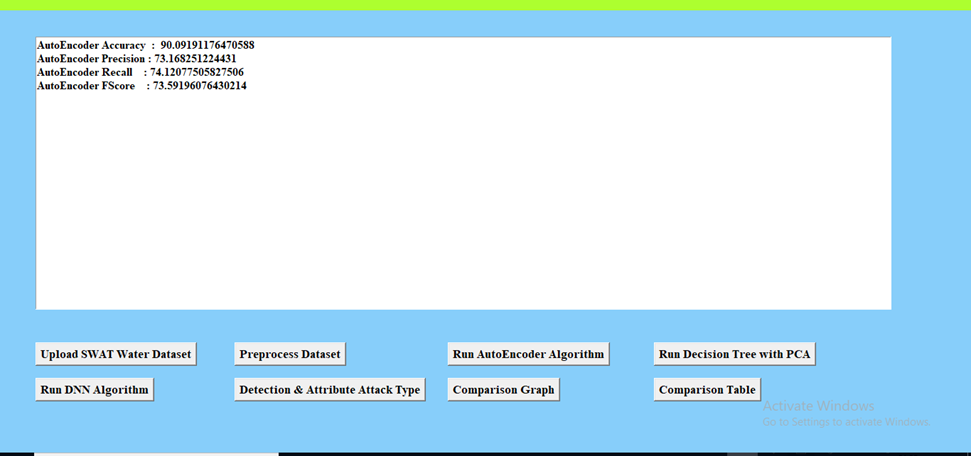
In above screen selecting and uploading SWAT dataset file and then click on ‘Open’ button to load dataset and get below output



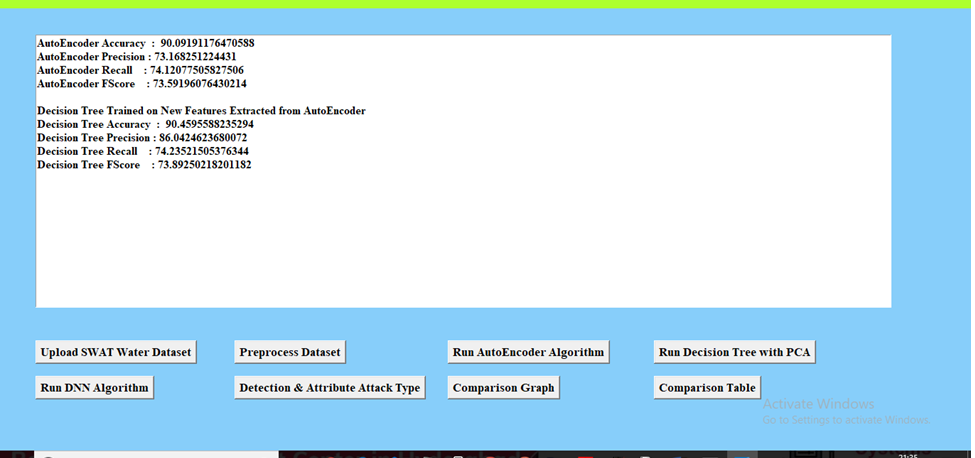
In above screen dataset loaded and in graph x-axis contains ATTACK NAME and y-axis contains count of those attacks found in dataset and we can see ‘NORMAL’ class contains so many records and other attacks contains very few records so it will raise data imbalance problem which can be solved using AutoEncoder, Decision Tree and DNN. Now close above graph and then click on ‘Preprocess Dataset’ button to remove missing values and then normalized values with MIN-MAX algorithm



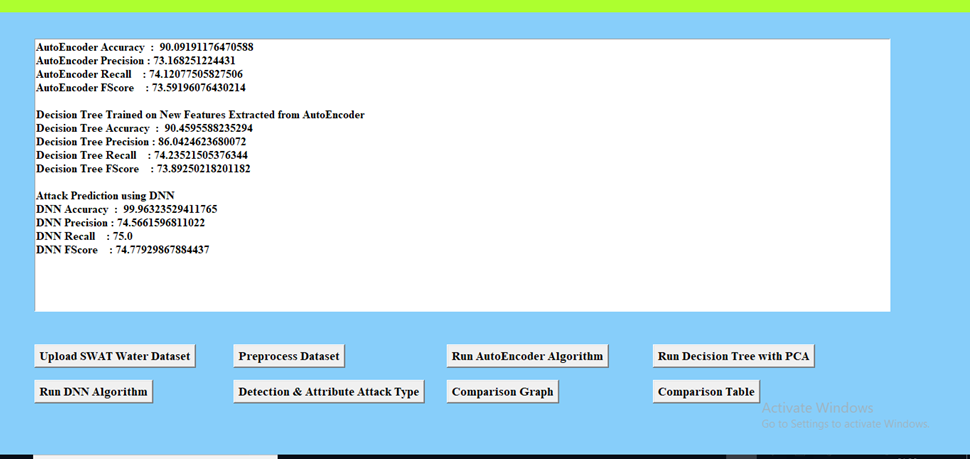
In above screen all values are normalized ( converting data between 0 and 1 called as normalization) and then we can see total records in dataset and then dataset train and test split records count also displaying. Now dataset is ready and now click on ‘Run AutoEncoder Algorithm’ button to train dataset with AutoEncoder and get below accuracy



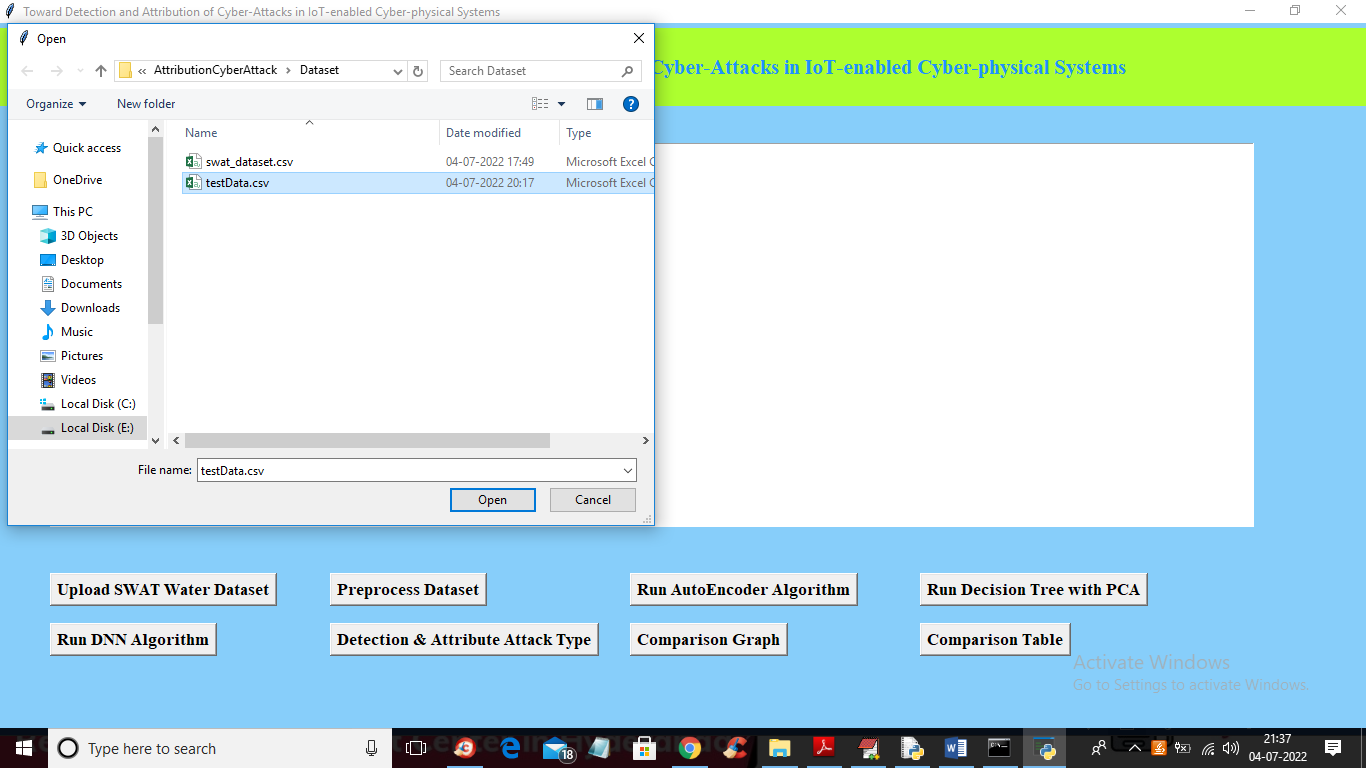
In above screen with AutoEncoder we got 90% accuracy and this accuracy can be enhance by implementing Decision Tree with PCA algorithm and now click on ‘Run Decision Tree with PCA’ button to get below output



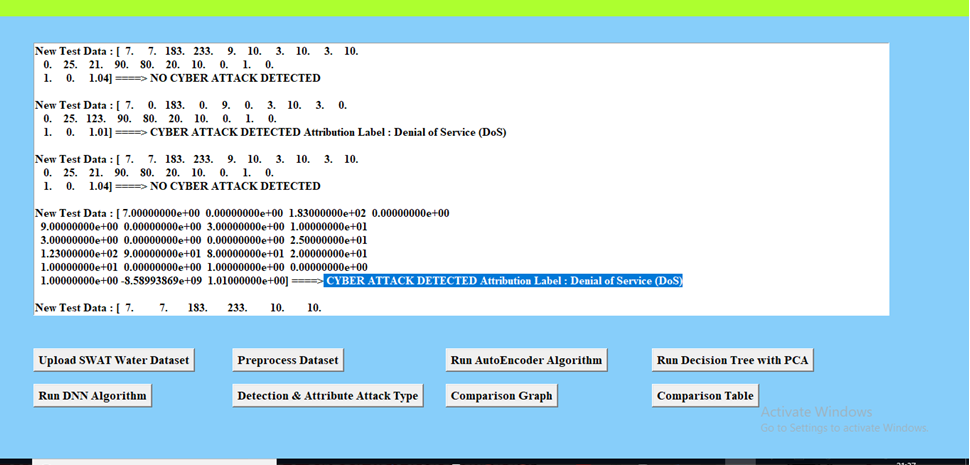
In above screen we can see with decision tree accuracy and precision value is enhanced and now click on ‘Run DNN Algorithm’ button to further enhance accuracy and get below output



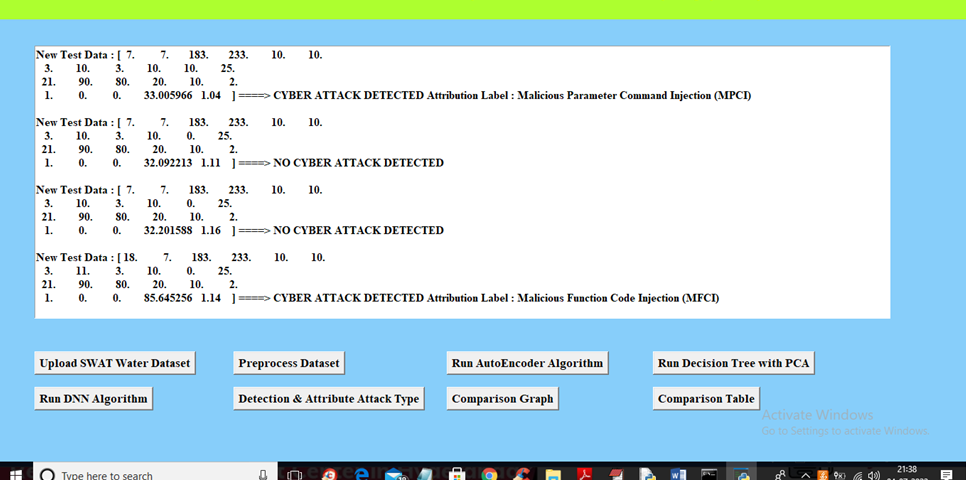
In above screen with DNN we got 99% accuracy and now click on ‘Detection & Attribute Attack Type’ button to upload test DATA and detect attack attributes



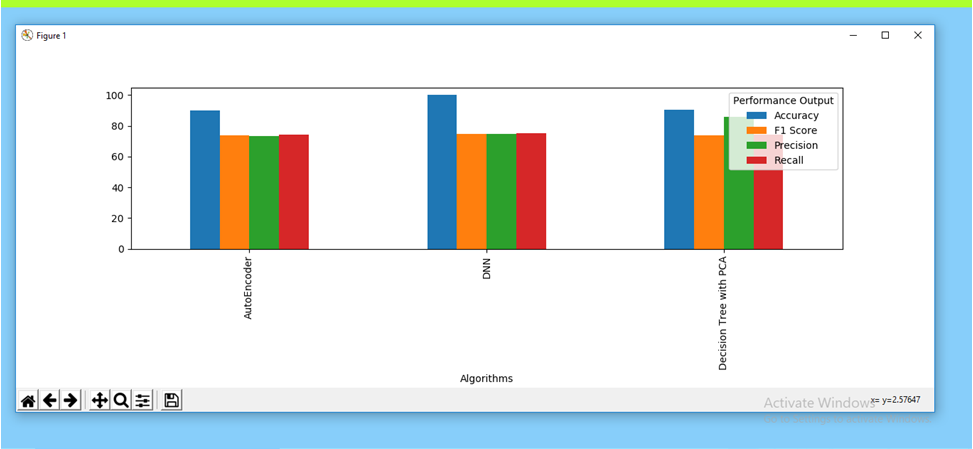
In above screen selecting and uploading ‘TEST DATA’ file and then click on ‘Open’ button to get below output



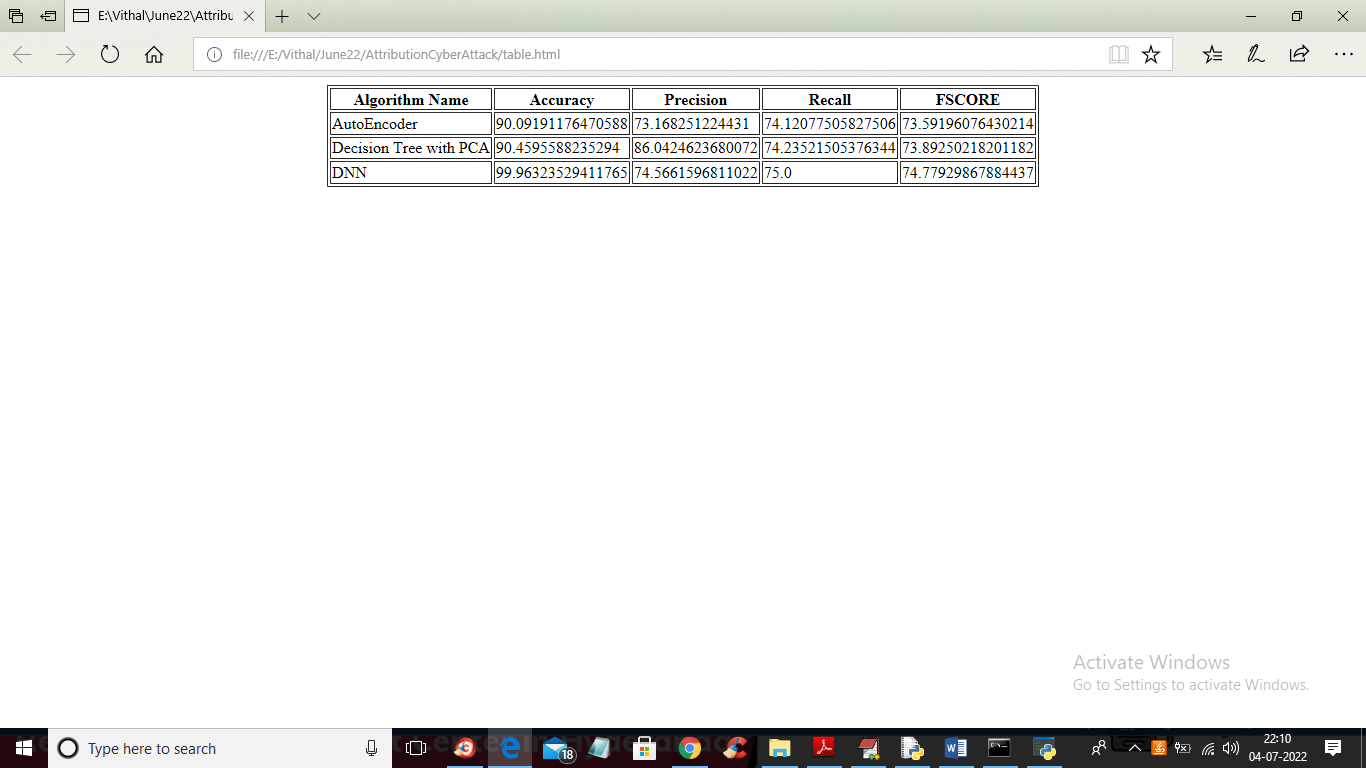
In above screen in square bracket we can see TEST data values and after arrow =🡺 symbol we can see detected ATTACK TYPE and scroll down above text area to view all detection



In above screen we can see detected various attacks and now click on ‘Comparison Graph’ button to get below graph



In above graph x-axis represents algorithms names and y-axis represents different metric values such as precision, recall, accuracy and FSCORE with different colour bars and in all algorithms DNN got high accuracy and now close above graph and then click on ‘Comparison Table’ to get below comparison table of all algorithms



In above table we can see algorithm names and its metrics values such as accuracy and precision and other.

**CHAPTER 10**

**CONCLUSION**

**10. CONCLUSION**

In this paper, we proposed a novel method for predicting students’ future performance in degree programs given their current and past performance. A latent factor model-based course clustering method was developed to discover relevant courses for constructing base predictors. An ensemble-based progressive prediction architecture was developed to incorporate students’ evolving performance into the prediction. These data-driven methods can be used in conjunction with other pedagogical methods for evaluating students’ performance and provide valuable information for academic advisors to recommend subsequent courses to students and carry out pedagogical intervention measures if necessary. Additionally, this work will also impact curriculum design in degree programs and education policy design in general. Future work includes extending the performance prediction to elective courses and using the prediction results to recommend courses to students.

**Future Enhancements:**

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

* As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
* Based on the future security issues, security can be improved using emerging technologies like single sign-on.

**REFERENCES**

[1] The White House, “Making college affordable,” https:// www:whitehouse:gov/issues/education/higher-education/ making-college-affordable, 2016.

[2] Complete College America, “Four-year myth: Making college more affordable,” http://completecollege:org/wp-content/uploads/2014/ 11/4-Year-Myth:pdf, 2014.

[3] H. Cen, K. Koedinger, and B. Junker, “Learning factors analysis–a general method for cognitive model evaluation and improvement,” in International Conference on Intelligent Tutoring Systems. Springer, 2006, pp. 164–175.

[4] M. Feng, N. Heffernan, and K. Koedinger, “Addressing the assessment challenge with an online system that tutors as it assesses,” User Modeling and User-Adapted Interaction, vol. 19, no. 3, pp. 243–266, 2009.

[5] H.-F. Yu, H.-Y. Lo, H.-P. Hsieh, J.-K. Lou, T. G. McKenzie, J.-W. Chou, P.-H. Chung, C.-H. Ho, C.-F. Chang, Y.-H. Wei et al., “Feature engineering and classifier ensemble for kdd cup 2010,” in Proceedings of the KDD Cup 2010 Workshop, 2010, pp. 1–16.

[6] Z. A. Pardos and N. T. Heffernan, “Using hmms and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset,” Journal of Machine Learning Research W & CP, 2010.

[7] Y. Meier, J. Xu, O. Atan, and M. van der Schaar, “Personalized grade prediction: A data mining approach,” in Data Mining (ICDM), 2015 IEEE International Conference on. IEEE, 2015, pp. 907–912.

[8] C. G. Brinton and M. Chiang, “Mooc performance prediction via clickstream data and social learning networks,” in 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015, pp. 2299– 2307.

[9] KDD Cup, “Educational data minding challenge,” https://pslcdatashop: web:cmu:edu/KDDCup/, 2010.

[10] Y. Jiang, R. S. Baker, L. Paquette, M. San Pedro, and N. T. Heffernan, “Learning, moment-by-moment and over the long term,” in International Conference on Artificial Intelligence in Education. Springer, 2015, pp. 654–657.

[11] C. Marquez-Vera, C. Romero, and S. Ventura, “Predicting school failure using data mining,” in Educational Data Mining 2011, 2010.

[12] Y.-h. Wang and H.-C. Liao, “Data mining for adaptive learning in a tesl-based e-learning system,” Expert Systems with Applications, vol. 38, no. 6, pp. 6480–6485, 2011.

[13] N. Thai-Nghe, L. Drumond, T. Horvath, L. Schmidt-Thieme ´ et al., “Multi-relational factorization models for predicting student performance,” in Proc. of the KDD Workshop on Knowledge Discovery in Educational Data. Citeseer, 2011.

[14] A. Toscher and M. Jahrer, “Collaborative filtering applied to educational data mining,” KDD cup, 2010.

[15] R. Bekele and W. Menzel, “A bayesian approach to predict performance of a student (bapps): A case with ethiopian students,” algorithms, vol. 22, no. 23, p. 24, 2005.

[16] N. Thai-Nghe, T. Horvath, and L. Schmidt-Thieme, “Factorization mod- ´ els for forecasting student performance,” in Educational Data Mining 2011, 2010.

[17] Y. Meier, J. Xu, O. Atan, and M. van der Schaar, “Predicting grades,” IEEE Transactions on Signal Processing, vol. 64, no. 4, pp. 959–972, Feb 2016.

[18] N. Cesa-Bianchi and G. Lugosi, Prediction, learning, and games. Cambridge university press, 2006.

[19] Y. Koren, R. Bell, C. Volinsky et al., “Matrix factorization techniques for recommender systems,” Computer, vol. 42, no. 8, pp. 30–37, 2009.

[20] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in NIPS, vol. 20, 2011, pp. 1–8.

[21] M.-C. Yuen, I. King, and K.-S. Leung, “Task recommendation in crowdsourcing systems,” in Proceedings of the First International Workshop on Crowdsourcing and Data Mining. ACM, 2012, pp. 22–26.

[22] K. Christakopoulou and A. Banerjee, “Collaborative ranking with a push at the top,” in Proceedings of the 24th International Conference on World Wide Web. ACM, 2015, pp. 205–215.

[23] Y. Xu, Z. Chen, J. Yin, Z. Wu, and T. Yao, “Learning to recommend with user generated content,” in International Conference on Web-Age Information Management. Springer, 2015, pp. 221–232.

[24] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk, “Sparse factor analysis for learning and content analytics.” Journal of Machine Learning Research, vol. 15, no. 1, pp. 1959–2008, 2014.

[25] N. Thai-Nghe, L. Drumond, A. Krohn-Grimberghe, and L. SchmidtThieme, “Recommender system for predicting student performance,” Procedia Computer Science, vol. 1, no. 2, pp. 2811–2819, 2010.

[26] J. I. Lee and E. Brunskill, “The impact on individualizing student models on necessary practice opportunities.” International Educational Data Mining Society, 2012.

[27] T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic, “Offline policy evaluation across representations with applications to educational games,” in Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, 2014, pp. 1077–1084.

[28] E. Brunskill and S. Russell, “Partially observable sequential decision making for problem selection in an intelligent tutoring system,” in Educational Data Mining 2011, 2010.

[29] J. Xu, T. Xing, and M. van der Schaar, “Personalized course sequence recommendations,” IEEE Transactions on Signal Processing, vol. 64, no. 20, pp. 5340–5352, Oct 2016.

[30] W. Hoiles and M. van der Schaar, “Bounded off-policy evaluation with missing data for course recommendation and curriculum design,” in Proceedings of The 33rd International Conference on Machine Learning, 2016, pp. 1596–1604.

[31] M. Cucuringu, C. Marshak, D. Montag, and P. Rombach, “Rank aggregation for course sequence discovery,” arXiv preprint arXiv:1603.02695, 2016.

[32] C. Tekin, J. Yoon, and M. van der Schaar, “Adaptive ensemble lear.

**Sample Code**