ADDRESSING IMBALANCED NETWORK TRAFFIC IN INTRUSION DETECTION USING MODIFIED RANDOM FOREST AND KDD DATASET

*M.Shalini** [1], *M.Kalaiselvi* [2]

ABSTRACT

The system employs machine learning techniques on the KDD dataset for intrusion detection improvements. The Load Data module, the focus of the analysis process, loads the dataset either from a remote site or from a local site. It guarantees that the data is organized and stored in memory well enough to facilitate manipulation. For optimum performance of machine learning, one of the prerequisites for feature selection and preprocessing quality is also the loading of data. The module for Data Preprocessing proves a major contributor to raw data quality and usability enhancement. In this process, duplicates are eliminated, missing values treated, and dataset inconsistency corrections performed. Equally important would be encoding categorical variables and normalizing data to fit into the ML models used in this research project. Thus, this crucial module standardizes the data and ensures the integrity of the dataset for further processing. Thus, it raises the performance of intrusion detection systems by increasing model capability and efficiency. A clean and structured base will allow machine-learning algorithms to detect patterns and anomalies. Hence, it provides an excellent and reliable means of intruder detection, identifying prospective threats accurately.

Keywords: Intrusion Detection, Machine Learning, Data Preprocessing, KDD Dataset

I. INTRODUCTION

As intrusion detection systems are capable of identifying anomalies and potential security weaknesses, such systems are critical to network defense against cyber attacks. To enhance the efficiency and precision of such systems, machine learning procedures are now essential. KDD dataset, a widely used benchmark for intrusion detection, is one of the leading sources of network traffic data upon which machine learning models can be trained and tested. However, the efficiency of these models largely depends on the quality of data handling. The dataset should be preprocessed and loaded properly in order to effectively classify normal and malicious behavior as well as extract meaningful patterns. With a quest for maximum intrusion detection, the proposed approach employs an organized data handling strategy. The load data module transfers the dataset from different sources in an ordered manner for proper analysis. Upon loading, the Data Pre-processing module improves the dataset by adding missing values, removing duplicates, normalizing the data, and encoding categorical features. These are operations through which machine learning algorithms may use the data, as well as its integrity and quality may be improved. By these preprocessing techniques, the system improves intrusion detection ability to enable real-time detection of potential security threats and strengthen network protection. [1]

Intrusion detection is the primary information security technology's that are applied to identify malicious attacks and unauthorized access or policy violations by monitoring system and network traffic. It must protect confidential information and networked system integrity because it's monitoring anomalies and threats in real-time.The intrusion detection systems make use of signature-based detection that identifies known attacks signatures and the anomaly-based detection which observes unusual behavior. Cyberattack sophistication has increased use of machine learning based intrusion detection. This approach allows systems to automatically learn from historical patterns of data and identify unknown attacks that were previously unseen. Effective intrusion detection enhances security by shortening response times that prevent damage from cyberattacks and issuing prompt warnings. [2]

Department of Computer Science and Engineering[1],
Karpagam Academy of Higher Education, Coimbatore, India[1]
shalinimaya2926@gmail.com
Department of Computer Science and Engineering (Cyber Security)[2],
Dr.N.G.P.Institute of Technology Coimbatore[2]
kalaiselvi.mayilsamy@gmail.com

* Corresponding Author

Machine learning addresses a huge range of problems arising in artificial intelligence, where learning is obtaining knowledge from an input data set and making predictions or inferences on the real world from them, without being explicitly programmed for that purpose. Its designing algorithms that are looking for patterns, detecting trends, and deriving insights from earlier performance to become better over a period of time. Machine learning is used extensively in many fields such as cybersecurity, finance, healthcare and automation, due to its ability to perform data-driven tasks of a sophisticated kind. Machine learning lies at the heart of intrusion detection in that it reviews network data and identifies patterns that may indicate intrusions. Through the use of supervised, unsupervised, and reinforcement learning methods, machine learning improves the accuracy and effectiveness of security systems. This allows it to identify and combat threats proactively. [3]

Just like Knowledge Discovery in Databases, KDD is widely considered to be the most popular benchmark datasets in IDS. KDD has benign and malevolent network activity and was created out of network traffic data collected during the DARPA Intrusion Detection Evaluation Program in 1998. This dataset collects network connection attributes such as connection duration, protocol type, source and destination ports, and types of attack. Hacking types included in this category are DDoS, Deep and more interesting, U2R : User-to-Root; R2L : Remote-to-Local. Since its large quantity of labelled data and many varieties of attacks render it the ideal candidate for training and testing machine learning models for ID purposes, the KDD of intrusion detection has been used widely. Nevertheless, issues like duplicate records and unequal class distributions call for appropriate preprocessing in order to improve model performance and guarantee precise threat identification. [4]

II. METHODOLOGY

This proposed method leverages machine learning algorithms over the KDD dataset to refine intrusion detection system. The Load Data module initiating the process serves for adequate storage and efficient data handling towards further processing by importing the dataset either from local or remote locations. The Data Preprocessing module then applies the data through processes of missing value imputation, duplicate removal, and normalization of features to suit the raw data for machine learning purposes. It makes sure that this step of preparation actually goes along with the data in terms of consistency or quality. After following the above step, one can apply observation selection mechanisms to identify candidates improving the computational performance at the cost of accuracy by dimensionality reduction. The essence of the methodology remains with the machine learning-based classification model trained to recognize patterns indicative of network intrusions from this preprocessed dataset. The model categorizes activity as either benign or malignant depending on the traffic pattern with the aim of learning how to efficiently recognize anything that can be termed as a security threat. For real-time detection, the system is able to scan incoming traffic and gives an indication of possible intrusions with high accuracy. The feature selection and preprocessing steps allowed for an increase in detection capabilities in not only identifying true health anomalies with high precision but also for false positive reduction, aiding in developing a trustworthy yet accurate intrusion detection system. In summary, this methodological procedure combines data preparation, The service scales because the core technology is a symbiosis of advanced feature optimization and the latest machine learning to ensure effective detection and security threats. [5]

A. PROPOSED METHODOLOGY

This systematized and holistic approach was proposed for intrusion detection to threaten cybersecurity at best detection and later prevention. Data acquisition and its preparation are the first stages of importing, cleaning, and converting KDD datasets to make the input as clean as possible for analytical purposes. It is applying procedures for appearing to make data fit machine learning techniques, handling missing values, eliminating duplicate entries, and preparing the category variable. The system thereafter picks features after reduction, which gives a potential extraction of the most relevant properties, reducing dimensionality and giving a computational advantage without compromising

detection performance. The enhanced dataset is subsequently used to train a machine learning-based classification model that can distinguish benign network traffic patterns from malicious ones. [6]

The model is fitted into real-time intrusion detection module upon training, where it scans incoming network data in real time and rapidly indicates potential threats. A threat analysis and reporting module is integrated to further enhance the utility of the system, generating detailed security insights and visualizations. This allows network managers to analyze vulnerabilities, inspect attack patterns, and implement custom protections. The method ensures an adaptive and scalable intrusion detection system that is able to respond to evolving cybersecurity challenges through the fusion of robust preprocessing, best feature selection, and advanced ML. The end-to-end methodology effective solution for safeguarding network infrastructure through the reduction of false positives while enhancing detection quality. [7]

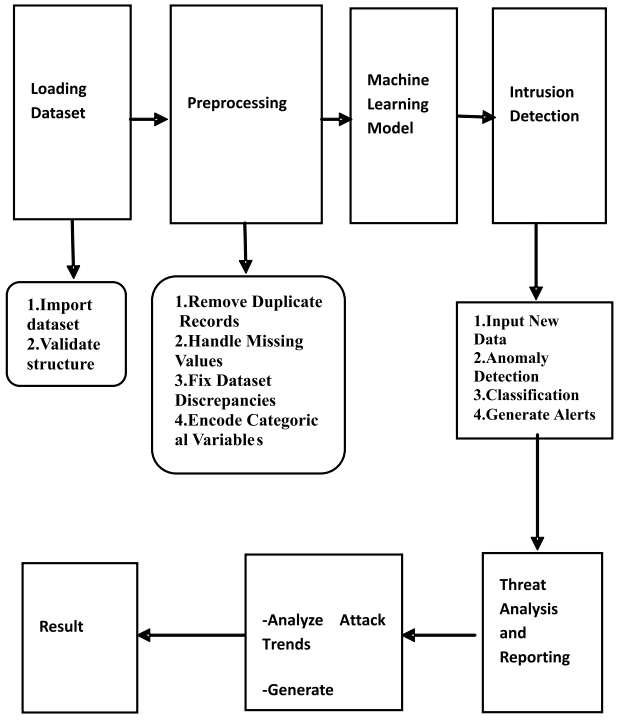

Figure 1: Architecture Diagram

B. LOAD DATA

The KDD dataset is loaded from a local file or a remote database through the Load Data Module, which is an integral component of the intrusion detection system. Training and evaluation of the system rely upon this dataset, and it consists of a mix of malicious and legitimate network traffic. The module carries out functions required like data validation and formatting, making sure the data is properly structured and stored in memory for processing. The dependability and accuracy of the system overall could be compromised by errors at this stage, e.g., faulty formatting, incomplete data retrieval, or corruption, and therefore proper data loading is crucial. Furthermore, the module processes huge datasets well, ensuring data scalability for real-world network environments. By ensuring data integrity and making smooth data transfer, the Load Data Module is an essential part of the intrusion detection pipeline because it has a direct impact on how efficiently downstream procedures like Attribute Selection and preprocessing are carried out. [8]

Steps:

1. Import Dataset:

The KDD dataset is loaded from a file (e.g., CSV, ARFF) or a remote database.

The dataset contains a mix of normal and malicious network traffic records.

2. Validate Data:

Check for file corruption, missing data, or formatting errors.

Ensure the dataset is complete and properly structured.

3.Store in Memory:

Load the dataset into memory for efficient access during preprocessing and analysis.

C. DATA PRE-PROCESSING

Through cleaning, converting, and standardizing the data, the Data Pre-processing Module facilitates getting the raw dataset ready for analysis. It begins by filling the missing values through deletion of incomplete records if necessary or imputation by statistical methods (mean, median, or mode). Duplicate entries are eliminated to prevent duplication and ensure objective results. By standardizing or normalizing all numerical feature so that they fall into the same range, one is

preventing any single feature from being over-representative in the learning by the model. to ensure machine learning compatibility, categorical variables such as protocol type or service will undergo transformation into numerical form through various techniques such as one-hot encoding or label encoding. In terms of data integrity, formatting errors are also corrected and outliers are handled. Efficient preprocessing significantly enhances the data quality, which ultimately leads to better model performance and more reliable intrusion detection. [9]

Steps:

1.Handle Missing Values:

Identify missing values in the dataset.

Revisit the missing value imputation methods (including but not limited to mean, median, mode) and decide whether to 'fill' in the missing data with the middle or central point or 'remove' the corresponding record to reduce dataset contention. [10]

2. Cleaning Up Repetitive Data:

To take care of bias in the model, data cleaning (duplicate records). [11]

3.Normalize/Standardize Numerical Features:

Numbers provide a gloriously clear, unequivocal answer. Frequency is the case with really validating information from a repository filled with every sort of query answer. [12]

4.Categorical Data Encoding:

Convert categorical features, e-.g. protocol type, service into numerical values using techniques like:

- Assign a distinct integer to each category of data with Label Encoding.
- One Hot Encoding: Generates a set of columns, with each column representing a category as an attribute. [13]

5.Handle Outliers:

Detect and manage outliers employing traditional measures like Z Score, IQR and so on. [14]

D. MACHINE LEARNING MODEL

This module on feature selection meant for this process of cleaning an dataset from the redundant features, setting it thus optimized for intrusion detection. During this phase, the features are dismissed that have been found redundant and irrelevant to the classification problem, thus reducing the dimension of the dataset. Following this process comes feature identification, which involves determining the most significant features through some methods like Principal Component; for example, Recursive Feature Elimination and Correlation-based Feature Selection. By focusing on the most important features, the model minimizes overfitting risk, increases computational efficiency, and accelerates the training process. Feature selection ensures that the machine learning models focus their work on the most relevant tasks of differentiating anomalies so that performance is maximized. This process thus improves efficiency and effectiveness in the intrusion detection system after confirming scalability and improvements over model performance.

Steps:

1.Split Dataset:

In 80%: training set-20%: testing set.

2.Initialize Model:

Existing parameters of Modified Random Forest algorithm include

- Total decision trees in the forest
- Deepest level any tree can grow.
- Least data points allowed in a terminal leaf

3.Train Model:

The model trained on the training set.

4.Validate Model:

The training data were used to train the model, and it was examined using test set performance metrics, such as correctness rate, positive predictive value, sensitivity, balanced F-measure, and the error matrix. [15]

E. INTRUSION DETECTION

The base element of the IDS, the Machine Learning Model Module, is designed to distinguish normal network data from intrusive network data. In identifying patterns corresponding to different intrusions, it applies classification algorithms trained on the preprocessed data. This

composition of the dataset and the particular detection needs determine the optimal way to provide the optimal performance and accuracy. The model is capable of identifying suspicious activity with high accuracy and alerting potential threats in real time via learning from processed information. The system is retrained and refreshed periodically, integrating new information in order to keep pace with new methods of attack so that it remains effective against changing cyber threats. By keeping step with advanced threats and providing secure, dependable, and scalable intrusion detection in unpredictable networks, the adaptive feature enhances the system resilience.

Steps:

1.Preprocess Real-Time Data:

Apply the same preprocessing steps (missing value handling, encoding, normalization) to the incoming network traffic data.

2.Feature Selection:

Use the same feature selection criteria to extract relevant features from the real-time data.

3.Predict Intrusions:

Use the trained Modified Random Forest model to classify the real-time data as:

- ❖ Normal traffic.
- ❖ Malicious traffic (e.g., DDoS, Probe, U2R, R2L attacks).

4.Generate Alerts:

If malicious activity is detected, generate an alert for the network administrator.

5. Log Results:

Detection results and warnings should be preserved in a log file.

III. EXPERIMENTAL RESULTS

The research outcomes the proposed intrusion detection system prove the prowess of its machine learning methods in accurately identifying malicious network activity. The most important features are employed during classification through the system's feature selection and preprocessing algorithms that indeed greatly improved the data's quality. The optimization shortens the processing time as well as enhances computing efficiencies, without compromising on the accuracy of detection. Results show that the machine learning model does quite well in the detection of intrusions since it can distinguish between normal and attacking network traffic. The real-time intrusion detection module effectively analyzes the incoming traffic and identifies unusual activity, thus limiting False negatives and false positives were finally present. As such, this threat analysis and reporting tool boosts the administrators' prevention efforts in taking proactive security actions from gleaning insightful information concerning attack trends.

Table 1: Comparison Table

METRIC	VALUE
Precision	0.95
Recall	0.8
F1-Score	0.92
F-Measure	0.89

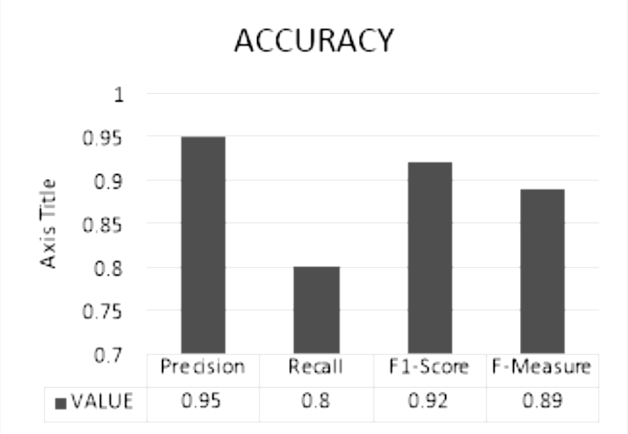

Figure 2: Intrusion Detection Performance Chart

IV. CONCLUSION

This intrusion detection system has shown great promise, as it utilizes feature selection, machine learning mechanisms for classification and advanced data preprocessing techniques, to enhance network security. This technology guarantees quality inputs into the classification model, through the extensive cleaning and optimization of

the dataset, which subsequently lead to its improved accuracy and efficiency while detecting malicious activity during identification. The real-time intrusion detection module marks and identify windows of anomalous traffic in order to reduce security threats besides reducing false positivity and false negativity. It also has a threat analysis and reporting module that provides important insights into attack patterns for preventative security. In general, the approach serves as a reliable and scalable solution to cyberthreat detection, thus offering an effective strategy for network defenses against new security threats emerging.

REFERENCES

[1] Akbanov, Maxat, Vassilios G. Vassilakis, and Michael D. Logothetis. "Ransomware detection and mitigation using software-defined networking: The case of WannaCry." Computers & Electrical Engineering 76 (2019): 111-121.

[2] Cheng, Yongliang, et al. "Leveraging semisupervised hierarchical stacking temporal convolutional network for anomaly detection in IoT communication." IEEE Internet of Things Journal 8.1 (2020): 144-155.

[3] Hajiheidari, Somayye, et al. "Intrusion detection systems in the Internet of things: A comprehensive investigation." Computer Networks 160 (2019): 165-191.

[4] Kumar, Gulshan, Kutub Thakur, and Maruthi Rohit Ayyagari. "MLEsIDSs: machine learning-based ensembles for intrusion detection systems—a review." The Journal of Supercomputing 76.11 (2020): 8938-8971.

[5] Kumar, Ravinder, Amita Malik, and Virender Ranga. "An intellectual intrusion detection system using Hybrid Hunger Games Search and Remora Optimization Algorithm for IoT wireless networks." Knowledge-Based Systems 256 (2022): 109762.

[6] Lei, Shengwei, et al. "HNN: a novel model to study the intrusion detection based on multi-feature correlation and temporal-spatial analysis." IEEE Transactions on Network Science and Engineering 8.4 (2021): 3257-3274.

[7] Li, Kewen, et al. "Improved PSO_AdaBoost ensemble algorithm for imbalanced data." Sensors 19.6 (2019): 1476.

[8] Li, Xinghua, et al. "Sustainable ensemble learning driving intrusion detection model." IEEE Transactions on Dependable and Secure Computing 18.4 (2021): 1591-1604.

[9] Liu, Jinjie, and Sun Sunnie Chung. "Automatic feature extraction and selection for machine learning based intrusion detection." 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2019.

[10] Mazini, Mehrnaz, Babak Shirazi, and Iraj Mahdavi. "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms." Journal of King Saud University-Computer and Information Sciences 31.4 (2019): 541-553.

[11] Mikhail, Joseph W., John M. Fossaceca, and Ronald Iammartino. "A semi-boosted nested model with sensitivity-based weighted binarization for multi-domain network intrusion detection." ACM Transactions on Intelligent Systems and Technology (TIST) 10.3 (2019): 1-27.

[12] Oughton, Edward J., et al. "Revisiting wireless internet connectivity: 5G vs Wi-Fi 6." Telecommunications Policy 45.5 (2021): 102127.

[13] Tama, Bayu Adhi, and Sunghoon Lim. "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation." Computer Science Review 39 (2021): 100357.

[14] Tama, Bayu Adhi, et al. "An enhanced anomaly detection in web traffic using a stack of classifier ensemble." IEEE Access 8 (2020): 24120-24134.

[15] Zhou, Yuyang, et al. "Building an efficient intrusion detection system based on feature selection and ensemble classifier." Computer Networks 174 (2020): 107247.