**TrackIt**

Project report submitted in the partial fulfillment of the requirement for the degree of

**Bachelor of Technology**

in

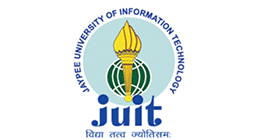
**Computer Science and Engineering**

By

**Tarun Gupta (171382)**

Under the supervision of

**Dr. Kapil Sharma, Mr. Rishabh Sharma and Mr. Bopaana.J**

to

Department of Computer Science and Engineering and Information Technology

**Jaypee University of Information and Technology Waknaghat, Solan-173234, Himachal Pradesh**

**CERTIFICATE**

**Candidate’s Declaration**

We hereby declare that the work presented in this report entitled **“TrackIt”** in partial fulfillments of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science and Engineering and Information Technology, Jaypee University of Information and Technology Waknaghat is an authentic record of my own work carried out over a period from July 2019 to November 2019 under the supervision of **Dr. Kapil Sharms,** **Mr. Rishabh Sharma (System Engineer ,Infosys Ltd.** and **Mr. Boppana J (Senior Systems Engineer, Infosys Ltd.**).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Tarun Gupta

171382

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

Dr. Kapil Sharma

Associate Professor

Department of CS & IT

Dated:

Rishabh Sharma

Systems Engineer

Infosys Ltd.

Dated:

**ACKNOWLEDGEMENT**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to Dr. Kapil Sharma, Mr. Rishabh Sharma and Mr. Bopanna J. for their guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

We would like to express our gratitude towards our respective family members for their kind co-operation and encouragement which help me in completion of this project.

Our thanks and appreciation also goes to our friends in developing the project and people who have willingly helped us out with their abilities.

Table of Contents

[CHAPTER-1 1](#_Toc58250843)

[1.1 GENERAL INTRODUCTION 1](#_Toc58250844)

[1.2 PROBLEM STATEMENT 3](#_Toc58250845)

[1.3 OBJECTIVES 4](#_Toc58250846)

[1.4 METHODOLOGY 5](#_Toc58250847)

[1.5 ORGANIZATION 7](#_Toc58250848)

[CHAPTER-2 LITERATURE SURVEY 8](#_Toc58250849)

[2.1 NETWORK SECURITY 8](#_Toc58250850)

[2.2 INTRUSION DETECTION SYSTEM 10](#_Toc58250851)

[2.3 NETWORK TRAFFIC ANOMALIES 14](#_Toc58250852)

[2.4 ANOMALY DETECTION USING MACHINE LEARNING 15](#_Toc58250853)

[2.4.1 K-nearest neighbors 15](#_Toc58250854)

[2.4.2 K-means Clustering 16](#_Toc58250855)

[2.4.3 SVM Algorithm 17](#_Toc58250856)

[2.4.4 Random Forest Classifier Algorithm 18](#_Toc58250857)

[2.4.5 Artificial Neural Network 19](#_Toc58250858)

[2.5 DATASET DESCRIPTION 20](#_Toc58250859)

[CHAPTER-3 SYSTEM DEVELOPMENT 22](#_Toc58250860)

[3.1 REQUIREMENT ANALYSIS 22](#_Toc58250861)

[3.1.1 Purpose 22](#_Toc58250862)

[3.1.2 Intended Audience and Reading Suggestions 22](#_Toc58250863)

[3.1.3 Project Scope 22](#_Toc58250864)

[3.1.4 Functional Requirements 22](#_Toc58250865)

[3.1.5 Non-Functional Requirements 23](#_Toc58250866)

[3.1.6 Design Constraints 23](#_Toc58250867)

[3.2 ARCHITECTURAL DESIGN 24](#_Toc58250868)

[3.2.1 Conceptual Design 24](#_Toc58250869)

[3.2.2 Flow Chart 26](#_Toc58250870)

[3.3 WORKING 27](#_Toc58250871)

[3.4 BENEFITS 28](#_Toc58250872)

[3.5 TOOLS AND TECHNOLOGIES 29](#_Toc58250873)

[3.6 MODULES 30](#_Toc58250874)

[3.6.1 Data Fetching 30](#_Toc58250875)

[3.6.2 Dataset Used 30](#_Toc58250876)

[3.6.3 Data Visualization 31](#_Toc58250877)

[3.6.4 Pre-processing and Selection of Features 33](#_Toc58250878)

[3.4.6 Neural Network Model 37](#_Toc58250879)

[CHAPTER-4 PERFORMANCE ANALYSIS 38](#_Toc58250880)

[4.1 PREDICTIONS MADE FOR INPUT 38](#_Toc58250881)

[4.2 TESTING FOR DOS ATTACK 39](#_Toc58250882)

[4.3 TESTING FOR R2L ATTACK 40](#_Toc58250883)

[4.4 TESTING FOR U2R ATTACK 41](#_Toc58250884)

[4.5 PROBING ATTACK 42](#_Toc58250885)

[4.6 COMPARISON 43](#_Toc58250886)

[CHAPTER-5 CONCLUSION 44](#_Toc58250887)

[5.1 SYSTEM CONCLUSION 44](#_Toc58250888)

[5.2 FUTURE SCOPE 44](#_Toc58250889)

[REFERENCES 45](#_Toc58250890)

**LIST OF ABBREVIATIONS**

* IDS – Intrusion Detection System
* NIDS – Network Intrusion Detection System
* HIDS – Host Intrusion Detection System
* KNN – K-Nearest Neighbor
* SVM – Support Vector Machine
* ANN – Artificial Neural Network
* ReLU – Rectified Linear Unit
* DNS – Domain Name Service
* DMZ – Demilitarized Zone
* DoS – Denial of Service
* U2R - User to Root Attack
* R2L - Remote to Local Attack

**LIST OF FIGURES**

[Figure 1 Network Intrusion Detection System 2](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250656)

[Figure 2 Stages of Agile Development 5](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250657)

[Figure 3 Network Topology of an Organization 9](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250658)

[Figure 4 IDS Categorization 10](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250659)

[Figure 5 Host IDS 11](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250660)

[Figure 6 Network IDS 11](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250661)

[Figure 7 Detection of Anomalies using different Techniques 13](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250662)

[Figure 8 Categorization of Network Anomalies 14](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250663)

[Figure 9 Machine Learning Techniques for Detection 15](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250664)

[Figure 10 KNN Clustering Algorithm 16](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250665)

[Figure 11 K-means clustering 16](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250666)

[Figure 12 SVM Algorithm 17](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250667)

[Figure 13 Random Forest Classifier 18](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250668)

[Figure 14 Artificial Neural Network 19](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250669)

[Figure 15 Conceptual Model for Anomaly-based NIDS 25](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250670)

[Figure 16 Flow Chart of Anomaly-based NIDS 26](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250671)

[Figure 17 Working of Anomaly-based NIDS 27](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250672)

[Figure 18 Advantages of IDS 28](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250673)

[Figure 19 Real Time Data Fetching 30](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250674)

[Figure 20 Snapshot of used Dataset 30](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250675)

[Figure 21 Dataset Visualization 31](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250676)

[Figure 22 Histogram showing classes of different attacks 31](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250677)

[Figure 23 Sanpshot to specify different categories 32](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250678)

[Figure 24 Feature Label categories 32](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250679)

[Figure 25 Distribution of Categories 32](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250680)

[Figure 26 Categorical Representation 33](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250681)

[Figure 27 Categorical Values 33](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250682)

[Figure 28 33](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250683)

[Figure 29 DoS Attack Features 34](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250684)

[Figure 30 R2L Attack 34](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250685)

[Figure 31 U2R Attack 34](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250686)

[Figure 32 U2R Features 35](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250687)

[Figure 33 Feature for all the categories of attack 35](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250688)

[Figure 34 DoS Features sorted by their rank using Random Forest Classifier 36](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250689)

[Figure 35 36](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250690)

[Figure 36 Neural Network Model 37](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250691)

[Figure 37 Trainable and Non-trainable Parameters 37](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250692)

[Figure 38 Predictions 38](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250693)

[Figure 39 Accuracy for DOS Attack 39](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250694)

[Figure 40 Accuracy for R2L Attack 40](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250695)

[Figure 41 Accuracy for U2R Attack 41](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250696)

[Figure 42 Accuracy for Probing Attack 42](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250697)

[Figure 43 Comparison Graph 43](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250698)

**LIST OF TABLES**

[Table 1 Basic features of individual TCP connections 20](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250640)

[Table 2  Traffic features computed using a two-second time window 21](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250641)

[Table 3 Content features within a connection suggested by domain knowledge 21](file:///C:\Users\HP!!\Documents\IDS%20Report-11.docx#_Toc58250642)

[Table 4 Comparison table for Performance Analysis 43](#_Toc58250643)

**ABSTRACT**

The pandemic's online mode has altered the structured communication's entire dynamics. Trainings and Internships going on, in online mode is a challenge which India is facing in the current situation. In a country with one of the largest populations, it is a very tough task to track all the information and daily to daily updates of the trainees, interns and the mentor. So, basically a training management system is a portal that is explicitly designed to monitor employee and trainee’s information so that the employees and trainees of an organization are aware of each other's details and day-to-day activities.

Trackit is a dashboard application that aims to provide all the training related activities and information that is required for the ETA. Trackit mainly includes a trainee portal where they can get all the data that is specific to their training. Along with that it also includes a batch owner or the mentor dashboard which allows them to get all the data of the trainees, such as their personal details, their performance graph and day-to-day tasks completed. It also provides a chat engine for the trainees or interns and the mentors so that they can communicate with each other on a regular basis which can be a very important feature during the pandemic.

The tech stack that is used for Trackit is Java Spring Boot for backend, Angular for frontend and Mongodb as the database. The tech stack used is based on the features that need to be implemented. The three main functionalities are already being implemented and other future works are being decided.

# CHAPTER-1

## GENERAL INTRODUCTION

A training management system is a portal specifically designed to track details of the employee and trainee with which the employees and trainees of a company would be able to know about the details and daily to daily activities of each other.

Most of the Training Management Systems also contains e-learning data that trainees can use for their learning purpose. The main motive of this is to get all the learning content at a single place in a systematic manner. However, most of the e-learning portals and Training Management Systems do not provide any way to know the details of the trainees and the batch owners. Also, they do not provide any way of communication between them.

In the recent times, during the pandemic, we all have deeply observed and realized that interaction between people working together has been nullified due to work from home. This is a major issue and might be a case even after the pandemic situation. Along with that, for most of the organizations it has been a bit challenging for the mentors and trainees to know the details of each other in one single medium.

The idea of Trackit is inspired by a typical Training Management System and adding to it a perspective of necessary interaction between the mentors and trainees in a professional manner. Trackit is a dashboard application that aims to provide all the training related activities and information that is required for the ETA.

Trackit mainly includes a trainee portal where they can get all the data that is specific to their training. Along with that it also includes a batch owner or the mentor dashboard which allows them to get all the data of the trainees, such as their personal details, their performance graph and day-to-day tasks completed. It also provides a chat engine for the trainees or interns and the mentors so that they can communicate with each other on a regular basis which can be a very important feature during the pandemic.

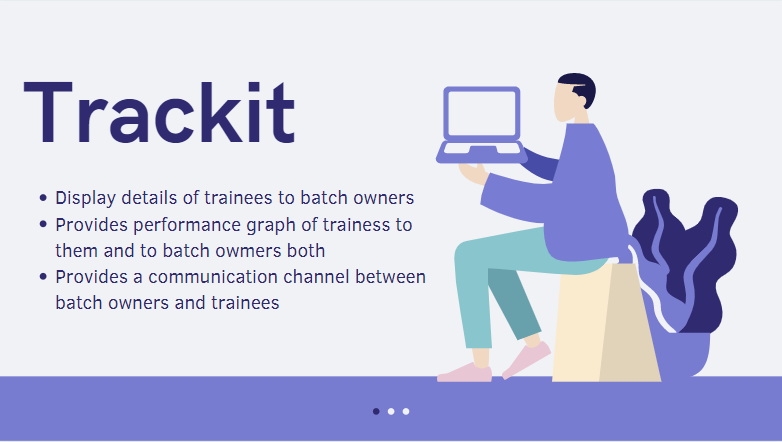


Figure 1 Diagrammatic overview of Trackit

Advantages of Trackit are as follows-

* Provides secure Batch Owners Login.
* Provides secure trainees Login.
* Provides personal details of all the trainees to the Batch Owners (a list is present in the batch owners portal through which a batch owner can select any of the trainee and view their details and their performance graph).
* A trainee can view his or her details in the portal and can edit them. They can also view their performance graph in the provided dashboard.
* Trackit provides a chat bot as a medium of communication between batch owners and trainees.

## PROBLEM STATEMENT

The pandemic situation across the world has created a lot of hassle. Most of the companies and organizations have started work from home due to which employees are away from the working environment. This has resulted in different kind of complexities in the work culture. For example, the details of trainees and their complete information needs to be present at a single platform, so that it is not an issue for their managers or mentors to track their performance and details. Many companies have designed their internal products as per their requirement. However, still there are many companies and organisations who are facing this kind of situation after the recruitments.

Another problem is that due to work from home interactions among the new trainees and their mentors or their managers has been nullified. People around the world have either started different ways of communication among each other or they have distant themselves from the working environment. This is a major issue for both the employees and the organization. A company or an organization is because of its employees who works as a team to achieve organization’s goals. The necessary work from home environment can lead to lack of team bonding and thus interest of an individual from the work. This will ultimately lead to loss of the company or the organization.

Trackit is inspired from a typical training management system with addition of a chat bot that is used as a medium of communication between the batch owners and the trainees. Trackit solves the above stated problems in such a way that it provides a common platform for getting details of both batch owners and trainees, performance of the trainees and allowing them to interact among each other so that they can develop the spirit of team bonding from the very start.

## OBJECTIVES

The main objective of this project is to ease the process of online training. Trackit aims to achieve the goal of getting all the management that is required for different kind of trainings in a single platform. the main aim is to design two portals, one for the batch owners and other for the trainees. The batch owner’s portal is designed for the managers or the trainers. This portal aims to display different kind of information related to the trainees. as of now, it aims to display a list of all the trainees from where a single trainee can be selected and batch owner can view his/ her personal details , marks earned in different phases of the training and a bar graph that displays the overall performance of the selected trainee. On the other hand, the trainee’s portal is designed so that all the trainees can have their personalized dashboard and can view training related material. Along with that it also allows a trainee to view the details of its batch owner. Another objective of this application is to solve the problem of no interaction happening among the co-workers. this problem is solved by Trackit since it comprises of a chat engine which allows both the trainee and the batch owner to communicate with each other. With this feature a batch owner can Have a one on one interaction with the trainee and can analyse the performance of the trainee. this can be a much professional and feasible way of communicating at an organization's level.

## METHODOLOGY

The project is based on the agile methodology for development. Agile model is an incremental and iterative model of development which means that each build of the software would be independent, but the final delivered product will be incremented with that build. In this model, at every iteration different teams of planning, requirement analysis, design, coding, unit testing and acceptance testing works simultaneously. Figure 2 defines the various stages of the Agile Development.

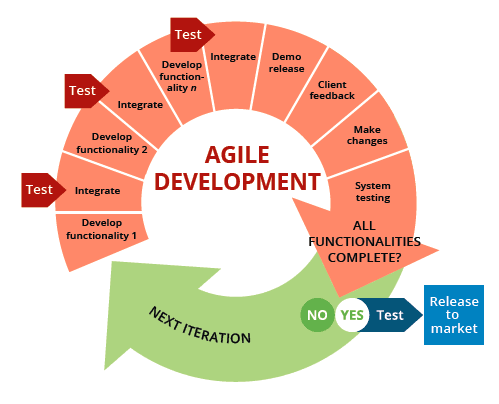


Figure 2 Stages of Agile Development

The motive of developing our project based on this model is that TrackIt Application requires frequent updates due to updates in the features and its new behaviors. Using this model will lead to develop updates and their testing independently and then integrating them with the existing product to test the final product. Along with that it is comparatively easy in agile development than in waterfall approach to fix the bugs or to update the system as per the customer’s requirement.

## ORGANIZATION

The Project Report follows the detailed analysis of how the project work is completed. The complete report is organized into the following chapters:

CHAPTER 1: Chapter 1 of this report gives a brief introduction about what is TrackIt Application and why we need to use it. The chapter describes about the use of Application and the benefit it gives to the users.

CHAPTER 2: This chapter gives details about the work done previously in this field. It briefs about the technologies and algorithms already used in this field with their significance.

CHAPTER 3: This chapter gives the details of the process followed to develop the system. It contains the conceptual models of the system and algorithms used in the system.

CHAPTER 4: In this chapter the result calculated by the proposed algorithm is presented and described. Performance is analyzed with the result.

CHAPTER 5: The conclusion on the basis of results and performance of the system is made in this chapter. This chapter also defines the future scope of this project. At last the report contains the references that are used in the project.

# CHAPTER-2 LITERATURE SURVEY

Work from home has altered the dynamics of structured contact as a result of the pandemic. Trainings and Internships going on in online mode is a challenge which India is facing in the current situation. In a country with one of the largest populations, it is a very tough task to track all the information and daily to daily updates of the trainees, interns and the mentor.

## 2.1 Spring Framework

The Spring Framework is a Java platform application framework and inversion of control container. The Spring Framework is a Java-based enterprise application configuration model. It comes with around 20 modules that can be used depending on the needs of the application.

### 2.1.1 Container Core

The Core Container is made up of four modules: Core, which provides IoC and Dependency Injection, Beans, which provides BeanFactory, Context, which provides the ApplicationContext interface, and SpEL, which provides an expression language for querying and manipulating an object graph at runtime.

### 2.1.2 Data Access/Integration

The Data Access/Integration layer consists of the JDBC, ORM, OXM, JMS and Transaction modules.

The JDBC module provides a JDBC-abstraction layer that removes the need to do tedious JDBC coding and parsing of database-vendor specific error codes.

JPA, JDO, Hibernate, and iBatis are just a few of the common object-relational mapping APIs supported by the ORM module. All of those O/R-mappers can be used in conjunction with all of the other Spring features, such as the easy declarative transaction management function described earlier, using the ORM bundle.

The OXM module acts as a layer of abstraction for a variety of Object/XML mapping implementations. JAXB, Castor, XMLBeans, JiBX, and XStream are among the technologies supported.

Spring's support for the Java Messaging Service is provided by the JMS module. It has capabilities for both sending and receiving messages.

Not only for classes implementing special interfaces, but for all your POJOs, the Transaction module offers a way to do programmatic as well as declarative transaction management (plain old Java objects).

### 2.1.3 WEB

The Web layer consists of Web module that provides basic web-oriented integration features and the initialization of the IoC container using servlet listeners and a web-oriented application context, the Web-MVC module that contains Spring's model-view-controller (MVC) implementation, the Web-Socket module provides support for two-way communication between client and server in web applications and Web-Portlet module which provides the MVC implementation to be used in a portlet environment and mirrors the functionality of Web-Servlet module.

## 2.2 Spring MVC

The Spring MVC Framework is a model-view-controller framework for creating loosely coupled Java web applications. It includes input logic, business logic, and user interface logic, all of which are interdependent to some extent.

Some of the main features of Spring MVC are-

* The Model encapsulates the application data and will, in most cases, be made up of POJO.
* The View renders the model data and produces HTML output for the client to view.
* The front-end development technologies used in this application are JavaScript, HTML, CSS, and jQuery.
* The Controller is in charge of processing user requests and directing the flow of application logic between the view and the model.

With the support of DispatcherServlet, Spring MVC offers an elegant solution for using MVC in the spring system. DispatcherServlet is a class that receives requests and routes them to the appropriate resources, such as controllers, templates, and views.

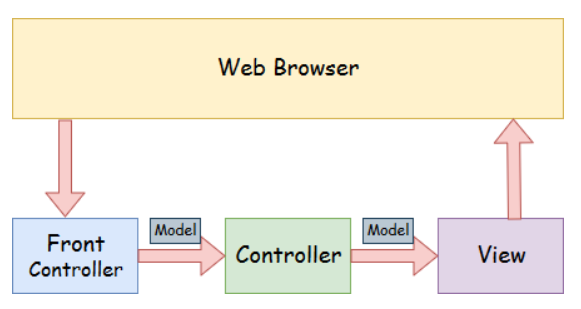


Figure 3 Spring Web Model-View Controller

Advantages of Spring MVC Framework are as follows –

* Each function in Spring MVC is separated, with specialized objects fulfilling the model object, controller, command object, view resolver, DispatcherServlet, validator, and so on.
* It develops and deploys the application using a light-weight servlet container.
* It includes simple referencing across contexts, such as from site controllers to business objects and validators, and it provides a comprehensive configuration for both system and application classes.
* The Spring MVC facilitates fast and parallel development.
* Instead of creating new objects, it allows us to use the existing business objects.
* In Spring, generally we create JavaBeans classes that enable you to inject test data using the setter methods.
* It provides the specific annotations that easily redirect the page.

## 2.3 Spring Micro-services

Micro Services allow large systems to be designed from a collection of interconnected components. It implements loosely coupled processes rather than loosely coupled components at the process level, as Spring has always done at the component level.

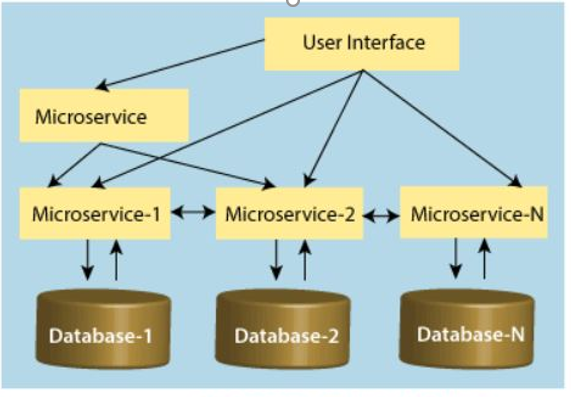


Figure 4 Microservices Architecture

The above figure shows a Microservice Architecture. A Service Oriented Architecture is a Microservice Architecture. There are a lot of microservices in the microservice architecture. It creates a large service by integrating all of the microservices. All of the services in a microservice architecture interact with one another.

## 2.4 ANGULAR

## 2.5 MONGODB

# CHAPTER-3 SYSTEM DEVELOPMENT

## 3.1 REQUIREMENT ANALYSIS

### 3.1.1 Purpose

The purpose of this project is to build an application that provides personalized dashboard for batch owners and trainees in order to maintain training related data in a single portal. The dashboard aims to provide details of both trainees and batch owners. It also aims to design a chart engine so that the problem of communication between batch owners and trainees can be resolved.

### 3.1.2 Intended Audience and Reading Suggestions

This project is mainly inspired from training management systems along with some extra functionalities, and it is mainly targeted for the organisations Who provides trainings and internships to the newly hired ones. This has been implemented under the guidance of college professors and one of the managers at Infosys Pvt. Ltd.

### 3.1.3 Project Scope

The purpose of this project is to build an application that provides personalized dashboard for batch owners and trainees in order to maintain training related data in a single portal. The dashboard aims to provide details of both trainees and batch owners. It also aims to design a chart engine so that the problem of communication between batch owners and trainees can be resolved. The project in now is developed and tested by using Spring Boot for backend, Angular for frontend and mongodb as the database.

### 3.1.4 Functional Requirements

The three major high-level requirements for the developed application are as follows-

* Secure Login – Login is the simplest functionality that every portal requires. For this application there should be two login flows, one for the batch owner another for the training. With successful login a trainee or a batch owner can view their dashboard respectively. The login must be secure and should have a session timeout.
* Personalized Dashboard – There should be a personalized dashboard for the trainee and the batch owner. This dashboard must contains all the necessary details and can only be accessed after the successful login.
* Trainee Details – All the trainee details should be present in the dashboard of the respective batch owner. It should be present in a list format so that whenever a batch owner selects any of the trainee, it can view the details of the trainee. The details must contain the personal information of the trainee and the performance of the trainee.
* Chat Engine – A chat engine needs to be implemented so that the batch owners and the trainees can communicate with each other. This chat engine should be restricted to batch owners and a trainee chat only.

### 3.1.5 Non-Functional Requirements

There are many non-functional requirements which are traded off between each another e.g. increased performance often comes at an increased total cost of ownership. Non-functional requirements for this application include-

* Scalability – The system should be scalable in terms of the following factors- the number of data feeds that the system can process at a time; in this case the number of trainees and batch owners that the portal can handle at a time.
* Performance - Performance is computed by the comparison the amount of work done to the time and resources that are required to do that work. The system should have quick response times i.e., it should response immediately in case of a request.
* Modifiability - Modifiability is the ease factor with which the system can be updated with new changes. The application should have easy process for updating new behaviours and data processing because in this field, updates and introduction of new features is quite common.
* Reliability – The application should be accurate and dependable so that it could get more correct outputs for the performance details of the trainees, for example, performance graph.
* Fault tolerance – The system should be able to tolerate the fault if occurred. This is similar to reliability, but a system must be reliable to use even after a fault has occurred.

### 3.1.6 Design Constraints

The software language used to implement our system is Python. Python is slower than C# and C++, but is widely used in quant trading because it is a high-level language. Along with that it has in-built libraries which make it a better performing and easy to use programming language. Research and prototyping are carried out much more easily due to Python’s high-performing libraries.

The development tools that we have used in the project are as follows-

* NumPy or Numerical Python is in-built Python package which helps in implementing large multi-dimensional arrays.
* Pandas is a vast Python library that is used for the purpose of data processing.
* Matplotlib is a Python library used for plotting 2D structures like graphs, charts, histogram, scatter plots etc.
* Keras is a Python library for implementing deep learning models.
* Tensorflow is a Python library used for numerical computations and creating deep learning models.

## 3.2 ARCHITECTURAL DESIGN

The architectural designs are the designs which are made to get a complete picture of the system to be developed in order to analyze the full process before actual development. The two designs we have used are Conceptual design and Flow Diagram.

### 3.2.1 Conceptual Design

The conceptual design of the developed system provides an overview of what the system is doing and the process which is followed. It is the first design that was made in the process of development of this system. It basically shows interaction between different phases of the system, strategies of the system and the complete process to get the output.

The conceptual model for Anomaly-based Network Intrusion Detection System is shown in Figure 15 on the next page. The diagram shows how the process is followed through different states of the system. The states which are included in the conceptual model are monitoring, data fetching or data collection, pre-processing of the fetched data, recognizing intrusions or anomalies in the fetched data by using intrusion models and finally generating alarms if anomaly in the incoming data packet is detected.

There are two main phases in the conceptual model of Anomaly-based NIDS, the first one is monitoring phase and the other one is implementing intrusion detection technique using machine learning. In the monitoring phase, the system is designed such that it regularly checks for the incoming data packets (mainly the header information of these data packets). This monitoring is done at different high traffic points in the network, so that no data packet is left and could cause threat to the computer network system in the near future. Header information collected from the incoming data packets while regular monitoring is used to analyze whether that activity performed by the packet is normal or anomalous. This is done using Machine Learning Algorithms by analyzing the past behaviour. If any kind of activity having anomalous behaviour is detected, then the system generates an alarm and sent this alarm to the system admin.

In case of any new behaviour causing threat to the system, it is analyzed by the system so that any kind of similar activity could be detected in the future scenario.

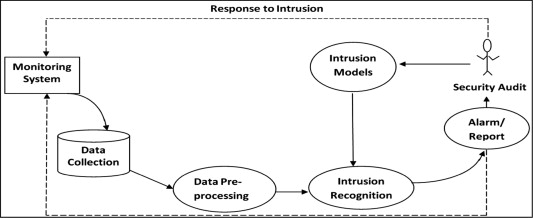


Figure 11 Conceptual Model for Anomaly-based NIDS

### 3.2.2 Flow Chart

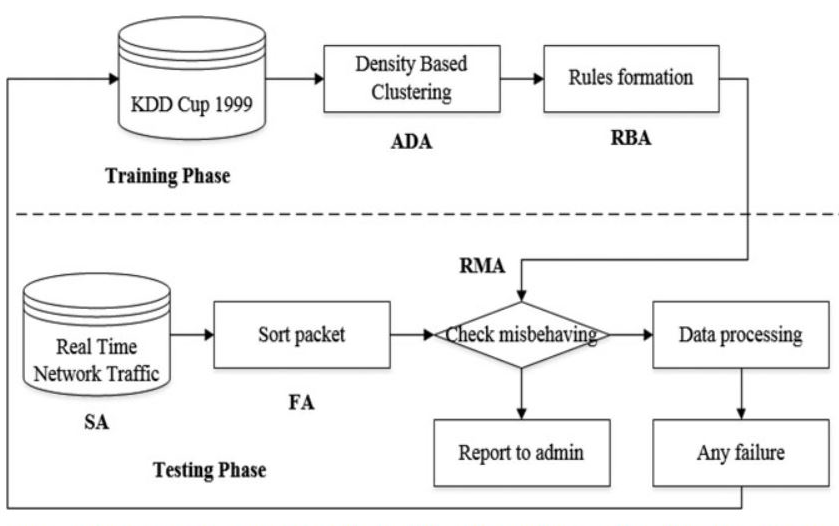
The below figure depicts the flow chart for the Anomaly-based Network Intrusion Detection System. Flow chart basically shows the working of a system in a step-by-step process in a diagrammatic format. We have used KDD CUP 99 dataset in this project.

Figure 12 Flow Chart of Anomaly-based NIDS

## 3.3 WORKING

The working of the Anomaly-based Network Intrusion Detection System can be explained using the proposed flow chart of the system. The working is categorized into two phases- Training phase and testing phase.

For the training phase, we have used KDD CUP 99 dataset which is a standard data set for the analysis of anomaly detection in a network system. This dataset consists of 41 features and 82 additional features have been extracted using this dataset. This dataset has been trained using different Classification and Clustering Algorithms of Machine Learning. The trained data is used to analyze the further incoming traffic in the testing phase. This is how classification is done that whether the activity done by the data packet is anomalous or normal.

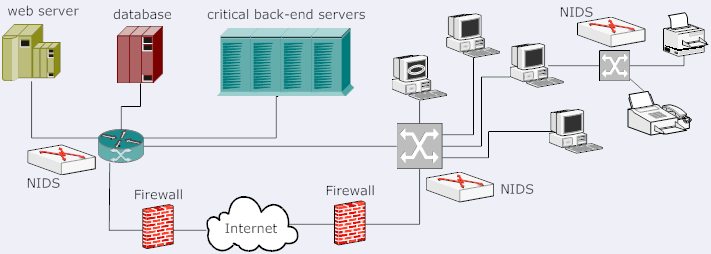
In the testing phase, till now we have used some part of KDD CUP 99 dataset for testing the efficiency of the proposed model. However, considering real time scenario, real time data from the incoming traffic in our machine can be extracted and its header information can be used to detect anomalies and threats. These packets in this phase are tested as per the past behavior of the activities of data packets in training phase. If the packet tested comprises of any kind of anomalous or malicious activity then an alarm is sent to the admin of the system. This behavior is also preprocessed so that in case of a new behavior system can be updated.

Figure 13 Working of Anomaly-based NIDS

## 3.4 BENEFITS

Anomaly-based Network Intrusion Detection System has the following benefits-

* One of the major benefits include that anomaly detection technique is not bounded to detect only the pre-known attacks, rather it is designed and efficient in a way that it can detect threats and anomalies having new or changed behavior.
* Anomaly detection technique solves the most common limitation of the misuse detection technique.
* There could be several perspectives of a normal activity as per different clients and their requirements. Anomaly detection system addresses this issue and provides customized way to design normal activities as per the requirements.
* This complete process of customization of normal activities makes the network system more secure as it becomes more difficult for a malicious person in such a scenario to attack without being detected by the system. Thus, making the network system more secure.
* Anomaly detection systems do have the concept of detecting zero-day attack and insider attacks.

The figure below gives an overview of the advantages of IDS-

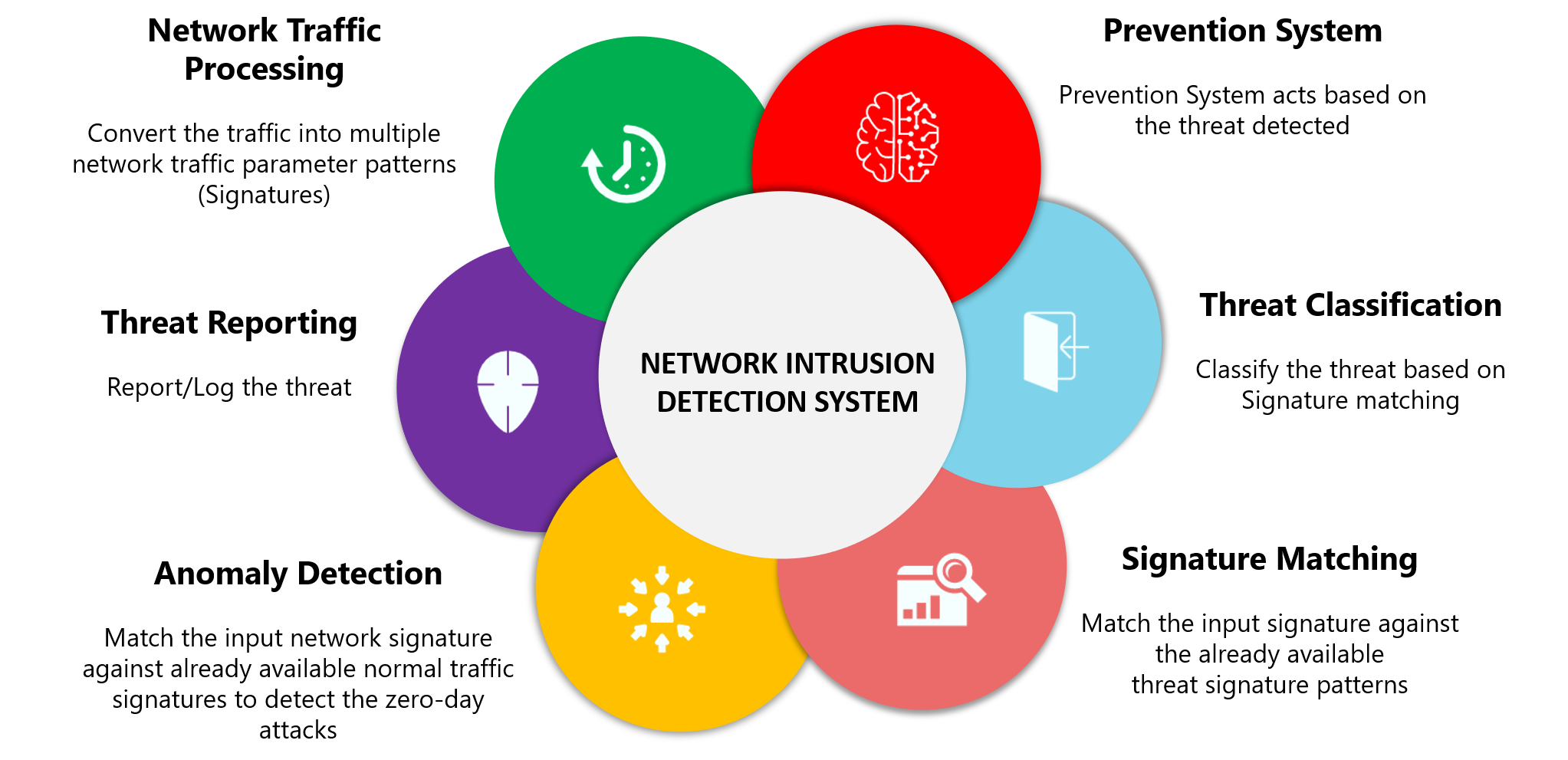


Figure 14 Advantages of IDS

## 3.5 TOOLS AND TECHNOLOGIES

The tools and technologies used in our project are as follows-

* Data cleaning of the packet data is done using NumPy and Pandas modules of Python programming language.
* We have used Anaconda Jupyter IDE for developing the system.
* Data visualization is done using the Visualization Tools of Python.
* Python matplotlib is used for plotting graphs.
* Machine Learning Algorithms-
  + Regression algorithms are tested on the fetched data but they do not reach to a great accuracy as they tries to detect the anomalies with less accuracy rather than predicting the behavior of the data packets and their trends.
  + KNN classification algorithm is used for predicting whether an activity is anomalous or not. This algorithms works good with small amount but its execution time increases and accuracy decreases as the dataset becomes large.
  + SVM algorithm can also be used, but, there could be scenario of large dataset, therefore, it’s not a good option to use SVM as we have random forest classifier to manage large dataset and give effective results.
  + Random Forest Classifier is used for classification, to avoid over-fitting of data and to deal with missing values in the data. It is also used to analyze the behavior of data packets in the dataset and thus it tends to more precise decision making.
* We have used Scikit-learn Python Module to implement machine learning algorithms.
* Artificial Neural Network is finally used to implement this project.

## 3.6 MODULES

### 3.6.1 Data Fetching

The figure below shows the code snippet of data fetching for real time purpose.

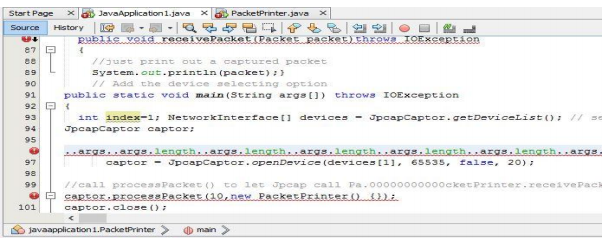


Figure 15 Real Time Data Fetching

### 3.6.2 Dataset Used

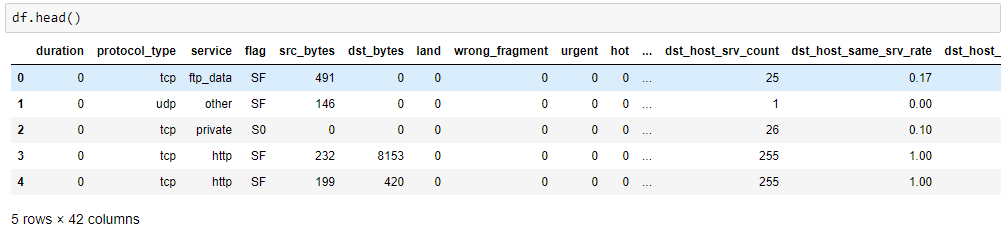
The dataset used in the project is KDD CUP 99 dataset which is obtained from Kaggle.

Figure 16 Snapshot of used Dataset

### 3.6.3 Data Visualization

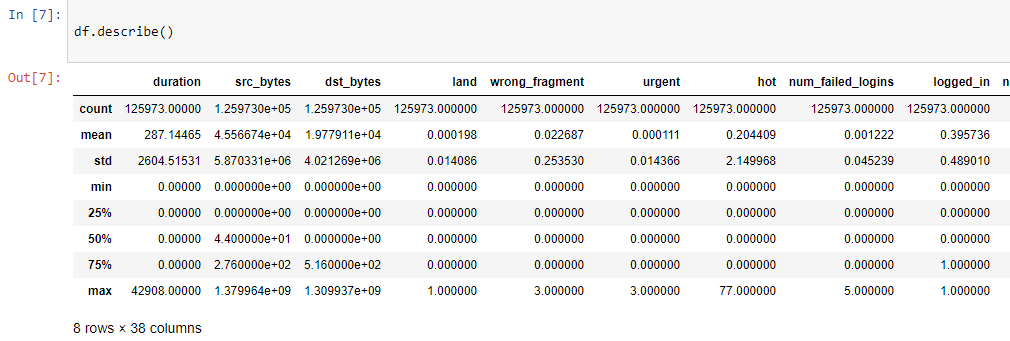
The dataset for incoming data packets can be visualized using Python Visualization Tools. Following are the figures which represents both the types of data.

Figure 17 Dataset Visualization



Figure 18 Histogram showing classes of different attacks

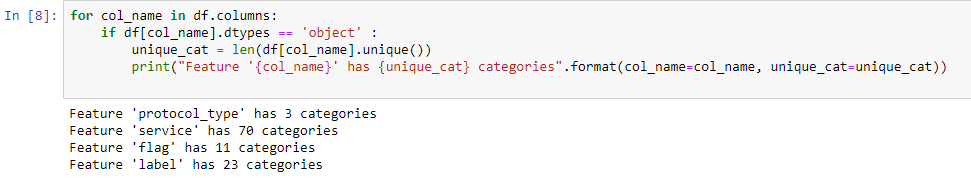


Figure 19 Sanpshot to specify different categories

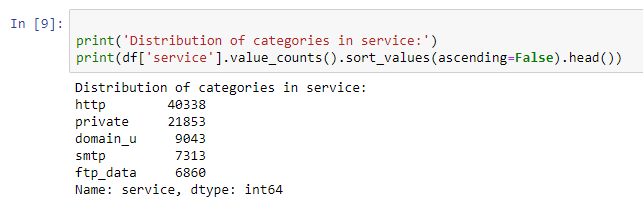
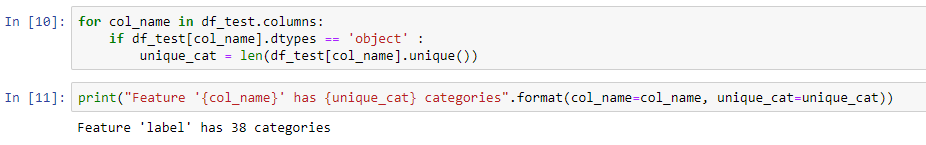


Figure 20 Feature Label categories

Figure 21 Distribution of Categories

### 3.6.4 Pre-processing and Selection of Features

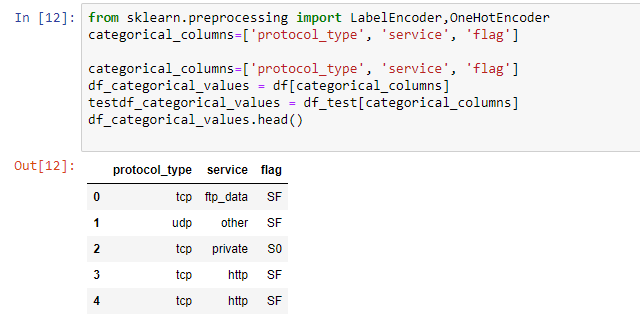
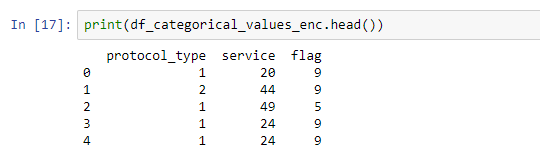


Figure 22 Categorical Representation

Figure 23 Categorical Values

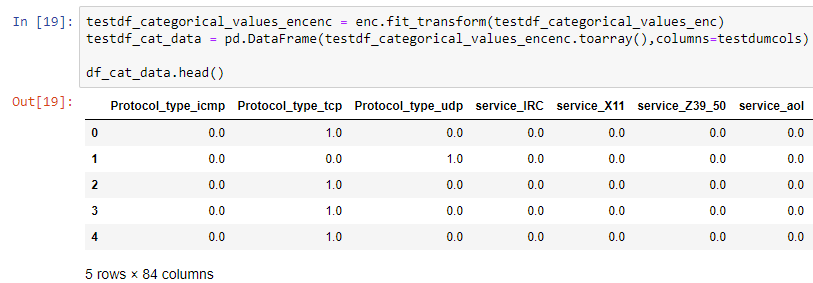


Figure 24

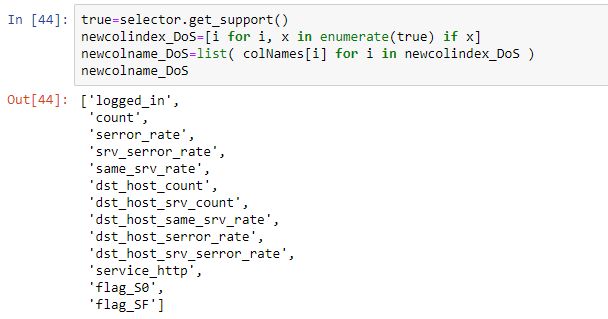
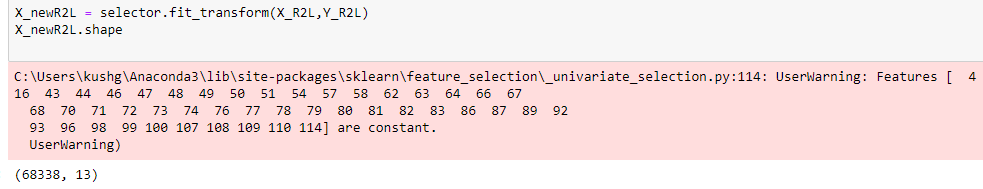


Figure 25 DoS Attack Features

Figure 26 R2L Attack

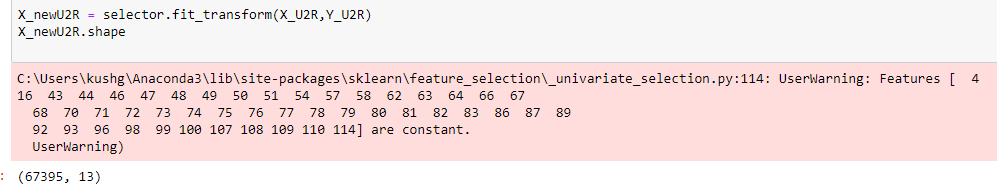


Figure 27 U2R Attack

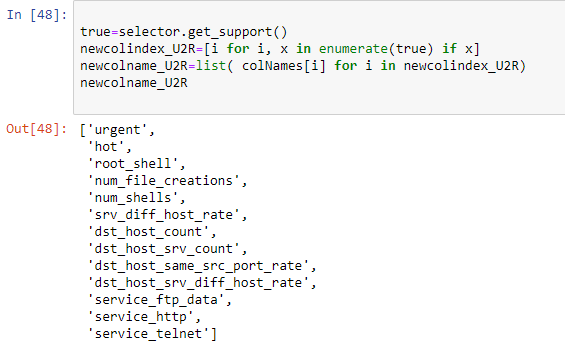


Figure 28 U2R Features

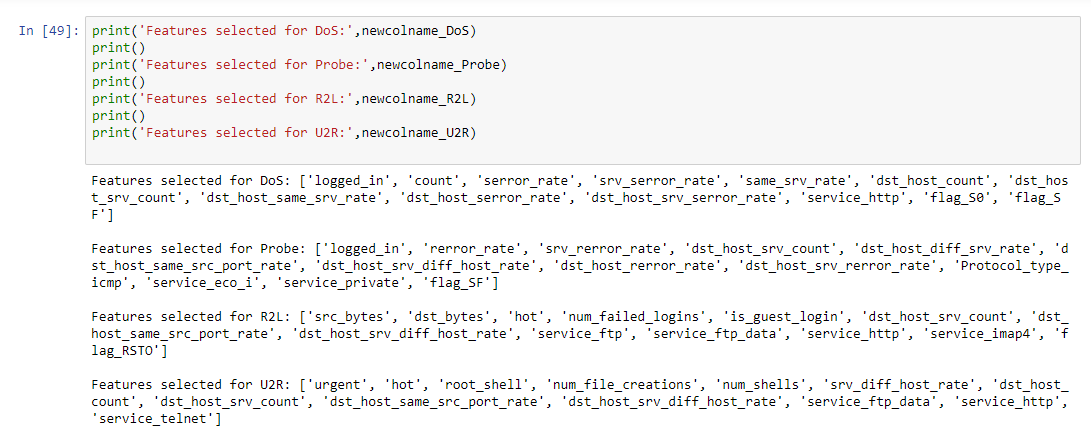


Figure 29 Feature for all the categories of attack

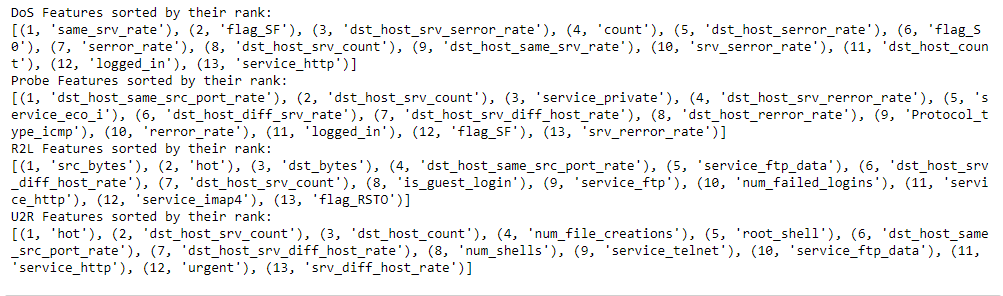


Figure 30 DoS Features sorted by their rank using Random Forest Classifier



Figure 31

### 3.4.6 Neural Network Model

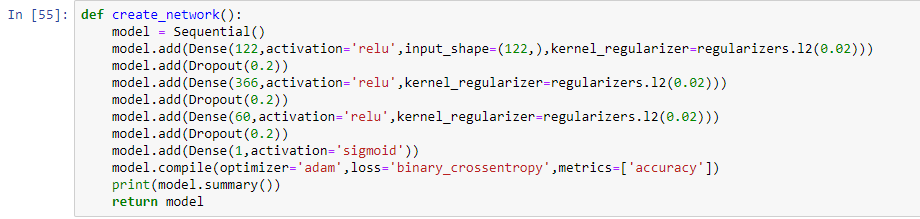


Figure 32 Neural Network Model

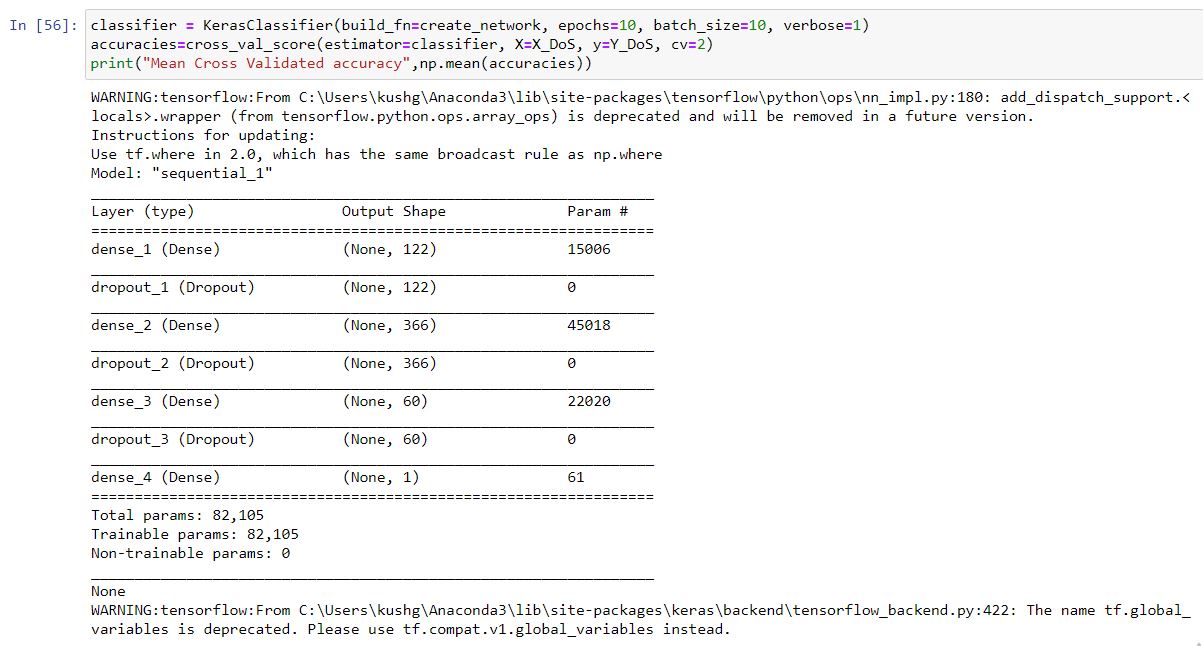


Figure 33 Trainable and Non-trainable Parameters

# CHAPTER-4 PERFORMANCE ANALYSIS

The project completely targets for improving the performance of the system to a better extent in order to solve the problem of network security to a greater extent. In our project we have implemented Anomaly Detection Technique using Machine Learning algorithms and have calculated accuracy rate for the predictions made to analyze the performance.

## 4.1 PREDICTIONS MADE FOR INPUT

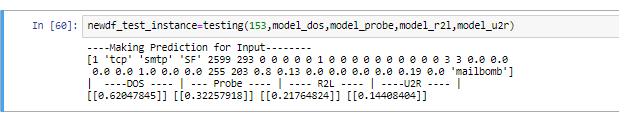
The performance of the system is tested on the following attacks-

Figure 34 Predictions

* Denial of Service Attack (DoS)
* User to Root Attack (U2R)
* Remote to Local Attack (R2L)
* Probing Attack

## 4.2 TESTING FOR DOS ATTACK

Figure 35 Accuracy for DOS Attack

## 4.3 TESTING FOR R2L ATTACK

The accuracy obtained for R2L attack type is 85.57%.

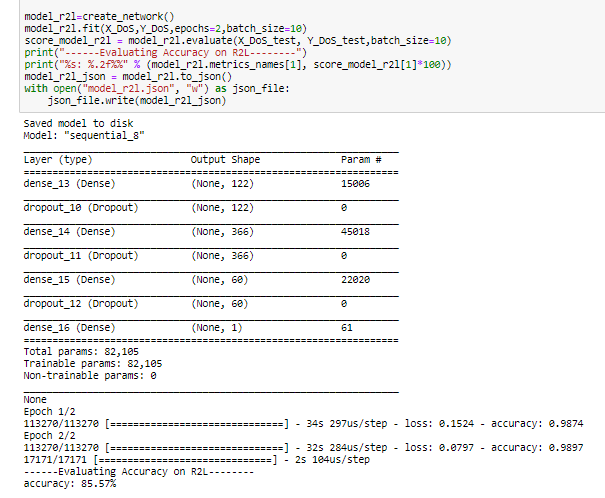


Figure 36 Accuracy for R2L Attack

## 4.4 TESTING FOR U2R ATTACK

The accuracy obtained for U2R attack type is 88.81%.

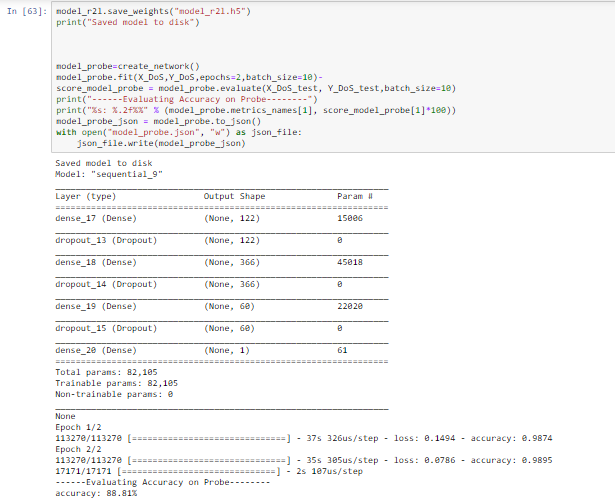


Figure 37 Accuracy for U2R Attack

## 4.5 PROBING ATTACK

The accuracy obtained for probing attack type is 75.98%.

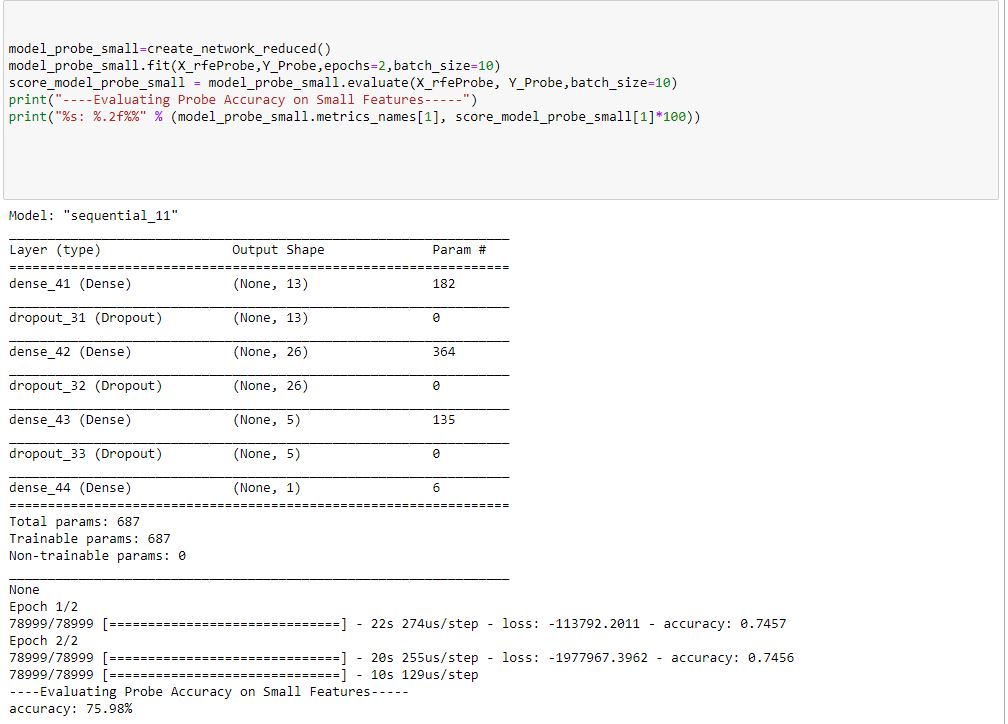


Figure 38 Accuracy for Probing Attack

## 4.6 COMPARISON

The table below shows the comparison of accuracies of different types of attack classes by using the developed Anomaly-based Network Intrusion Detection System.

|  |  |  |  |
| --- | --- | --- | --- |
| ATTACKS CLASS | NUMBER OF INSTANCES | NUMBER OF FEATURES | ACCURACY |
| DOS | 113270 | 13 | 89.56 |
| PROBE | 78999 | 13 | 75.98 |
| R2L | 68338 | 13 | 85.57 |
| U2R | 67395 | 13 | 88.81 |

Table 4 Comparison table for Performance Analysis

The graphical form of above comparison is shown in the figure below-

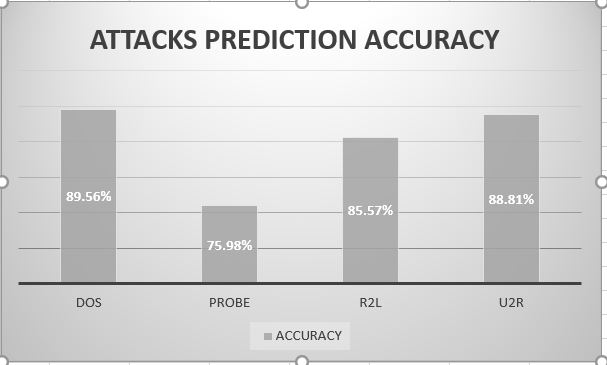


Figure 39 Comparison Graph

# CHAPTER-5 CONCLUSION

## 5.1 SYSTEM CONCLUSION

The implementation of the Anomaly-based IDS is done. The system aims for getting the optimal decision making of whether the activity performed by the incoming data packet is normal or anomalous. Over all, the system predicts the behavior of different normal and anomalous packets which can be used to make a decision. The dataset used in this project is KDD CUP 99 dataset which is

In this project, we have used different Machine Learning Algorithms to implement the system and compared these algorithms. The best suited one for this project came out to be Artificial Neural Network. We have used ANN for the final development of the system since ANN performed well for the packets whose behavior are new and are not pre-known.

## 5.2 FUTURE SCOPE

Till now, we have tested the developed system on a standard dataset of KDD CUP 99. To make the project useful in the real world, we are trying to extract real time data packet information. The training dataset would be same for this purpose and the testing data will be changed. Calculating accuracy in this scenario would be more beneficial and helpful. Real time data extraction can be done using Wireshark. Packet Sniffer can also be used to serve this purpose.

# REFERENCES

[1] <https://www.idrbt.ac.in/assets/alumni/PT-2015/Vigneshwaran.pdf>

[2] <https://www.intechopen.com/books/computer-and-network-security/anomaly-based-intrusion-detection-system>

[3] <https://www.researchgate.net/publication/339032478_Anomaly-Based_Network_Intrusion_Detection_System_A_Machine_Learning_Approach>

[4] <https://www.sciencedirect.com/science/article/pii/S1877050917328594>

[5] <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

[6] <https://en.wikipedia.org/wiki/K-means_clustering>

[7]<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>

[8]<https://towardsdatascience.com/random-forest-classification-and-its-implementation-d5d840dbead0>

[9] <https://www.investopedia.com/terms/a/artificial-neural-networks-ann.asp#:~:text=An%20artificial%20neural%20network%20(ANN)%20is%20the%20piece%20of%20a,by%20human%20or%20statistical%20standards>.

[10] <https://kdd.ics.uci.edu/databases/kddcup99/task.html>