Evaluating ML Algorithms for Network Intrusion Detection: Insights from CICIDS2017 Dataset

K. RAMESH
Kakatiya Government College, Hanumakonda

Abstract*— The main aim of this study is to investigate the effectiveness of machine learning (ML) models in detecting network intrusions using the CICIDS2017 dataset. This dataset contains various instances of network traffic, encompassing different types of cyber-attacks. The study begins by consolidating and refining multiple datasets to create a unified dataset for analysis. Subsequently, an exploratory analysis reveals the distribution patterns of different attacks within the dataset. Data preparation involves optimizing the dataset for modeling by applying feature scaling and selection techniques. Several ML algorithms, such as Logistic Regression, Naive Bayes, and Decision Tree classifiers, are trained and rigorously evaluated using cross-validation methods. The evaluation metrics include accuracy measures and cross-validation mean scores, providing a comprehensive assessment of the models' performance. Moreover, the study employs the Random Forest classifier to identify and prioritize significant features aiding in intrusion detection. This research endeavors to contribute significantly to the field of cyber security by showcasing the potential of ML algorithms in detecting and categorizing diverse network intrusions. The findings highlight the feasibility of deploying robust ML-based intrusion detection systems, strengthening real-time network security applications and fortifying defenses against evolving cyber threats.

Indexed Terms- CICIDS2017, Intrusion detection, Logistic Regression, Naive Bayes, and Decision Tree classifiers.

I. INTRODUCTION

In the landscape of cyber security, safeguarding networks against unauthorized access and malicious activities remains a crucial concern. As technology advances, so does the sophistication of cyber threats, necessitating effective intrusion detection measures to counter these risks[4]. The detection and prevention of network intrusions remain critical in ensuring the security and integrity of modern information systems. With the proliferation of cyber threats and sophisticated attack techniques, the need for effective intrusion detection systems has become paramount[11]. Machine learning (ML) algorithms have emerged as promising tools in cyber security, offering the potential to detect and classify diverse network intrusions accurately. In recent years, the landscape of cyber threats has evolved significantly, posing immense challenges to traditional intrusion detection mechanisms. Networks face a myriad of threats, ranging from Distributed Denial of Service (DDoS) attacks, port scanning, to infiltration and web-based assaults. These attacks exploit vulnerabilities in network protocols, applications, and systems, leading to data breaches, service disruption, and financial losses[14].

Traditionally, intrusion detection relied on rule-based systems, signatures, and heuristics[12]. However, the evolving nature of threats rendered these methods insufficient. This led to a shift towards utilizing machine learning (ML) algorithms.

ML algorithms gained prominence due to their ability to process vast data, detect patterns, and adapt to new threats. Unlike rule-based systems, ML algorithms can learn from data and autonomously detect anomalies or suspicious activities, offering a more dynamic defense against emerging threats.ML models employ diverse techniques like supervised, unsupervised, and semi-supervised learning. Supervised algorithms classify network traffic based on labeled data, while unsupervised methods detect deviations in unlabeled data. Semi-supervised learning combines aspects of both to enhance accuracy.ML algorithms also enable predictive models that evolve with new information, improving detection accuracy over time. They contribute not only to detection but also to threat intelligence, incident response, and proactive security measures. In essence, ML algorithms play a crucial role in enhancing intrusion detection by boosting speed, accuracy, and adaptability. As cyber threats evolve, the integration of ML in cyber security

becomes increasingly pivotal in fortifying network defenses against malicious intrusions.

The CICIDS2017 dataset stands as a comprehensive resource in the realm of cyber security, comprising diverse instances of network traffic and simulated cyber-attacks. This dataset amalgamates various scenarios, providing a rich repository of information for analyzing and understanding different intrusion patterns. Leveraging this dataset, this study aims to harness the power of ML algorithms to build robust intrusion detection systems capable of accurately identifying and categorizing these attacks.

The objectives of the paper as follows:

1. Conduct a comparative analysis to identify strengths and weaknesses among classifiers for accurately detecting network intrusions.
2. Analyze feature importance extracted from network traffic data to understand key factors contributing to intrusion classification.
3. Utilize Cross Validation Mean Score and Model Accuracy metrics to assess the effectiveness of classifiers in intrusion detection.
4. Evaluate and compare the performance of Logistic Regression, Naive Bayes, and Decision Trees for intrusion detection using the CICIDS2017 dataset.
5. Extract insights to guide the enhancement of intrusion detection systems based on comparative analysis findings.
6. Contribute insights to advance intrusion detection methodologies for improved cyber security measures.

II. RELATED WORKS

Addressing the escalating network threats, recent research has concentrated on developing effective intrusion detection systems (IDS). Krsteski et al. [1]focused on constructing a PyCaret-based machine learning IDS over the CICIDS 2017 dataset. Their comprehensive data analysis led to the removal of redundant data. Their classification approach highlighted Random Forest as the most proficient classifier, boasting a remarkable 99.6% accuracy and a notable F1-Macro score of 0.917. The application of clustering and anomaly detection through PyCaret underscored the challenges, with clustering achieving a silhouette score of 0.90 and accuracy ranging between 0.54% and 0.75%, pinpointing areas for enhancement.

In their review, Ravipati and Abualkibash [2]explored diverse machine learning algorithms for IDS, emphasizing KNN's high false rate and AdaBoost's superior detection rate and algorithmic speed. They aim to delve into unsupervised algorithms to identify potential superior alternatives.

Atefi, Hashim, and Kassim[3] addressed the inadequacies of older datasets like Kddcup'99 in accurately detecting intrusions. Their anomaly analysis focused on employing K-Nearest Neighbors (KNN) and Deep Neural Network (DNN), with DNN outperforming KNN significantly, scoring 0.9293% compared to 0.8824%.

Jaradat, Barhoush, and Easa [5]highlighted the importance of network security and proposed a machine learning-based approach for intrusion detection using the CICIDS2017 dataset. Leveraging KNIME analytics platform and classifiers like SVM, RProp, and decision tree, they aimed to build a robust IDS system.

Leon, Markovic, and Punnekkat [6]examined supervised and unsupervised algorithms for intrusion detection across different benchmark datasets (KDD99, NSL-KDD, UNSW-NB15, and CIC-IDS-2017). Their findings favored Random Forest as the most suitable algorithm considering accuracy and execution time.

Hidayat, Ali, and Arshad [7] focused on ML-based intrusion detection using the TON_IoT dataset. Their work showcased high accuracy of decision tree and AdaBoost algorithms, reaching approximately 99.6%. Deep Learning (DL) techniques like MLP and LSTM also exhibited high accuracy of nearly 99.2% and 99% respectively. Panwar, Raiwani, and Panwar emphasized the complexity of maintaining network security due to increased internet usage. Their study incorporated eight supervised classification techniques on the CICIDS-2017 dataset, revealing the importance of various intrusion detection strategies.

III. MATERIALS AND METHODOLOGY

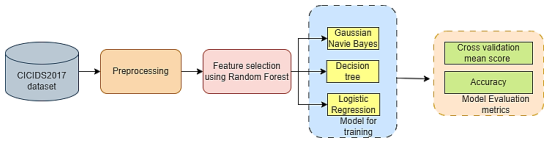

Fig 1. Block Diagram of Proposed Methodology.

a. Dataset

The CICIDS2017 dataset, known as the Canadian Institute for Cyber security Intrusion Detection System 2017 dataset, is a valuable resource widely used in cyber security research and the development of intrusion detection systems. It comprises multiple CSV files, each depicting various network traffic scenarios, including both normal and simulated cyber-attack behaviors[8].

In this dataset:

- Rows represent individual network traffic instances, while columns contain attributes like source/destination IP addresses, port numbers, protocol types, and packet sizes.
- Diverse cyber-attack types, such as DDoS attacks, port scans, web attacks, and infiltration attempts, are labeled within the dataset.

Researchers use this dataset to Evaluate intrusion detection systems and machine learning models, Analyze attack patterns and network behavior, Develop and test algorithms for anomaly detection and network security. The dataset offers a mix of benign and attack traffic, aiding in understanding various cyber threats. Researchers typically preprocess the data by handling missing values, scaling features, and splitting it for model training.CICIDS2017 is available for research purposes and can be accessed from repositories or cyber security research platforms, often accompanied by documentation describing its contents and format [9].This dataset significantly contributes to cyber security research by providing real-world network traffic data for testing and improving intrusion detection systems and security algorithms.

3.2 Methodology

This research aims to develop a robust system for identifying and categorizing network intrusions using machine learning methods. The methodology involves several key phases: data collection, preprocessing, exploratory analysis, feature selection, model development, and evaluation.

3.2.1. Data Collection

The dataset used in this study comprises multiple .csv files that capture various types of network traffic, including DDoS attacks, port scans, and web attacks. These files, obtained from [specify the source or repository], represent network activity during different time frames and days of the week. The dataset is combined to create a comprehensive dataset covering diverse network intrusions.

3.2.2. Data Preprocessing

After merging the dataset, an initial check is conducted to ensure data quality. This phase involves handling missing values, removing duplicates, and addressing infinite values. The goal is to ensure the dataset's integrity by cleaning it and preparing it for analysis.

3.2.3. Exploratory Analysis

Exploratory Data Analysis (EDA) is performed to understand the dataset's characteristics and distributions. Descriptive statistics are calculated to explore numerical attributes' central tendencies and variability. Additionally, an analysis of packet attacks across different types of intrusions helps understand the dataset's class distribution and potential imbalances.

3.2.4. Feature Selection

Identifying influential attributes is crucial for effective classification[13]. A Random Forest Classifier is utilized to determine feature importance. This process ranks attributes based on their contribution to classifying network intrusions, aiding in selecting the most informative attributes.

3.3.5. Model Development

Several machine learning models, including Logistic Regression, Gaussian Naive Bayes, and Decision Tree Classifier, are chosen and trained on the preprocessed dataset. The dataset is divided into training and testing sets for model training and evaluation. Model parameters are optimized for improved performance.

Logistic Regression (LR) for Multiclass Classification: Logistic Regression is a linear classification method used for both binary and multiclass problems. When handling multiclass tasks, LR employs the "one-vs-rest" (OvR) or "one-vs-all" approach. In OvR, LR trains separate models for each class to discern that specific class from the rest. The model calculates probabilities for each class and

combines them to make the final prediction.In the multiclass setting using OvR, LR predicts the probability of an input belonging to each class using the softmax function. This function computes the probability distribution over multiple classes by applying the exponential function to the weighted sum of input features, normalized by the sum of exponentiated weighted sums across all classes.

$$P(y = i \mid x) = \frac{e\ wi\ \cdot x}{\sum_{j=1}^{K} \quad e\ wj \cdot x} \quad (1)$$

- $P(y=i|x)$ represents the probability of the input sample x belonging to class i.
- w_i denotes the weights associated with class i.
- K is the total number of classes.

Gaussian Naive Bayes (GNB) for Multiclass Classification: Gaussian Naive Bayes is a probabilistic classifier based on Bayes' theorem. In multiclass problems, GNB assumes feature independence within each class. It calculates the probability of an input belonging to each class using Gaussian distributions. For continuous features, GNB estimates the likelihood of features given a class by assuming a Gaussian (normal) distribution, using mean and variance parameters. The class probability is computed through Bayes' theorem by combining individual feature probabilities.

For continuous features, GNB assumes a Gaussian distribution:

$$P(x\ i\ \mid y = c) = \frac{1}{\sqrt{2\pi\sigma\ c,i\ ^2}} e^{-\left(\frac{(xi - \mu\ c,i\ )2}{2\sigma\ 2\ c,i}\right)} \quad (2)$$

- $P(x_i \mid y=c)$ is the probability of feature xi given class c.
- $\mu_{c,i}$ and $2\sigma_{c,i}{}^2$ are the mean and variance of feature xi in class c, respectively.

The class probability is computed using Bayes' theorem by combining individual feature probabilities. Decision Trees (DT) for Multiclass Classification: Decision Trees partition the feature space into segments and assign class labels based on these segments. In multiclass classification, DT constructs a tree-like structure, employing algorithms like CART or C4.5. The tree structure recursively splits the dataset based on features to minimize impurity or maximize information gain. For instance, using Gini impurity, DT measures node homogeneity by calculating the probability distribution of classes at each node and aiming to reduce impurity in subsequent splits.

In multiclass DT, the Gini impurity or entropy is used to measure the homogeneity of the nodes. For Gini impurity:

$$Gini(t) = 1 - \sum_{i=1}^{K} \quad p(i \mid t)2 \quad (3)$$

- $Gini(t)$ represents the Gini impurity at node t.
- $p(i|t)$ is the probability of class i at node t.

3.3.6. Model Evaluation

The performance of each model is evaluated using cross-validation techniques to ensure reliability and prevent overfitting. K-fold cross-validation with 10 folds is employed to assess metrics such as accuracy. These metrics provide insights into how accurately the models classify network intrusions.

ALGORITHM 1: ALGORITHM FOR MODEL EVALUATION

Input: X_train, Y_train
Output: Accuracy
Initialization:
1: Define a list of models with their respective names.[Naive Bayes Classifier, Decision Tree Classifier, Logistic Regression Classifier]
2: Define an empty list to store model scores.
Loop Process:
3: for each *model* in the *list of models* do
4: Perform cross-validation with 10 folds:
5: Calculate the scores using *cross_val_score* method.
6: Compute accuracy using *metrics.accuracy_score.*
7: Print the model evaluation results:
8: Cross Validation Mean Score.
9: Model Accuracy.
10: end for
Return: Performance metrics (such as accuracy) for different models after evaluation.

IV. RESULTS

The investigation into the CICIDS2017 dataset revealed a diverse distribution of network traffic instances across different attack categories.

Table 1. Class distribution chart

Label	Value count
BENIGN	628203
DoS Hulk	51981
DDOS	38187
PortScan	27505
DoS GoldenEye	3123
FTP-Patator	1763
DoS slowloris	1633
DoS Slowhttptest	1606
SSH-Patator	964
Bot	600
Web Attack -Brute Force	446
Web Attack- XSS	214
Infiltration	8
Web Attack -Sql Injection	6
Heartbleed	1

As shown in table 1,the dataset primarily consisted of benign network traffic, accounting for 628,203 instances. Notably, instances of 'DoS Hulk' and 'DDOS' were observed, with 51,981 and 38,187 occurrences, respectively. 'PortScan' instances were identified 27,505 times, while 'DoS GoldenEye' was detected 3,123 times. Additionally, the dataset encompassed several other attack types, including 'FTP-Patator' (1,763 instances), 'DoS slowloris' (1,633 instances), 'DoS Slowhttptest' (1,606 instances), 'SSH-Patator' (964 instances), 'Bot' (600 instances), 'Web Attack - Brute Force' (446 instances), 'Web Attack - XSS' (214 instances), 'Infiltration' (8 instances), 'Web Attack - SQL Injection' (6 instances), and a minimal occurrence of 'Heartbleed' (1 instance).

Model	Cross Validation Mean Score:	Model Accuracy:
Naive Baye Classifier Model	82.97%	82.98%
Decision Tree Classifier Model	99.86%	99.99%
Logistic Regression Model	95.49%	95.49%

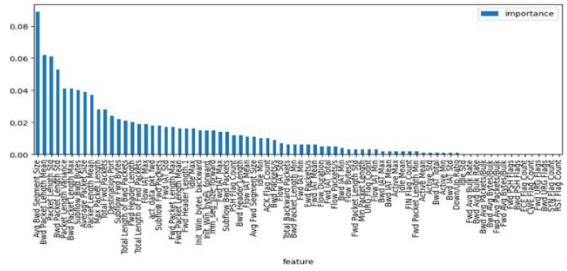

Fig 2. Feature Importance

Fig 2. Random Forest calculates the importance of each feature by evaluating how much the inclusion of a feature contributes to reducing prediction errors across all the trees in the forest. Higher importance scores signify a greater impact of the feature on the overall predictions.

Table 2. Model Evaluation

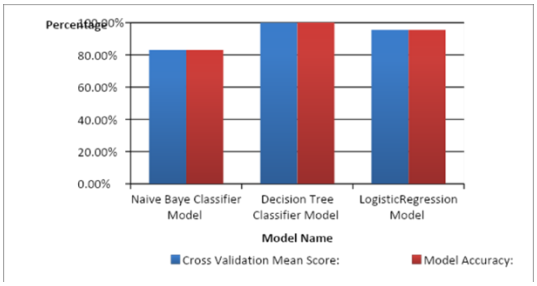

Fig 3. Comparison graph (model vs. Accuracy and Cross Validation Score)

As shown in Table 2 and Fig 3. the Decision Tree Classifier exhibits exceptional performance with nearly perfect accuracy (99.99%) and a very high cross-validation mean score (99.86%). This indicates its strong ability to capture patterns within the dataset. The Logistic Regression Model also performs well, demonstrating a cross-validation mean score of 95.49% and an accuracy of 95.49%. This suggests good generalization and accuracy on the given dataset. The Naive Baye Classifier Model shows slightly lower performance compared to the other models, with a cross-validation mean score and accuracy of 82.97% and 82.98%, respectively.

CONCLUSION

The investigation into intrusion detection methods using diverse classifiers on the CICIDS2017 dataset has yielded notable findings. Employing a systematic approach involving data preprocessing, feature

selection, model training, and assessment, this study aimed to compare classifiers for effective intrusion identification within network traffic. Various classifiers such as Logistic Regression, Gaussian Naive Bayes, and Decision Trees underwent evaluation using key metrics like Cross Validation Mean Score and Model Accuracy. The comparative analysis revealed distinctive differences among the models. The Decision Tree classifier exhibited exceptional performance, demonstrating a Cross Validation Mean Score approaching 99.86 and achieving an impressive Model Accuracy close to 99.99. Conversely, the Logistic Regression model showed respectable performance, with a Cross Validation Mean Score of approximately 95.49, aligning closely with its Model Accuracy of 95.49. Meanwhile, the Gaussian Naive Bayes classifier displayed moderate performance, yielding a Cross Validation Mean Score around 82.98, consistent with its Model Accuracy of 82.98.These findings emphasize the significance of classifier selection in accurately detecting and categorizing network intrusions within the scope of the CICIDS2017 dataset. This study provides valuable insights, showcasing the effectiveness of different approaches in intrusion detection without specific technology references, offering potential directions for the development of resilient cyber security systems to address contemporary network security challenges.

REFERENCES

[1] Krsteski, S., Tashkovska, M., Sazdov, B., Radojichikj, L., Cholakoska, A., Efnusheva, D. (2023). Intrusion Detection with Supervised and Unsupervised Learning Using PyCaret Over CICIDS 2017 Dataset. In: Silhavy, R., Silhavy, P. (eds) Artificial Intelligence Application in Networks and Systems. CSOC 2023. Lecture Notes in Networks and Systems, vol 724. Springer, Cham. https://doi.org/10.1007/978-3-031-35314-7_12

[2] Ravipati, Rama Devi and Abualkibash, Munther, (June 2019), Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper ,International Journal of Computer Science & Information Technology (IJCSIT) Vol 11, No 3, June 2019, Available at SSRN: https://ssrn.com/abstract=3428211 or http://dx.doi.org/10.2139/ssrn.3428211

[3] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa and C. F. M. Foozy, 2021,"Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," in IEEE Access, vol. 9, pp. 22351-22370 , doi: 10.1109/ACCESS.2021.3056614.

[4] K. Atefi, H. Hashim and M. Kassim, 2019, "Anomaly Analysis for the Classification Purpose of Intrusion Detection System with K-Nearest Neighbors and Deep Neural Network," 2019 IEEE 7th Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia, , pp. 269-274, doi: 10.1109/ICSPC47137.2019.9068081

[5] Jaradat, A. S., Barhoush, M. M., & Easa, R. B. (2022). Network intrusion detection system: machine learning approach. Indonesian Journal of Electrical Engineering and Computer Science, 25(2), 1151-1158.

[6] M. Leon, T. Markovic and S. Punnekkat, 2022 "Comparative Evaluation of Machine Learning Algorithms for Network Intrusion Detection and Attack Classification," 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 01-08, doi: 10.1109/IJCNN55064.2022.9892293.

[7] Hidayat, I., Ali, M. Z., & Arshad, A. (2023). Machine Learning-Based Intrusion Detection System: An Experimental Comparison. Journal of Computational and Cognitive Engineering, 2(2), 88-97.

[8] S. S. Panwar, Y. P. Raiwani and L. S. Panwar, "An Intrusion Detection Model for CICIDS-2017 Dataset Using Machine Learning Algorithms," 2022 International Conference on Advances in Computing, Communication and Materials (ICACCM), Dehradun, India, 2022, pp. 1-10, doi: 10.1109/ICACCM56405.2022.10009400.

[9] T. Elmasri, N. Samir, M. Mashaly and Y. Atef, 2020 "Evaluation of CICIDS2017 with Qualitative Comparison of Machine Learning Algorithm," IEEE Cloud Summit, Harrisburg,

PA, USA, 2020, pp. 46-51, doi: 10.1109/IEEECloudSummit48914.2020.00013.

[10] Abbas, A., Khan, M.A., Latif, S. et al. (2022). A New Ensemble-Based Intrusion Detection System for Internet of Things. Arab J Sci Eng 47, 1805–1819 .https://doi.org/10.1007/s13369-021-06086-5

[11] Azidine Guezzaz, Said Benkirane, Mourade Azrour, Shahzada Khurram, 2021. "A Reliable Network Intrusion Detection Approach Using Decision Tree with Enhanced Data Quality", Security and Communication Networks, vol. 2021, Article ID 1230593, 8 pages. https://doi.org/10.1155/2021/1230593

[12] R. Lalduhsaka & Nilutpol Bora & Ajoy Kumar Khan, 2022. "Anomaly-Based Intrusion Detection Using Machine Learning: An Ensemble Approach," International Journal of Information Security and Privacy (IJISP), IGI Global, vol. 16(1), pages 1-15, January.

[13] Thakkar, A., & Lohiya, R. (2022). A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artificial Intelligence Review, 55(1), 453-563.

[14] Al-Imran, M., Ripon, S.H. (2021). Network Intrusion Detection: An Analytical Assessment Using Deep Learning and State-of-the-Art Machine Learning Models. Int J Comput Intell Syst 14, 200 https://doi.org/10.1007/s44196-021-00047-4.