

A Major Project On

INTRUSION DETECTION IN NETWORK AND SERVER

Submitted in partial fulfillment of the requirements for the award of the

Bachelor of Technology

In

Department of Computer Science and Engineering

By

Mohd Muzammil	19241A0587
Jakka Arun	20245A0511
Sidharth Nookala	19241A05A7
Raj Mohan	19241A05B3

Under the Esteemed guidance of

Dr. K. Butchi Raju

Professor



Department of Computer Science and Engineering

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)



GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Autonomous)

CERTIFICATE

This is to certify that the major project entitled “**Intrusion Detection in Network and Server**” is submitted by **Mohd Muzammil (19241A0587)**, **Jakka Arun (20245A0511)**, **Sidharth Nookala (19241A05A7)**, **Raj Mohan (19241A05B3)** in partial fulfillment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering during academic year 2022-2023.

INTERNAL GUIDE

Dr. K. Butchi Raju
Professor

HEAD OF THE DEPARTMENT

Dr. K. MADHAVI
Professor

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First we would like to express our deep gratitude towards our internal guide **Dr. K. Butchi Raju , Prof.** Department of CSE for his support in the completion of our dissertation. We wish to express our sincere thanks to **Dr. K. Madhavi, HOD, Department of CSE** and to our principal . **Dr. J. Praveen** for providing the facilities to complete the dissertation. We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

Mohd Muzammil (19241A0587)

Jakka Arun (20245A0511)

Sidharth Nookala (19241A05A7)

Raj Mohan (19241A05B3)

DECLARATION

We hereby declare that the industrial major project entitled **“Intrusion Detection in Network and Server”** is the work done during the period from **16th Dec 2022 to 3rd June 2023** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad). The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

Mohd Muzammil (19241A0587)

Jakka Arun (20245A0511)

Sidharth Nookala (19241A05A7)

Raj Mohan (19241A05B3)

	Table of Contents	
Chapter	TITLE	Page No
	Abstract	
1	Introduction	
	1.1 Existing System	
	1.1.1 Limitations in Existing System	
	1.2 Proposed System	
	1.2.1 Advantages over Existing System	
2	Literature Survey	
3	Software Requirement Specifications	
	3.1 IEEE Standard 1. Introduction 1.1 Purpose of the requirements document 1.2 Scope of the product 1.3 Definitions, acronyms and abbreviations 1.4 Overview of the remainder of the document 2. General description 2.1 Product perspective 2.2 Product functions 2.3 User characteristics 2.4 General constraints 2.5 Assumptions and dependencies 3. Specific requirements 3.1 Functional Requirements, 3.2 Non-Functional Requirements 3.3 Interface Requirements	
	3.2 Feasibility Study	
	3.2.1 Technical Feasibility	
	3.2.2 Economic Feasibility	
	3.2.3 Operational Feasibility	
4	Design	
	4.1 Project Description	

	4.1.1 System Use case Diagrams	
	4.1.2 Class Diagrams	
	4.1.3 Data Flow Diagram(DFD)	
5	Implementation	
	5.1 Source code (Main)	
	5.2 Screen shots (Results)	
6	Testing	
	Test Cases	
7	Conclusion& Future Scope	
8	References	
9	List of Figures List of Tables Acronyms	

ABSTRACT

The most sensitive and important data are stored on servers; strong security is required to avoid data theft and misuse. When an intrusion occurs in a system, an intrusion detection system (IDS) is used to identify it and alert the admin. A network and devices are inspected by an IDS for malicious activity or policy breaches. Any unlawful behaviour or a violation is often captured continuously using a security information and event management system and notified to an admin. In order to differentiate between hostile behaviour and spurious reports, a SIEM system aggregates outputs from many sources and use event filtering algorithms. In order to track traffic to and from all networked devices, intrusion detection systems (IDS) are installed at one or more strategically located locations inside the network. Our study is on UNSW_NB15 dataset which comprises of different attacks.

Keywords:-- Security, Computer Network, System.

1. INTRODUCTION

Security is very essential and important need in today's digital world . Cyber security is the technique of protecting networks, electronic devices and data against malicious intrusions. Information security is the protection of internet-connected systems from online attacks. In today's society, the majority of data is in digital form and kept on internet-connected digital devices, cloud servers. In India there has been continuous increase in online payments. Customers have trust on the web applications and apps only when their data is secured and privacy is maintained. The field of computer science completely depends on cyber security , If there happens to be no security then no body will use those applications. The big gaint companies have made a positive trust and continuously improve their products from the threats. The attackers are using new techiques to get the information from server, cloud. Login information, encryption keys and banking data will be the majority of the data that is always at risk. Therefore, one must have an intrinsic security plan to safeguard the privacy of information. IDS using machine learning is a software that evaluates the packet data of incoming traffic and indicates whether or not it is a genuine packet. Whenever a malicious packet is received, a notification type signal is sent to admin.

An intrusion detection (IDS) checks the network traffic for unusual behaviour and issues notifications when it detects. It is software that scans a network or system for potentially dangerous behaviour and rule breaches. Generally, intrusions are of two varieties : Signature based and Anomaly based. The Signature based method can quickly identify threats for whom the patterns has been present in the system. The amount of bytes, number of ones, or number of zeros in the network traffic are only a few examples of the specific patterns. The Anomaly based method detects malware depending on deviations from normal behaviour. False negatives from IDS are an issue since they allow threats to easily pass through the system and network because they are mistaken as for genuine traffic. As a result of this issue, nobody will be aware of any intrusions that have occurred, which can occasionally pose major hazards, loss to the company. Our technique makes easier and reduces the cost of loss in the company.

In the project we have employed machine learning (ML) models, and the forecast was made using the model with the best accuracy. In this project, two different datasets were employed. One for training and the other dataset for testing. Prediction is made using the additional dataset. Several models, including the Decision tree classifier, Naïve Bayes logistic regression, gradient boosting classifier, and support vector machine (SVM) for supervised learning, were used in this research. The incoming packet information is categorised into 10 distinct categories of attack type. Our machine learning-based IDS built a method to distinguish 10 various forms of cyberattacks and malware with a detection rate of more than 89%.

1.1 Existing System

The current existing is signature-based intrusion detection system are being loaded with some fixed number of attack types and are used to detect those. The amount of bytes, number of ones, or number of zeros in the network traffic are only a few examples of the particular patterns that signature-based IDS uses to identify attacks. Additionally, it identifies based on the malware's already-known harmful instruction sequence. Signatures are the patterns that the IDS has identified. A network packet contains a lot of information, many mathematical calculations have to be done in order to process the code for a new attack category. In the current world the replications of new viruses, worms with very little configuration changes are becoming common in the cyber world.

1.1.1 Limitation in Existing System

- Developing such a system requires a lot of coding effort
- Detection of any type of new attack category becomes a very hectic task.
- In the current world the replications of new viruses, worms with very little configuration changes are becoming common in the cyber world. So, it is so hard to deliberate calculations for many intrusions.
- Cost of maintenance is expensive
- The Signature Library needs to be continually updated to detect the threats.
- It is incapable of identifying unknown threats or even known attack variations

1.2 Proposed System

We developed a machine learning-based intrusion detection with a very much reduced coding effort. Introducing a new anomaly or attack type into the system is very easy. Just training the system with the new attack category one time is enough. If any intrusions of that type happen the system detects it a good accuracy of approximately 90%. The system can perfectly determine which packet is normal. So whenever there happens any intrusion, the system may not detect the type of intrusion perfectly but it can perfectly differentiate the packet as an anomalous packet and raises an alert. True positivity rate of predicting a normal packet is high.

1.2.1 Advantages over Existing System

- The technique based on machine learning is more generic.
- Cost of maintenance is low.
- It is effective in identifying unknown threats or even known attack variations.
- Provides a additional layer of protection with more security.
- Developing such a system requires a less of coding effort

2. LITRATURE SURVEY

[1] “A critical review of intrusion detection systems in the internet of things”.The paper has given info on attackers that are targeting IoT devices.The IoT technology is evolving continuously in the digital world.It gives a comprehensive review of IDS and overview of techniques,validation strategy,deployment strategy.It provides up to date taxonomy with machine learning techniques to make IoT IDS.

[2] “Network Intrusion Detection System using Deep Learning”. This paper gave us insight and knowledge of deep learning techniques that are employed.We have learned different attack categories.It tells about the strengths and limitation of the security system that were developed without using the intelligence.

[3] “Network Intrusion Detection System using Neural Network”The paper gave us an insight on how the neural network learns from the raw data provided. The network adjusts the weights according to the target.The paper also tells that the attacks are evolving but the neural network solution provides a generalized way of identifying the IDS .

[4] “Intrusion Detection : A comprehensive review”.This paper gave us understanding on IDS and intrusion prevention,It is about the classification of IDS into wireless based.The network data is used to identify IDS.

[5] “A Review of Intrusion Detection System Using Neural Network and Machine Learning Technique This paper gives insights and clear understanding of the working of an IDS, what classifies an IDS and how is IDS different from other network security systems. It also gives perspectives on the different types of ids and their features and flaws. This paper also provided knowledge on how neural network perform on models trying to build an ids

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 IEEE Standard

The IEEE (Institute of Electrical and Electronics Engineers) has published a standard for software system requirements called IEEE Std 830-1998. This standard provides guidelines for writing and documenting software requirements specifications (SRS) for a software system.

The IEEE Std 830-1998 standard covers various aspects of software requirements such as: Introduction, General Description and Specific Requirements.

1. Introduction

1.1 Purpose of the requirements document

The purpose of the requirements document for the Intrusion Detection System (IDS) project using machine learning is to clearly define and document the expectations, specifications, and constraints of the IDS project. It serves as a comprehensive reference that outlines the desired functionalities, features, and characteristics of the IDS, and provides a clear understanding of what is expected from the system. The requirements document acts as a communication tool between different stakeholders, including project managers, developers, users, and other relevant parties, to ensure a shared understanding of the project's scope, objectives, and deliverables. It also serves as a basis for validating the system's performance, conducting testing and quality assurance, and managing changes or updates throughout the project's lifecycle. A well-defined requirements document is essential for guiding the development and implementation of the IDS, ensuring that it meets the needs and expectations of the stakeholders and achieves its intended goals.

1.2 Scope of the product

The scope of the product for the Intrusion Detection System (IDS) project using machine learning encompasses the boundaries and extent of the system's functionalities, features, and capabilities. It includes the specific goals and objectives of the IDS, the types of intrusions it is designed to detect, the network or system environments it is intended to operate within, and the expected outcomes and deliverables of the project. The scope may also define any limitations or exclusions of the IDS, such as the specific types of attacks or intrusions that may not be covered, the operational conditions under which the IDS may not be effective, and any dependencies or constraints that may impact the system's performance. Clarifying the scope of the product is essential for ensuring a clear understanding of the project's boundaries, goals, and deliverables among stakeholders, and for managing expectations throughout the project's lifecycle.

1.3 Definitions, acronyms and abbreviations

IDS - Intrusion Detection System: A security system that monitors and analysis network or system activities to detect and respond to potential unauthorized access, attacks, or intrusions.

ML - Machine Learning: A subset of artificial intelligence (AI) that involves the use of algorithms and statistical models to enable systems to learn from and make predictions or decisions based on data without being explicitly programmed.

NIDS - Network-based Intrusion Detection System: A type of IDS that monitors and analysis network traffic for signs of intrusions or attacks, typically deployed at the network perimeter or within the internal network to detect abnormal or malicious activities.

HIDS - Host-based Intrusion Detection System: A type of IDS that monitors and analysis activities on individual host systems, such as servers or endpoints, to detect potential intrusions or attacks that may not be visible at the network level.

SIEM - Security Information and Event Management: A system or platform that combines security information management (SIM) and security event management (SEM) capabilities to collect, correlate, and analyse security-related data from various sources, including IDS, log files, and other security tools, to detect and respond to security incidents in a centralized manner.

1.4 Overview of the remainder of the document

The remainder of this document provides a comprehensive overview of the Intrusion Detection System (IDS) project that utilizes machine learning techniques. It includes a detailed description of the project's objectives and scope, highlighting the importance of IDS in today's cybersecurity landscape. The document discusses the methodology employed in developing the IDS, including data collection, pre-processing, and feature extraction. It further elaborates on the machine learning algorithms used for intrusion detection, such as anomaly detection and classification. The evaluation metrics and performance analysis of the IDS are also presented, along with discussions on potential improvements and future directions. The document concludes with key findings and recommendations for further research and development in the field of intrusion detection using machine learning.

2. General description

2.1 Product perspective

The product perspective for the Intrusion Detection System (IDS) project using machine learning is to develop a robust and effective system for detecting and mitigating intrusions in a network or system. The IDS aims to provide real-time monitoring and analysis of network traffic or system logs to identify abnormal behaviours that may indicate potential intrusions or

cyber-attacks. The product perspective includes designing and implementing machine learning algorithms and techniques, integrating data collection and pre-processing modules, creating a user-friendly interface for system administrators, and ensuring scalability and performance of the IDS. The ultimate goal is to deliver a reliable and efficient IDS solution that enhances the security posture of the target system or network, protecting it against potential security breaches and minimizing the risk of unauthorized access or data compromise.

2.2 Product functions

The product functions of the Intrusion Detection System (IDS) project using machine learning include real-time monitoring of network traffic or system logs, anomaly detection using machine learning algorithms, classification of detected anomalies, data collection and pre-processing, reporting and visualization, scalability and performance, a user-friendly interface, integration and interoperability with existing security tools, maintenance and updates for adaptability, and ensuring security and privacy best practices are followed. The IDS aims to deliver a robust and efficient system for detecting and mitigating intrusions in real-time, providing comprehensive reports and visualizations, being scalable and interoperable, and adhering to security and privacy standards to enhance the security posture of the target system or network.

2.3 User characteristics

The user characteristics for the Intrusion Detection System (IDS) project using machine learning typically include system administrators or security personnel who are knowledgeable in network security and have experience in managing and securing computer networks or systems. These users are expected to have a good understanding of intrusion detection concepts, network protocols, and security best practices. They are skilled in interpreting the alerts and reports generated by the IDS, making decisions based on the severity and frequency of detected intrusions, and taking appropriate actions for incident response and mitigation. They may also have expertise in machine learning or data analysis, allowing them to fine-tune the IDS algorithms, perform model retraining, and optimize the system's performance. Overall, the users of the IDS are expected to have a strong technical background and proficiency in network security to effectively operate and manage the system.

2.4 General constraints

The general constraints for the Intrusion Detection System (IDS) project using machine learning may include limited or biased data availability, computational resource constraints, lack of model interpretability, security and privacy concerns, challenges in balancing false positives and

false negatives, time constraints for real-time processing, budget limitations, legal and regulatory compliance, human factors such as user acceptance and usability, and the dynamic nature of the threat landscape. These constraints need to be carefully considered and addressed to ensure the successful development and deployment of an effective IDS that can accurately detect and mitigate intrusions in real-time while adhering to security, privacy, and regulatory requirements, and meeting the operational needs of system administrators.

2.5 Assumptions and dependencies

Assumptions and dependencies for the Intrusion Detection System (IDS) project using machine learning may include the availability of labelled training data, dependency on effective feature engineering techniques and domain expertise, assumption of representative test data for evaluating system performance, dependency on accurate and reliable machine learning model performance, and the assumption of adequate computing resources for training and testing the IDS. These assumptions and dependencies play a crucial role in the effectiveness and accuracy of the IDS, and careful consideration and management of these factors are necessary during the development and deployment of the system to ensure its optimal performance in detecting and mitigating intrusions in real-world scenarios.

3. Specific Requirements

3.1 Functional Requirements

Functional requirements for the Intrusion Detection System (IDS) project using machine learning may include the ability to accurately detect and classify various types of network or system intrusions in real-time, provide alerts or notifications to system administrators or security personnel, support multiple machine learning algorithms for intrusion detection, handle large volumes of network traffic or system logs efficiently, provide a user-friendly interface for system configuration and management, allow customization of detection rules or policies, support integration with existing security infrastructure, provide logging and reporting capabilities for auditing and analysis purposes, and ensure system scalability, reliability, and performance to meet the operational needs of the organization. These functional requirements are essential to the successful implementation and operation of the IDS and its effectiveness in detecting and mitigating intrusions in a timely and accurate manner.

3.2 Non-Functional Requirements

Non-functional requirements for the Intrusion Detection System (IDS) project using machine learning may include aspects such as system performance, scalability, reliability, security, privacy, usability, and maintainability. System performance requirements may include response time, throughput, and resource utilization to ensure efficient processing of network traffic or

system logs in real-time. Scalability requirements may address the system's ability to handle increasing amounts of data and users over time. Reliability requirements may include system availability, fault tolerance, and error handling mechanisms. Security and privacy requirements may encompass data encryption, access controls, and compliance with relevant regulations. Usability requirements may focus on the user interface, system configuration, and management. Maintainability requirements may involve system documentation, code maintainability, and upgradeability. These non-functional requirements are essential for ensuring the overall quality, robustness, and reliability of the IDS during its development, deployment, and operation.

3.3 Interface Requirements

Interface requirements for the Intrusion Detection System (IDS) project using machine learning may include the design and implementation of interfaces between various system components or modules, as well as interfaces with external systems or users. This may include interfaces between data sources, such as network sensors or system logs, and the IDS for data collection and pre-processing. It may also include interfaces between different machine learning algorithms or models for classification or anomaly detection. Interfaces with external systems or users may involve alert notifications or reporting mechanisms to system administrators or security personnel. These interface requirements need to be carefully designed and implemented to ensure seamless communication and interoperability among different components of the IDS, as well as effective interaction with external systems or users for smooth operation and management of the system.

3.4 Feasibility Study

A feasibility study for the Intrusion Detection System (IDS) project using machine learning is a systematic evaluation of the project's viability and potential success. It typically involves assessing various aspects, including technical, economic, operational, legal, scheduling, and resource considerations, to determine if the project is feasible and worth pursuing. Technical feasibility examines the system's technical requirements, such as the availability of suitable machine learning algorithms, data sources, and computing resources. Economic feasibility evaluates the financial implications of the project, including costs, benefits, and return on investment. Operational feasibility assesses the system's practicality and compatibility with existing infrastructure and processes. Legal feasibility considers compliance with relevant laws, regulations, and ethical considerations. Scheduling feasibility involves determining the project timeline and deadlines. Resource feasibility assesses the availability of necessary resources, including human expertise, tools, and technologies. A thorough feasibility study helps project stakeholders make informed decisions about the project's viability, risks, and potential benefits.

3.4.1 Technical Feasibility

Technical feasibility for the Intrusion Detection System (IDS) project using machine learning refers to the assessment of the system's technical requirements and capabilities to determine if they can be effectively implemented and integrated into the existing environment. This may include evaluating the availability and suitability of machine learning algorithms for intrusion detection, the availability of labelled training data for model training, the performance and scalability of the system in handling large volumes of network traffic or system logs, the availability of adequate computing resources for training and testing the machine learning models, and the compatibility of the IDS with existing hardware, software, and networking infrastructure. Technical feasibility is crucial to ensure that the IDS can be effectively implemented and operated within the technical constraints of the environment, and that it can meet the performance, scalability, and reliability requirements for effective intrusion detection.

3.4.2 Economic Feasibility

Economic feasibility for the Intrusion Detection System (IDS) project using machine learning refers to the evaluation of the financial implications of the project to determine if it is economically viable and financially justifiable. This may include assessing the costs associated with the development, deployment, operation, and maintenance of the IDS, such as hardware, software, data storage, personnel, and training costs. It also involves analysing the potential benefits or returns on investment (ROI) of the IDS, such as the anticipated reduction in losses due to security breaches, cost savings from early detection and prevention of intrusions, and potential revenue gains from improved customer trust and confidence. Economic feasibility also considers factors such as the project's funding sources, budget constraints, and the availability of financial resources to support the project throughout its lifecycle. A thorough economic feasibility analysis is critical to ensure that the project is financially viable and can provide a positive economic impact for the organization, taking into account both the costs and benefits associated with the IDS implementation.

3.4.3 Operational Feasibility

Operational feasibility for the Intrusion Detection System (IDS) project using machine learning refers to the evaluation of the practicality and compatibility of the system with existing operational processes, procedures, and resources. It involves assessing whether the IDS can be effectively integrated into the existing operational environment without disrupting ongoing operations or requiring significant changes to existing processes. This may include evaluating the ease of deployment and integration of the IDS within the organization's network architecture, the compatibility of the IDS with existing security policies and procedures, the

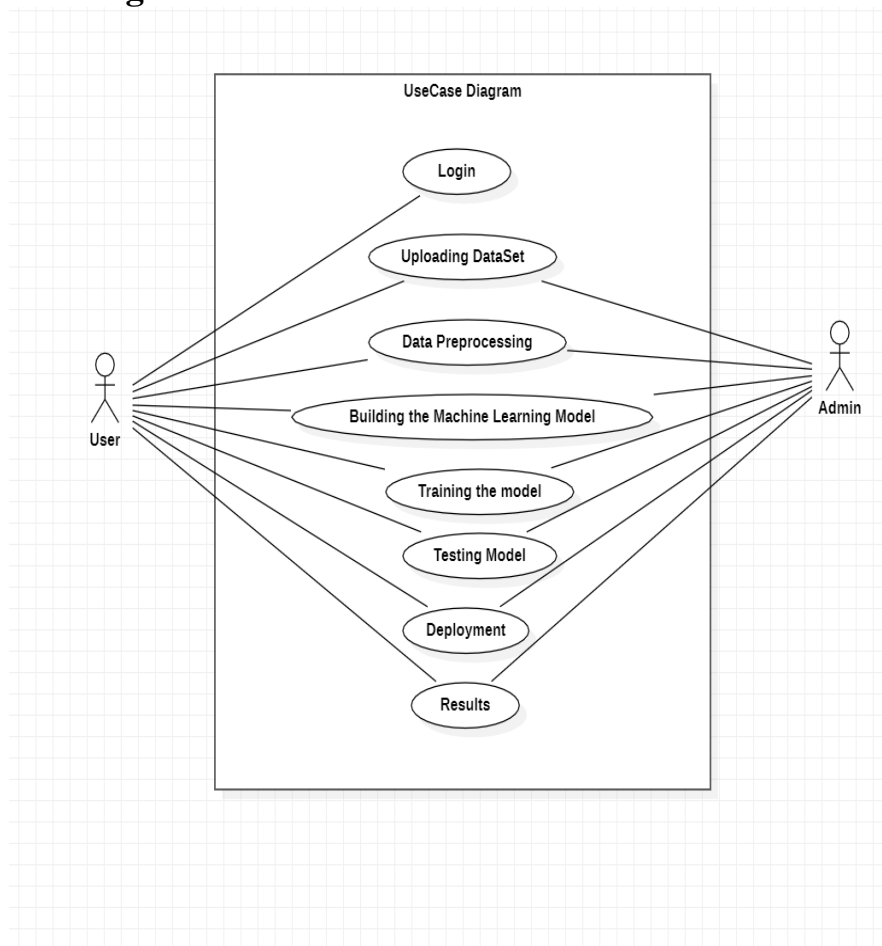
availability of skilled personnel to operate and maintain the IDS, and the impact of the IDS on day-to-day operations. Operational feasibility is crucial to ensure that the IDS can be effectively implemented and integrated into the existing operational workflows, and that it can be operated and maintained in a practical and efficient manner without negatively impacting the organization's ongoing operations.

4.DESIGN

4.1 Project Description

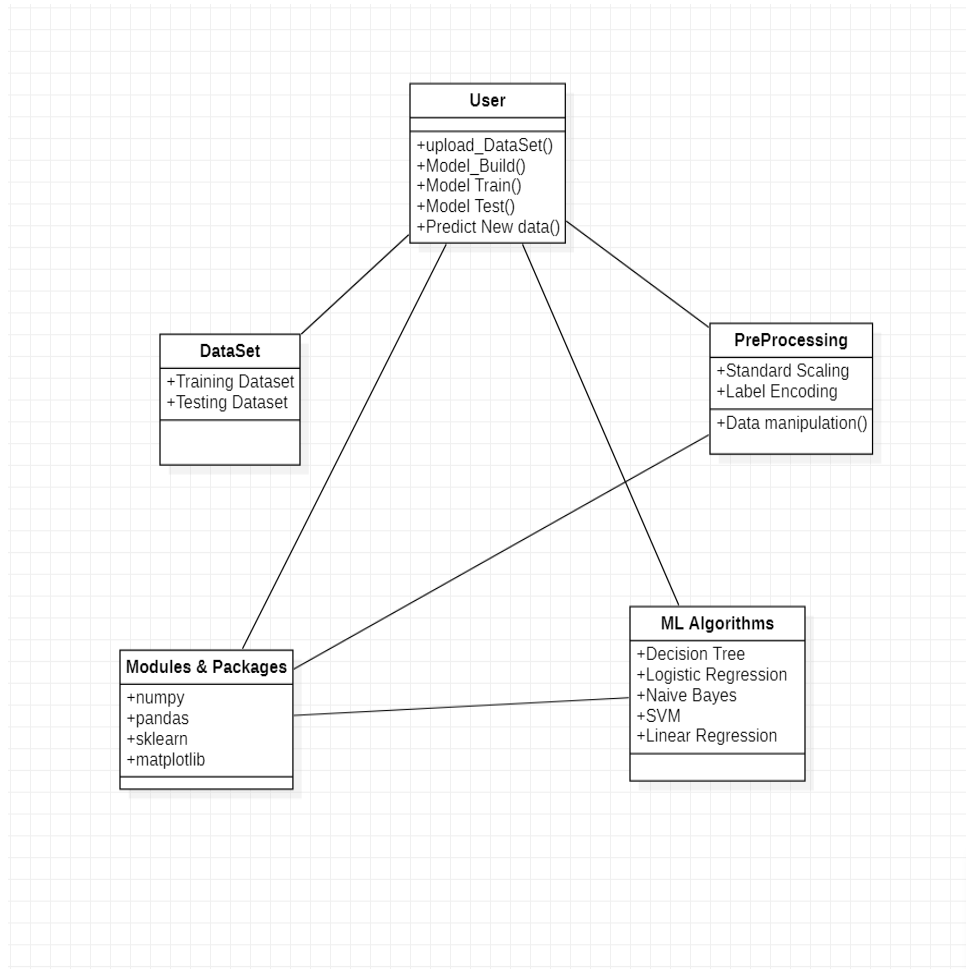
The first step in designing an IDS is to identify the types of threats that the system should detect. This can include malware, unauthorized access attempts, or other suspicious behavior. Overall, designing an IDS requires careful consideration of the organization's security needs, as well as the available technologies and resources. With a well-designed IDS in place, organizations can improve their ability to detect and respond to potential security threats. We use UML diagrams to represent the design.

4.1.1 Use Case Diagram



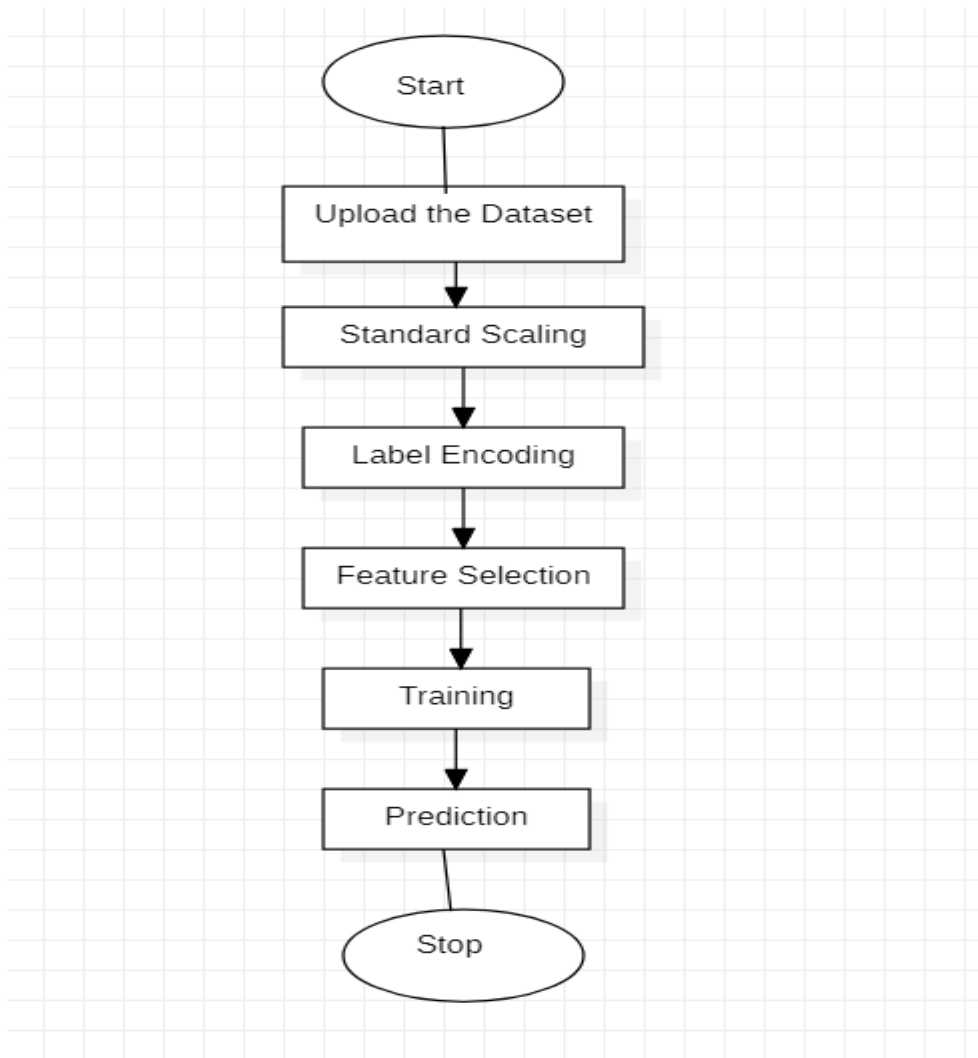
In the above use case diagram we have user and admin, the user has the data collected over the network and uploads into our website, now the data is processed and the ML algorithm detects any intrusions and the results are displayed.

4.1.2 Class Diagram



The relationships between these classes can be represented using various types of associations, such as inheritance, composition. It provides a high-level view of the system's components and their relationships. This diagram can be useful for understanding the system's structure, identifying potential areas for improvement,

4.1.3 Data Flow Diagram



In the above data flow diagram we can see how the data is being processed in the intrusion detection . A DFD is an important diagram for developing and maintaining effective intrusion detection systems, as it provides a clear and detailed picture of data flows through the system and can help to identify potential vulnerabilities and sources of false alarms.

5.IMPLEMENTATION

5.1 Source Code (Main)

Importing Modules

```
# Importing Libraries
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd #Data manipulation and analysis
import numpy as np #Performs high level manipulation
import sklearn # provides efficient tools for predictive data analysis

# Preprocessing purpose
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder

# For getting the importances
from sklearn.ensemble import RandomForestClassifier

# Feature Extraction
from sklearn.decomposition import PCA

# Splitting Data
from sklearn.model_selection import train_test_split

# For accuracy, Classification Report, Confusion Matrix
from sklearn import metrics

# For training different ML models
from sklearn import tree
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import BernoulliNB
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVC
from sklearn.ensemble import GradientBoostingClassifier

# Ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

Uploading DataSet

```
# Uploading datasets for training, testing and prediction
train = pd.read_csv('E:/MajorProject/Code/DataSet1/UNSW_NB15_training-set.csv')
test = pd.read_csv('E:/MajorProject/Code/DataSet1/UNSW_NB15_testing-set.csv')
```

Preprocessing

Standard Scaling

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

# extract numerical attributes and scale it to have zero mean and unit variance
cols = train.select_dtypes(include=['float64', 'int64']).columns
sc_train = scaler.fit_transform(train.select_dtypes(include=['float64', 'int64']))
sc_test = scaler.fit_transform(test.select_dtypes(include=['float64', 'int64']))

# turn the result back to a dataframe
sc_traindf = pd.DataFrame(sc_train, columns = cols)
sc_testdf = pd.DataFrame(sc_test, columns = cols)
```

Label Encoding

```
LE = LabelEncoder()

# extract categorical attributes from both training and test sets
obj_train = train.select_dtypes(include=['object']).copy()
obj_test = test.select_dtypes(include=['object']).copy()
#print(obj_train)
#print(obj_pred)

# encode the categorical attributes
LE_obj_train = obj_train.apply(LE.fit_transform)
LE_obj_test = obj_test.apply(LE.fit_transform)

# separate target column from encoded data
enctrain = LE_obj_train.drop(['attack_cat'], axis=1)
#print(enctrain)
encntest = LE_obj_test.drop(['attack_cat'], axis=1)
#print(encntest)
test_target = test['attack_cat']
#lir_tar_train = LE_obj_train['attack_cat']
```

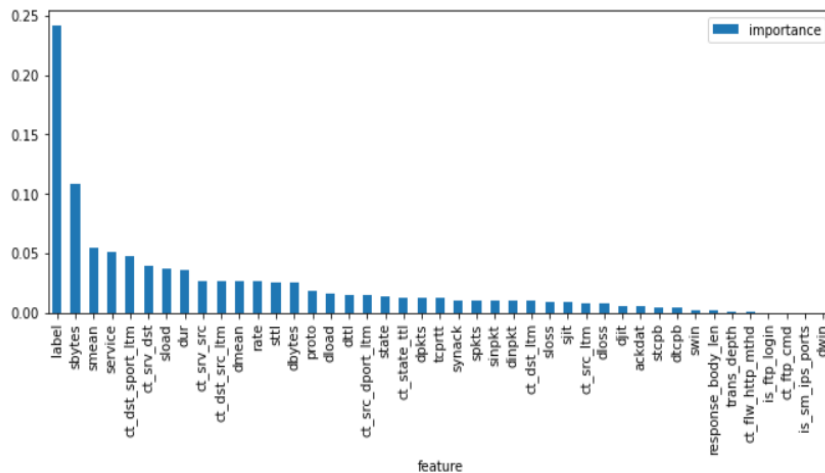
Feature Selection

```
rfc = RandomForestClassifier();

# fit random forest classifier on the training set
rfc.fit(train_x, train_y);

# extract important features
score = np.round(rfc.feature_importances_,3)
importances = pd.DataFrame({'feature':train_x.columns,'importance':score})
importances = importances.sort_values('importance',ascending=False).set_index('feature')

# plot importances
plt.rcParams['figure.figsize'] = (11, 4)
importances.plot.bar();
```



```
from sklearn.feature_selection import RFE
import itertools
#rfc = RandomForestClassifier()

# create the RFE model and select 10 attributes
rfe = RFE(rfc, n_features_to_select=12)
rfe = rfe.fit(train_x, train_y)

# summarize the selection of the attributes
feature_map = [(i, v) for i, v in itertools.zip_longest(rfe.get_support(), train_x.columns)]
selected_features = [v for i, v in feature_map if i==True]

selected_train = train_x.loc[:, selected_features]
#print()
selected_test = test_df.loc[:, selected_features]

selected_features
```

Building Machine learning models for comparative study

```
X_train,X_test,Y_train,Y_test = train_test_split(selected_train,train_y,train_size=0.80, random_state=2)

#Training different Machine Learning models for comapritive analysis
DTC_Classifier = tree.DecisionTreeClassifier(criterion='entropy', random_state=0) #Decision Tree Classifier

LGR_Classifier = LogisticRegression(n_jobs=-1, random_state=0) #Logistic Regression

BNB_Classifier = BernoulliNB() #Naive Bayes Algorithm

gradient_booster = GradientBoostingClassifier(learning_rate=0.1)

SVM_Classifier = SVC(kernel = 'poly',C = 75) #Support Vector Machine


DTC_Classifier.fit(X_train, Y_train)

LGR_Classifier.fit(X_train, Y_train)

BNB_Classifier.fit(X_train, Y_train)

gradient_booster.fit(X_train,Y_train)

SVM_Classifier.fit(X_train,Y_train)
```

Training

```
for i, v in models:
    accuracy = metrics.accuracy_score(Y_train, v.predict(X_train))
    confusion_matrix = metrics.confusion_matrix(Y_train, v.predict(X_train))
    classification = metrics.classification_report(Y_train, v.predict(X_train))
    print()
    print('===== {} Model Evaluation ====='.format(i))
    print("Model Accuracy:" "\n", accuracy)
    print()
    print("Confusion matrix:" "\n", confusion_matrix)
    print()
    print("Classification report:" "\n", classification)
    print()
```

Testing

```
for i, v in models:
    accuracy = metrics.accuracy_score(Y_test, v.predict(X_test))
    confusion_matrix = metrics.confusion_matrix(Y_test, v.predict(X_test))
    classification = metrics.classification_report(Y_test, v.predict(X_test))
    print()
    print('===== {} Model Test Results ====='.format(i))
    print()
    print("Model Accuracy:" "\n", accuracy)
    print()
    print("Confusion matrix:" "\n", confusion_matrix)
    print()
    print("Classification report:" "\n", classification)
    print()
```

Prediction Interface

```
.box input[type="submit"]:hover {
    background: #347736;
}

</style>
</head>
<body>
    <h1>Intrusion Detection in Networks</h1>
    <form class="box" action="{% url 'result' %}">
        {% csrf_token %}

        <p>Total Duration : </p>
        <input type="number" name="dur" step="any">
        <br>

        <p>Source Bytes : </p>
        <input type="number" name="sbytes" step="any">
        <br>

        <p>Destination Bytes : </p>
        <input type="number" name="dbytes" step="any">
        <br>
```

5.2 Screen shots (Results)

===== Decision Tree Classifier Model Test Results :

Model Accuracy:

0.887004048582996

Confusion matrix:

```
[[ 15   9  42 104  35   0   0   0   0   0]
 [  4   5  15 101  46   2   0   1   0   0]
 [ 11   4 646 505  59  20   0   5   8   1]
 [ 28  15 655 2309 156  54   0  74  18   5]
 [ 24  16  85  281 1400  10   0   6   8   0]
 [  0   4  27   63  22 5536   0   3   5   0]
 [  0   0   0   0   0   0 11098   0   0   0]
 [  0   2  78  105   6   2   0  837   5   0]
 [  0   1  14   20  11   3   0   1  60   0]
 [  0   0   4   4   2   2   0   0   0   3]]
```

Classification report:

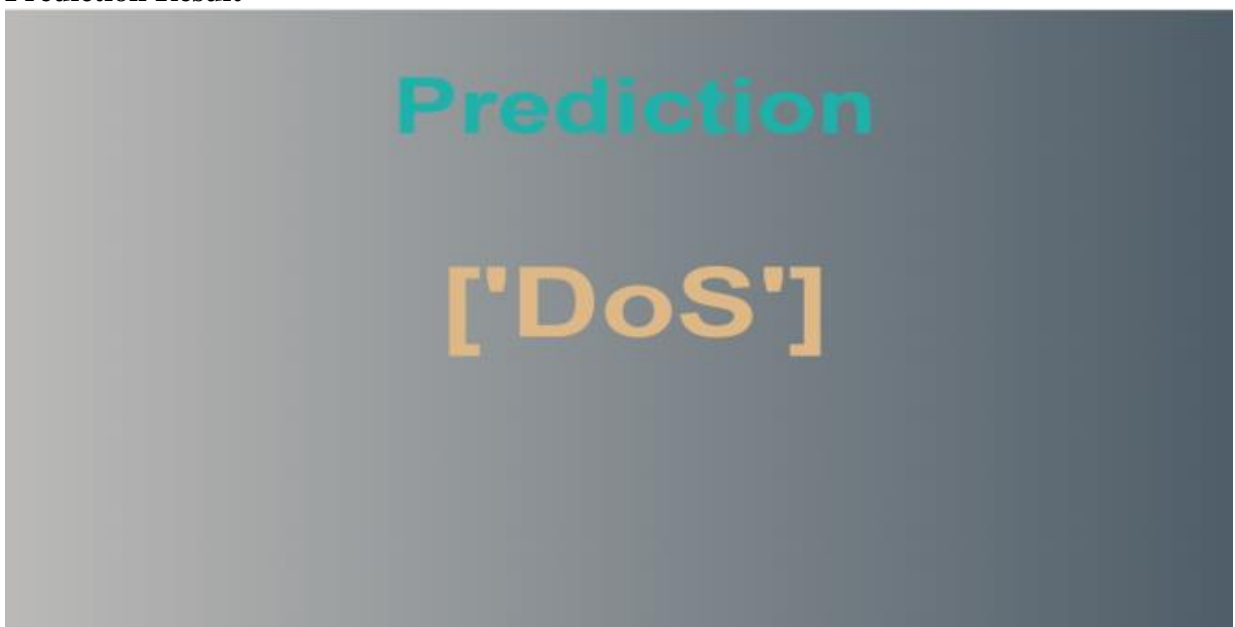
	precision	recall	f1-score	support
Analysis	0.18	0.07	0.10	205
Backdoor	0.09	0.03	0.04	174
DoS	0.41	0.51	0.46	1259
Exploits	0.66	0.70	0.68	3314
Fuzzers	0.81	0.77	0.78	1830
Generic	0.98	0.98	0.98	5660
Normal	1.00	1.00	1.00	11098
Reconnaissance	0.90	0.81	0.85	1035
Shellcode	0.58	0.55	0.56	110
Worms	0.33	0.20	0.25	15

Prediction Interface



The image shows a web interface titled "Intrusion Detection in Networks" in green text at the top. Below the title is a dark gray rectangular box containing four input fields. The first field is labeled "Total Duration :" in red text. The second field is labeled "Source Bytes :" in red text. The third field is labeled "Destination Bytes :" in red text. The fourth field is labeled "Rate :" in red text. All labels are positioned above their respective input boxes.

Prediction Result



The image shows a dark gray rectangular box with the word "Prediction" in large, bold, teal text at the top. Below it, the text "['DoS']" is displayed in large, bold, orange text.

6.TESTING

Test Cases

Test Scenario : Numeric values

Pre-condition : Processing the data

Expected Results : Predict if there was an intrusion in the network and its type

Actual Results : The actual label of the data taken from dataset.

Test Case ID	Test Data (file size kb)	Expected Results	Actual Results	Status Pass/Fail
1	2	Shellcode	Shellcode	Pass
2	1.1	Fuzzers	Fuzzers	Pass
3	1.6	DoS	DoS	Pass
4	1	Reconnaissance	Reconnaissance	Pass
5	2	Exploits	DoS	Fail
6	0.9	Worms	Worms	Pass
7	1	Fuzzers	Fuzzers	Pass
8	0.8	Exploits	Exploits	Pass
9	1	DoS	Fuzzers	Fail
10	1.3	Normal	Normal	Pass

7. CONCLUSION & FUTURE SCOPE

We have used machine learning based approach, and were able to identify the intrusions and find the type of attack. The models will work with good accuracy on generalized data and can identify normal connections also. We have worked on the UNSW_NB15 dataset taken from kaggle, which comprises of 10 different types of attacks. We have build the 5 ML models using different algorithms and deployed the best on user interface . The project can be deployed at strategic points on network.

The future of intrusion detection lies in the development of more advanced and intelligent systems that can adapt to changing threat landscapes. Another area of future development is the integration of intrusion detection with other security systems, such as firewalls and access control systems. By combining these technologies, organizations can create a comprehensive security infrastructure. As technology continues to advance it is essential for ensuring the security of our digital infrastructure.

8. REFERENCES

1. <https://www.n-able.com/blog/intrusion-detection-system>
2. <https://towardsdatascience.com/building-an-intrusion-detectionsystem-using-deeplearning- b9488332b321>
3. <https://www.codespeedy.com/intrusion-detection-model-usingmachine-learningalgorithm-in-python/>
4. https://www.researchgate.net/publication/281451813_Using_Machine_Learning_in_Networks_Intrusion_Detection
5. <https://cybersecurity.springeropen.com/articles/10.1186/s42400-019-0038-#availabilityof-data-and-materials>
6. <https://www.geeksforgeeks.org/intrusion-detection-systemusing-machine-learningalgorithms>
7. <https://www.geeksforgeeks.org/best-python-libraries-formachine-learning>
8. <https://anderfernandez.com/en/blog/code-decision-tree-pythonfrom-scratch>
9. <https://arxiv.org/ftp/arxiv/papers/2101/2101.05067>
10. <https://www.geeksforgeeks.org/principal-componentanalysis-with-python/>

9. LIST OF FIGURES

Figure No	Figure Title	Page No
1	Use Case diagram	11
2	Class diagram	12
3	Data flow diagram	13
4	Prediction Interface	18
5	Prediction Result	18

LIST OF TABLES

Table No	Table Title	Page No
1	Test Cases	19

ACRONYMS

IDS – Intrusion Detection System

ML – Machine Learning

NIDS – Network based Intrusion Detection