Anomaly Detection in Network Traffic Using Machine Learning and Deep Learning Techniques

Tirumali Sri Tejaswini[1], Vellanki Pranitha[2], Tirumala Kamala Shreya[3], Dr. R. Manivannan[4]
[1,2,3]*Department of Computer Science and Engineering Stanley College of Engineering & Technology for Women Hyderabad, India*
[4]*Associate Professor, Department of Computer Science and Engineering Stanley College of Engineering & Technology for Women Hyderabad, India*

***Abstract*—Network security has become an essential issue with the rapid development of internet communication and the emergence of cyber-attacks. This paper proposes a system that can detect anomalies in the network traffic using the NSL-KDD dataset with the help of machine and deep learning algorithms. The authors have used various preprocessing steps such as categorical encoding, data cleaning, and feature scaling in the proposed system. The authors have used an ensemble method with four algorithms Support Vector Machine, Random Forest, Deep Neural Network, and Extreme Learning Machine to detect anomalies in the network traffic. The performance of the proposed system is evaluated using various parameters such as accuracy, precision, recall, and confusion matrix. The results show that the proposed system can detect malicious patterns in the network traffic and improve security.**

***Index Terms*—Network Security, Anomaly Detection, Machine Learning, Deep Learning, NSL-KDD Dataset.**

I. INTRODUCTION

Significantly the growth of internet technologies has led to an increase in network traffic. At the same time, cyber-attacks such as hacking, malware, and unauthorized access have also increased. Detecting these activities is a major challenge for organizations. Traditional security measures, such as firewalls and intrusion detection systems often struggle to identify patterns of cyber-attacks. Network Traffic Anomaly Detection is crucial in finding network data patterns. Recently, machine learning has become an effective technique for detecting network anomalies.

Current systems use an algorithm for network anomaly detection. However, the accuracy of detecting network attacks using one algorithm is not high. So, there is a need for a system that uses multiple algorithms. To solve this problem this project proposes a machine learning-based network anomaly detection system that uses algorithms. The proposed system uses Support Vector Machine, Random Forest, Deep Neural Network, Extreme Learning Machine algorithms, for network anomaly detection.

The proposed system will help detect network attacks through a desktop application. Network Traffic Anomaly Detection plays a role here. The use of machine learning and multiple algorithms makes the system strong. This approach can improve the accuracy of detecting network attacks.

II. LITERATURE SURVEY

Researchers have been investigating machine learning and deep learning techniques for detecting anomalies in network traffic to improve intrusion detection systems, aiming to reduce false alarms. These techniques allow for more effective identification of unusual patterns in large network datasets compared to traditional methods currently available. M. Tavallaee et al. (2009) developed the NSL-KDD dataset, which enhanced the evaluation of intrusion detection system performance by eliminating duplicate records from the original KDD dataset. This dataset includes information on network traffic but does not reflect modern network communication patterns. W. Wang et al. (2017) applied deep learning techniques for intrusion detection, which improved their ability to extract features and identify threats; however, their system requires significant computational resources to function [1], [2].

J. Kim et al. (2016) developed a DNN system that classifies network traffic and achieves high detection accuracy. The model needs a lot of training data to work effectively. Y. Yin et al. (2017) used Recurrent Neural Networks (RNN) to process sequential network traffic data, which improved their detection performance. However, this added complexity to their models and increased training time [3], [4]. Z. Wang et al. (2020) researched the use of Support Vector Machine (SVM) and Random Forest as machine learning tools to spot unusual patterns in data. While these classification systems perform well, they struggle with noisy data. N. Moustafa and J. Slay (2015) created feature selection methods to enhance intrusion detection system performance, but their system cannot detect certain complex attack patterns [5], [6].

A. Javaid et al. (2016) created a deep learning intrusion detection system which identifies attack patterns with excellent precision but needs extensive training data and requires extended training periods. The network traffic classification system by K. Kim et al. (2014) applies Support Vector Machine which produces good results when dealing with basic patterns yet struggles to identify complex attack patterns [7], [8].

The existing methods show various limitations which the proposed system addresses by combining different machine learning and deep learning algorithms to achieve better detection results and improved system stability and performance.

III. PROPOSED SYSTEM

Machine learning / Deep learning technologies are used to detect anomalies in network traffic by classifying whether the traffic is either normal or an attack on the server or client. The system is written in Python and utilizes Tkinter to provide a basic desktop application interface.

Dataset Used is the NSL-KDD (which contains already labeled classifyable records of normal and attack instances).

Preprocessing Steps Include:

Encoding categorical features with Label Encoders (protocol_type, service, flag).

Cleaning data by removing invalid log entries. Scaling features using Standard Scaler. Split the data using an 80:20 ratio dividing it between Training and Testing datasets.

Algorithms Used Include:

1. Support Vector Machine (SVM) - finds the optimal hyperplane to separate classes of records.
2. Random Forest - uses multiple decision trees to determine which tree generates the best accuracy.
3. Deep Neural Network (DNN) - learns complex traffic patterns and will generate its own ideal classifiers just by using examples of non-malicious and malicious traffic.
4. Extreme Learning Machine (ELM) - uses a fast-training method while performing well against test data.

Metrics Used to Evaluate Algorithms are accuracy, precision, recall, confusion matrix, and overfitting gap. Functionality Available In GUI Includes uploading a dataset, pre-processing dataset, training the model, and visualizing results.

The goal of the overall project is to provide the end user with a solution that can reliably identify malicious activity on their network by using several techniques from both Machine Learning and Deep Learning to accomplish the tasks mentioned above.

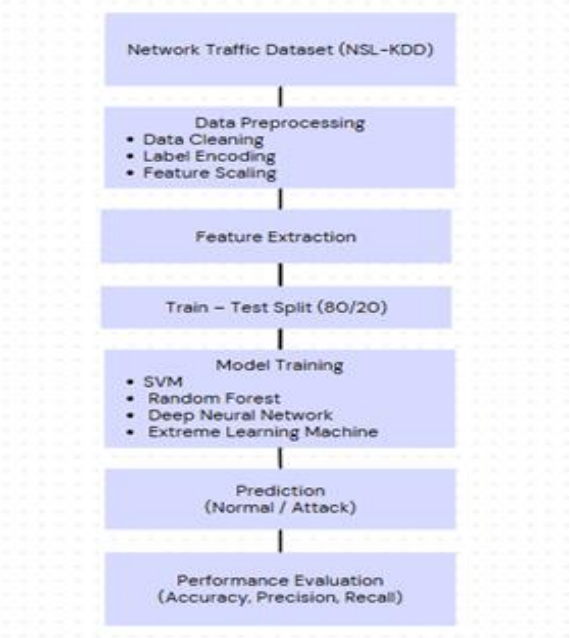

Fig. 1. shows the architecture of the proposed anomaly detection system, including data preprocessing, feature extraction, model training, and prediction stages.

IV. SYSTEM IMPLEMENTATION

Anomaly Detection System Anomaly detection system is developed as a Python based Desktop Application with Graphical User Interfaces similar to GUI's created in other languages. We use the NSL-KDD data set for the Detection of Normal or Malicious Network Traffic using Machine Learning and Deep Learning Algorithms.

A. Preprocessing of The Dataset

The data set has been preprocessed to increase the quality of the data and better Model Performance.

Categorical Encoding: Using Python's Label Encoder Method to convert the Protocol Type, Service and Flag features into numeric values.

Cleaning: Removing all invalid values with small classes to improve the integrity of the data and to make it look better.

Normalisation: Using the Scikit-learn StandardAero technique to standardize the features.

Splitting the Dataset: Create Training and Testing sets from the Data Set using an 80% - 20% Ratio.

B. Training the Model

Several different algorithms were used to detect anomalies.

Support Vector Machine (SVM): SVM identifies a set of hyperplanes (to identify a hyperplane that provides maximum separation of normal and abnormal traffic), and classifies network traffic based on those identified hyperplanes.

Random Forest (RF): Random Forests consist of multiple decision trees combined to increase the accuracy of the predictions made by the Classifier.

Deep Neural Network (DNN): Keras is used to implement a DNN with an Input Layer, Hidden Layers, and Output Layer, and is used to classify using only the neural layers.

Extreme Learning Machine (ELM): ELM Uses a Single Hidden Layer and was created to provide Fast Training and Efficient Performance.

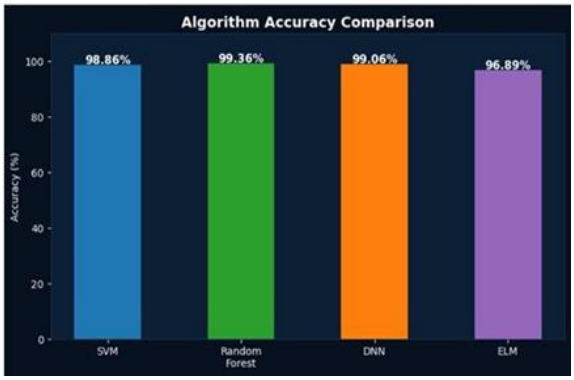

Fig. 2. Accuracy comparison of SVM, Random Forest, DNN, and ELM algorithms.

C. GUI - Graphical User Interface

The GUI is created using Tkinter to allow the user to Upload their own Data Sets for Anomaly Detection.

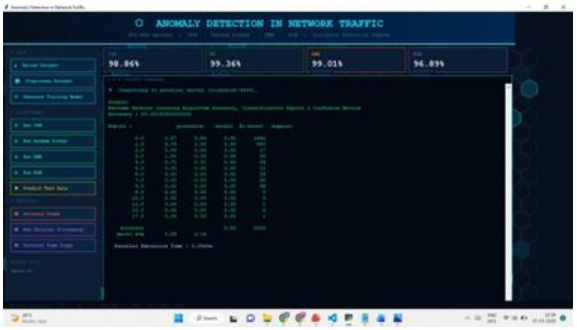

Fig. 4. shows the graphical user interface developed using the Tkinter library for dataset processing, model training, and performance visualization.

V. RESULTS AND ANALYSIS

The Anomaly Detection System is designed as a Python-based Desktop Application. There are GUI Applications of the same type built in other languages too, and the example used for this study is the NSL-KDD data set, which was used to Train and Test a Normal and Malicious Network Traffic Classification Model based on Machine Learning and Deep Learning Algorithms.

A. Dataset Preprocessing

The following Preprocessing techniques were applied to the data set to improve its data quality and enhance the resulting model performance.

1. Categorical Encoding: The Protocol_Type, Service and Flag features were encoded into numeric values using Python's LabelEncoder method.

2. Data Cleaning: All invalid values and classes with small counts were removed from the data set to enhance its Integrity and Aesthetic Appeal.
3. Data Normalisation: All of the features were standardised using the Scikit-learn StandardAero Technique.
4. Dataset Splitting: An 80% - 20% ratio was used to create Training and Testing sets from the data set.

B. Model Training
Anomaly detection was performed using Various Algorithms.
1. Support Vector Machine (SVM): The SVM algorithm identifies the Hyperplanes (and identifies the hyperplane that provides the Maximum Separation of Normal and Abnormal Extreme Data) and classifies the Network Traffic based upon the identified hyperplanes.
2. Random Forest (RF): Random Forests comprise multiple Decision Trees that are combined to enhance the accuracy of the predictions made by the Decision Tree.

VII. CONCLUSION

In our research, we created a machine-learning/deep learning system for identifying anomalous network traffic. Our prototype application is written in Python and is based on the NSL-KDD dataset.
We examined the traffic pattern of networks, and we classified these traffic patterns into normal and malicious groups. We accomplished this by using different algorithms, including: support vector machine (SVM), random forest (RF), deep neural network (DNN), and extreme learning machine (ELM).
Based on the experimental results, the random forest algorithm produced the greatest accuracy performance, and both SVM and DNN performed well also. The ELM produced fast training times but had a lower accuracy compared with the other algorithms.
Our proposed system makes additional improvements to detect and define malicious activities on a network.

ACKNOWLEDGMENT

The authors would like to thank the faculty members and project supervisors of the Department of Computer Science for their valuable guidance and support during the development of this research work.

REFERENCES

[1] Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.
[2] Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection systems," in Proc. 9th EAI Int. Conf., 2016.
[3] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," Expert Systems with Applications, vol. 41, no. 4, pp. 1690–1700, 2014.
[4] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive dataset for network intrusion detection systems," in Proc. Military Communications and Information Systems Conf., 2015.
[5] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Proc. IEEE Symp. Security and Privacy, 2010.
[6] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in Proc. IEEE Symp. Computational Intelligence, 2009.
[7] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks," IEEE Access, vol. 6, pp. 1792–1806, 2018.
[8] Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.