Intrusion Detection System For Dos Attacks Using Machine Learning

[1]Dr Pradeep Kumar, [2]K.Laxmikanth Reddy, [3]D.Pavan Kumar Nayak, [4]Shaik Ishaq

[1]Assistant Professor, [2,3,4]Student
Department of Artificial Intelligence
Anurag University, Hyderabad, India.

Abstract- **This paper presents a novel approach network traffic is at risk from hackers with various passive and aggressive attacks, compromising security. A strong Intrusion Detection System is crucial for swift and accurate attack identification by analyzing each packet in real-time. Utilizing machine learning enhances network security, demanding substantial data to uncover complex patterns effectively. Thus We focus on the NSL-KDD datasets, which eliminate certain redundant and more frequent records from the 1999 KDD Cup dataset that can still be used in machine learning techniques that are the primary tools for network traffic analysis and anomaly detection. Initially, features to be extracted from network traffic are pre-processed, which often involves the use of numerous mathematical techniques like removing unnecessary or undesired features to assemble the data for a machine learning model. Then the selected features are used for training the proposed models and a binary classification technique is used for prediction of normal or attack type. Eventually, the overall performance accuracy and error rate of our model is evaluated. Thus network traffic analysis will be used to expose invasions and forbid network attacks.**

Keywords: **Denial Of Service Attack, Deep Learning, Intrusion Detection System, Intrusion Prevention System.**

I. INTRODUCTION

The utilization of the internet has indeed permeated numerous aspects of modern life, offering immense benefits and opportunities for businesses and individuals alike. However, this widespread connectivity also brings with it significant security challenges and hazards. Critical information is constantly at risk of falling into the hands of unauthorized users, leaving networks vulnerable to breaches and attacks.

Intrusion, the unauthorized access to a system, is a persistent threat in the digital landscape. Identifying intruders can be a daunting task, as they often hide behind various IP addresses, operating systems, or applications. Network security, therefore, encompasses a range of measures and precautions administrators take to safeguard against such breaches, aiming to prevent hackers from gaining forbidden access to sensitive information.One pivotal technology in the realm of network security is the Intrusion Detection System (IDS). IDS serves as a vigilant watchdog, tasked with spotting and alerting administrators to potential attacks on accessible target systems. There are two primary types of IDS: signature-based and anomaly-based.

Signature-based IDS operates by identifying patterns or signatures of well-known attacks, allowing for their prompt detection and response. On the other hand, anomaly-based IDS focuses on detecting deviations from normal network behavior, flagging activities that are unusual or unexpected, which may indicate a potential intrusion.Despite these advancements, the ever-evolving landscape of cyber threats necessitates continual innovation in intrusion detection methods. In recent years, researchers have turned to machine learning (ML) techniques as a promising approach to bolstering network security.

Machine learning models, which can be categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning offer a dynamic and adaptable means of detecting intrusions. By analyzing vast amounts of data, ML algorithms can learn to recognize patterns and anomalies indicative of potential threats, enhancing the efficacy of intrusion detection systems.

In summary, while the internet provides unparalleled opportunities for connectivity and innovation, it also presents formidable challenges in terms of security. Intrusion detection systems, augmented by machine learning techniques, play a crucial role in safeguarding networks and preserving the integrity of sensitive information in an increasingly digitized world.

II. LITERATURE SURVEY

Machine learning is increasingly used to detect Denial-of-Service (DoS) attacks, which disrupt online services. Seminal works like Zargar et al. (2013) survey defense mechanisms against DoS attacks, including machine learning methods. Mishra and Phoha (2009) propose an approach inspired by the human immune system to detect DoS attacks in wireless

networks. Additionally, Kumar and Sharma (2018) review various machine learning techniques for DoS attack detection. Despite progress, challenges like handling complex data and adapting to new attack methods remain. Future research aims to create more effective and adaptable detection systems to combat evolving DoS attacks. Here are seminal works that provide insights into the landscape of ML-based approaches in this domain:

1. **"A Deep Learning-Based IDPS for Detecting and Preventing DOS Attacks", Uan Fernando Canola Garcia, Publication Year: 2022**

Uan Fernando Canola Garcia created a deep learning introduces Dique, an IDS/IPS system powered by deep learning to detect and prevent DoS attacks on web servers. Utilizing a Graphical User Interface (GUI), Dique offers real-time monitoring and classification of incoming packets as benign or malicious. It employs a multi-layered Deep Feed Forward neural network trained on the CICDDoS2019 Dataset, achieving a high accuracy of 0.994. Additionally, an offensive system called Diluvio is developed to test Dique, featuring seven DoS attack types, including novel ones not in the training dataset. Dique's effectiveness in mitigating these attacks demonstrates its potential as an advanced security solution.

2. **"A Multi-Layer Hybrid Intrusion Detection Method Based on Nb And SVM", Ruimin Wang, Publication Year: 2022**

Ruimun Wang introduced a multi-layer hybrid intrusion detection model (MLH-IDM) aimed at improving the detection of both low-frequency and high-frequency attacks in computer networks. Traditional intrusion detection systems struggle with unknown intrusions, leading to imbalances in detection accuracy. MLH-IDM addresses this issue by filtering redundant features and balancing the dataset using random downsampling. The detection process involves layered filtering using Naive Bayes and Support Vector Machine classifiers. Specifically, Naive Bayes is applied in the first and second layers to filter Probe and DoS attacks, while Support Vector Machine is used in the third layer to detect low-frequency attacks (U2R and R2L). Simulation experiments using the NSL-KDD dataset demonstrate improved F1 score, recall rate, and detection rates for various attack types compared to benchmark methods.

3. **"A hybrid machine learning approach for detecting unprecedented DDoS attacks", Mohammad Najafmehr, Sajjad Zarifzadeh, Seyedakbar Mostafavi, Publication Year: 2022**

This paper proposes a novel approach for DDoS detection, combining supervised and unsupervised algorithms. Initially, a clustering algorithm segregates anomalous traffic from normal data based on flow-based features. Subsequently, statistical measures are utilized for cluster labeling using a classification algorithm. Employing a big data processing framework, the method is evaluated on the CICIDS2017 dataset, and tested on the CICDDoS2019 dataset. Results show a significantly higher Positive Likelihood Ratio (LR+) compared to traditional ML classification algorithms, indicating a promising advancement in DDoS detection with approximately 198% higher LR+.

These studies collectively emphasize the pivotal role of machine learning-based approaches in Intrusion Detection Systems (IDS) for detecting Denial of Service (DoS) attacks. By harnessing machine learning techniques, researchers strive to bolster the precision and efficiency of DoS attack identification and mitigation systems, thereby making substantial strides in safeguarding network infrastructures and related domains from malicious cyber threats.

III. PROBLEM STATEMENT

The proposed project aims to develop a comprehensive intrusion detection system (IDS) capable of effectively identifying and mitigating various types of network attacks. This involves several key stages, including dataset selection representing real-world network traffic, thorough pre-processing to ensure data cleanliness and consistency, and rigorous feature extraction and selection to capture relevant patterns and anomalies. Additionally, the project entails the careful selection and implementation of machine learning algorithms tailored to the detection task, alongside the development of a robust framework for seamless integration and deployment of the IDS. Evaluation metrics such as accuracy, precision, recall, and F1-score will be employed to assess the system's performance, with the ultimate goal of achieving a reliable and scalable framework for the detection of diverse network attacks. The project will culminate in a detailed analysis and discussion of the evaluation results, providing insights into the effectiveness and practical applicability of the developed IDS in safeguarding network security

3.1 Existing Systems

Existing systems for pose identification, particularly in the realm of yoga practice, have explored diverse methodologies beyond Convolutional Neural Networks (CNNs). Here are some notable approaches utilized in this domain:

1. **CNN(Convolution Neural Network):** A deep learning model commonly used for image recognition tasks, CNNs utilize convolutional layers to extract features hierarchically, followed by pooling layers for dimensionality reduction, enabling effective pattern recognition.
2. **ANN(Artificial Neural Network):** A versatile machine learning model inspired by the human brain's neural network structure. ANN consists of interconnected nodes organized in layers, with each node performing a

weighted sum of inputs followed by a non-linear activation function, enabling complex pattern recognition and regression tasks.

3. **Naive Bayes:**A simple probabilistic classifier based on Bayes' theorem with the "naive" assumption of feature independence. Despite its simplicity, Naive Bayes performs well in text classification and is efficient with large datasets.
4. **RNN(Recurrent Neural Network):**Designed to handle sequential data, RNNs have recurrent connections that allow information to persist over time. They excel in tasks like time series prediction, natural language processing, and speech recognition due to their ability to capture temporal dependencies.
5. **K-Nearest Neighbors(KNN):** A non-parametric algorithm used for classification and regression tasks. KNN assigns a class label or value based on the majority vote or average of its k nearest neighbors in the feature space, making it simple yet effective for small to medium-sized datasets.

These existing systems demonstrate the versatility of machine learning and probabilistic models in addressing pose identification and correction challenges, complementing the capabilities of Convolutional Neural Networks (CNNs) in this domain.

3.2 Proposed System

The proposed system hybrid feature extraction method that selects the most pertinent features subset from the pre-processed dataset. We use the NSL-KDD dataset, which eliminates certain redundant and more frequent records from the 1999 KDD Cup dataset that can still be used in machine learning techniques that are the primary tools for network traffic analysis and anomaly detection. The selected features are then used to train SVM and Naive Bayesian machine learning models, which are tested on a dataset to gauge their performance in predicting if an attack will occur in the network. The system follows a systematic approach:

Step 1 – Collection of the data from different sites like kaggle. Step 2 – Applying the Data Pre-processing steps. Step 3 – Splitting dataset into train and test.

Step 4 – Model Training. Step 5 – Signup and signin.

Step 6 – Userinput and output is displayed whether the attack has taken place.

IV. DATASET

The Yoga Poses Dataset consists of images categorized into five classes representing different yoga poses. These classes include downward dog pose, goddess pose, tree pose, plank pose, and warrior pose. The dataset is divided into train and test subsets, with each subset containing five subfolders corresponding to the five yoga pose classes. Images were sourced from Bing using their API functionality, although some inaccuracies such as watermarks or text may exist in the images.

- **Number of Classes:** 5 yoga pose classes (downward dog, goddess, tree, plank, warrior).
- **Data Distribution:** Images are divided into train and test directories, with approximately equal numbers of images for each yoga pose class in both subsets.
- **Class Labels:** Each image is labeled with the corresponding yoga pose class.

V. PROPOSED METHODOLOGY

Step 1 – Collection of the data from different sites like kaggle. Step 2 – Applying the Data Pre-processing steps. Step 3 – Splitting dataset into train and test.

Step 4 – Model Training. Step 5 – Signup and signin.

Step 6 – Userinput and output is displayed whether the attack has taken place.

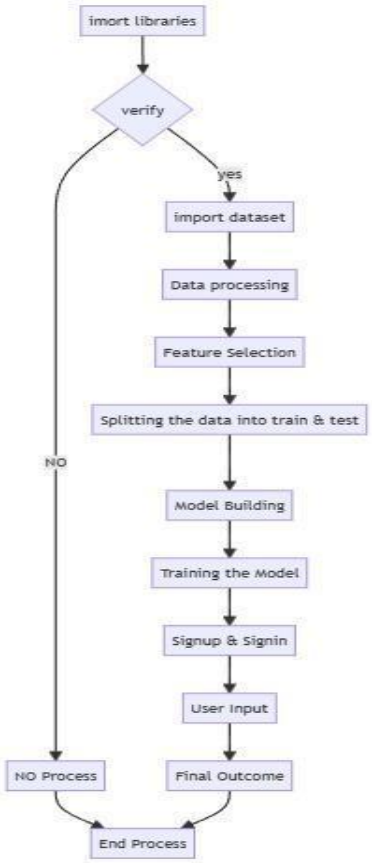

Figure: Flow Chart

5.1. Data Collection

The NSL-KDD dataset is a version of the well-known KDD Cup 99 dataset, which was widely used for evaluating intrusion detection systems. The NSL-KDD dataset was proposed as an improvement over the original KDD Cup 99 dataset to address some of its limitations, such as redundancy and skewed class distribution.It contains a collection of network traffic data generated in a simulated environment, including normal traffic as well as various types of attacks, such as DoS, Probe, R2L (Unauthorized access from a remote machine), and U2R (Unauthorized access to local superuser privileges). The dataset provides labeled instances, indicating whether each network connection is normal or malicious.The NSL-KDD dataset is widely used by researchers and practitioners for developing and evaluating intrusion detection systems using machine learning techniques. Its diverse range of attack types and realistic network traffic patterns make it a valuable resource for studying cybersecurity and testing the effectiveness of intrusion detection algorithms.

5.2. Data Preprocessing

Data preprocessing for the NSL KDD dataset involves several critical steps to prepare the network traffic data for training within intrusion detection systems (IDS). Initially, the dataset undergoes integrity checks to ensure data accessibility and quality, flagging any inconsistencies or missing values for further investigation. Following this, feature engineering techniques are applied to extract relevant information from the raw data, such as protocol type, service, flag, and other network attributes.

Next, categorical features are encoded into numerical values using techniques like one-hot encoding or label encoding to facilitate model training. Simultaneously, continuous features are scaled to a standard range to prevent any single feature from dominating the model's learning process. Additionally, the dataset may undergo dimensionality reduction techniques such as principal component analysis (PCA) to reduce computational complexity and improve model performance.

Once preprocessing is complete, the dataset is split into training and testing sets to evaluate model performance effectively. Careful consideration is given to maintaining the class distribution balance to prevent bias in the model's training. Finally, the preprocessed dataset is ready for training machine learning algorithms, including neural networks and other classifiers, to effectively detect and classify network intrusions with high accuracy and reliability.

5.3. Splitting The Dataset

Quality data is the bedrock of successful machine learning (ML) endeavors. In today's era of massive data creation, the challenge doesn't solely lie in acquiring vast datasets but in ensuring their quality for meaningful ML outcomes. While handling raw data demands adept engineering skills, the crux is effective utilization. Partitioning data for ML/DL models poses a crucial challenge. Despite its apparent simplicity, the intricacies of this task are profound. Improper division into training and testing sets can drastically impact model performance, leading to overfitting or underfitting issues. These challenges underscore the necessity of not just feeding large volumes of data but also meticulously organizing it to optimize model learning, ensuring unbiased and reliable outcomes.

The NSL-KDD is an imbalanced dataset with different types of attacks and normal instances, it's essential to perform stratified splitting. This ensures that the proportion of each class remains similar across the training, validation, and testing sets.In addition to splitting the data into training, validation, and testing sets, we choose to perform k-fold cross-validation on the training set to further validate your model's performance and robustness.

5.4. Model Training

The model building process is meticulously executed through a combination of feature selection techniques and diverse machine learning algorithms. Initially, methods like Chi2, RFE, and Lasso Regularization are employed to sift through features and prioritize those most pertinent to accurate classification. These techniques ensure that the model focuses on the most informative aspects of the data, enhancing its ability to discern patterns and make precise predictions.

A wide array of machine learning algorithms, including SVM, Naive Bayes, Random Forest, AdaBoost, Voting Classifier (RF + AB), and Stacking Classifier (RF + MLP with LightGBM), are then applied to the selected features. Each algorithm undergoes training to develop classifiers capable of accurately categorizing data instances. This comprehensive approach ensures that the model leverages the strengths of various algorithms, leading to robust and versatile classification capabilities.

Evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC score are employed to assess the performance of each model. Techniques like cross-validation and hyperparameter tuning further enhance model robustness and prevent overfitting. The best-performing models are selected for deployment in production environments, where continuous monitoring and updates are conducted to uphold their effectiveness and reliability over time, ensuring accurate classification across diverse contexts and applications.

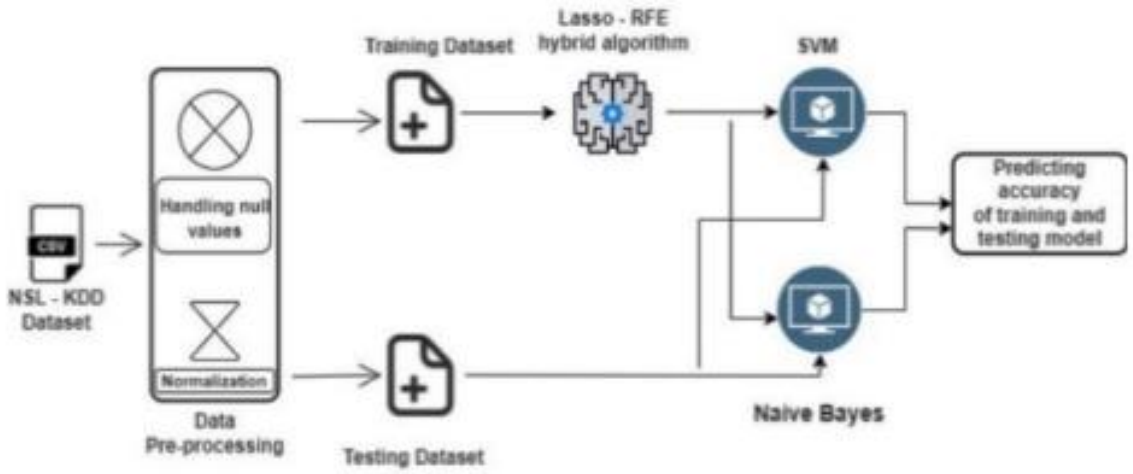

Figure: Model training NSL-KDD Data

VI. MODEL VISUALIZATION

The decision tree model provided is a hierarchical structure for classifying Iris flower species based on their petal and sepal measurements. It begins by assessing the 'petal length (cm)', where if it's less than or equal to 2.45 cm, the model assigns class 0. If 'petal length' exceeds 2.45 cm, the model further evaluates 'petal width (cm)'. Subsequently, it branches based on different thresholds of 'petal width' and 'petal length', leading to specific class assignments (1 or 2). These decisions are contingent upon additional features such as 'sepal length (cm)' and 'sepal width (cm)', ensuring comprehensive classification across various parameter ranges. This structured approach effectively delineates the

feature space, enabling precise predictions regarding Iris species classifications.

```
|--- petal length (cm) <= 2.45
|   |--- class: 0
|--- petal length (cm) >  2.45
|   |--- petal width (cm) <= 1.75
|   |   |--- petal length (cm) <= 4.95
|   |   |   |--- petal width (cm) <= 1.65
|   |   |   |   |--- class: 1
|   |   |   |--- petal width (cm) >  1.65
|   |   |   |   |--- class: 2
|   |   |--- petal length (cm) >  4.95
|   |   |   |--- petal width (cm) <= 1.55
|   |   |   |   |--- class: 2
|   |   |   |--- petal width (cm) >  1.55
|   |   |   |   |--- sepal length (cm) <= 6.95
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- sepal length (cm) >  6.95
|   |   |   |   |   |--- class: 2
|   |--- petal width (cm) >  1.75
|   |   |--- petal length (cm) <= 4.85
|   |   |   |--- sepal width (cm) <= 3.10
|   |   |   |   |--- class: 2
|   |   |   |--- sepal width (cm) >  3.10
|   |   |   |   |--- class: 1
|   |   |--- petal length (cm) >  4.85
|   |   |   |--- class: 2
```

Figure: Model Summary

VII. RESULTS

Decision Tree RESULTS

The Decision Tree model, after undergoing thorough training on a comprehensive dataset, has demonstrated commendable accuracy levels across both training and testing phases. This proficiency highlights its efficacy as a reliable classifier adept at accurately categorizing data points. The consistent and satisfactory performance not only affirms its dependability but also underscores its potential for real-world applications. Its robust performance positions it as a promising candidate for a variety of machine learning tasks, particularly in scenarios where transparent decision-making and interpretability are crucial.

With its inherent interpretability, the Decision Tree model offers clear insights into the decision-making process, making it advantageous in applications where understanding the rationale behind predictions is essential. Its ability to efficiently partition the feature space based on sequential decision rules facilitates intuitive understanding of the data's structure. This interpretability aspect renders the Decision Tree model suitable for applications where transparent and understandable models are preferred.

Furthermore, the Decision Tree's versatility in handling both numerical and categorical data, coupled with its computational efficiency, enhances its applicability across different domains and datasets of varying complexities. This adaptability positions the Decision Tree model as a valuable tool for researchers and practitioners seeking reliable and interpretable classification algorithms for diverse data analysis tasks.

In summary, the Decision Tree model's robustness in analyzing intricate datasets and its interpretability make it a valuable asset for addressing complex challenges across various domains, establishing it as a preferred choice for classification tasks where transparency and reliability are paramount.

VIII. CONCLUSION

Our study was dedicated to propose a hybrid feature extraction method that selects the most pertinent features subset from the pre-processed dataset.The network data is first gathered in order to be used as the model input. This information is gathered from the NSL-KDD dataset, an existing network dataset. The dataset's characteristics go through pre-

processing, which includes operations like normalization, scaling of values, handling of null values, and handling of the combination of categorical and numerical values. After the dataset has been preprocessed, we employ hybrid feature extraction methods to choose the features subset from the dataset that are the most pertinent. The SVM and Naive Bayesian machine learning models are trained using the retrieved features to learn the parameters. After the training phase, the models are put to the test on a dataset to gauge how well they performed using machine learning. It forecasts if an attack will occur in the network during the test phase. We examine the precision and error rate of this anticipated result. We use the model with the best score for further analysis based on the prediction accuracy.

IX. ACKNOWLEDGMENT

We want to express our deep-felt gratitude and sincere thanks to our guide Dr. Pardeep Kumar, Associate Professor, Department of AI, Anurag University, for his skilful guidance, timely suggestions, and encouragement in completing this project. We want to express our profound gratitude to all for having helped us in achieving this dissertation. Finally, we would like to express our heartfelt thanks to our parents, who were very financially and mentally supportive and for their encouragement to achieve our goals.

REFERENCES:

1. F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. Mohamed Chaabani and A. Taleb-Ahmed, "Network Intrusion Detection System Using Neural Network and Condensed Nearest Neighbors with Selection of NSL-KDD Influencing Features," 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2021, pp. 23-29, doi: 10.1109/IoTaIS50849.2021.9359689.
2. M. B. Shahbaz, Xianbin Wang, A. Behnad and J. Samarabandu, "On efficiency enhancement of the correlation-based feature selection for intrusion detection systems," 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON),2016,pp.1-7,doi: 10.1109/IEMCON.2016.7746286.
3. Ali Hussein Shamman Al-Safi , Zaid Ibrahim Rasool Hani, , Musaddak M. Abdul Zahra,"Using A Hybrid Algorithm and Feature Selection for Network Anomaly Intrusion Detection", Journal of Mechanical Engineering Research and Developments, ISSN: 1024-1752, CODEN: JERDFO, Vol. 44, No. 4, pp. 253-262. Published Year 2021.
4. Seyedakbar Mostafavi , Sajjad Zarifzadeh, Mohammad Najafmehr,"A hybrid machine learning approach for detecting unprecedented DDoS attacks",Accepted: 16 December 2021 / Published online: 7 January 2022 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021.
5. A Deep Learning-Based Intrusion Detection and Preventation System for Detecting and Preventing Denial-of-Service Attacks JUAN FERNANDO CAÑOLA GARCIA 1 AND GABRIEL ENRIQUE TABORDA BLANDON 2 1Grupo Éxito S.A., Envigado 055428, Columbia 2Research Group in Automation, Electronics and Computer Science, Instituto Tecnológico Metropolitano, Medellín 050036, Colombia Corresponding author: Juan Fernando Cañola Garcia (juancanola116639@correo.itm.edu.co)