**TrackIt**

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

**Bachelor of Technology**

in

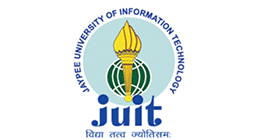
**Computer Science and Engineering**

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 for all activities and information required for the ETA. Trackit includes a trainee profile page where trainees can see all the data related to their training. This application should also automate a chatting engine, that helps trainees to interact with their Batch Owner.

# CHAPTER-1

## GENERAL INTRODUCTION

A training management system is a portal specifically designed to track details of 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.

A network attack is a kind of activity to attempt changes in the network of an organization in order to get its unauthorized access, perform changes in its network and for other malicious activities. Detecting these attacks and preventing the system from them is really important.

For serving this purpose both hardware and software methods can be used. Network Intrusion Detection System is a system or software that regularly monitors malicious activities within a network. They are classified as Signature-based IDS and Anomaly-based IDS. An NIDS generates an alarm whenever it finds any activity as malicious. These alarms can be false as well. These kinds of systems are placed within the network at different points so that they can monitor the incoming traffic regularly and detect the malicious activities. These systems can be developed easily with low cost and for every kind of network.

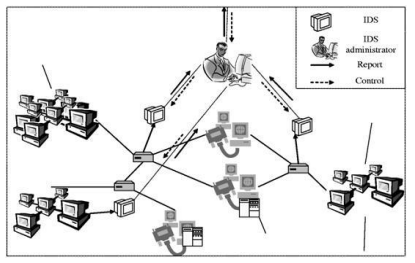


Figure 1 Network Intrusion Detection System

Advantages of Network IDS are as follows-

* The network is regularly monitored by the system for an attack.
* IDS can be changed according to the requirements of clients and can help outside as well as inner threats to the system and network.
* It efficiently prevents any kind of damage to the network.
* It provides user friendly interface which allows easy security management systems.
* Any alterations to files and directories on the system can be detected and reported.

## PROBLEM STATEMENT

According to the past scenario, hackers need to learn the techniques and vulnerabilities of the computer network system. This process takes a long time and was much difficult in the past. But due to fast growing technology and growth in the reach of the internet, hacking has become a very easy task. One only needs to download and install different hacking tools from the internet and then use them without knowing any insights and details.

With this growth nowadays, network security needs to be taken very seriously. With these developments in consideration, network threats can result in network system corruption, information manipulation, unauthorized access of the network, data corruption or data loss, identity threats of the customers, and other malicious activities. These activities could also be threat to the lives of the customers.

In order to detect these activities, IDS are used. They are placed within the network at different points to detect the threats and then they generate alarms so that these threats can be prevented. However, one of the easiest ways to solve this problem is to close the network completely from the outside world. This is due to the fact that closed networks do not provide any kind of connection to the public network; they provide connection to the trusted parties only. But, this only solves this problem when the network is small and does not require a lot of communication with other networks.

## OBJECTIVES

The main objective of this project is to detect the threats or the malicious activities in the incoming traffic in a computer network. Nowadays, it has become really important for small networks to prevent themselves from malicious activities. In order to prevent these small systems from getting their information reveled, we have developed Anomaly-based NIDS which aims to analyze the activities when placed at different points in the network. The system generates an alarm by analyzing the data packet from the already stored libraries. The system aims to provide maximum accuracy so as to provide a more secure network system. The classification between normal and anomalous is done on the basis of network behaviors that are pre-defined. If the system accepts it, then it belongs to the normal category, else it belongs to the malicious activity and thus generates an alarm.

## METHODOLOGY

In this project we have used the Agile Model of developing software. 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.

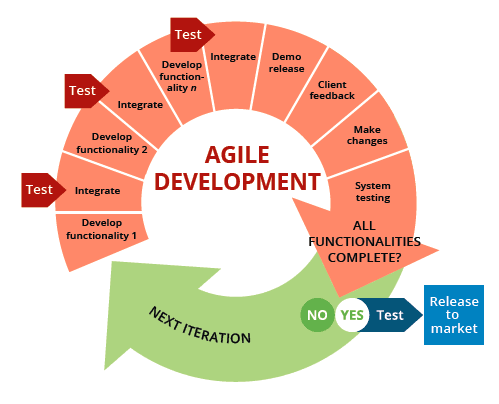


Figure 2 Stages of Agile Development

The motive of developing our project on the basis of 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 final 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

[2] 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 population, 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

[2] 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.

In depth,

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.

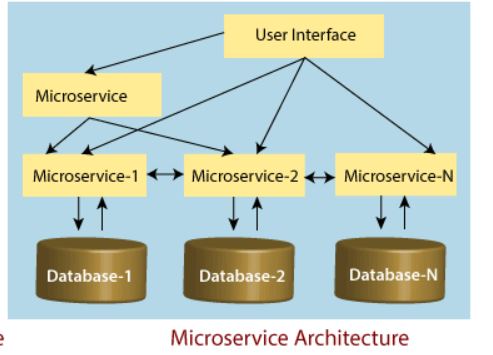
## 2.2 Spring MVC

[3] 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.

* 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.

## 2.3 Spring MicroServices

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.



## 2.4 ANOMALY DETECTION USING MACHINE LEARNING

[3]Machine Learning and its technique is very important aspect in this project. We have used clustering algorithms and classification algorithms to serve the purpose of network security. Figure 9 depicts the purpose of detecting the anomalies in the network traffic.

### 2.4.1 K-nearest neighbors

Figure 9 Machine Learning Techniques for Detection

[5]KNN is a supervised learning algorithm which is used as classification as well as regression algorithm. We have used KNN classification algorithm in our project. KNN algorithms use the information of the data packets and classify the new data packets either as normal or anomalous based on the similarity.

The process of classification is based on the majority vote to its neighbors. The accuracy of the algorithm is defined by variable ‘k’ and its value increases as the number of nearest neighbors increases.

The figure below depicts that the new data point added is categorized by the algorithm by checking the similarity from the existing data.

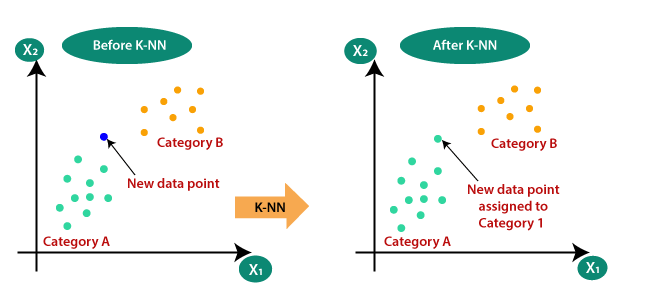


Figure 10 KNN Clustering Algorithm

KNN is a lazy learner algorithm i.e., when a new dataset comes in it does not get categorized from the training set immediately but it stores the dataset and at the time of classification, it performs an action on the dataset. This also requires the calculation of the value of k and computation of distance of each query instance to all training samples which takes time and thus slows down the execution process.

### 2.4.2 K-means Clustering

[6] K-means Clustering is an unsupervised learning algorithm that aims to divide n observations into k clusters and each of these observations belongs to the cluster with the nearest mean. Figure below shows the functioning of K-means clustering.

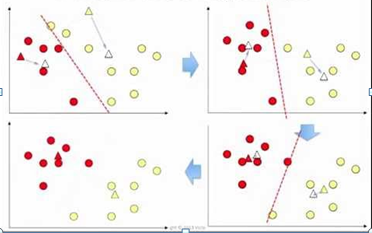


Figure 11 K-means clustering

K-means is not best suited for this purpose. The reason for this is that the clusters do not have regular shapes. Therefore, as soon as condition for shapes is not obtained, the model will not be able to successfully separate the clusters. Another reason is that anomaly points belonging to the clusters affect the center and the radius of the clusters which does not detect the anomalies while testing the data.

### 2.4.3 SVM Algorithm

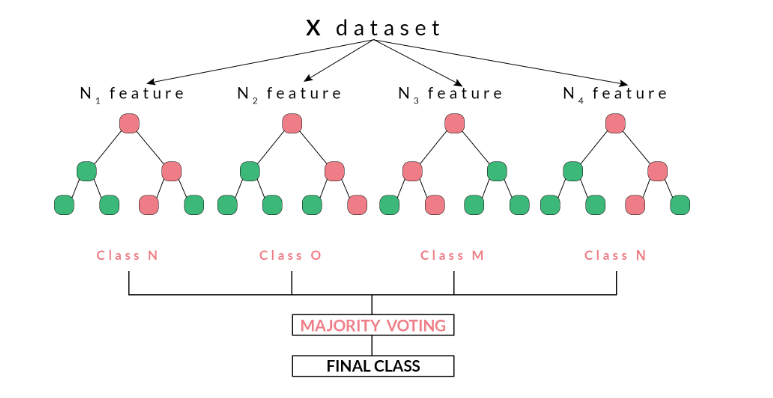
Figure 12 SVM Algorithm

[7] SVM is also a supervised learning algorithm which can be used for classification and regression both. In the SVM algorithm, each data item is plotted as a point in n-dimensional space, where n is the number of features, where the value of each feature is the value of a particular coordinate. Then, the classification is performed by finding the hyper-plane that differentiates the two classes very well. The figure depicts the two classes that are differentiated. In the below figure the support vectors drawn are the co-ordinates of the individual data points.

SVM algorithm with a non-linear kernel can be used in this project when the incoming traffic is not so huge since it requires a lot of time to train the large data. Since, there could be scenarios of high data traffic, therefore, it’s not a good option to use SVM as we have random forest classifier to manage large dataset and give effective results.

### 2.4.4 Random Forest Classifier Algorithm

[8]Random forest is a supervised classification machine learning algorithm. It is based on ensemble method i.e. to combine several decision trees and predict a better result than to utilize a single tree. The algorithm randomly select subset of training set and then creates a set of decision trees from it. It then aggregates the votes from different decision trees to decide the final class of the test object. Figure 13 depicts the working of Random Forest Classifier Algorithm.



Random forest makes a decision or prediction on the basis of maximum number of votes received from the decision trees (for categorizing data packet as anomalous or normal). The outcome which comes for the maximum number of times using various decision trees is considered as the final outcome by the random forest.

Figure 13 Random Forest Classifier

The main advantage of using Random Forest is that it can be implemented on a very large dataset (can also be used in case of high data traffic). The accuracy rate of using this algorithm is also comparatively more than the other algorithms.

### 2.4.5 Artificial Neural Network

[9]ANN has been developed to achieve performances as that of the human brain and for this particular reason; it has always out-performed the conventional computer systems. ANN revolves around the concept of working of human brain. As human brain has neurons, ANN also comprises of collection of connected nodes which are known as neurons. These neurons communicate with each other to transfer information or the message. These messages are a kind of input in the form of real numbers which passes from one artificial neuron to the other; and the output is computed by using some non-linear function (we have used RELU for developing this project). The ANN consists of an input layer, a hidden layer and an output layer. The figure below shows the basic structure of the neural network.

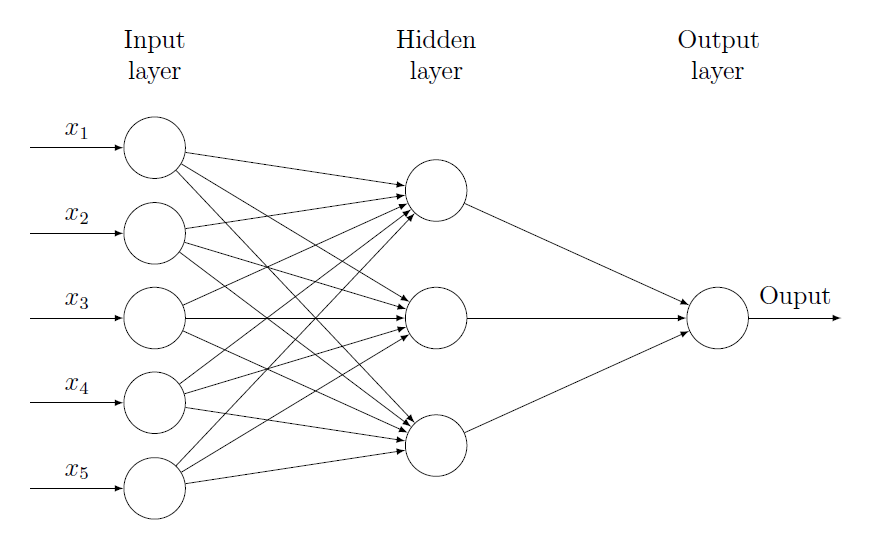
The ReLU Activation Function directly gives the output as positive or zero. In this case, the output will be positive, if the activity is found to be anomalous while the output will be zero, if the activity is found to be normal. The sigmoid and hyperbolic tangent activation functions can’t be used in neural networks having so many layers due to the problem of vanishing gradient.

Figure 14 Artificial Neural Network

## 2.5 DATASET DESCRIPTION

[10]The dataset that we have used in this project is a standard dataset KDD CUP 99. This dataset is mostly used to serve the purpose of analyzing the anomaly detection techniques. This data set is prepared by Stolfo et al. and is built based on the data captured in DARPA’98 IDS evaluation program.

The attacks fall into one of the following categories-

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

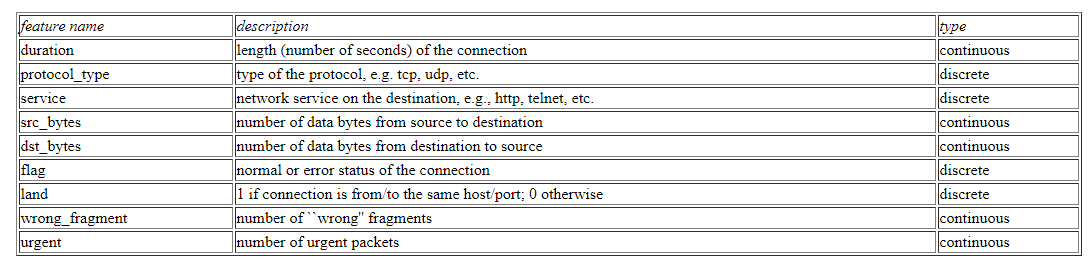
Description of the dataset is depicted in the following tables-

Table 1 Basic features of individual TCP connections

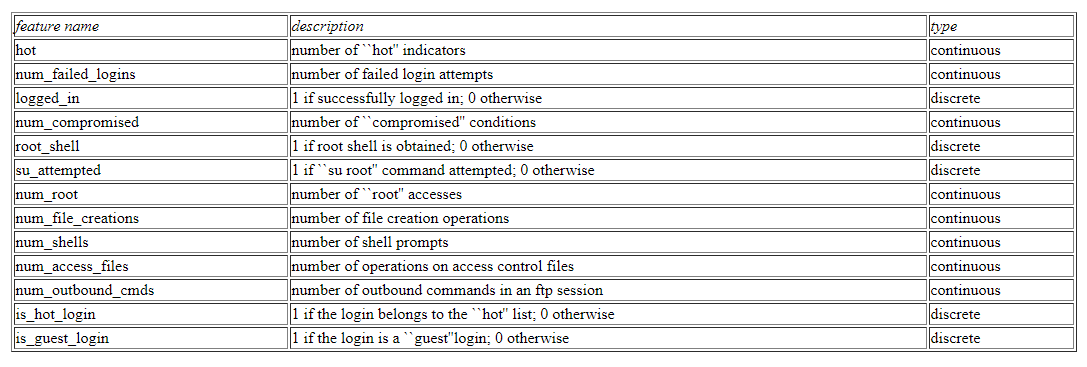
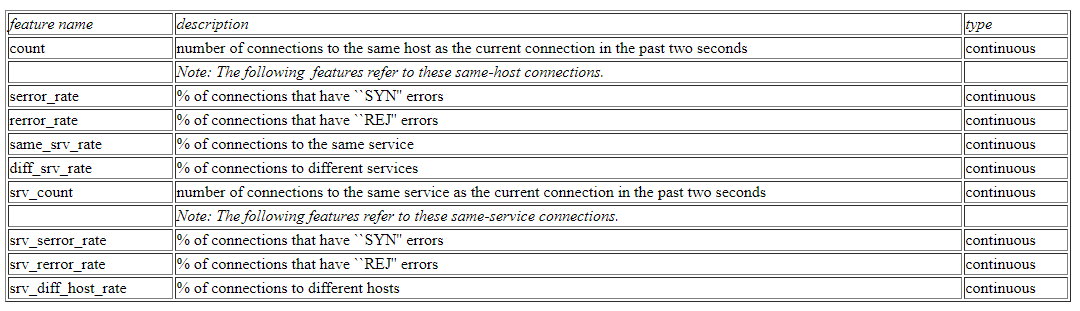


Table 2  Traffic features computed using a two-second time window

Table 3 Content features within a connection suggested by domain knowledge

# CHAPTER-3 SYSTEM DEVELOPMENT

## 3.1 REQUIREMENT ANALYSIS

### 3.1.1 Purpose

The purpose of this project is to build a software system which is capable to detect and analyze incoming data traffic in a computer network system in order to provide it network security and protect the system from different kinds of threats and malicious activities. It aims to maintain the integrity, accessibility and confidentiality of the system.

### 3.1.2 Intended Audience and Reading Suggestions

This project is a prototype for the anomaly-based network intrusion detection system and it is mainly targeted for the organizations with small networks having public connections; for example, education sites. This has been implemented under the guidance of college professors.

### 3.1.3 Project Scope

The purpose of the system developed a process of detecting network threats. This is done by categorizing the activities performed by the incoming data traffic in a network as normal or anomalous. This project has been implemented on a standard dataset till now to check for the accuracy of the proposed algorithm. The project further aims to test the scope of the project on some real time data.

### 3.1.4 Functional Requirements

The three major high level requirements for ‘Trade Decision Making’ are as follows-

* Get packet information – A standard dataset called KDD CUP 99 is used for analysis of the algorithms. Efforts have being made towards fetching real time data packets on a local machine so as to analyze the system’s accuracy for real data.
* Define detection technique - Specify and define the statistical model for new behaviours of the network anomalies. Finding the best suited technique to detect the anomalies in a network is an important task to perform.
* Categorizing the Activity – Each incoming data packet needs to be analyzed with its header information in order to classify it as normal or anomalous.

### 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 anomaly-based NIDS includes-

* Scalability - An IDS 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 data packets that it is managing to analyze for detecting the threats.
* Performance - Performance is computed by the comparison the amount of work done to the time and resources that are required to do that work. An IDS should have quick response times i.e., it should response immediately in case of an anomaly or a threat.
* Modifiability - Modifiability is the ease factor with which the system can be updated with new changes. An IDS should have easy process for updating new behaviours and data processing because in this field, updates and introduction of new threats is quite common.
* Reliability – The IDS should be accurate and dependable so that it could get more correct outputs for the inputs and generates less false positive alarms.
* 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.

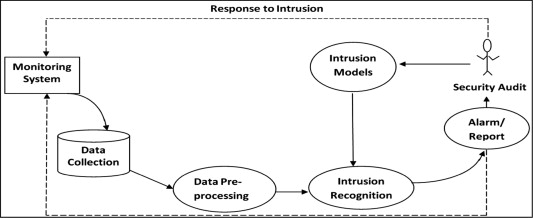


Figure 15 Conceptual Model for Anomaly-based NIDS

### 3.2.2 Flow Chart

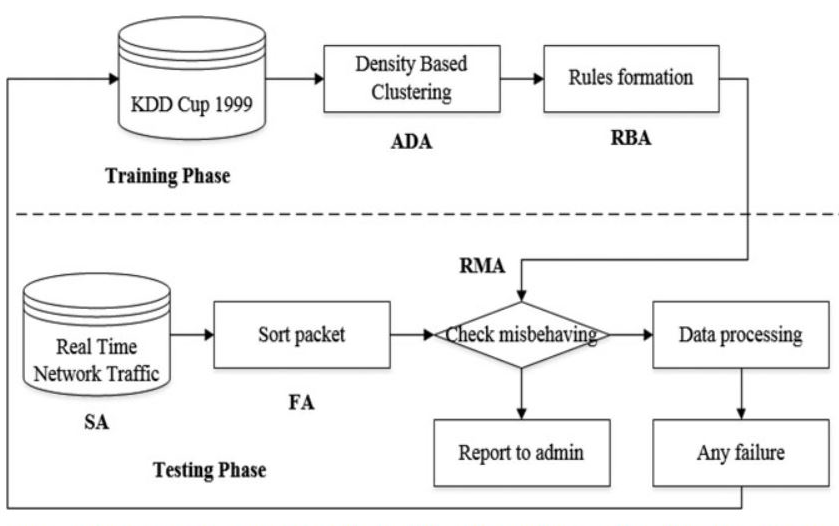
The below figure depicts the flow chart for the Anomaly-based Network Intrusion Detection System. Flow chart basically shows the working of a system in a step-by-step process in a diagrammatic format. We have used KDD CUP 99 dataset in this project.

Figure 16 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.

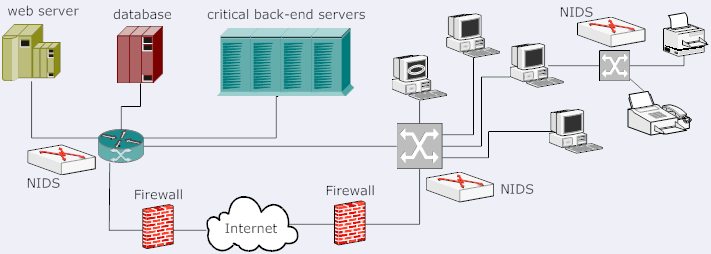
In the testing phase, till now we have used some part of KDD CUP 99 dataset for testing the efficiency of the proposed model. However, considering real time scenario, real time data from the incoming traffic in our machine can be extracted and its header information can be used to detect anomalies and 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 17 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-

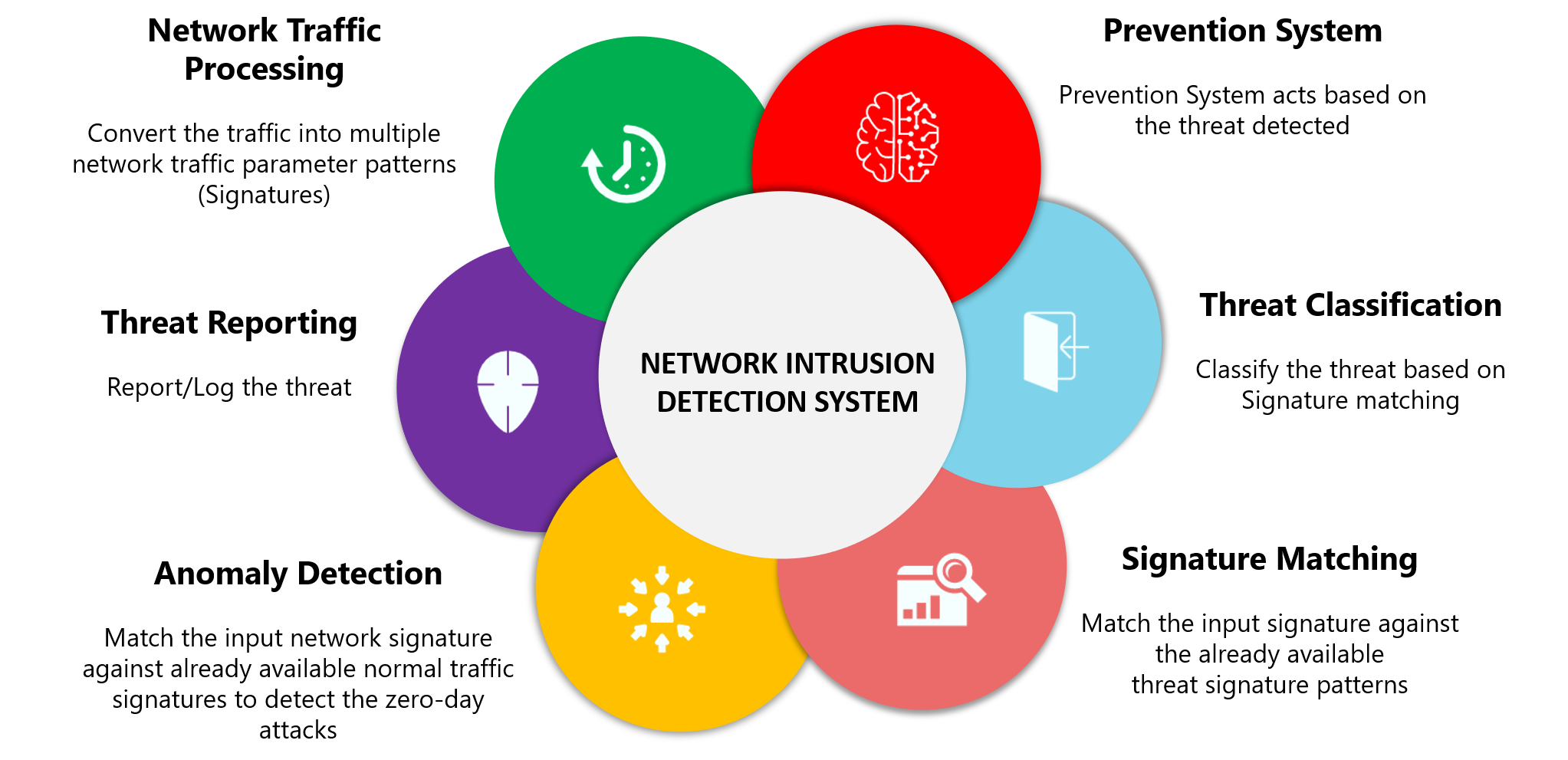


Figure 18 Advantages of IDS

## 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.

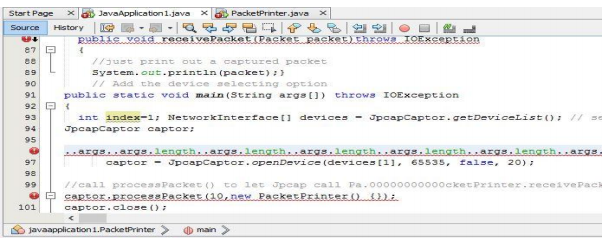


Figure 19 Real Time Data Fetching

### 3.6.2 Dataset Used

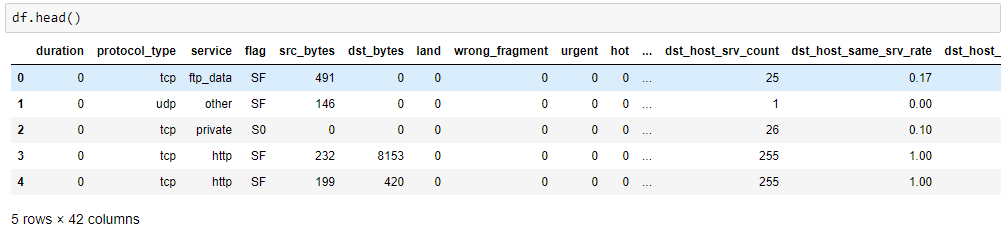
The dataset used in the project is KDD CUP 99 dataset which is obtained from Kaggle.

Figure 20 Snapshot of used Dataset

### 3.6.3 Data Visualization

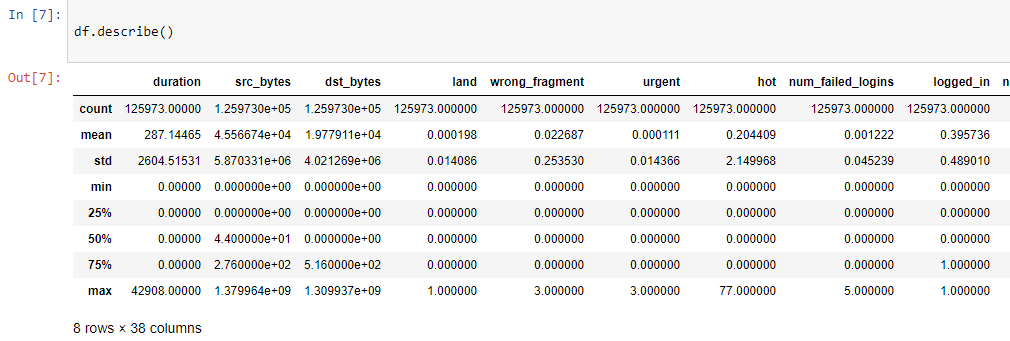
The dataset for incoming data packets can be visualized using Python Visualization Tools. Following are the figures which represents both the types of data.

Figure 21 Dataset Visualization



Figure 22 Histogram showing classes of different attacks

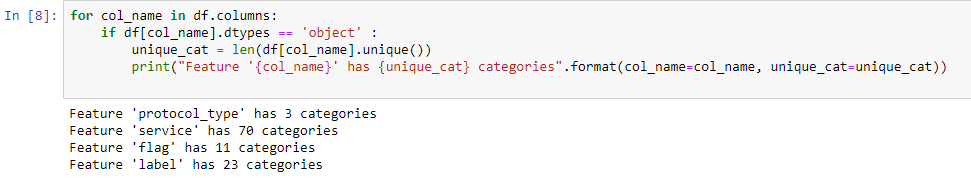


Figure 23 Sanpshot to specify different categories

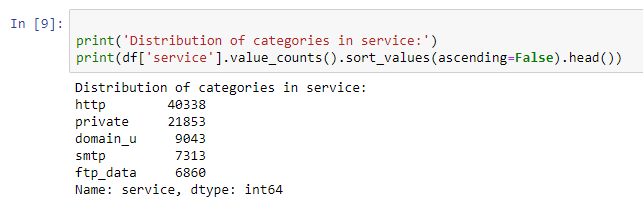
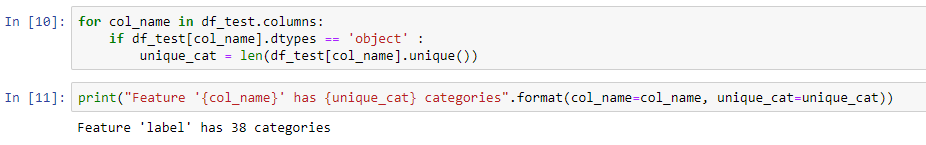


Figure 24 Feature Label categories

Figure 25 Distribution of Categories

### 3.6.4 Pre-processing and Selection of Features

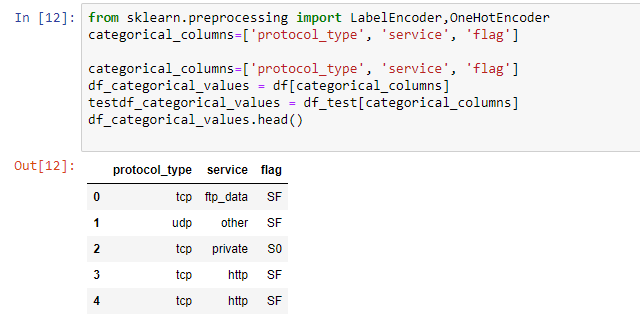
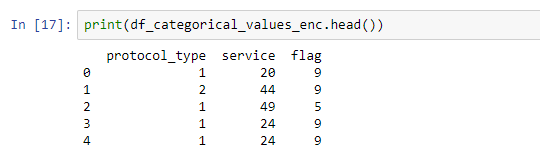


Figure 26 Categorical Representation

Figure 27 Categorical Values

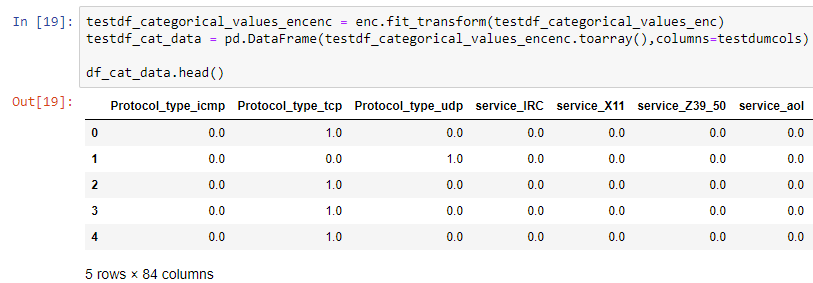


Figure 28

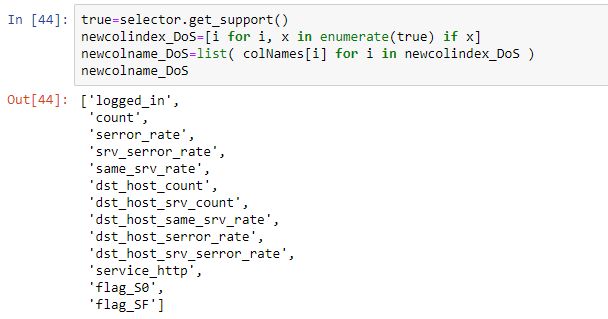
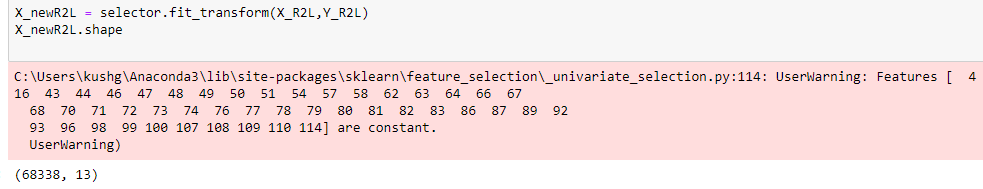


Figure 29 DoS Attack Features

Figure 30 R2L Attack

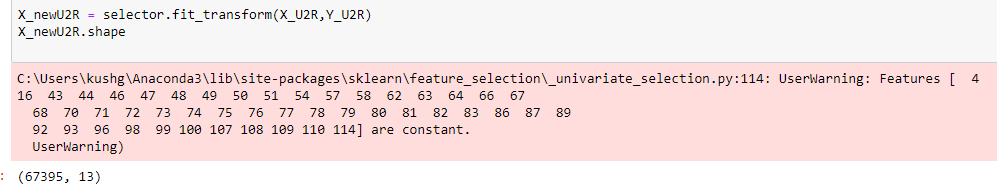


Figure 31 U2R Attack

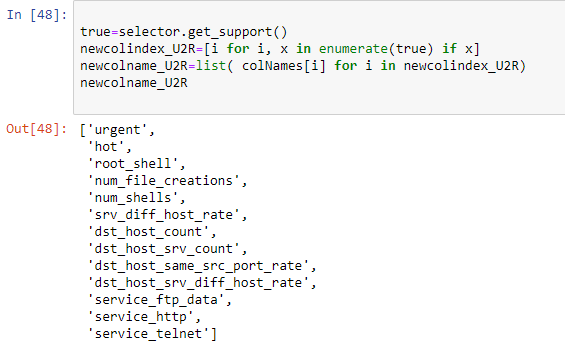


Figure 32 U2R Features

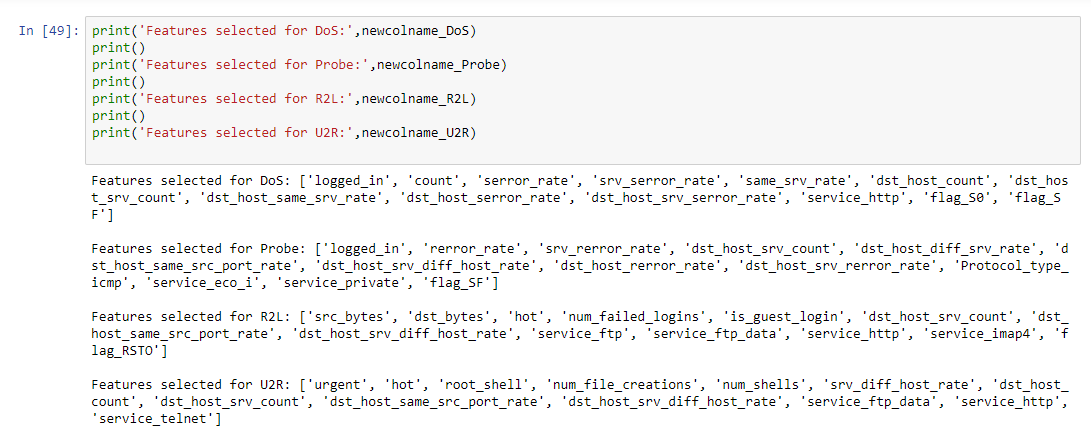


Figure 33 Feature for all the categories of attack

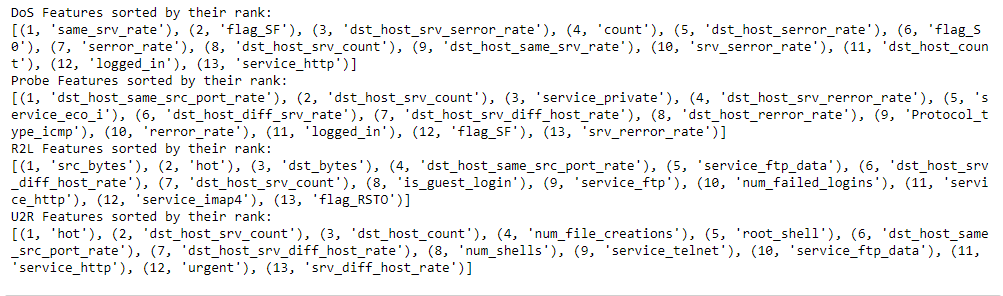


Figure 34 DoS Features sorted by their rank using Random Forest Classifier



Figure 35

### 3.4.6 Neural Network Model

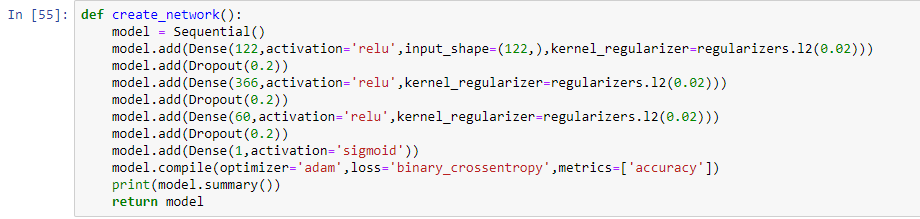


Figure 36 Neural Network Model

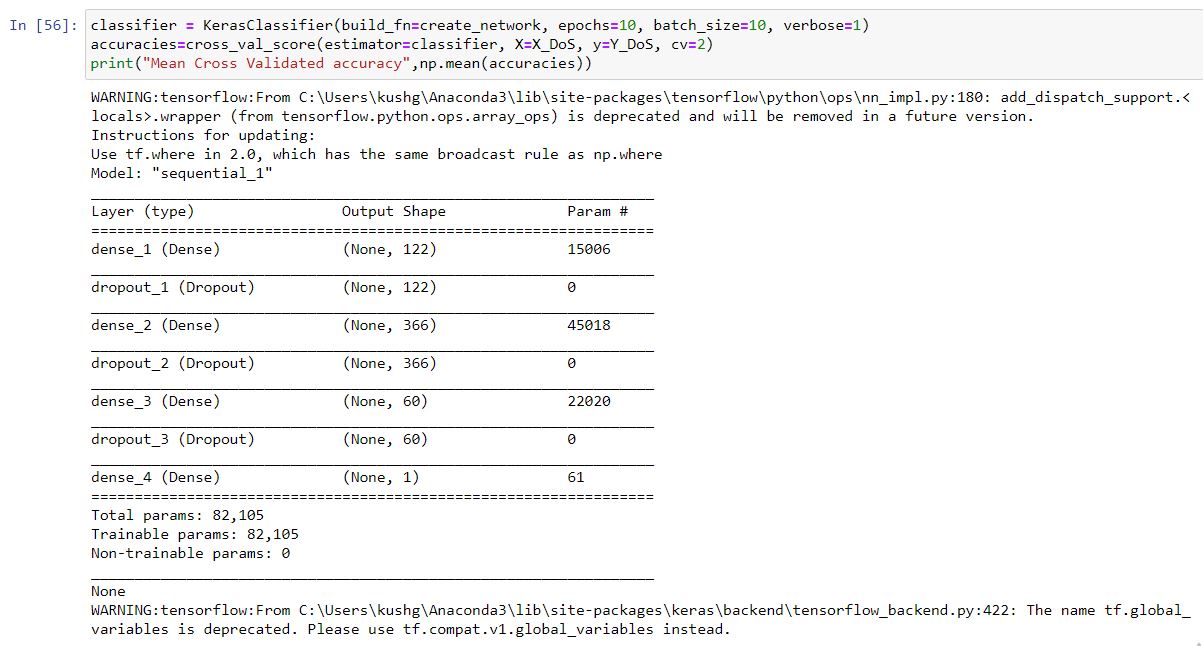


Figure 37 Trainable and Non-trainable Parameters

# 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

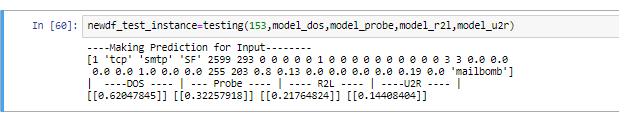
The performance of the system is tested on the following attacks-

Figure 38 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 39 Accuracy for DOS Attack

## 4.3 TESTING FOR R2L ATTACK

The accuracy obtained for R2L attack type is 85.57%.

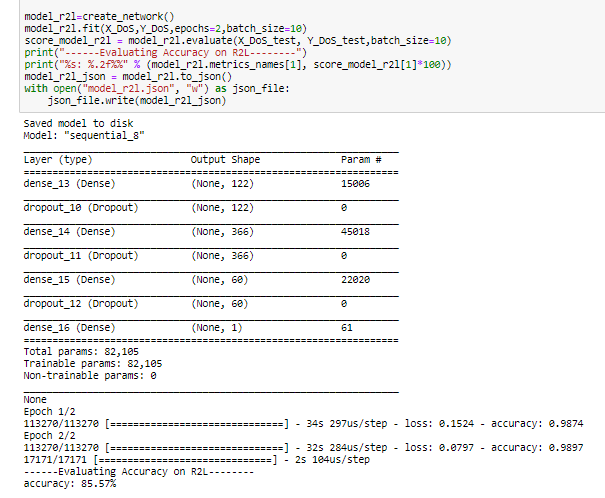


Figure 40 Accuracy for R2L Attack

## 4.4 TESTING FOR U2R ATTACK

The accuracy obtained for U2R attack type is 88.81%.

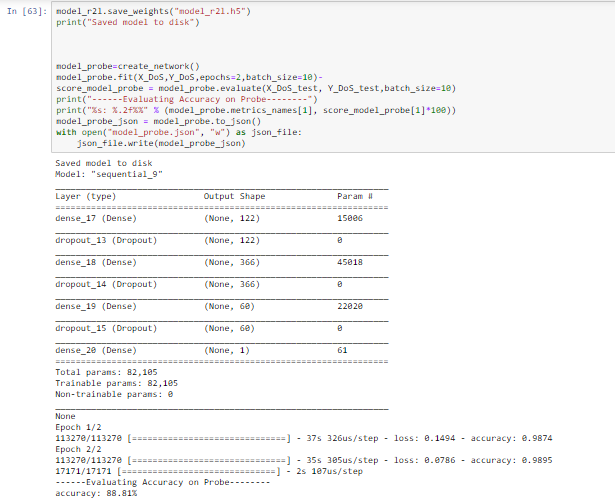


Figure 41 Accuracy for U2R Attack

## 4.5 PROBING ATTACK

The accuracy obtained for probing attack type is 75.98%.

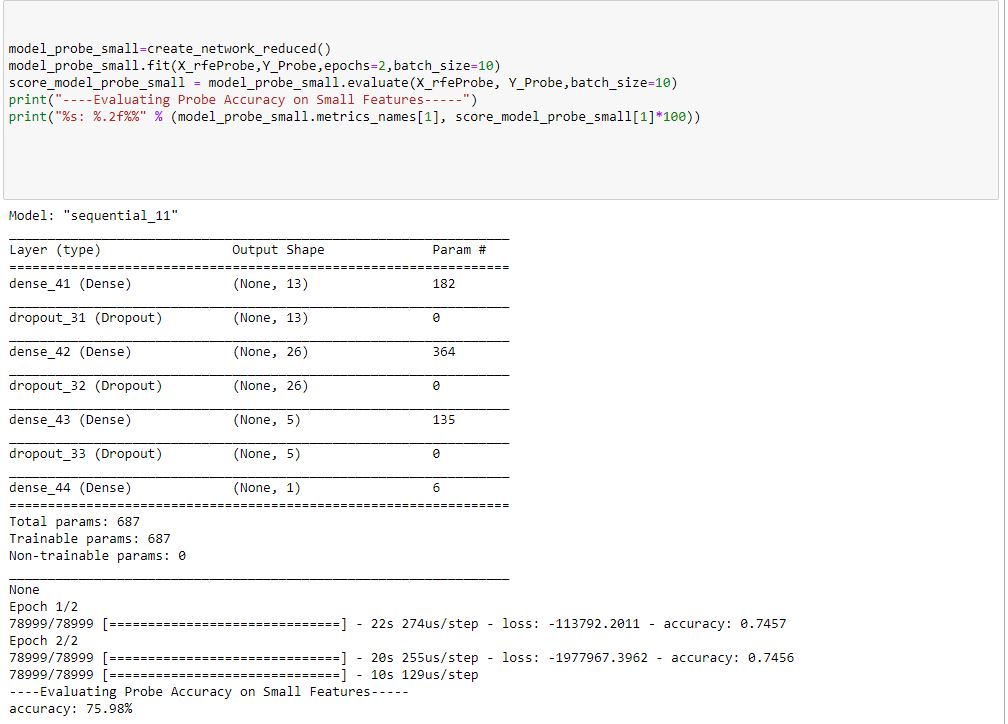


Figure 42 Accuracy for Probing Attack

## 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-

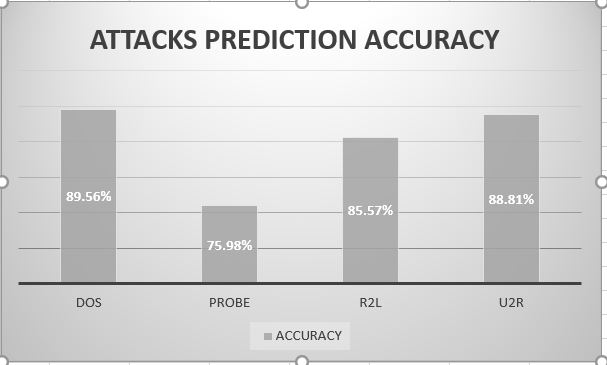


Figure 43 Comparison Graph

# 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>