Software Requirements Specification

For

INTRUSION DETECTION SYSTEM

28-03-2022

Prepared by

NAME	SAP ID	ROLL NO
ANUSHKA BANSAL	500067844	R164218014
SHREYA SHARMA	500068573	R164218070
SILVI	500069092	R164218072
ISHIKA AGRAWAL	500071154	R164218097

Department of Informatics

School Of Computer Science

UNIVERSITY OF PETROLEUM & ENERGY STUDIES,

DEHRADUN- 248007. Uttarakhand

Table of Contents

Revision History

Date	**Change**	**Reason for Changes**	**Mentor Signature**

1.0. Introduction

1.1. Purpose of the project

Intrusion Detection System (IDS) is a detective device designed to detect malicious (including policy-violating) actions. An Intrusion Prevention System (IPS) is primarily a preventive device designed not only to detect but also to block malicious actions.

Depending on their physical location in the infrastructure, and the scope of protection required, the IDS and IPS fall into two basic types: network-based and host-based. Both have the same function and the specific type deployed depends on strategic considerations.

The IDS and IPS devices employ technology, which analyses traffic flows to the protected resource in order to detect and prevent exploits or other vulnerability issues.

These exploits can manifest themselves as ill-intended interactions with a targeted application or service. The goal is to interrupt and gain control of an application or a machine, thus enabling the attacker to disable the target causing a denial-of-service situation, or to gain access to rights and permissions available through the target.

1.2. Target Beneficiary

Keeping your network safe from intrusion is one of the most vital parts of system and network administration and security. If your network is penetrated by a malicious attacker, it can lead to massive losses for your company, including potential downtime, data breaches, and loss of customer trust.

An intrusion detection system (IDS) is a tool or software that works with your network to keep it secure and flag when somebody is trying to break into your system. There are several different types of IDS and numerous tools on the market and figuring out which one to use can be daunting.

1.3. Project Scope

An intrusion detection system (IDS) is a system that monitors network traffic for suspicious activity and alerts when such activity is discovered. While anomaly detection and reporting are the primary functions of an IDS, some intrusion detection systems are capable of taking actions when malicious activity or anomalous traffic is detected, including blocking traffic sent from suspicious Internet Protocol (IP) addresses. Intrusion detection systems offer organizations several benefits, starting with the ability to identify security incidents. An IDS can be used to help analyze the quantity and types of attacks. Organizations can use this information to change their security systems or implement more effective controls. An intrusion detection system can also help companies identify bugs or problems with their network device configurations. These metrics can then be used to assess future risks. Intrusion detection systems can also help enterprises attain regulatory compliance. An IDS gives companies greater visibility across their networks, making it easier to meet security regulations. Additionally, businesses can use their IDS logs as part of the documentation to show they are meeting certain compliance requirements.Intrusion detection systems can also improve security responses. Since IDS sensors can detect network hosts and devices, they can also be used to inspect data within the network packets, as well as identify the OSes of services being used. Using an IDS to collect this information can be much more efficient than manual censuses of connected system.

1.4. References

Sim Hoong Kok, Azween Abdullah, Noor Zaman,M. Supramaniam on “A review of intrusion detection system using machine learning approach”

Taylor's University

(https://www.researchgate.net/publication/332260496_A_review_of_intrusion_detection_system_using_machine_learning_approach)

Mrutyunjaya Panda, Ajith Abraham, Swagatam Das, Manas Ranjan Patra on “Network intrusion detection system: A machine learning approach” *Utkal University, Machine Intelligence Research Labs, Indian Statistical Institute,*

Berhampur university

(https://www.researchgate.net/publication/220468036_Network_intrusion_detection_system_A_machine_learning_approach)

Usman Shuaibu Musa, Sudeshna Chakraborty, Muhammad M. Abdullahi, Tarun Maini on “A Review on Intrusion Detection System using Machine Learning Techniques” *School of Engineering & Technology, Sharda University,*

Gr. Noida, UP, India

(https://ieeexplore.ieee.org/document/9397121

2.0. Project Description

2.1. Reference Algorithm

Due to the advancement in technology and digitization of information, there has been an increase in network data traffic. This also brings in the scope of increased network attacks and poses a threat to computer systems and data. The vulnerabilities in network security seem to increase in direct proportion with the use of the internet. Intrusion Detection Systems prove to be an effective method to detect unauthorized access and attacks in a network and safeguard it from intruders. In this project, we use the KDD dataset to develop an intrusion detection system using machine learning algorithms and ensemble techniques. The dataset is first preprocessed to obtain clean and non-redundant data which is then tested against an ensemble model involving three classifiers namely Gaussian Naive Bayes, Decision Tree, and XGBoost.

The goal is to build an IDS to classify attacks as malicious or normal connections. Here is the entire implementation in four steps:-

- Load the dataset and apply pre-processing.
- Perform Exploratory Data Analysis on the dataset.
- Train and test following classifiers - Decision Tree, Gaussian Naive Bayes, XGBoost.
- Make predictions using ensemble techniques.

Steps Involved

Preprocessing

This step includes data integration, data reduction through feature and instance selections, data transformation by converting symbolic features to numeric and class to nominal), data cleaning to remove outlier and extreme values. Also, the dataset had all types of forms: continuous, discrete, and symbolic with varied ranges and resolutions, most of which cannot be processed by a pattern classification method, hence preprocessing of data was necessary before we could build a classification model.

Exploratory data analysis (EDA)

EDA involves visualizing a dataset to summarize the main characteristics using statistical graphics and other data visualization methods. This provides a deeper understanding of the dataset to extract impartial experiments.

Building Classifiers

Decision Tree

A decision tree is very much popular as a single classifier because of its simplicity and easier implementation. It is a classification model that uses a tree-like structure to perform a decision and is commonly used in operation research and intrusion detection because it gives better performance compared to other algorithms.

Gaussian Naive Bayes

Gaussian Naive Bayes is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data. Naive Bayes is a group of supervised machine learning classification algorithms based on the Bayes theorem. It produces good results in the classification where there exist simpler relations.

XGBoost

eXtreme gradient boosting (XGBoost) is a boosting technique that is a part of the ensemble-based approach. A sequential decision tree is constructed, which is also called a sequential ensemble technique. This method provides a result that contains bias that is low and high invariance because the model has a better ability to fit the training data.

Classification using ensemble techniques

Classification of the network traffic data as normal or malicious using Ensemble Learning is a method that combines multiple learners to solve the specific problems to improve the accuracy of classifiers. We will use the max voting method that is generally used for classification problems. In this technique, multiple models are used to make predictions for each data point. The predictions by each model are considered as a 'vote'. The predictions which we get from the majority of the models are used as the final prediction.

2.3. SWOT Analysis

2.4. Product Features

1. It must run continually without human supervision. The system must be reliable enough to allow it to run in the background of the system being observed. However, it should not be a "black box". That is, its internal workings should be examinable from outside.

2. It must be fault tolerant in the sense that it must survive a system crash and not have its knowledge-base rebuilt at restart.

3. On a similar note to above, it must resist subversion. The system can monitor itself to ensure that it has not been subverted.

4. It must impose minimal overhead on the system. A system that slows a computer to a crawl will simply not be used.

5. It must observe deviations from normal behaviour.

6. It must be easily tailored to the system in question. Every system has a different usage pattern, and the defense mechanism should adapt easily to these patterns.

7. It must cope with changing system behaviour over time as new applications are being added. The system profile will change over time, and the IDS must be able to adapt.

8. Finally, it must be difficult to fool.

2.5. User Classes and Characteristics

The goal is to build an IDS to classify attacks as malicious or normal connections. Here is the entire implementation in four steps:-

- Load the dataset and apply pre-processing.
- Perform Exploratory Data Analysis on the dataset.
- Train and test following classifiers - Decision Tree, Gaussian Naive Bayes, XGBoost.
- Make predictions using ensemble techniques.

2.6. Design and Implementation Constraints

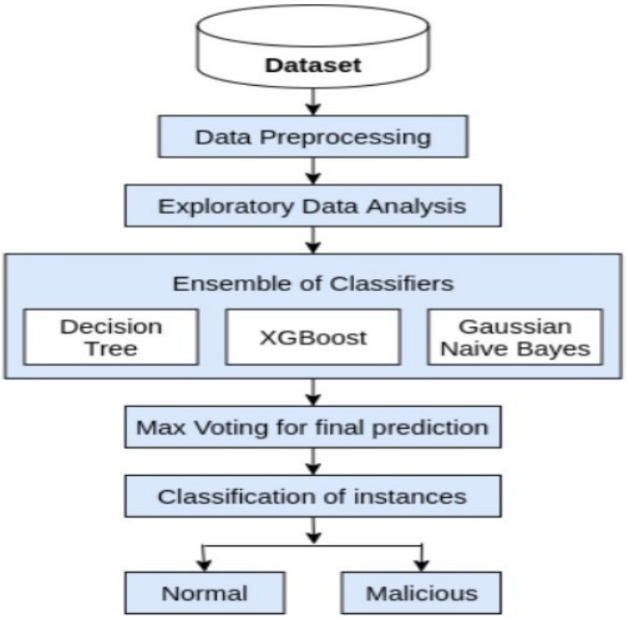
· Need to have proper internet connection and some datasets for symptoms, disease and doctors details.

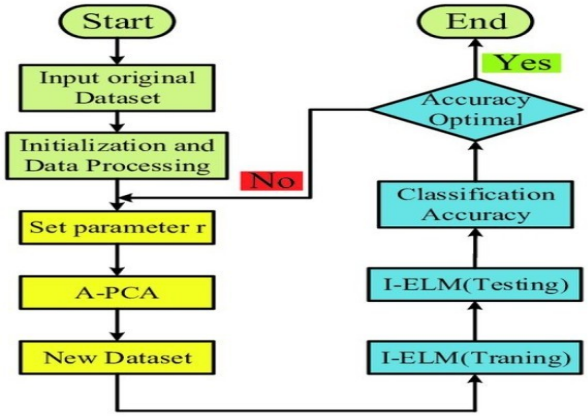
· Need to have a laptop with 8GB of RAM.

2.7 Design diagrams

Flowchart

Intrusion Detection System





9. Schedule

2.8. Assumption and Dependencies

Let us assume that this application is used in the following application:

The goal is to build an IDS to classify attacks as malicious or normal connections. Here is the entire implementation in four steps:-

- Load the dataset and apply pre-processing.
- Perform Exploratory Data Analysis on the dataset.
- Train and test following classifiers - Decision Tree, Gaussian Naive Bayes, XGBoost.
- Make predictions using ensemble techniques.

3.0. System requirements

One personal computer with:

◦ Minimum 4 gigabytes of

RAM Software Requirements:

◦ Windows 10
◦ VS code
◦ Jupyter Notebook
◦ A dataset (csv file)

3.1. Software Interface

Software used	Description
Operating system	We have chosen windows operating system for its best support and user-friendliness.
Juypter notebook	Platform on which we have written our code compiled and tested.

Table 1:Software Interfaces

3.2. Protocols

- browser
- the definition of message formatting.

4.0. Non Functional requirements

4.1. Performance requirements

Firebase : Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment

The goal is to build an IDS to classify attacks as malicious or normal connections.

4.2. Security requirements

Access of administrative staff to the bot management system needs to be tightly controlled through built in role based security and multi-user management.

4.3. Software Quality Attributes

Performance :

- Avoid inappropriate utterances and be able to perform damage control.
- Chatbot should be able to answer all the common questions asked by the user.

Functionality :

- Interprets commands accurately
- Contains breath of knowledge, is flexible in interpreting

it Accessibility :

- Can detect meaning or intent
- Responds to social cues

Affect :

- Give conversational cues

Appendix A: Glossary

This document uses the following conventions.

DB	Database
User	User can login, register and ask queries related to the health issues they are facing.