Network Threat Hunting and Detection System using Machine Learning

Savitha.J[1], Sankari. A[1], Mini Bala. G[1], AnbuRaja. R[1,] , Mr. R .Ponneela Vignesh[2]
UG Scholar[1] , HOD [2]
[1,2]Department of Computer Science and Engineering
Tamilnadu College of Engineering, Coimbatore, Tamil Nadu, India.

***Abstract* — In the modern digital landscape, traditional intrusion detection systems (IDS) are increasingly inadequate to address sophisticated cyber threats. This study introduces a Network Threat Hunting and Detection System powered by machine learning (ML) to address the existing challenges. Using the KDD Cup 1999 dataset, the study explores various ML models, including Logistic Regression, Decision Tree, Random Forest, Naive Bayes, K-Nearest Neighbours (KNN), and Support Vector Machine (SVM). Data preprocessing techniques such as normalization, encoding, and feature selection were employed to enhance model performance. Among the models evaluated, the Random Forest classifier demonstrated superior results, achieving an accuracy of over 99%. This system effectively distinguishes between normal and malicious network traffic, offering a scalable, adaptive, and highly accurate approach to intrusion detection. Future enhancements include integrating real-time detection, deploying deep learning models, and using explainable AI for greater transparency.**

***Index Terms*— Network Security, Intrusion Detection System, Machine Learning, Random Forest, Cybersecurity.**

I. INTRODUCTION

In the contemporary digital era, the pervasive integration of information technology into daily life, commerce, governance, and critical infrastructure has profoundly transformed the world. However, this digital transformation has also significantly increased the attack surface for cyber threats. Organizations, institutions, and individuals now face sophisticated and persistent cybersecurity challenges that threaten data confidentiality, system integrity, and service availability. These growing threats underscore the urgent need for intelligent, dynamic, and robust cybersecurity solutions capable of proactive threat detection and response.

Traditional security solutions, such as firewalls, antivirus software, and rule-based Intrusion Detection Systems (IDS), primarily rely on predefined signatures or heuristic rules to identify malicious activities. Network Threat Hunting is a proactive approach to cybersecurity where the focus is on actively seeking out threats that go unnoticed by standard detection methods. Threat hunters utilize data analytics, behavioral patterns, and threat intelligence to detect indicators in a network setting.
The importance of intelligent network threat detection is further amplified in modern distributed and dynamic computing environments. The proliferation of Internet of Things (IoT) devices, cloud computing, mobile workforces, and remote collaboration platforms has blurred traditional network perimeters, making endpoint-centric and perimeter-based defense strategies insufficient. Attackers exploit this expanded attack surface using advanced tactics, techniques, and procedures (TTPs), necessitating adaptive and context-aware defense mechanisms.
This paper proposes a Network Threat Hunting and Detection System using machine learning techniques, aiming to address the limitations of traditional IDS and enhance cybersecurity resilience. The KDD Cup 1999 dataset, a well-established standard in intrusion detection research, is utilized for training and assessing different supervised machine learning models to identify the most efficient algorithm for running the model effectively.

II. LITERATURE SURVEY

Intrusion Detection Systems (IDS) are essential components of network security architectures, designed to detect and respond to unauthorized access or malicious activities. Traditionally, IDS have relied on signature-based detection mechanisms, which identify attacks based on predefined patterns or signatures. This literature survey reviews the evolution of IDS, the role of ML in network threat detection, and key research contributions in this domain, with a focus on methodologies relevant to the proposed system.

This application was developed with inspiration from the following papers:
In their study "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms," Dhanabal and Shantharajah (2015) investigate the effectiveness of several machine learning classification algorithms. The dataset is specifically designed to address issues of data imbalance and redundancy. The authors compare multiple algorithms, including Naive Bayes, J48 (a variant of Decision Trees), Random Forest, and Support Vector Machines (SVM). The paper highlights the importance of utilizing updated and balanced datasets like NSL-KDD for accurate evaluations and stresses the significant impact of feature selection on classifier performance. The authors also suggest that future work should explore hybrid models to further enhance detection accuracy.[3]
In their survey "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," Buczak and Guven (2016) offer an extensive review of various data mining. They emphasize that ensemble methods, deep learning, and hybrid systems show considerable potential in overcoming the limitations of traditional machine learning approaches. The paper calls for the development of scalable and adaptive systems capable of functioning effectively in real-time environments, presenting a comprehensive comparison of algorithms and suggesting future research directions in the field of intelligent IDS.[4]

In their analysis, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection," Revathi and Malathi (2013) explore the performance of various machine learning techniques on the NSL-KDD dataset. The paper also delves into the role of individual features in model predictions, emphasizing the optimization of feature selection for future IDS systems. The study underscores the effectiveness of ensemble methods and stresses the importance of dataset quality in the development of robust and effective intrusion detection models.[5]

In their study "Intrusion Detection System using Support Vector Machine with Modified K-Means Clustering," Reddy and Reddy (2016) propose a hybrid intrusion detection system that combines Support Vector Machines (SVM) with a modified K-Means clustering algorithm. The methodology involves clustering similar data instances using a modified K-Means algorithm before training an SVM on the resulting clusters. The paper highlights the advantages of combining unsupervised learning (clustering) with supervised techniques to leverage both the structure of the data and the label information. The authors suggest that such hybrid methods can be refined further to adapt to dynamic network environments, thereby improving intrusion detection systems.[6]

III. METHODOLOGY

The proposed system utilizes machine learning algorithms to detect network intrusions on the KDD Cup 1999 dataset. The key algorithms used are Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Art (ANN), Naive Bayes Classifier, and Support Vector Machine (SVM). Feature selection techniques are employed to enhance the model's accuracy and reduce false positives.

The System model has five phases:

- Data Preprocessing: Clean and normalize the KDD dataset, handle missing values, and select relevant features for training.
- Model Training: Train the classifiers (Logistic Regression, Decision Tree, Random Forest, ANN, Naive Bayes, SVM) on the preprocessed data.
- Model Evaluation: Evaluate the models using metrics such as accuracy, precision, recall, and false positive rate.
- Optimization: Tune hyperparameters and perform cross-validation to improve model performance.
- Deployment: The deployment process involves implementing the trained model for real-time network intrusion detection.

ALGORITHM:

Step 1: Preprocess the KDD dataset (handle missing values, normalization, feature selection).
Step 2: Train the models using Logistic Regression, Decision Tree, Random Forest, KNN, Naive Bayes, and SVM.

Step 3: Evaluate testing set model's performance.
Step 4: Optimize hyper parameters to improve results.
Step 5: Plot and evaluate the model performance with graphs

This methodology ensures accurate and efficient intrusion detection using the KDD Cup 1999 dataset.

IV. SYSTEM MODEL

The system aims to detect network intrusions by using machine learning models on the KDD Cup 1999 dataset. The system is designed to distinguish between "good normal connections" and "bad malicious connections."

The proposed system has the following modules:

- Import the Dataset: The KDD Cup 1999 dataset, containing network connection records with 41 features, is used for training and assessment.
- Dataset Preprocessing: During dataset preprocessing, categorical variables are handled using Label Encoding and One-Hot Encoding.
- Model Training: Various machine learning algorithms are trained on the preprocessed dataset.
- Classification and Evaluation: The trained models are tested on the test set, and evaluation metrics like accuracy, precision, recall, F1-score, and confusion matrix are used to assess their performance.

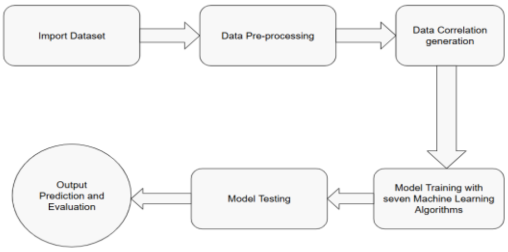
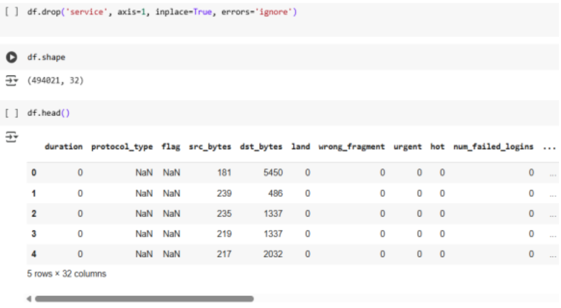

Import Dataset
Data Pre-processing
Data Correlation generation
Model Training with seven Machine Learning Algorithms
Model Testing
Output Prediction and Evaluation

Fig 1: Workflow



```
[ ] df.drop('service', axis=1, inplace=True, errors='ignore')

df.shape
(494021, 32)

[ ] df.head()

   duration protocol_type flag src_bytes dst_bytes land wrong_fragment urgent hot num_failed_logins ...
0         0           NaN  NaN       181      5450    0              0      0   0                 0 ...
1         0           NaN  NaN       239       486    0              0      0   0                 0 ...
2         0           NaN  NaN       235      1337    0              0      0   0                 0 ...
3         0           NaN  NaN       219      1337    0              0      0   0                 0 ...
4         0           NaN  NaN       217      2032    0              0      0   0                 0 ...
5 rows × 32 columns
```

Fig 2: Working Source code

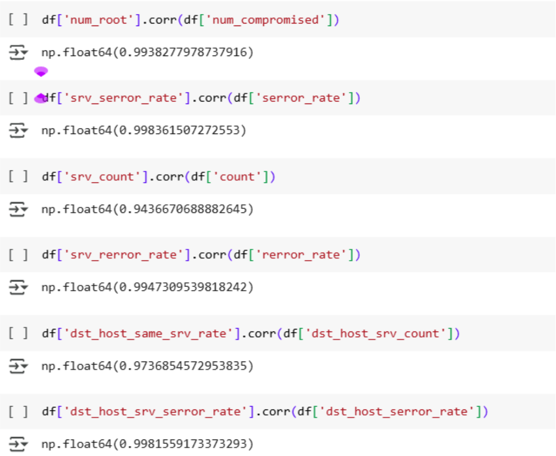
Machine Learning Algorithms

The system utilizes the following machine learning models to classify network intrusions:

- Logistic Regression: Used for binary classification tasks, providing simplicity and interpretability.
- Decision Tree Classifier: Splits the dataset based on feature values.
- Random Forest Classifier: An ensemble method that combines the outcomes of multiple decision trees to improve accuracy.
- Naive Bayes: Naive Bayes, a probabilistic classifier based on Bayes' Theorem, is employed.
- Support Vector Machine (SVM): Finds the optimal hyperplane separating classes in high-dimensional space.

The KDD Cup 1999 dataset, which contains network connection records, requires several preprocessing steps to ensure the data is suitable for training machine learning models. Initially, categorical features like protocal_type, service, and flag are encoded using Label Encoding and One-Hot Encoding to convert them into numerical formats.

Once the data is preprocessed, several machine learning models are trained on the training dataset. These models include Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Artificial Neural Network (ANN), Naive Bayes, and Support Vector Machine (SVM).



```
[ ] df['num_root'].corr(df['num_compromised'])
np.float64(0.9938277978737916)

[ ] df['srv_serror_rate'].corr(df['serror_rate'])
np.float64(0.998361507272553)

[ ] df['srv_count'].corr(df['count'])
np.float64(0.9436670688882645)

[ ] df['srv_rerror_rate'].corr(df['rerror_rate'])
np.float64(0.9947309539818242)

[ ] df['dst_host_same_srv_rate'].corr(df['dst_host_srv_count'])
np.float64(0.9736854572953835)

[ ] df['dst_host_srv_serror_rate'].corr(df['dst_host_serror_rate'])
np.float64(0.9981559173373293)
```

Fig 3: Data Preprocessing

The Gaussian Naive Bayes (gNB) classifier utilizes probability theory, achieving a high accuracy rate of 99.3%. The Decision Tree (DT) classifier maintaining a testing accuracy of 99.3%. The

Random Forest (RF) model improves accuracy by constructing multiple decision trees and combining their results, offering the highest performance with a testing accuracy of 99.9%. The Support Vector Machine (SVM) maximizes the margin between different classes, achieving 99.8% accuracy at a high cost, which is not ideal for real-time applications. The Logistic Regression (LR) classifier, a simpler linear model, maintains accuracy at 99.2% with minimal computational requirements. The Gradient Boosting Classifier (GBC) builds an ensemble of weak learners to improve accuracy iteratively, producing a testing accuracy of 99.9%, though at a higher computational cost. Lastly, the Artificial Neural Network (ANN) uses multiple layers to model complex relationships within the data, achieving 99.8% accuracy.

The evaluation of these models is based on key performance. For example, Random Forest and Gradient Boosting Classifier perform exceptionally well with near-perfect accuracy of 99.9%, while other models like Logistic Regression and Gaussian Naive Bayes still achieve impressive accuracies above 99%. Random Forest, despite its high accuracy, requires more time (6.3 seconds) for training, while simpler models like Gaussian Naive Bayes and Logistic Regression are significantly faster. The SVM has a much higher training time (599.6 seconds) and testing time (62.21 seconds), which makes it less practical for time-sensitive applications.

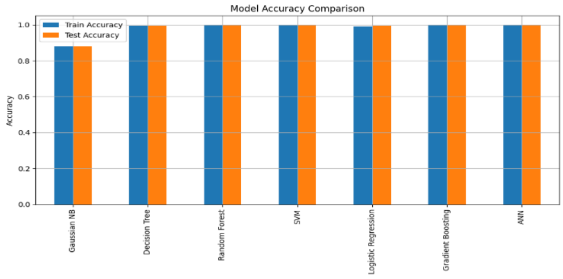


Fig 8: Training and Testing Accuracy

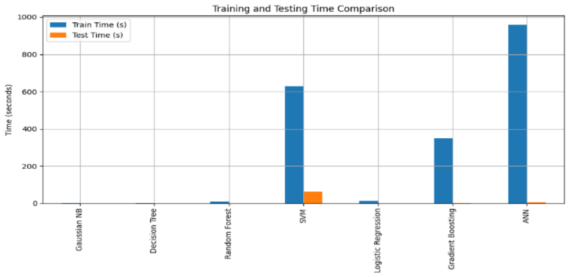


Fig 9: Training and Testing Time

V. FUTURE WORK

Future work will focus on enhancing the system with more advanced models like deep neural networks . The goal is to improve real-time intrusion detection capabilities and optimize the model for faster predictions.

VI. CONCLUSION

In conclusion, this intrusion detection system successfully identifies network threats with high accuracy across multiple machine learning models. The Random Forest model emerged as the most efficient, achieving the highest testing accuracy of 99.9%, while maintaining relatively reasonable training and testing times. The Support Vector Machine (SVM) model, though highly accurate (99.8%), showed a considerable computational delay. In practice, once a potential intrusion is detected, the system flags it and alerts the network administrator for further action. These results demonstrate that the system can not only detect a wide range of network intrusions with high precision but also balance accuracy with computational efficiency for real-time threat detection. The insights suggest that Random Forest offers the most reliable solution for intrusion detection, with its superior accuracy and moderate computational demands.

REFERENCES

[1] Debra Anderson, Thane Frivold, and Alfonso Valdes, "NIDES Next-generation Intrusion Detection Expert System (NIDES)", A Summary, Computer Science Laboratory, SRI-CSL-95-07, May 1995

[2] Dhanabal, L., & Shantharajah, S. P. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. International Journal of Advanced Research in Computer and Communication Engineering, 4(6), 446–452.

[3] Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176.

[4] Revathi, S., & Malathi, A. (2013). A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. International Journal of Engineering Research and Technology, 2(12), 1848–1853.

[5] Reddy, A. R., & Reddy, B. E. (2016). Intrusion Detection System using Support Vector Machine with Modified K-Means Clustering. International Journal of Computer Applications, 144(5).

[6] Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. Expert Systems with Applications, 36(10), 11994–12000.

[7] James P. Anderson, "Computer security threat monitoring and surveillance," Technical Report 98-17, James P. Anderson Co., Fort Washington, Pennsylvania, USA, April 1980.

[8] D. E. Denning, "An intrusion detection model," IEEE Transaction on Software Engineering, SE13(2), 1987, pp. 222-232.

[9] Alanoud Alsaleh, Wojdan Binsaeedan, The Influence of Salp Swarm Algorithm-Based Feature Selection on Network Anomaly Intrusion Detection, August 2021, IEEE Access PP (99):1-1, DOI:10.1109.

[10] Ajay Shah; Sophine Clachar; Manfred Minimair; Davis Cook, Building Multiclass Classification Baselines for Anomaly-based Network Intrusion Detection Systems, 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), DOI: 10.1109/DSAA49011.2020.00102, 6-9 October, 2020.

[11] Daniel Barbará, Julia Couto, SushilJajodia, Leonard Popyack and Ningning Wu, "ADAM: Detecting intrusion by data mining," IEEE Workshop on Information Assurance and Dr.V.Suganthi,*1 , P. K. Manoj Kumar 2 2018 E-J. 1 (2018) 24 Security, West Point, New York, June 5-6, pp. 11-16, 2001.

[12] Te-Shun Chou and Tsung-Nan Chou, "Hybrid Classified Systems for Intrusion Detection," Seventh Annual Communications Networks and Services Research Conference, pp. 286-291, 2009.

[13] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naïve Bayes vs. decision trees in intrusion detection systems," Proc. of 2004 ACM Symposium on Applied Computing, 2004, pp. 420-424.

[14] Nasrin Sultana, Naveen Chilamkurti, Naveen Chilamkurti, Wei PengWei, PengRabei Alhadad, Survey on SDN based network intrusion detection system using machine learning approaches, Springer, DOI: 10.1007/s12083-017-0630-0.

[15] Suchet Sapre; Khondkar Islam; Pouyan Ahmadi, A Comprehensive Data Sampling Analysis Applied to the Classification of Rare IoT Network Intrusion Types, 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), DOI: 10.1109/CCNC49032.2021.9369617.