International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Intrusion Detection Using Machine Learning: A Random Forest-Based Approach

Ch. Sai Sampath[1], Dr. P. Anuradha[2]

[1]Student, CSE, GITAM School of Technology, Visakhapatnam, India
[2]Associate Professor, GITAM School of Technology, Visakhapatnam, India

Abstract

It can be very difficult and time-consuming to pinpoint network traffic behaviours that frequently disrupt services. A researcher must check out all the massive and off-course data to discover the chain of network link interruptions. A system administrator is to be notified by an intrusion detection system (IDS) each time an intruder attempts to breach the network. Using a table of harmful signatures, an IDS that has been mishandled inhibits attacks. An alarm is triggered if a persistent exercise matches a signal on the chart. These kinds of systems are used by countless organisations and institutions worldwide. They are simple to use, let administrators tailor the sign table, and help identify the real facts of events. An Intrusion Detection System (IDS) has been developed that employs various machine intelligence techniques to automatically identify assaults on intricate networks and systems. Principal component analysis (PCA), along with several classification algorithms including Support Vector Machines, Random Forest, and K-Nearest Neighbor, is used in an effort to increase the capabilities of IDS. Attack detection is the principal function of an intrusion detection system. Nonetheless, identifying intrusions as soon as possible will help to lessen their damage.

Keywords: Principal Component Analysis, Ensemble Methodologies, Anomaly Detection, Intrusion Detection, and Supervised Learning.

1. Introduction

Telecom development in the 20th century evolved through a number of key phases, from circuit and packet switched networks to all-IP networks that are solely based on the Internet. This transition has resulted in a unified environment in which communication between applications and services, such as voice and data, is carried out over IP.

Despite the fact that the advancement of communication networks has resulted in improved technological qualities, it has also resulted in unfavourable possibilities. Radio access networks are increasingly susceptible to threats that were previously exclusively present in mounted networks. Once it was recognized that these threats are becoming more sophisticated and well-developed. As a result, the defence systems require more intelligence.

Several robust security solutions, such as firewalls and malware scanners, are ineffective against the wide range of sophisticated web attacks that are on the rise. Network security can be improved by including intrusion detection systems in the levels of protection.

Systems and networks are impacted by multiple threats. This system or network is aware of attacks such as hollow attacks and gray hole attacks. Such network or system assaults ultimately try to take the data and information from the network or system being attacked.

Intrusion detection systems (IDS) were developed in an effort to prevent such attacks from occurring. An intrusion detection system can be used to monitor and stop these kinds of attacks.

2. Literature Review

[1] "A Proposed Wireless Intrusion Detection Prevention and Attack System", by Jafar Abo Nada and Mohammad Rasmi Al-Mosa:

In this paper, the authors propose a wireless intrusion detection prevention and attack system that aims to enhance the security of wireless networks. The authors highlight the vulnerabilities of wireless networks and the need for effective intrusion detection and prevention systems to address these vulnerabilities. They discuss the different types of attacks that can target wireless networks, including denial-of-service (DoS), man-in-the-middle (MITM), and rogue access point (RAP) attacks. The authors then present the proposed system, which consists of three main components: a wireless IDS sensor, a wireless prevention unit, and a wireless attack unit. The IDS sensor is responsible for detecting and analysing network traffic, while the prevention and attack units are designed to prevent and respond to attacks, respectively. The authors describe the implementation of the proposed system and present experimental results showing its effectiveness in detecting and preventing attacks on wireless networks.

[2] "Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm", by Kinam Park, Youngrok Song, and Yun-Gyung Cheong:

The prime focus is on developing a machine learning algorithm for intrusion detection systems (IDS). The authors discuss the limitations of traditional rule-based IDS approaches, which need help to detect complex and evolving attack patterns. They highlight the potential of machine learning algorithms to improve the accuracy and effectiveness of IDS. The authors present the proposed system, which uses a Random Forest algorithm to classify attacks based on features extracted from network traffic data. They describe the experimental setup and present results showing the effectiveness of the proposed system in detecting and classifying different types of attacks, including denial-of-service (DoS), probing, user-to-root (U2R), and remote-to-local (R2L) attacks. The authors also discuss the scalability of the proposed system and its potential for real-time intrusion detection in large-scale network environments.

[3]"Deep Learning-Based Intrusion Detection for IoT Networks" by Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, and Antonio Robles-Kelly:

The paper proposes a deep learning-based intrusion detection system for IoT networks using a convolutional neural network (CNN) architecture. The system aims to improve detection accuracy and reduce false alarms compared to traditional methods. The authors conducted experiments using the CICIDS2017 dataset and achieved promising results, outperforming some state-of-the-art methods. Overall, the paper highlights the potential of using deep learning techniques for intrusion detection in IoT networks.

3. Problem Statement and Objectives

Problem Identification:

Models in the current system are introduced using human detection or rule-based detection.

To detect such fraudulent activity, substantial costs and a large number of competent professionals are needed. The last thing is that intrusions are more likely to become apparent after the attack has begun rather than prior to it. This challenge has the following drawbacks: low precision, time consuming, high complexity, difficult to scale, and expensive.

Objective:

For intrusion detection this system employs Machine Learning Algorithms such as Support Vector Machines, Random Forest, XgBoost, and K-Nearest Neighbor. These strategies can detect the likelihood of an attack faster than existing methods, allowing an even more rapid response to a threat. Principal Component Analysis is utilised to minimise the large cardinality in this system.

4. System Methodology

Design:

Intrusion Detection System is the system that will be created, and users can access it. Users can enter information about themselves, such as the duration, src_bytes, dst_bytes, logged_in, count, srv_count, dst_host_count, protocol type, service, flag and the server will gather the information, extract its features, match the values, classify it, and ultimately predict the intrusion and provide a report to the user.

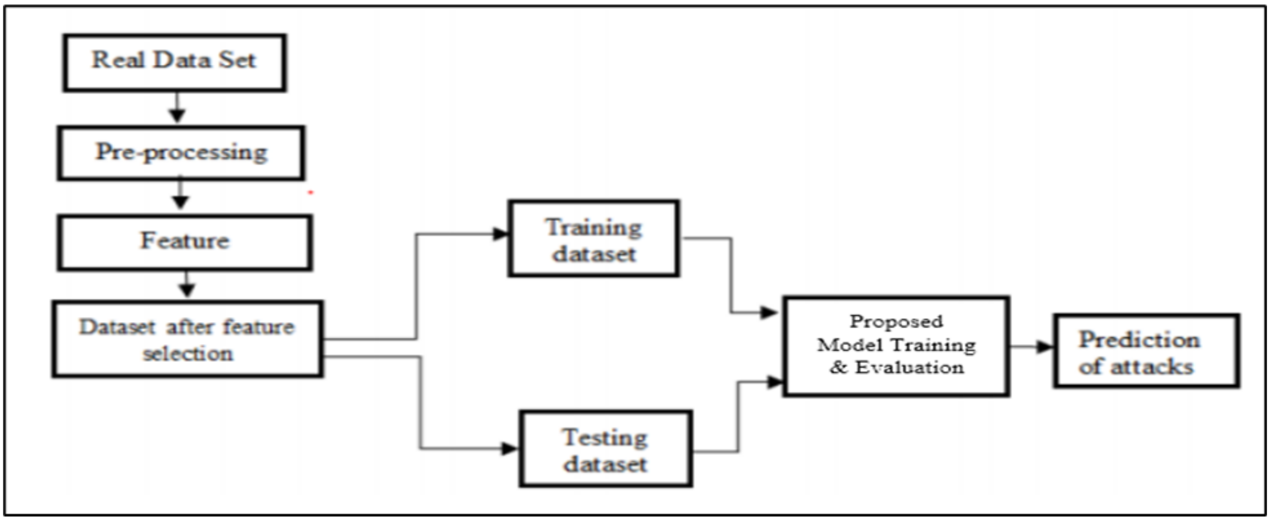


Fig 1: Flowchart

System Side Process:

Store Dataset: The user-provided dataset is stored by the system.

Model Training: The System gets the data from the user and feeds that data to the selected model.

Model Predictions: The system uses the information provided by the user to forecast the results.

User Side Process:

Load Dataset: The user can load the dataset on which he or she want to work.

View Dataset: The dataset can be viewed by the user.

Model selection: The user can apply the model to the dataset to test its correctness.
Evaluation: Model performance can be evaluated by the user.

Advantages:
There are various advantages to using intrusion detection systems (IDS), including:

Identifying both known and unknown attacks: IDS can detect both known (signature-based detection) and previously unknown (anomaly-based detection) threats by monitoring network traffic and finding patterns that are outside of the system's typical behaviour.

Real-time monitoring: IDS can monitor network traffic in real time and notify security workers of potential attacks as they occur. This can help to mitigate the impact of an attack and shorten the response time of security teams.

IDS can also help to improve network performance by identifying and correcting issues that can slow down network traffic or create network downtime.

5. Overview of Technologies

PYTHON:

Python is an advanced programming language. It is a multilingual, object-oriented, interpreted language. Python was invented between 1984 and 1989 by Guido van Rossum. Python, like Perl, has its source code accessible under the GNU General Public License.
Python has some highly useful features, including: The interpreter in Python processes programs at runtime. To make scripts, it is simple to interface directly with the interpreter. It is also an example of Object-Oriented programming. Those who are new to programming will benefit greatly from this language.

PANDAS:

Pandas provides us with a plethora of series and Data Frames. It allows us to easily explore, organise, represent, and manipulate data.Pandas smart alignment and indexing functions provide ideal data structure and labelling.
Pandas' particular features enable us to handle missing data or values with associated measurements.Pandas has very clear code that is simple to deal with. Individuals with no programming experience can easily utilise and work with it.
Pandas also provides users with a diverse set of built-in capabilities for reading and writing data in various databases. It also aids in reading and writing to many online services and data structures. With the help of Pandas, it is also possible to integrate multiple databases.

NUMPY:

Numpy is a sophisticated mathematical implementation for large data sets. Numpy makes any procedure involving such a large amount of data simple and painless.

It supports both normal array objects and masking arrays. It includes various functions such as using and altering logical forms, general linear algebra, and many others.

For integration, the Numpy package contains a number of essential utilities. Numpy is simple to integrate with the C, C++, and FORTRAN programming languages.

Numpy can alter any N-dimensional array by producing updated arrays and deleting older ones as they change.

The functions of MATLAB are similar to those of NUMPY. Numpy and MATLAB both accelerate operations.

FLASK:

Flask is a popular Python web framework that helps developers to create online apps rapidly. It is a lightweight framework that prioritizes simplicity and versatility, making it an ideal choice for designing small to medium-sized applications.

Flask is based on the Werkzeug toolkit and the Jinja2 template engine and comes with a built-in development server and debugger. Extensions that add functionality, such as database connectivity, user authentication, and form validation, are also supported.

Flask uses the Model-View-Controller (MVC) architectural design, however developers can easily utilize alternative patterns. Flask is popular among developers because of its simplicity, versatility, and ease of usage.

SKLEARN:

Scikit-learn/Sklearn is one of Python's most useful machine learning libraries. It is also accessible as open source software.

A variety of ML algorithms are currently available with default parameters in the SKlearn machine learning library, so they will work right away.

Regression, dimensionality reduction, classification, and clustering are some of the most useful and widely used methods. They are primarily employed in machine learning and statistical modeling.

SKlearn is one of the most user-friendly and dependable solutions for predictive data analysis.

Sklearn is based on the Python libraries NumPy, SciPy, and Matplotlib.

ALGORITHMS USED:

1. RANDOM FOREST REGRESSION:

This is a classification learning procedure that employs a regression function. During the training phase, it includes a number of different decision trees in order to achieve a common result/output. It is used to distinguish between items by comparing their properties and calculating the mean. This strategy corrects the problem of overfitting. Decision trees outperform random forests in terms of performance, but their accuracy suffers in contrast to gradient-boosted trees. This performance is determined by the quality of the data.

When all of the decision trees are joined, the variation is modest. Individually, it has an extensive variety of variations. While resolving the classified issue, the ultimate decision is determined by taking into account the majority of the classifier's votes. This is an ensemble method for performing classification and regression tasks.

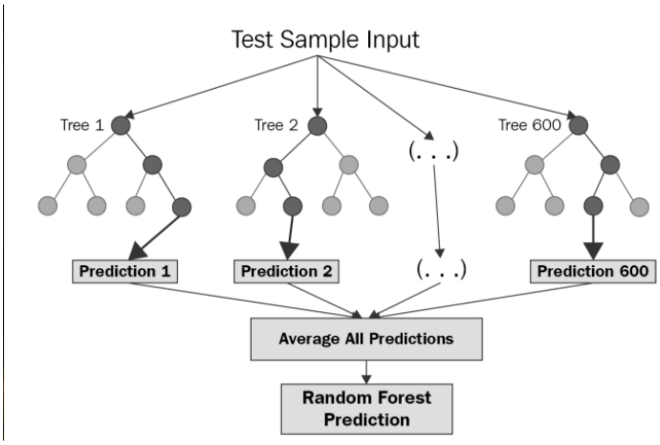


Fig 2: Random Forest Regression

2. SUPPORT VECTOR MACHINES:

To determine the data points on a hyperplane, this classification technique is employed. This is accomplished in an N-dimensional space. SVM is an abbreviation for Support vector machine. In terms of categorizing the data and generating a standard deviation for it, this performs the same tasks as Random Forest Regression. It is primarily used for classification. Data points that fall on the same hyperplane are deemed to be in the same class, whereas data that do not are ignored. This improves accuracy.

To precisely pinpoint the location and inclination of the hyperplane, support vectors are required. This is used to expand the categorization method's boundary. The plane's orientation will be adjusted by eliminating this process.

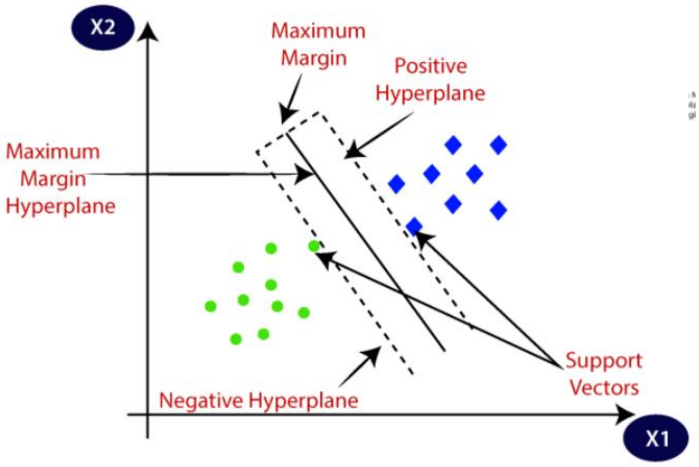


Fig 3: Support Vector Machines

3. DECISION TREES:

A decision tree is a common machine learning technique that can be used for classification and regression applications. It's a tree-like structure in which each core node represents a decision based on a feature's value and each leaf node represents a predicted output or a class label.

The decision tree algorithm constructs the tree by recursively partitioning the data at each node based on the feature that maximises information gain or lowers impurity. This method is repeated until all of the data in a leaf node is of the same class. Or when the number of instances in a leaf node hits a predefined minimum.

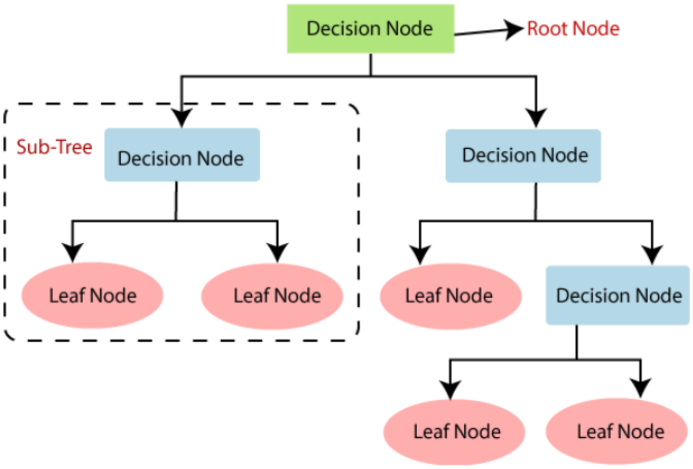


Fig 4: Decision Tree

4. K-NEAREST NEIGHBOR:

The K Nearest Neighbor (KNN) algorithm is a basic and effective machine learning approach that can be used for classification and regression applications. It is a non-parametric algorithm, which means it makes no assumptions about the data's underlying distribution.

The KNN method finds the K data points in the training set that are closest to the input data point. Based on a distance measure such as Euclidean distance, this is calculated. A majority vote of the K nearest neighbors is then used to decide the expected output or class label.

K is a hyperparameter that can be tweaked to improve the model's performance. A low K number may result in overfitting, whereas a high K value may result in underfitting.

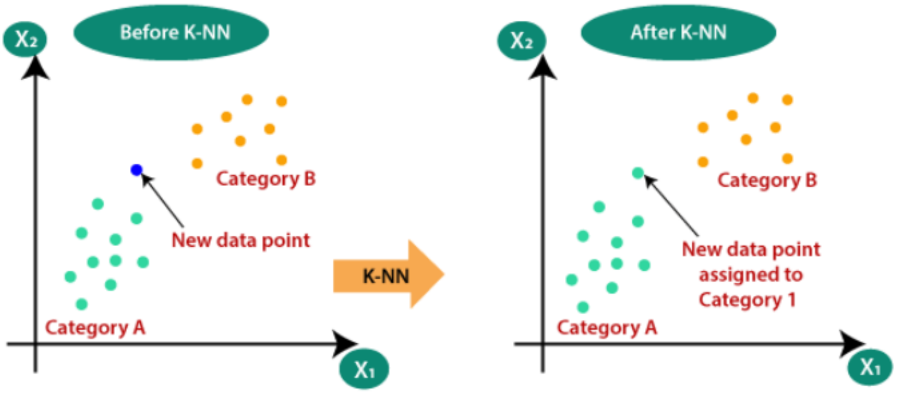
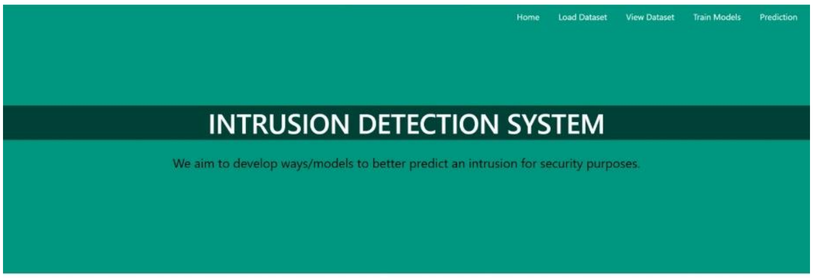


Fig 5: K-Nearest Neighbor

6. Results and Discussions

Four machine learning classifiers were used to train the model and took into account the following features:'src bytes', 'duration', 'logged in', 'count','srv count', 'dst host count', 'protocol type', 'dst bytes', 'service', and 'flag' check for any intrusion identified.

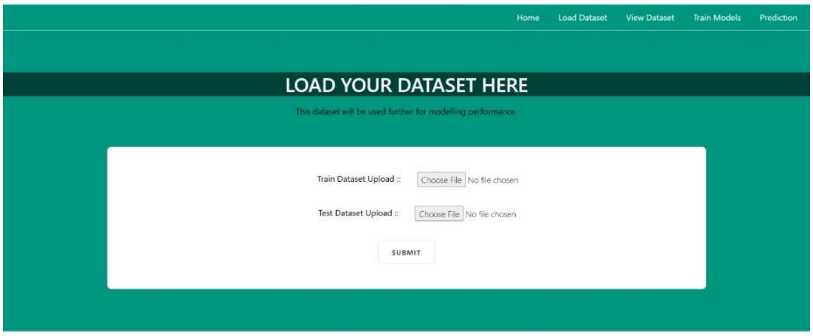
UI OUTPUT:



INTRUSION DETECTION SYSTEM

We aim to develop ways/models to better predict an intrusion for security purposes.

Fig 6: Home page



LOAD YOUR DATASET HERE

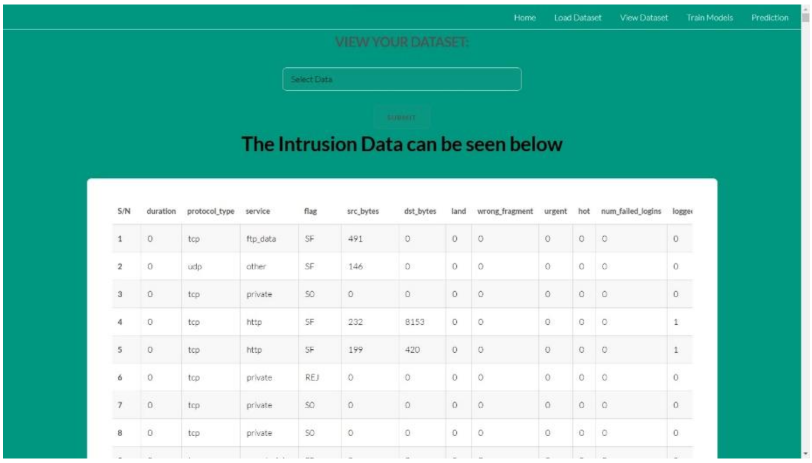
This dataset will be used further for modelling performance.

Train Dataset Upload :: Choose File No file chosen

Test Dataset Upload :: Choose File No file chosen

SUBMIT

Fig 7: Dataset Uploading page

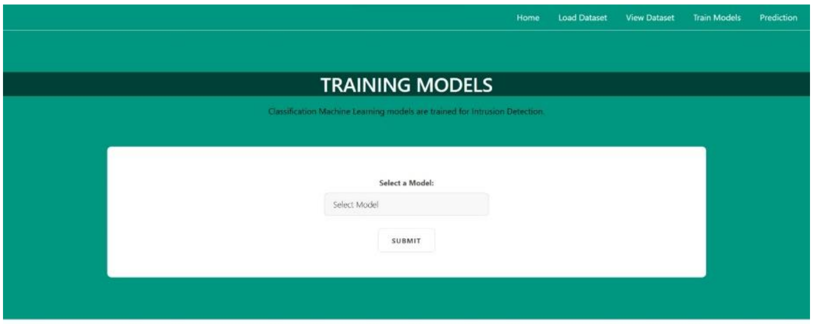


VIEW YOUR DATASET:

The Intrusion Data can be seen below

S/N	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logge
1	0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0
2	0	udp	other	SF	146	0	0	0	0	0	0	0
3	0	tcp	private	S0	0	0	0	0	0	0	0	0
4	0	tcp	http	SF	232	8153	0	0	0	0	0	1
5	0	tcp	http	SF	199	420	0	0	0	0	0	1
6	0	tcp	private	REJ	0	0	0	0	0	0	0	0
7	0	tcp	private	S0	0	0	0	0	0	0	0	0
8	0	tcp	private	S0	0	0	0	0	0	0	0	0

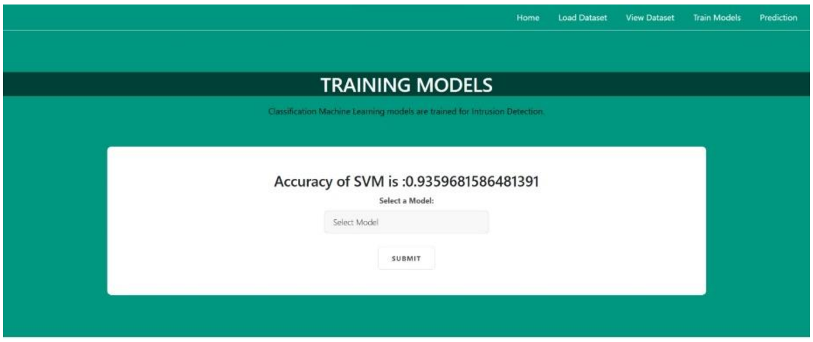
Fig 8: Dataset view page



Classification Machine Learning models are trained for Intrusion Detection.

Fig 9: Model Selection page

SUPPORT VECTOR MACHINE(SVM):

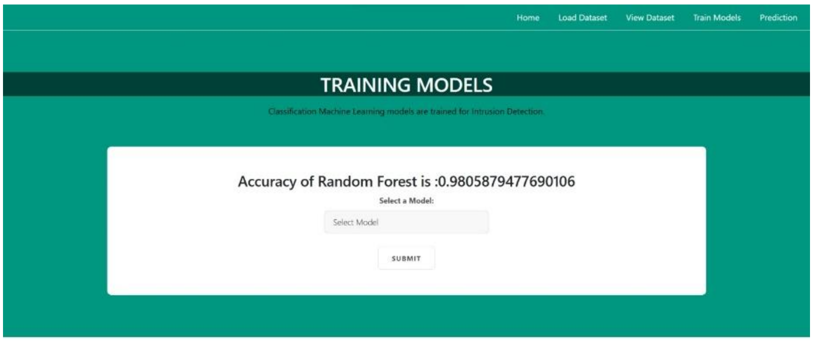


Classification Machine Learning models are trained for Intrusion Detection.

Accuracy of SVM is :0.9359681586481391

Fig 10: SVM Accuracy page

RANDOM FOREST REGRESSION:

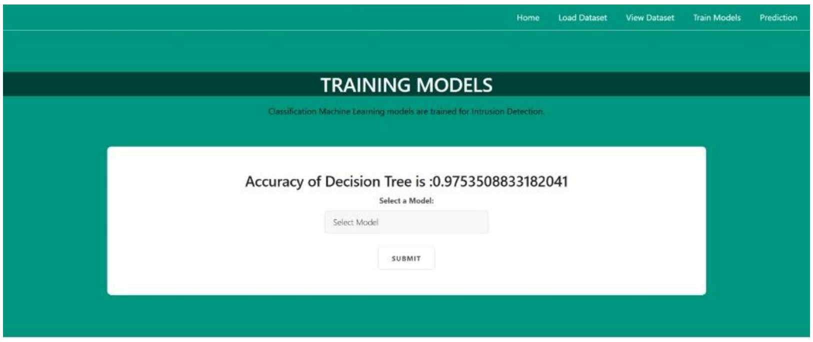


Classification Machine Learning models are trained for Intrusion Detection.

Accuracy of Random Forest is :0.9805879477690106

Fig 11: Random Forest Accuracy page

DECISION TREE:



TRAINING MODELS

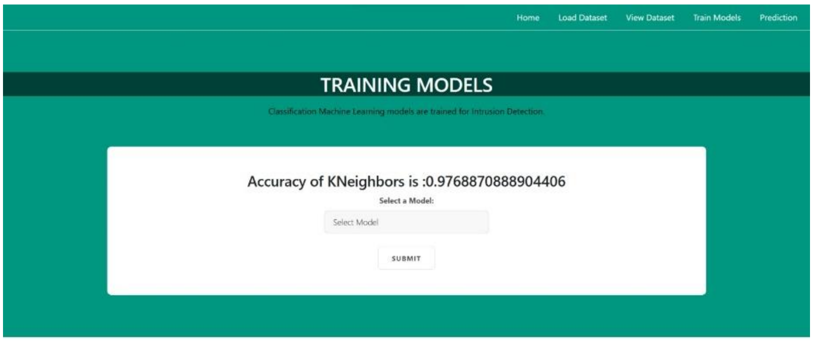
Classification Machine Learning models are trained for Intrusion Detection.

Accuracy of Decision Tree is :0.9753508833182041

Select a Model:

Fig 12: Decision Tree Accuracy page

K-NEAREST NEIGHBOR:



TRAINING MODELS

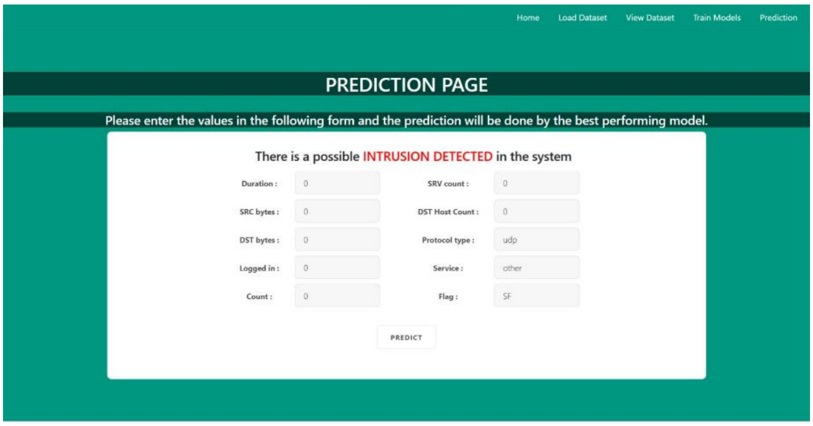
Classification Machine Learning models are trained for Intrusion Detection.

Accuracy of KNeighbors is :0.9768870888904406

Select a Model:

Fig 13: KNN Accuracy page

OUTPAGE:



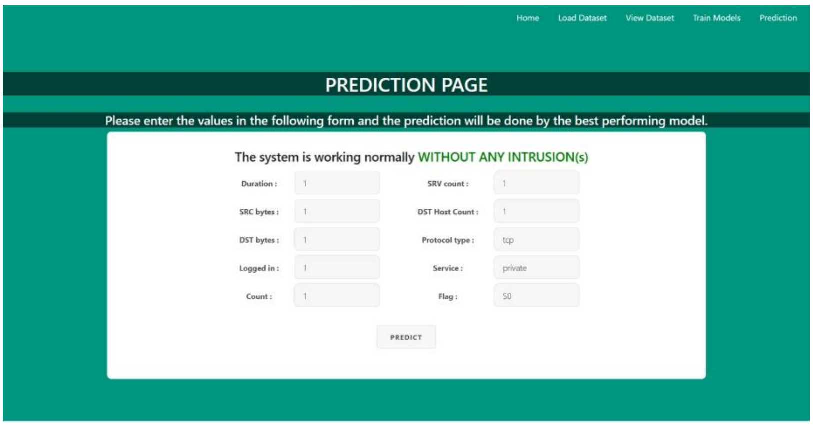
PREDICTION PAGE

Please enter the values in the following form and the prediction will be done by the best performing model.

There is a possible INTRUSION DETECTED in the system

Duration :	0	SRV count :	0
SRC bytes :	0	DST Host Count :	0
DST bytes :	0	Protocol type :	udp
Logged in :	0	Service :	other
Count :	0	Flag :	SF

Fig 14: Intrusion Detected page



PREDICTION PAGE

Please enter the values in the following form and the prediction will be done by the best performing model.

The system is working normally WITHOUT ANY INTRUSION(s)

Duration :	1	SRV count :	1
SRC bytes :	1	DST Host Count :	1
DST bytes :	1	Protocol type :	tcp
Logged in :	1	Service :	private
Count :	1	Flag :	S0

Fig 15: No Intrusion Detected page

7. Conclusions and Future Scope

Computer network security is largely dependent on intrusion detection systems (IDS). Students are made to recognise and react to online dangers that can jeopardise the availability, confidentiality, and integrity of critical data. In light of the foregoing, a conclusion is drawn that IDS is and will remain a key part of the cybersecurity ecosystem.

IDS's capacity to monitor network traffic in real-time and find anomalies or suspicious activity that may point to a potential cyberattack is one of its main advantages. IDS will need to develop and adapt in order to stay up with the evolving threat landscape as the quantity and sophistication of cyber attacks increase.

With accuracy rates of 93.5%, 98.1%, 97.4%, and 97.6%, an intrusion detection system is successfully constructed in this application using Support Vector Machines, Random Forest Regression, Decision tree, and K-Nearest Neighbor. Our model's cardinality can be reduced with the use of principal component analysis, and a web application based on flask is created to provide access to the system.

8. Reference

Journals/Articles :

1. “A Proposed Wireless Intrusion Detection Prevention and Attack System”, by Jafar Abo Nada and Mohammad Rasmi Al-Mosa.
2. “Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm”, by Kinam Park, Youngrok Song, and Yun-Gyung Cheong.
3. "Deep Learning-Based Intrusion Detection for IoT Networks" by Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, and Antonio Robles-Kelly.

e-websites / downloads :

1. Kaggle, https://www.kaggle.com/
2. Course Hero, www.coursehero.com