[1]Veera V Rama Rao M

[2]Anuj Rapaka

[3] M Prasad

[4]Raja Rao PBV

[5]P T Satyanarayana murty

[6]Kiran Sree Pokkuluri

Enhancing Network Security: Leveraging Machine Learning for Intrusion Detection

Abstract: - In order to precisely identify and categorise different kinds of network attacks, this study focuses on applying machine learning approaches for network intrusion detection. Data collection, preprocessing, feature scaling, model definition, feature selection, and assessment metrics are all part of the methodology. Different machine learning models, including Decision Tree Classifier and Random Forest Classifier, are considered, along with the use of all features or some part of features for each attack category. Evaluation is performed using K-fold cross-validation, with metrics such as accuracy, precision, recall, and F1-score analysed. Results indicate the efficiency of Random Forest Classifier in handling high-dimensional datasets and improving detection accuracy, making it a superior choice for network intrusion detection tasks.

Keywords: Machine Learning, Random Forest Classifier, Feature Selection, Intrusion Detection, K-fold Cross-Validation

I. INTRODUCTION

Due to the prevalence of cyber-attacks, safeguarding computer networks is crucial to ensure data security. Traditional detection methods are becoming less effective against increasingly complex cyber threats. Therefore, advanced techniques are required to identify these threats promptly. This study explores the use of machine learning, a computational approach, to assist in the identification of cyber threats within network data.

Machine learning empowers computers to identify patterns and forecast results from data. When applied to vast amounts of network data, it can detect hidden patterns that traditional analysis techniques may overlook. Moreover, it has the ability to continuously learn and refine its capabilities, making it highly adept at identifying both known and emergent threats.

This study aims to assess the efficiency of machine learning techniques in detecting network attacks. Various network data features, such as data patterns, origin, and destination addresses, will be utilized. Machine learning models will be trained on both normal and attack data to enhance their ability to differentiate between legitimate and malicious network traffic.

To enhance the efficiency of intrusion detection systems, we will explore strategies for optimizing feature selection and model choice. By evaluating various combinations of features and employing different algorithms, we aim to ascertain the most efficient methods for detecting diverse attack types. Additionally, we will utilize techniques such as Recursive Feature Elimination to identify the most critical features for threat detection.

[1] *Corresponding author: Assistant Professor, Dept of CSE, Shri Vishnu Engineering College for Women

[2] Assistant Professor, Dept of CSE, Shri Vishnu Engineering College for Women

[3] Associate Professor, Dept of CSE, Shri Vishnu Engineering College for Women

[4] Associate Professor, Dept of CSE, Shri Vishnu Engineering College for Women

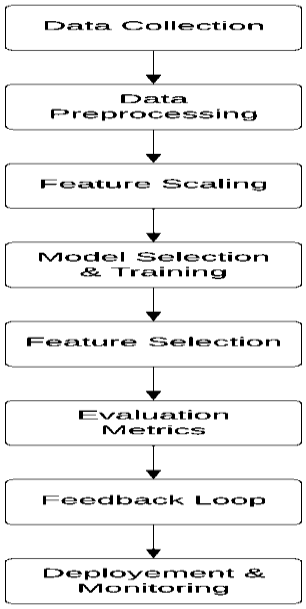
[5] Assistant Professor, Dept of CSE, Shri Vishnu Engineering College for Women

[6] Professor, Dept of CSE, Shri Vishnu Engineering College for Women

This study strives to enhance network security through the implementation of machine learning, a cutting-edge technology designed to detect and mitigate cyber threats. By employing adaptable and precise threat models, we endeavor to safeguard computer networks and protect sensitive data from potential breaches.

II. METHODOLOGY

The process involves several key steps: 1. Gather data from network traffic, encompassing both regular and suspicious activity. 2. Separate the data into two portions: one for training the detection model and one for evaluating its performance. 3. Prepare the data by transforming categorical information into numerical values and assigning specific values to each attack type. 4. Generate distinct data sets for each type of attack to facilitate targeted model training



.

Figure 1 Architecture for Enhancing Network Security

Feature scaling using StandardScaler ensures that features have comparable scales, enhancing model performance. Various machine learning models, including Decision Tree Classifier and Random Forest Classifier, are considered, Recursive Feature Elimination (RFE) is used in feature identification in order to find the most pertinent features. Through K-fold cross-validation, evaluation metrics including accuracy, precision, recall, and F1-score are used to evaluate the working of the model.

1. Data Collection:

Data collection involves gathering network traffic data comprising both normal and intrusive instances. The dataset is organized into features that give inputs into various aspects of network activity.

The dataset containing network traffic data is collected with a comprehensive array of attributes that provide detailed insights into various aspects of network activity. These attributes include "duration", indicating the duration of the connection; "protocol_type", specifying the type of protocol used such as TCP or UDP; "service", representing the type of service requested; and "flag", indicating the status of the connection (e.g., normal or error). Other attributes include "src_bytes" and "dst_bytes", denoting the number of bytes sent from the source and to the destination respectively. Additionally, attributes like "land", "wrong_fragment", "urgent", and "hot" provide further details such as whether the connection is the quantity of wrong pieces, urgent packets, hot indicators, and from/to the same host/port, respectively. Various attributes also capture security-related information like number of failed login attempts ("num_failed_logins"), whether a user is logged in ("logged_in"), and indicators of compromise such as "num_compromised" and "root_shell". Moreover, the dataset encompasses features related to connection counts, error rates, and rates of connections to the same or different services and hosts, among others. Every occurance in the dataset is labeled with a "label" indicating the type of network activity, and the "last_flag" attribute represents the last control flag. This comprehensive collection of network attributes facilitates thorough analysis and modeling for network intrusion detection and security purposes.

The dataset is then split into train and test data sets to facilitate model training and evaluation.

2. Data Preprocessing:

In network intrusion detection, data preprocessing is crucial to make computers comprehend it. We convert categorical data like protocol types and services into numerical values using LabelEncoder and OneHotEncoder techniques. OneHotEncoder specifically transforms categories into binary codes, assigning a unique code to each category. This allows computers to distinguish different categories and identify anomalous patterns in the network traffic.

LabelEncoder assigns unique numerical labels to different attack types. For instance, it could assign 0 to regular traffic, 1 to a specific attack type, 2 to another attack type, and so on. By doing this, the computer can easily distinguish between normal behavior and potential threats. Once this labeling is complete, we create separate data sets for each attack type. This allows us to concentrate on each attack type separately and ensure the accuracy of our detection techniques.

3. Making New Datasets:

To generate distinct datasets, we follow these steps: 1. Label Extraction: We identify the labels associated with data points in the original dataset, indicating the type of activity they represent. 2. Numerical Mapping: We assign a unique numerical value to each attack label. For example, we might assign 0 to normal traffic, 1 to a specific type of attack, and so on. This standardizes the representation of different attack types, simplifying their analysis.

After mapping the data, we divide it into distinct datasets based on attack types. We exclude instances with labels associated with different attack types from each dataset. As a result, each dataset only includes instances of a specific attack category, enabling us to concentrate our analysis and model development on each type of attack separately.

Overall, these steps help us prepare the data for analysis and model development, making sure that our intrusion detection systems are accurate and effective.

4. Feature scaling of New Datasets:

To enhance the efficiency of machine learning algorithms, it's crucial to scale the features in our datasets. This entails ensuring that the various network traffic data attributes have similar scales. To achieve this, we use the StandardScaler technique, which alters each feature to have a mean of 0 and a standard deviation of 1. This process standardizes the scales of the features, prohibiting any individual feature from having an undue impact on the learning process due to its size.

Using StandardScaler to scale features in intrusion detection models: - Enhances efficiency of algorithm training, resulting in better model performance. - Minimizes the effects of outliers and data noise, boosting model accuracy and reliability.

5. Model Description:

Finally, we explain the models we use to classify various types of network attacks based on different combinations of features and classifiers.

1.1 Using Decision Tree Classifier with all Features:

Let's say we want to use a Decision Tree Classifier to classify network attacks. This classifier will start with the entire set of features available for each attack category. It will then repeatedly split the data into smaller groups, based on the feature that best separates the different types of attacks. This allows the classifier to create complex boundaries that can distinguish between different attacks.

1.2 Using Decision Tree Classifier with Subset of Features:

This approach involves selecting some part of features for each attack category and training a Decision Tree Classifier. Feature selection is crucial for improving model efficiency and reducing overfitting. By identifying the most informative features using models like Recursive Feature Elimination (RFE), the classifier focuses on the most discriminative attributes, leading to a more parsimonious model with improved generalization capability.

1.3 Using Random Forest Classifier with All Features:

In this approach, a Random Forest Classifier is employed, considering all features for each attack category. Random Forest is an ensemble learning technique that builds multiple decision trees and combines their predictions through voting or averaging. By leveraging the diversity of multiple decision trees, Random Forest enhances the robustness and accuracy of the classification model, particularly in scenarios with high-dimensional and complex datasets.

1.4 Using Random Forest Classifier with Subset of Features:

Similar to the previous approach, this strategy involves selecting some part of features for each attack category and training a Random Forest Classifier. Focusing on fewer features helps the classifier pick out very important information for telling apart different types of network attacks. This makes it easier for the computer to do its job well without getting overwhelmed by too much data.

In summary, these model descriptions offer different ways to make intrusion detection systems work better. By trying out different combinations of features and classifiers, the goal is to find the best method for accurately spotting different types of network attacks, which ultimately makes our computer networks safer.

6. Feature Selection:

Recursive Feature Elimination (RFE) is a method used to figure out which features in our dataset are the most important. It does this by repeatedly training a computer program on different sets of features and then seeing which ones it finds most useful. This way, we can pick out just the features that are really good at telling the difference between regular network traffic and the sneaky stuff that might be an attack.

III. EVALUATION METRICS

K-fold cross-validation is utilized to evaluate the performance of intrusion detection models. This technique involves dividing the dataset into K subsets or folds, training the model K times using K-1 folds for training and the remaining fold for validation. Performance metrics such as accuracy, precision, recall, and F1-score are then averaged across all K iterations to obtain a robust estimate of the model's performance. K-fold cross-validation ensures rigorous assessment of the model's ability to generalize to unseen data and aids in selecting optimal hyperparameters and features for intrusion detection models.

IV RESULTS

The results demonstrate the efficiency of Random Forest Classifier in handling high-dimensional datasets and improving detection accuracy compared to Decision Tree Classifier. When considering all features or a subset of features for each attack category, Random Forest Classifier consistently outperforms its counterparts, showcasing its robustness and efficiency in detecting network intrusions.

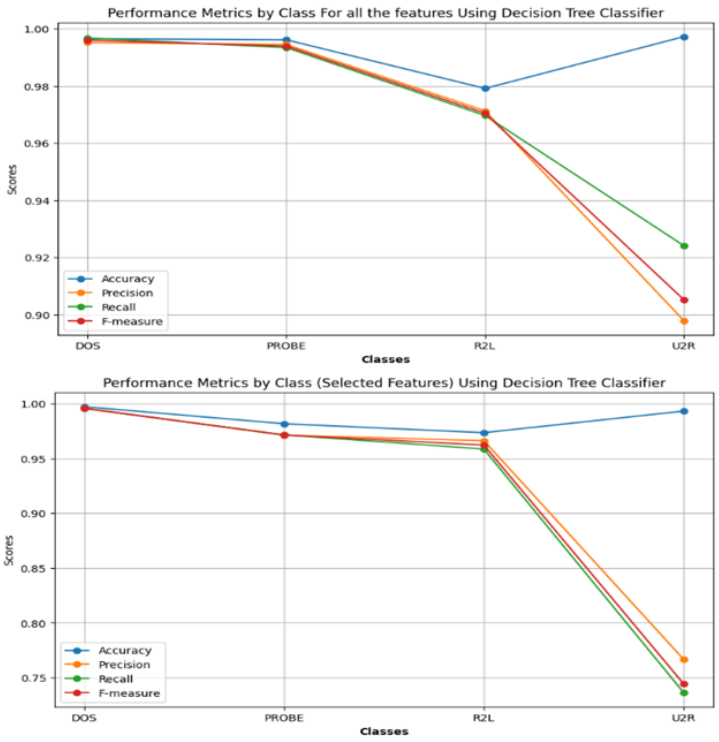


Figure 2 Performance Metrics by class for all features & Selected Features using Decision Tree Classifier

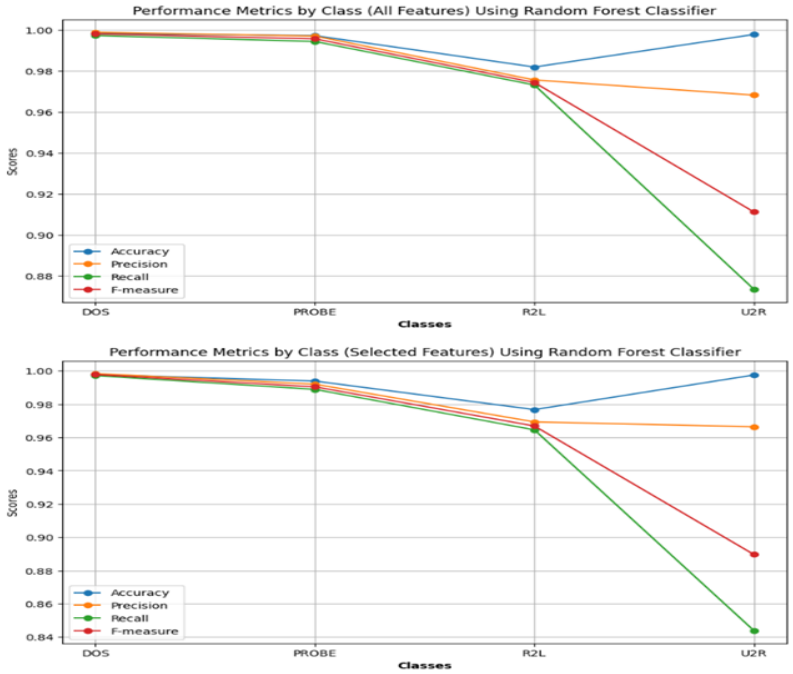


Figure 3 Performance Metrics by class all Features & Selected Features using Random Forest Classifier

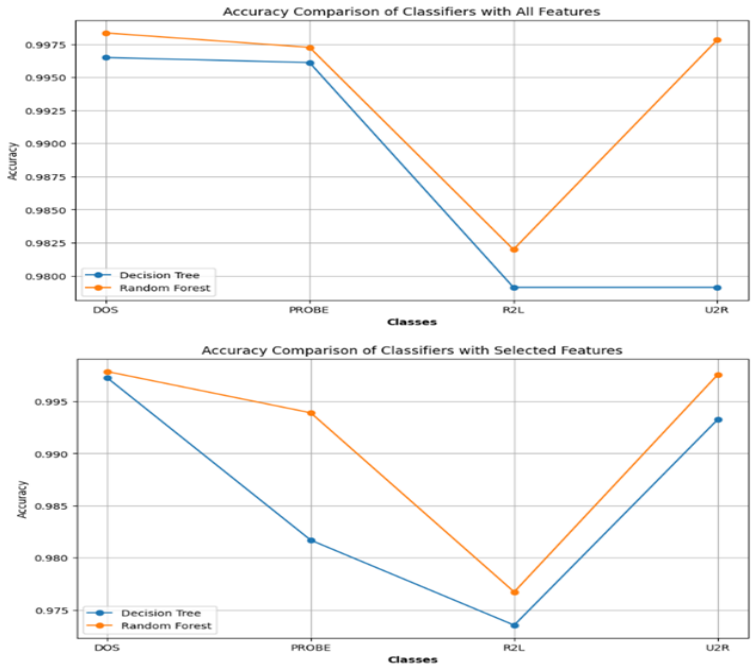


Figure 4Accuracy Comparison of classifiers with all features & Selected Features

Table 1 Comparison of Network Attack Detection Performance

Classifier	DOS	PROBE	R2L	U2R
Decision Tree (All Features)	0.99651 (+/- 0.00353)	0.99613 (+/- 0.00391)	0.97912 (+/ - 0.01100)	0.97912 (+/- 0.01100)
Random Forest (All Features)	0.99837 (+/ - 0.00186)	0.99728 (+/ - 0.00266)	0.98198 (+/ - 0.00619)	0.99785 (+/ - 0.00213)
Decision Tree (Selected Features)	0.99724 (+/ - 0.00259)	0.98170 (+/ - 0.00776)	0.97356 (+/ - 0.00870)	0.99325 (+/ - 0.00345)
Random Forest (Selected Features)	0.99785 (+/- 0.00213)	0.99390 (+/- 0.00340)	0.97674 (+/ - 0.00727)	0.99755 (+/- 0.00306)

V. CONCLUSION:

In conclusion, this study highlights the pivotal role of machine learning techniques in network intrusion detection, emphasizing the superiority of the Random Forest Classifier for handling complex datasets and improving detection accuracy. By using machine learning technology, companies can strengthen their computer network defenses and stop different types of attacks. The Random Forest Classifier is a powerful tool for this, especially when it comes to spotting complicated patterns and connections between different parts of the network.

When we compare how well the Random Forest classifier works to another method called the Decision Tree classifier, we see some big differences. This is especially true when we look at all the features of the data or just a specific set of 13 features for each type of attack. The Random Forest Classifier consistently does better. It does this because it uses a group of decision trees that each look at different parts of the data. This way, it's less likely to make mistakes based on just one tree's view. This group approach helps it to work well even with messy data and makes it better at finding the right answer.

The Random Forest Classifier also does a great job of understanding how different parts of the network data relate to each other, especially when there's a lot of noise or unusual data. Whether it's using all the features or just a

selected few, the Random Forest model is good at picking out the very important ones for spotting different types of attacks. This ability, along with its group learning method, makes the Random Forest Classifier really accurate and efficient at spotting network intrusions.

Overall, this study shows that the Random Forest Classifier is a fantastic tool for making computer networks more secure. It's great at stopping all kinds of attacks, which makes it a vital part of modern cybersecurity.

REFERENCES

[1] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking," in 10th IEEE International Conference on Data Mining (ICDM), 2010, pp. 1189-1194.

[2] M. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, "A Framework for Network Intrusion Detection Using a Hybrid Model," Journal of Network and Computer Applications, vol. 35, no. 3, pp. 1035-1043, 2012.

[3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[4] G. E. Hinton et al., "Improving Neural Networks by Preventing Co-adaptation of Feature Detectors," arXiv preprint arXiv:1207.0580, 2012.

[5] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann, 2011.

[6] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[7] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in IEEE Symposium on Security and Privacy, 2010, pp. 305-316.

[8] P. N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining. Pearson Addison Wesley, 2005.

[9] D. Dua and C. Graff, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2019. [Online]. Available: [https://archive.ics.uci.edu/ml].

[10] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, 2009.

[11] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised Machine Learning: A Review of Classification Techniques," Emerging Artificial Intelligence Applications in Computer Engineering, pp. 3-24, 2007.

[12] Gandomi, M. Haider, and A. R. Alavi, "Beyond the Hype: Big Data Concepts, Methods, and Analytics," International Journal of Information Management, vol. 35, no. 2, pp. 137-144, 2015.

[13] M. Tavallaee et al., "A Detailed Analysis of the KDD CUP 99 Data Set," in Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 1-6.

[14] Bhattacharya and D. Chakrabarti, "Performance Analysis of Different Machine Learning Algorithms for Network Intrusion Detection: A Comparative Study," International Journal of Computer Applications, vol. 53, no. 17, pp. 36-42, 2012.

[15] Y. S. Sone and J. M. Kang, "A Study on Machine Learning-Based Network Intrusion Detection System," Cluster Computing, vol. 23, no. 1, pp. 455-464, 2020.

[16] Y. Yin et al., "Recent Advances in Deep Learning-Based Intrusion Detection Systems: A Comprehensive Review," IEEE Access, vol. 8, pp. 33703-33722, 2020.

[17] S. Bhattacharya et al., "A Survey on Intrusion Detection with Machine Learning Techniques," Journal of Network and Computer Applications, vol. 168, p. 102790, 2021.

[18] S. K. Parida and A. Jena, "A Survey on Intrusion Detection System using Machine Learning Techniques," in Proceedings of the 5th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2020, pp. 1-6.

[19] H. Hajibabaei et al., "Intrusion Detection Systems: A Comprehensive Review," Journal of Network and Computer Applications, vol. 164, p. 102818, 2020.

[20] R. Alhichri and E. S. El-Alfy, "A Comparative Study of Intrusion Detection Techniques," Computers & Security, vol. 105, p. 102217, 2021.

[21] M. Kamble and D. P. Chaudhari, "A Comprehensive Review on Intrusion Detection System Using Machine Learning Techniques," in Proceedings of the 2020 4th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 135-139.