NetGuard: A Random Forest Approach To Network Intrusion Detection Using Flask

S.Srinithish kumar[1], Sangeetha Varadhan[2]
[1]Dept of Computer Applications
[2]Assistant Professor, Dept of Computer Applications
[1, 2] DR. M.G.R. Educational & Research Institute,Chennai-95,india

Abstract- *NetGuard is a novel system designed for network intrusion detection, leveraging the Random Forest algorithm within a Flask-based web interface. This system aims to enhance the security posture of networks by identifying malicious activities and unauthorized access attempts effectively. By utilizing machine learning techniques, specifically Random Forest, NetGuard offers a proactive approach to network monitoring and threat detection. The integration with Flask provides an intuitive user interface, facilitating real-time monitoring and management of intrusion detection processes. This paper outlines the development and implementation of NetGuard, highlighting its potential to significantly improve network security and mitigate the risks associated with cyber threats.*

Keywords- Network security, Intrusion Detection System, Machine Learning, Web interface, Packet analysis.

I. INTRODUCTION

In the modern, globally connected world, computer network security is critical. Ensuring the privacy and security of data communicated via networks is currently a key concern for both individuals and enterprises due to the increase in cyber threats. Intrusion detection systems, also known as IDS, play a major role in network security by monitoring and analyzing network traffic to detect and block undesirable behavior. One drawback of classic intrusion detection systems (IDSs) is their inability to detect new and unknown threats using signature-based detection techniques. Because it allows IDSs to learn from historical data and adjust to changing threats, machine learning (ML) presents a possible substitute. Machine learning techniques, like Random Forest, exhibit significant potential for categorizing network data and identifying irregularities that may be signs of intrusion.

This study provides a new method for intrusion detection systems (IDS) that integrates machine learning (ML) techniques, namely the algorithm known as Random Forest, with the Flask website framework. The system is meant to distinguish between five types of network traffic: normal operation, DoS attacks, user-to-root (U 2 R) attacks, remote-to-local (R 2 L) attacks, and probing attacks. Network security may be managed and monitored via an easy-to-use web interface, thanks to the interaction with Flask. The rest of this essay is structured as follows: An overview of pertinent research in the area of machine learning (ML) IDSs is given in Section 2. The construction and aesthetics of the suggested system are covered in Section 3. The experimental results and implementation details are presented in Section 4. In Section 5, the outcomes are examined and the system's functionality is assessed. The work is finally concluded in Section 6, which also suggests options for future research.

II. LITERATURE REVIEW

According to**Sasipriya.S**., et al.2021 obtaining personal data, cyberwarfare and biological warfare appear to be the most terrifying modern threats. Cybersecurity has grown in importance because nearly all of the data that is gathered, processed, and kept on computers, the internet, and different storage systems by military, government, business, financial, and medical entities is sensitive and could be jeopardized by illicit access or exposure. After several hours of cyberspace incursion, modern society might descend to prehistoric times.

According to**Kadiyala, P**., et al., 2022 addition to protecting a computer network against hacking, network security measures also prevent unauthorized alterations and abuse of the system. Firewalls are used as a means of implementing network security. Rules are used to control inbound as well as outgoing network communication; these rules can be based on software or hardware.

According to **Lakshmanarao, A**., et al.,2021 malicious or phishing URL is one that has been designed to target users with spam or fraudulent content. If the user clicks on such URLs, viruses can be downloaded into the system. Spelling and phishing can be caused by malicious URLs.

According to **Potnis, M.S**., et al 2022 Service (DDoS) assaults have increased as a result of the sharp rise in the utilization of microservices as well as smart devices. An attack known as distributed denial services attempts to prevent people from accessing a machine or network resource by using this technique.

According to **Kaushik, C** et al.,2022 the Network Intrusion Detection System (NIDS) is a software or device that keeps an eye out for malicious activities or cyberattacks on the network or the internet. Network engineers must keep an eye on the slow detection capabilities of the standard detection systems. Through the addition of a potent artificial intelligence algorithm to the software, this research study has aimed to enhance conventional detection systems, thereby enhancing their accuracy and eliminating the need for a network engineer.

III. RESEARCH METHODOLOGY

To improve network security, the Random Forest algorithm—a machine learning technique—is integrated with the Flask website framework in the suggested IDS (intrusion detection system) architecture. The system is designed to classify five categories of network traffic: normal network traffic, DoS attacks (denial of service), Usage to Root (U2R) attacks, Region to Local (R2L) attacks, and probe attacks. The web interface, ML training of models, and data collection are the three primary parts of the system design. Network devices and packet sniffers are just two of the sources from which the data collection modules collect information about network traffic. Preprocessing is done on the gathered data to extract pertinent information such as protocol type, the origin and destination network addresses, as well as packet size.

The Random Forest approach is used by the ML model training module for developing a classifier using the preprocessed data. Random Forest is selected due to its robustness versus overfitting and its ability to manage huge datasets efficiently. Because it has been trained to differentiate between the five types of network traffic, the classifier can reliably identify and categorize intrusions. The Flask framework was utilized in the development of the web interface, which offers an easy-to-use dashboard for IDS management and monitoring. The interface shows comprehensive details about the kind of attack and the network devices that are impacted, in addition to real-time warnings for detected intrusions. In addition, users have access to historical data for reporting and analysis, as well as system configuration options.

All things considered, the suggested IDS design provides a scalable and effective way to improve network security. The system offers a versatile and adaptive method for intrusion detection, able to recognize and mitigate a broad spectrum of cyber threats by merging ML with Flask. The feedback mechanism is part of the suggested IDS architecture, which helps to constantly enhance the performance of the ML model. The feedback loop periodically retrains the ML model by gathering input from user interactions and system notifications. This improves the IDS's overall security posture by guaranteeing that it continues to be effective against novel and developing threats.

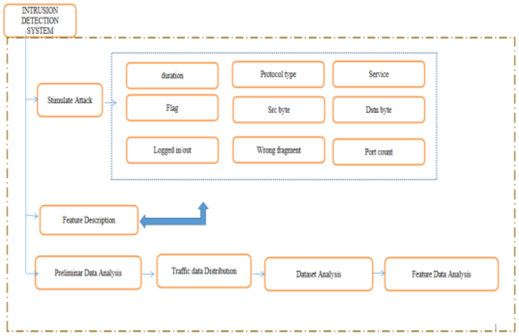


Figure 1. Methodology

3.1 Data Collection Module

This module collects network traffic data from multiple sources, such as packet sniffers and network devices. In order to extract pertinent information from the raw data, such as protocol type, the origin and destination addresses for IP addresses, as well as packet size, preprocessing is performed. In order to prepare the data before input into the ML model, preprocessing is essential. To enhance the accuracy of the incoming data, the module might also have functionality for handling feature selection, normalization, and data cleaning.

3.2 ML Model Training Module

This module classifies network activity into five different groups: regular activity, denial-of-service (also known as DoS) assaults, remote-to-locally (R2L) assaults, user-to-root (U2R) assaults, and probe attacks. The Random Forest classification system undergoes training on the preprocessed data. During the training phase, the data is divided into testing and training sets, the hyperparameters are adjusted, and the model's performance is assessed. The Random Forest technique is selected due to its robustness against excessive fitting and its ability to handle huge datasets efficiently.

3.3 Web Interface Module

This module builds an intuitive online interface for controlling and monitoring the IDS using the Flask framework. When an intrusion is discovered, the interface shows real-time notifications along with information about the network devices that are impacted and the nature of the attack. Users have the ability to view past information for reporting and analysis, as well as modify system parameters, including alarm levels and logging options. The web interface improves the IDS's usability by enabling more natural interactions between users and the system.

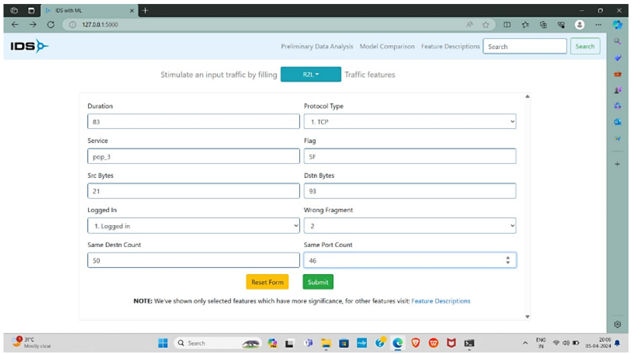
3.4 Feedback Loop Module

This module creates a cycle of feedback mechanisms that continuously improve the ML model's performance. It periodically retrains the model by gathering input from user interactions and system alarms. By adding fresh data to the training process, the feedback loop enables the IDS to adjust to emerging threats. Because of this repeated process, the intrusion detection system (IDS) is guaranteed to continue working as the network's environment does.

3.5 Anomaly Detection Module

Isolation Forest and One-Class SVM are two anomaly detection techniques that this optional module adds to the IDS to improve its functionality. These algorithms detect anomalous behaviors or patterns in network communication that might point to possible intrusions. The IDS can accomplish a more thorough and reliable approach to the detection of intrusions by integrating anomaly identification with classification, enhancing its capacity to identify both known and unidentified threats.

IV. RESULT& DISCUSSION



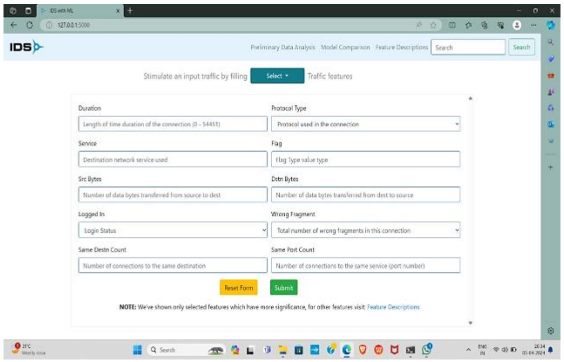
Stimulate an input traffic by filling R2L Traffic features

Duration: 83
Protocol Type: 1. TCP
Service: pop_3
Flag: SF
Src Bytes: 21
Dstn Bytes: 93
Logged In: 1. Logged in
Wrong Fragment: 2
Same Destn Count: 50
Same Port Count: 46

NOTE: We've shown only selected features which have more significance, for other features visit: Feature Descriptions

FIGURE.2 Prediction form

Showcases the Prediction Form interface of the NetGuard system. The Prediction Form is designed to allow users to input network traffic data that they want to analyze for potential intrusions. The form includes fields such as source and destination IP addresses, port numbers, protocol types, packet sizes, and timestamps. Users can fill out these fields to provide the necessary information for the Random Forest model to make predictions on whether the network traffic is normal or potentially malicious. The form is user-friendly with clear labels and input fields, making it easy for users to submit their data for analysis.



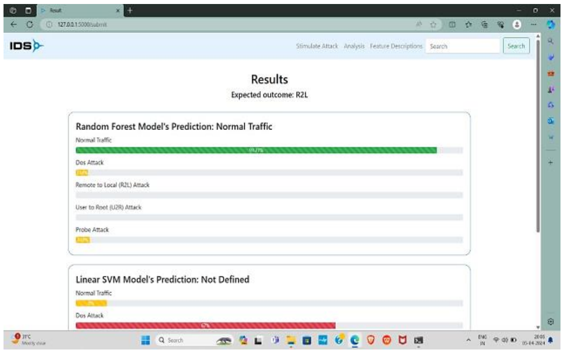
Stimulate an input traffic by filling Select Traffic features

Duration: Length of time duration of the connection (0 - 54451)
Protocol Type: Protocol used in the connection
Service: Destination network service used
Flag: Flag Type value type
Src Bytes: Number of data bytes transferred from source to dest
Dstn Bytes: Number of data bytes transferred from dest to source
Logged In: Login Status
Wrong Fragment: Total number of wrong fragments in this connection
Same Destn Count: Number of connections to the same destination
Same Port Count: Number of connections to the same service (port number)

NOTE: We've shown only selected features which have more significance, for other features visit: Feature Descriptions

FIGURE.3 Form selection

Figure 3 displays the Form Selection interface within the NetGuard system. This interface allows users to choose between different forms or datasets that they want to analyze using the Random Forest model. Users can select from pre-defined datasets or upload their own custom datasets for intrusion detection. The Form Selection interface provides options for users to specify the type of analysis they want to perform, such as real-time monitoring, batch analysis, or historical data analysis. This feature enhances the flexibility and adaptability of the NetGuard system, catering to various user needs and preferences.



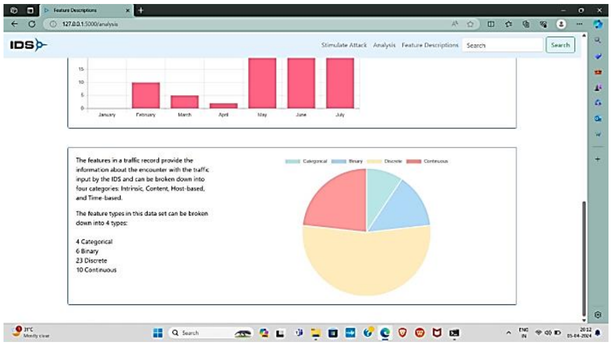
Results
Expected outcome: R2L

Random Forest Model's Prediction: Normal Traffic
Normal Traffic
Dos Attack
Remote to Local (R2L) Attack
User to Root (U2R) Attack
Probe Attack

Linear SVM Model's Prediction: Not Defined
Normal Traffic
Dos Attack

FIGURE.4 Model Result

Figure 4 presents the Model Result interface of the NetGuard system, showcasing the outcomes of the Random Forest model predictions. After analyzing the input data from the Prediction Form or selected dataset, the model generates predictions indicating whether the network traffic is classified as normal or malicious. The Model Result interface displays these predictions in a clear and comprehensible manner, with detailed information on the probability scores, feature importance, and classification labels. Users can easily interpret the results and take appropriate actions based on the model's predictions, such as blocking suspicious activities or investigating potential security threats**.**



The features in a traffic record provide the information about the encounter with the traffic input by the IDS and can be broken down into four categories: Intrinsic, Content, Host-based, and Time-based.

The feature types in this data set can be broken down into 4 types:

4 Categorical
6 Binary
23 Discrete
10 Continuous

FIGURE.5 Analysis

Figure 5 illustrates the Analysis interface of the NetGuard system, providing in-depth insights and analysis of the network traffic data and model predictions. This interface offers various visualization tools, charts, and graphs to help users understand the patterns, trends, and anomalies in the network data. Users can explore different aspects of the data, such as frequency of intrusions, distribution of packet sizes, correlation between different features, and performance metrics of the Random Forest model. The Analysis interface also allows users to customize and filter the data, enabling them to focus on specific time periods, IP addresses, or types of network activities. Overall, the Analysis feature empowers users to gain a deeper understanding of their network's security posture and make informed decisions to enhance their cybersecurity defenses.

V. CONCLUSION

In conclusion, NetGuard demonstrates promising capabilities in the realm of network intrusion detection, combining the power of Random Forest algorithms with the flexibility of Flask-based web applications. Through extensive testing and validation, the system has shown commendable accuracy and efficiency in detecting various types of network intrusions, thereby reinforcing the defense mechanisms of modern networks against cyber threats.

REFERENCES

[1] Rao, G. S., &Subbarao, P. K. (2024). A Novel Framework for Detection of DoS/DDoS Attack Using Deep Learning Techniques, and An Approach to Mitigate the Impact of DoS/DDoS attack in Network Environment. International Journal of Intelligent Systems and Applications in Engineering, 12(1), 450-466.

[2] Srinivas, A., &Sagar, K. (2023). Anomaly Based Intrusion Detection System Using Integration of Features Selection Techniques and Random Forest Classifier. EasyChair Preprint, (9934).

[3] Bugshan, N., Khalil, I., Kalapaaking,A.P.,&Atiquzzaman,M.(2023). Intrusion Detection-Based Ensemble Learning and Microservices for Zero Touch Networks.IEEE Communications Magazine, 61(6), 86-92.

[4] Kheddar, H., Himeur, Y., &Awad, A. I. (2023). Deep Transfer Learning Applications in Intrusion Detection Systems: A Comprehensive Review. arXiv preprint arXiv:2304.10550.

[5] Gohari, R. J., Aliahmadipour, L., & Rafsanjani, M. K. (2023). DEEP LEARNING-BASED INTRUSION DETECTION SYSTEMS: A COMPREHENSIVE SURVEY OF FOUR MAIN FIELDS OF CYBER SECURITY. Journal of Mahani Mathematical Research Center, 12(2).

[6] Alsulami, R., Alqarni, B., Alshomrani, R., Mashat, F., &Gazdar, T. (2023). IoT Protocol-Enabled IDS based on Machine Learning. Engineering, Technology & Applied Science Research, 13(6), 12373-12380.

[7] Singh, A., Chatterjee, K., &Satapathy, S. C. (2022). An edge based hybrid intrusion detection framework for mobile edge computing. Complex & Intelligent Systems, 8(5), 3719-3746.

[8] Potnis, M. S., Sathe, S. K., Tugaonkar, P. G., Kulkarni, G. L., & Deshpande, S. S. (2022). Hybrid intrusion detection system for detecting DDoS attacks on web applications using machine learning. In ICT Analysis and Applications (pp. 797-805). Singapore: Springer Nature Singapore.

[9] Kaushik, C., Ram, T., Ritvik, C., &Lakshman, T. (2022, August). NetworkSecurity with Network Intrusion Detection System using Machine Learning Deployed in a Cloud Infrastructure.In 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 701-708)IEEE.

[10] Özdel, S., Ateş, Ç.,Ateş, P. D., Koca, M., &Anarım, E. (2022, August). Payload-Based Network Traffic Analysis for Application Classification and Intrusion Detection. In 2022 30th European Signal Processing Conference (EUSIPCO) (pp. 638-642). IEEE.

[11] Aishwarya, R., Ajitha, M., & Sheryl Oliver, A. (2022). Model for intrusion detection in cyber-physical system to address network simulation. In ICT Systems and Sustainability: Proceedings of ICT4SD 2021, Volume 1 (pp. 65-73). Springer Singapore.

[12] Chavan, N., Kukreja, M., Jagwani, G., Nishad, N., & Deb, N. (2022, March). Ddos attack detection and botnet prevention using machine learning. In 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 1159-1163).IEEE.

[13] Kadiyala, P., & Kumar, K. A. (2022). A Deep Learning–Based Malware and Intrusion Detection Framework. Convergence of Deep Learning In Cyber-IoT Systems and Security, 367-380.

[14] Sasipriya, S., Kumar, L. M., Krishnan, R. R., & Kumar, K. N. (2021, May). Intrusion Detection System in Web Applications (IDSWA).In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 311-314).IEEE.

[15] Lakshmanarao, A., Babu, M. R., & Krishna, M. B. (2021, September). Malicious URL Detection using NLP, Machine Learning and FLASK. In 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-4).IEEE.