NETWORK TRAFFIC SAMPLING AND SECURITY WITH MACHINE LEARNING

DR.CK.Gomathy
dept.Computer Science and Engineering
SCSVMV Deemed to be University
Kancheepuram, India

DR.V.Geetha
dept.Computer Science and Engineering
SCSVMV Deemed to be University
Kancheepuram, India

Vajjala Vijaysimha
dept.Computer Science and Engineering
SCSVMV Deemed to be University
Kancheepuram, India

Perumalla Vyshnavi Naga Satya Sreya
dept. Computer Science and Engineering
SCSVMV Deemed to be University
Kancheepuram, India

Abstract— This project aims to develop a web application using Flask, which integrates machine learning functionalities for network intrusion detection. The application will provide a user interface enabling users to load data, preprocess it, select models, and predict intrusions. Leveraging libraries such as Pandas, NumPy, and Scikit-learn, the application facilitates tasks such as data handling, encoding categorical features, splitting datasets for training and testing, and evaluating models. The machine learning algorithms available include Decision Trees, Random Forests, Support Vector Machines, and a pre-trained Neural Network. Users will be able to input network traffic features to predict intrusions at a certain time, with the system classifying potential intrusions into distinct attack categories. The ultimate goal of this project is to offer a user-friendly tool for network security analysts to identify and classify potential network intrusions.

Keywords- Random Forest, Decision Tree, Neural Network and Support vector machine, ML techniques, evaluation.

I. INTRODUCTION

The project endeavors to craft a user-friendly web application for network intrusion detection, leveraging machine learning techniques. Powered by Flask, this application empowers users to execute various pivotal tasks. Initially, users can upload datasets containing network activity details. Subsequently, the application preprocesses this data by encoding categorical variables and partitioning it into training and testing sets. The primary objective is to prepare the data for training machine learning models. Following this, users are presented with a suite of machine learning algorithms, including Decision Trees, Random Forests, Support Vector Machines, and a pre-trained Neural Network. The aim is to enable users to assess the performance of these algorithms in intrusion detection. Models are trained on the preprocessed data, and their accuracies are assessed using the test set. The project encompasses the development of a robust and interactive web application for network intrusion detection through machine learning. Its core focus is on building a user-friendly interface that facilitates diverse functionalities. Additionally, the project involves user engagement via an intuitive interface, allowing individuals to input specific network-related features and receive predictions regarding intrusion nature. Furthermore, it includes user-input prediction, enabling individuals to forecast intrusion nature using a Random Forest Classifier based on specific network-related features. The overarching goal is to deliver an accessible, adaptable, and effective tool for network intrusion detection, catering to both novices and experts in the realms of cybersecurity and machine learning.

II. LITERATURE SURVEY

1) A Survey on Network Intrusion Detection using Convolutional Neural Network-Antanios Kaissar, Ali Bou Nassif, MohammadNoor Injadat-2022: How Artificial Intelligence, specifically Convolutional Neural Networks, contributes to protecting networks through Intrusion Detection Systems. It looks at 81 research articles that focus on using CNN in network security. Among these, 28 articles explore hybrid models combining CNN with other methods for intrusion detection. The survey also identifies 21 different ways researchers evaluate the effectiveness of these models and highlights the use of 12 diverse datasets in these studies.

2) Deep Learning-Based Intrusion Detection Systems: A Systematic Review-Jan Lansky, Saqib Ali, Mokhtar Mohammadi, Mohammed Kamal Majeed, Sarkhel H. Taher Karim -2021:

They focus on these deep learning approaches within Intrusion Detection Systems, which are crucial for protecting networks and hosts. It starts by explaining the basics of IDS architecture and the different deep learning techniques. Then, it categorizes various intrusion detection approaches based on the specific deep- learning methods they use. They explain how these deep learning networks accurately identify intrusions, enhancing the overall securitymeasures.

3) Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: T. Saranya a, S. Sridevi b, C. Deisy c, Tran Duc Chung d, M.K.A.Ahamed Khan-2020: How technology growth has made life easier but also brought more security problems. It mentions a system called Intrusion Detection System that helps keep things safe by watching for suspicious activity online. Recently, smart computer programs called Machine Learning algorithms have been used in IDS to spot and sort out security threats.

4) Intrusion Detection Systems: A Comprehensive Survey- Ke He, Dan Dongseong Kim, Muhammad Rizwan Asghar-2019: A system that protects computer networks from bad things. It mentions that a type of smart system, called Deep Neural Networks, is used in this protection. But sometimes, sneaky changes in network data can fool these smart systems, causing them to make mistakes and letting in harmful things.

5) Machine Learning Methods for Network Intrusion Detection- Mouhammad Alkasassbeh, Mohammad Almseidin-2018: How people who work in network security, like Network Security Engineers, make sure that online services stay available and safe from intruders. They use a system called Intrusion Detection System to spot any weird or bad actions happening online.

III. PROPOSED SYSTEM

The application presents users with a user-friendly interface, comprising a welcoming homepage and intuitive navigation to sections such as About, Data Upload, Preprocessing, Model Selection, and Prediction. It empowers users to upload network intrusion data in CSV format, which then undergoes preprocessing. This preprocessing involves the removal of irrelevant columns, encoding of categorical features, and division of the dataset into training and testing subsets. In the Model Selection section, users can select from a variety of machine learning algorithms, including Decision Trees, Random Forests, Support Vector Machines, and Neural Networks. The accuracy scores for each model on the testing set are displayed to aid in decision-making. The Prediction section enables users to input network activity parameters for intrusion prediction, utilizing a trained Random Forest Classifier to predict intrusion types. Notable enhancements include model persistence for future use without the need for retraining, visualization aids such as accuracy charts or confusion matrices for performance illustration, and deployment on a server for online accessibility, catering to users of all technical levels.

IV. ALGORITHM

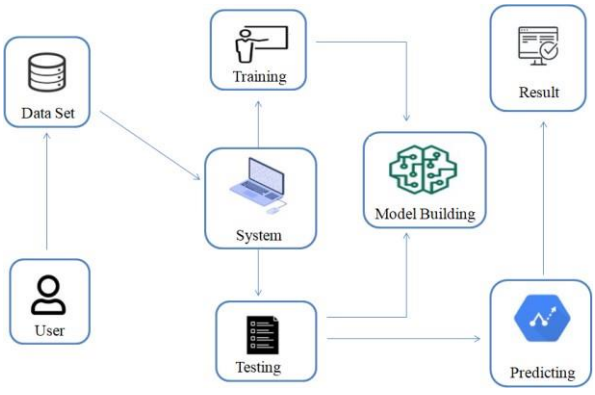
1. RANDOM FOREST**:** The random forest algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or meanof the output from various trees. Increasing the number of trees increases the precision of the outcome. A random forest eradicates the limitations of a decision tree algorithm. It reduces the over-fitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like Scikit- learn).

2. DECISION TREE: A decision tree can be used to visually and explicitly represent decisions and decision- making. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal, the feature's importance is clear, and relationships can be easily viewed.

3. SUPPORT VECTOR MACHINES: The primary goal of SVM is to find a hyperplane that best separates the data into different classes. The hyperplane chosen is the one that maximizes the margin, i.e., the distance between the hyperplane and the nearest data points (support vectors) of each class.

4. NEURAL NETWORKS: ANNs are inspired by the structure and functioning of the human brain, consisting of interconnected nodes (neurons) organized into layers (input, hidden, and output). ANNs find applications in a wide range of fields, including image and speech recognition, natural language processing, and many other areas where complex patterns need to be recognized.

V. SYSTEM ARCHITECTURE



This project entails the development of a Network Traffic Sampling and Security system via a Flask-based web application. The process is intricately designed to detect network intrusions seamlessly. It begins with users uploading a CSV file containing network intrusion data, which then undergoes preprocessing. During this phase, irrelevant columns are removed, and the dataset is split into training and testing subsets. Subsequently, users are presented with a user-friendly interface to choose from multiple machine learning algorithms for model training, including Decision Trees, Random Forests, Support Vector Machines, and a Neural Network. The selected model is trained on the preprocessed training data, and its accuracy is evaluated against the testing dataset. Users can input parameters related to network activities for intrusion prediction. These inputs are fed into a trained Random Forest Classifier, which predicts the type of intrusion, providing instant feedback on the web interface. The Flask framework underpins the entire process, facilitating the seamless integration of backend Python functionalities with an intuitive web-based interface. Ultimately, this system aims to empower users with a dependable and easily accessible tool for network intrusion detection, offering a smooth journey from data handling and model training to real-time prediction, all accessible through a standard web browser.

VI. MODULES

USER:

- View Home page: The user views the home page of the network's application.
- View About page: Users can learn more about the network's platform.
- View Load_data page: In the Load_data page, the user will load the dataset for modeling.
- Input Model: The user must provide input values for fields to get results.
- View Results: The user views the generated results from the model.
- View score: Users can view the accuracy score in %.

SYSTEM:

- Working on the dataset: The system checks whether data is available or not and loads the data in CSV files.
- Pre-processing: Data are according to the models to increase the accuracy and better information about the data.
- Training the data: The data will be split into two parts for training and test data before training with the given algorithms.
- Model Building: To create a model that predicts the personality with better accuracy, this module will help the user.
- Generated Score: User views the score in %.
- Generate Results: We train the machine learning algorithm and predict the result.

VII. TESTING TO BE USED

1) Data Loading and Pre-processing:

- Testing Dataset Loading: A dataset will upload a file. The code reads the CSV file using Pandas and then pre-processes the data.
- Data Loading and Pre-processing: Testing Dataset Loading: A dataset will upload a file. The code reads the CSV file using Pandas and then pre-processes the data.
- Verification: The code includes print statements to verify the size and shape of the training and testing datasets after splitting.

2) Model Training and Evaluation:

- Model Selection: Users can choose from various machine learning algorithms (Decision Tree, Random Forest, Support Vector Machine, and Neural Network) for training.
- Model Training: Once an algorithm is selected, the code initializes the chosen model, fits it to the training data, and generates predictions for the testing dataset.
- Accuracy Measurement: The accuracy of each trained model is assessed using the accuracy score metric from sci-kit-learn, comparing the model's predictions with the actual test labels. These accuracy scores are then displayed to the user in the web interface.

3) Prediction:

- Input Validation: Users are presented with a form to input various parameters related to network intrusions. After submitting the form, the input parameters are utilized along with the trained models to predict the type of network intrusion based on the provided input. The predicted intrusion type is then displayed on the web interface
- Model Prediction: After submitting the form, the input parameters are utilized along with the trained models to predict the type of network intrusion based on the provided input. The predicted intrusion type is then displayed on the web interface.

4) General Methodology:

- Web Interface Testing: Verification of Flask routes (/about, /load, /pre-process, /model, /prediction) to ensure correct template rendering and appropriate responses to user actions (GET and POST requests).
- Functionality Verification: Validation of each function within the Flask routes to ensure proper operation, handling of various requests, and generation of expected outputs.
- Limitations: The code lacks extensive model hyper-parameter tuning, cross-validation, or in-depth evaluation metrics beyond accuracy.

VIII. RESULTS

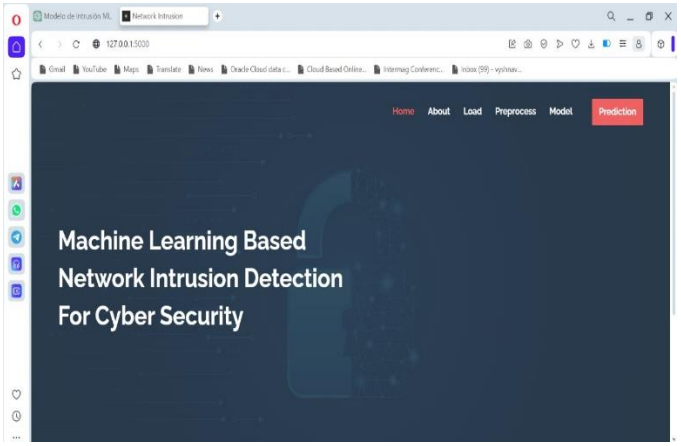


Fig.no.1 Home Page

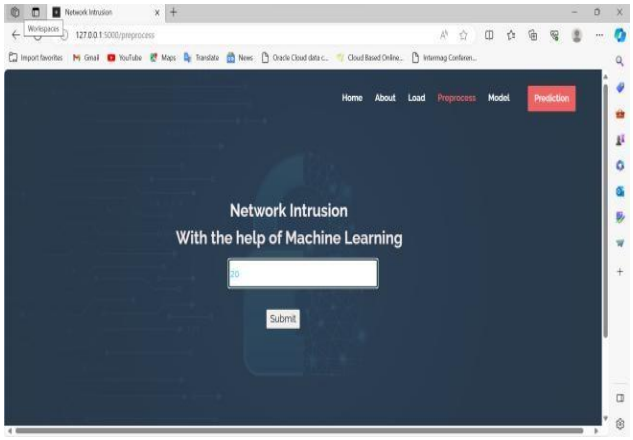


Fig.no.4 Preprocess the data

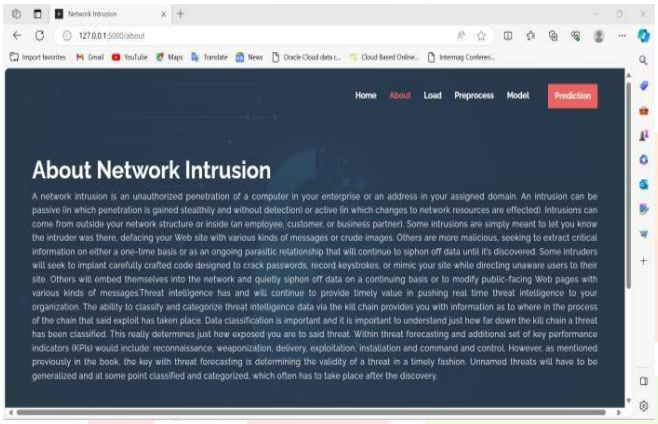


Fig.no.2 about our network intrusion.

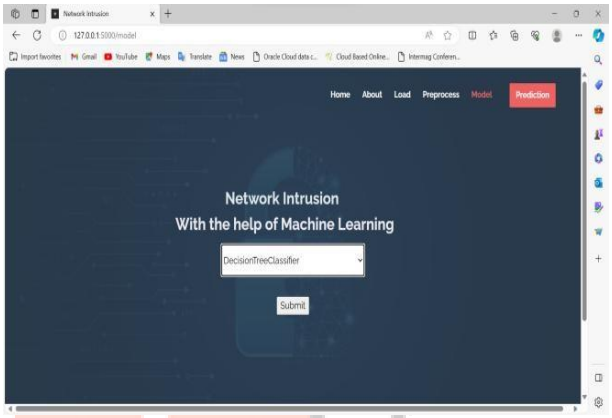


Fig.no.5 Selecting model

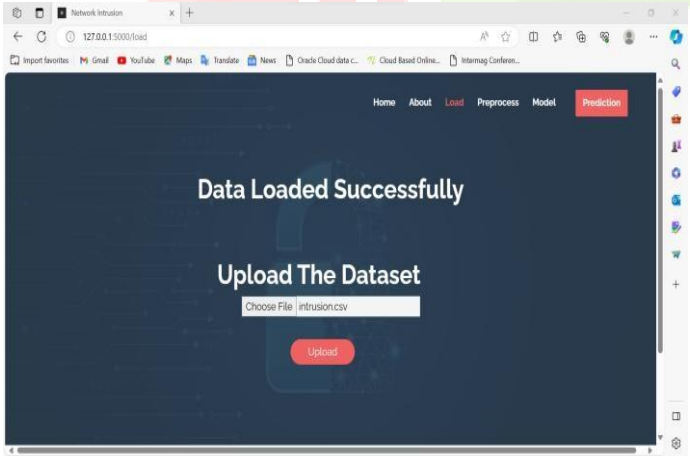


Fig.no.3 Upload Dataset

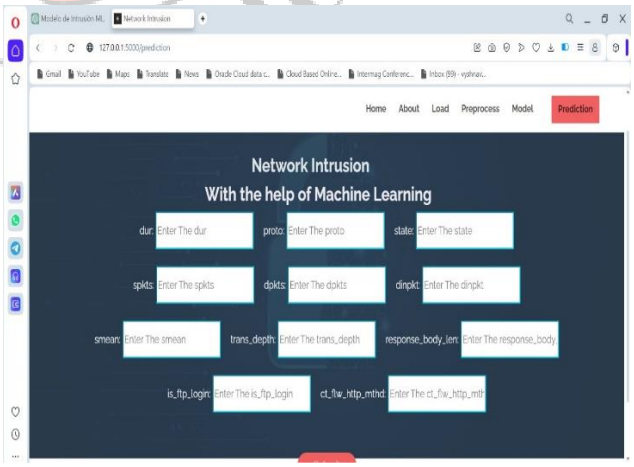


Fig.no.6 Result

IX. CONCLUSIONS

This paper encompasses essential functionalities like data uploading, preprocessing, model training, and real-time prediction via an intuitive web interface. Leveraging well-known machine learning algorithms such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks, the web application empowers users to select and assess models according to their intrusion detection requirements. The integration of Flask streamlines user interaction with backend functionalities, enhancing system usability. However, it's crucial to acknowledge potential limitations, such as relying solely on accuracy as the evaluation metric and using a fixed dataset for training and testing. Future enhancements might entail incorporating additional evaluation metrics, enabling dynamic dataset uploads, and further refining the user interface to create a more comprehensive and adaptable Network Intrusion Detection System.

X. FUTURE ENHANCEMENTS

Enhance the system's capability to accommodate dynamic dataset uploads, granting users the flexibility to utilize various datasets for intrusion detection tasks. Extend the system to support real-time intrusion detection and monitoring, enabling prompt responses to ongoing network threats. Introduce user authentication and additional security measures to safeguard sensitive intrusion data and ensure secure system access. Optimize machine learning models for improved performance and scalability, enabling efficient handling of larger datasets. Integrate visualizations such as graphs or charts to present insights and detailed reports on intrusion patterns and system performance. Implement mechanisms for the system to continually learn and adapt to evolving intrusion patterns, ensuring readiness to counter new threats.

XI. REFERENCES

- 2018-" Machine Learning for Network Intrusion Detection: A Review"-Mahmood.A.N & Hu.J
- 2019-"Intrusion Detection Systems: A Comprehensive Survey"-Chen.X & Li.W
- 2020-" A Comprehensive Review of Network Intrusion Detection Using Machine Learning Techniques"- Gupta.S & Sharma.R
- 2021-"Deep Learning-Based Intrusion Detection Systems: A Review"-Lee.C & Kim.D
- 2022-"A Survey of Machine Learning Techniques in Intrusion Detection"-Smith.J & Johnson.A
- 2019-"Deep Learning Approach for Intelligent Intrusion Detection System"- R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. AlNemrat and S. Venkatraman.
- 2021- Network Intrusion Detection on UNSWNB15."- V. Kumar.[Online] Available: https://github.com/vinayakumarr/NetworkIntrusionDetection/blob/master/UNSW-NB15/CNN/multiclass/cnn2.py.
- 2019-"Survey on SDN based network intrusion detection system using machine learning approaches," -N. Sultana, N. Chilamkurti, W. Peng and R. Alhadad Peer-to-Peer Netw.
- 2017- "Flow-based intrusion detection: Techniques and challenges"- M. F. Umer, M. Sher and Y. Bi Comput. & Secur., vol. 70, pp. 238–254, , doi: 10.1016/j.cose.2017.05.009.