E-XAI: Evaluating Black-Box Explainable AI Frameworks for Network Intrusion Detection

Ankit Kumar[1], Rishav Sikdar[2], Trisha U[3], Raghavendra Singh Negi[4], Yogesh K M[5], and Vedavati Bhandari[6]

[1,2,3,4,5]Department of Computer Science and Engineering M S Ramaiah University of Applied Sciences, Bengaluru, India
[6]Department of Computer Science AJ Institute of Engineering and Technology, Mangaluru, India
ankit19286@gmail.com, rishisikdar02@gmail.com, trishareddy126@gmail.com, raghavendrasn2003@gmail.com, yogeshkm.mrj@gmail.com, vedavatibh@gmail.com

Abstract

In many real-world applications, such as sports analytics, smart settings, and healthcare monitoring, Human Activity Recognition (HAR) is indispensable. The objective of this study is to leverage time-domain information taken from the HARTH dataset to enhance HAR performance.In order to improve the precision and effectiveness of activity recognition, the study looks into various machine learning classifiers.The objective of this work is to create a more effective method for differentiating between different physical activities by analyzing important time-domain properties as mean, variance, skewness, kurtosis, and signal magnitude area (SMA). To determine the best classifier for HAR tasks, a number of machine learning models are assessed, including Support Vector Machines (SVM), Random Forest, and Decision Trees. Based to the results, well-designed time-domain features greatly improve classification performance while lowering computing cost and preserving high accuracy.

Keywords: XAI Intrusion detection SHAP LIME Network security Black-box models NSL-KDD CICIDS 2017 SIMARGL 2021

1 INTRODUCTION

The increasing sophistication and frequency of network intrusions have driven the evolution of intrusion detection systems (IDS) powered by artificial intelligence (AI). AI has been widely applied in IDS to automate the detection of

malicious activities, leveraging techniques such as neural networks, support vector machines, decision trees, naive Bayes, and random forests. However, many of these methods are inherently black-box models, where the decision-making process remains opaque to security analysts. This lack of interpretability hinders their practical deployment, as understanding the reasoning behind an AI model's decision is crucial for ensuring reliability and trust in security-critical applications.Neupane et al. [9] highlighted the lack of explainability in AI-driven IDS as a critical barrier for adoption in operational security environments like CSoCs. While high classification accuracy remains the focus of many AI-based IDS studies, there has been limited exploration into providing clear and understandable explanations of AI model behavior and reasoning, which are essential for making informed decisions in real-time network monitoring and incident response.

The growing need for transparency has fueled the emergence of explainable AI (XAI), a field dedicated to developing methods that can make black-box AI models more interpretable. XAI aims to provide security analysts with insights into how AI systems arrive at their decisions, thereby enhancing trust, usability, and accountability . Recent studies have highlighted the potential of XAI in improving IDS by offering both global and local explanations of model predictions, as shown by Mane and Rao [12].

Several works have already explored XAI in the context of IDS. For example, some research has designed human-in-the-loop approaches to integrate explainability into IDS, ensuring that security personnel can interact with the system to validate and refine its decisions. In these approaches, simple models, like decision trees, have been used to provide transparent decision rules for easier interpretation, but these methods often fall short when applied to more complex black-box models [8]. Furthermore, the integration of explainable methods such as LIME with more sophisticated models has been limited, with some studies focusing on a single model type and dataset, such as support vector machines and the NSL-KDD dataset similar to the work seen in Gaspar et al.[10]. Despite these efforts, the application of XAI to a broader range of IDS models and datasets remains an area of ongoing research, with the need for more robust and generalizable explainability methods still unmet based on paper by Sauka et al. [11] and Apruzzese et al. [13]. This highlights the importance of advancing XAI techniques for enhancing IDS effectiveness, accuracy, and trust in security environments.

2 LITERATURE SURVEY

The literature increasingly highlights the role of explainable AI (XAI) in bridging this gap. Below, we summarize key contributions in this domain.

2.1 S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok, and M. Guizani

[1] present a comprehensive survey of intrusion detection systems (IDS) in the Internet of Things (IoT) domain, highlighting current methods and proposing future directions. The authors classify existing IDS techniques into four main categories: data-driven, trust-based, mathematical modeling, and blockchain-based systems. They emphasize that most research has been focused on the IoT device layer while largely neglecting the fog and cloud layers.

2.2 A. K. Balyan, S. Ahuja, U. K. Lilhore, K. Sharma, P. Manoharan, A. D. Algarni, H. Elmannai, and K. Raahemifar

[2] proposed a hybrid intrusion detection model that combines the strengths of Enhanced Genetic Algorithm (EGA) and Particle Swarm Optimization (PSO) for optimal feature selection, along with an improved Random Forest classifier to enhance detection accuracy. Their approach, published in Sensors, demonstrates significant improvements in identifying attacks while reducing false positives in IoT environments.

2.3 Z. A. El Houda, B. Brik and S.M. Senouci

[3] introduced a novel IoT-based intrusion detection framework that integrates deep learning with explainable AI (XAI) techniques. The model ensures high detection performance while also providing transparency in decision-making, addressing the black-box nature of traditional deep learning models. Their work focuses on enhancing trust and interpretability in security-critical IoT systems.

2.4 S. Neupane, J. Ables, W. Anderson, S. Mittal, S. Rahimi, I. Banicescuand M. Seale (2022)

[4] conducted a survey on Explainable Intrusion Detection Systems (X-IDS), reviewing the state-of-the-art methods, tools, and challenges in applying explainable AI to cybersecurity. The authors identify key limitations in current IDS models, such as lack of user trust and interpretability, and highlight research opportunities for integrating XAI techniques to bridge the gap between performance and transparency.

2.5 Y. Chen, Q. Lin, W. Wei, J. Ji, K.-C. Wong and C.A.C. Coell (2024)

[5] proposed an intrusion detection approach using a Multi-objective Evolutionary Convolutional Neural Network (MECNN) tailored for fog computing environments in the IoT. The model optimizes both accuracy and computational efficiency, making it suitable for distributed systems. Their work, published

in Knowledge-Based Systems, contributes to the development of lightweight, high-performance IDS for edge-based infrastructures.

2.6 Arreche, O., Guntur, T., & Abdallah, M. (2024). XAI-IDS

[6] introduced XAI-IDS, an encapsulated artificial intelligence framework designed to enhance network intrusion detection using explainable AI principles. The framework aims to improve model transparency, reduce complexity, and support better decision-making by security analysts. The study, published in Applied Sciences, emphasizes modularity and explainability in next-generation IDS design.

2.7 Tritscher, J., Wolf, M., Hotho, A., & Schlör, D. (2023, July)

[7] explored the evaluation of feature relevance through explainable AI techniques in the context of network intrusion detection. Presented at the World Conference on Explainable Artificial Intelligence (2023), their research investigates how different features contribute to model predictions, offering insights into improving both the accuracy and interpretability of IDS.

3 METHOD AND METHODOLOGY

The proposed system enhances IDS explainability and performance by evaluating black-box AI models within a comprehensive framework. It uses feature selection, trains multiple ML models, and selects the best one based on validation metrics like accuracy, precision, recall, and F1-score. Taking inspiration from Barnard et al. [8] who proposed a two-stage pipeline using XGBoost with SHAP explanations and an autoencoder, achieving robust and explainable intrusion detection on NSL-KDD.

i) Dataset Collection:

The dataset collection consists of CIC-IDS 2017, NSL KDD, and SIMARGL 2021, which are used for evaluating various intrusion detection models. These datasets contain network traffic data, including various attributes like packet counts, protocol types, and flow characteristics, helping to train and assess machine learning algorithms for network security tasks.

Fig.2 Dataset Collection Table - CIC-IDS 2017

CIC-IDS 2017: The CIC-IDS 2017 dataset contains more than 2 million records with 78 features such as flow duration, packet statistics, protocol flags (e.g., SYN, ACK) and inter arrival times. It includes labeled benign and attack traffic, offering a diverse environment for training and evaluating intrusion detection systems. Figure 1 shows a segment from the CIC-IDS 2017 dataset.

Fig.3 Dataset Collection Table - NSL KDD

	Protocol	Flow Duration	Total Fwd Packets	Total Backward Packets	Fwd Packets Length Total	Bwd Packets Length Total	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	...	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	6	4	2	0	12	0	6	6	6.00000	0.000000	...	20	0.0	0.0	0	0	0.0	0.0	0	0	Benign
1	6	1	2	0	12	0	6	6	6.00000	0.000000	...	20	0.0	0.0	0	0	0.0	0.0	0	0	Benign
2	6	3	2	0	12	0	6	6	6.00000	0.000000	...	20	0.0	0.0	0	0	0.0	0.0	0	0	Benign
3	6	1	2	0	12	0	6	6	6.00000	0.000000	...	20	0.0	0.0	0	0	0.0	0.0	0	0	Benign
4	6	609	7	4	484	414	233	0	69.14286	111.967896	...	20	0.0	0.0	0	0	0.0	0.0	0	0	Benign

5 rows × 78 columns

Figure 1: Fig.2 Dataset Collection Table - CIC-IDS 2017

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serr
0	0	udp	other	SF	146	0	0	0	0	0	...	0.00	0.60	0.88	0.00	
1	0	tcp	private	S0	0	0	0	0	0	0	...	0.10	0.05	0.00	0.00	
2	0	tcp	http	SF	232	8153	0	0	0	0	...	1.00	0.00	0.03	0.04	
3	0	tcp	http	SF	199	420	0	0	0	0	...	1.00	0.00	0.00	0.00	
4	0	tcp	private	REJ	0	0	0	0	0	0	...	0.07	0.07	0.00	0.00	

5 rows × 43 columns

Figure 2: Fig.3 Dataset Collection Table - NSL KDD

NSL-KDD: The NSL-KDD data-set includes 125,972 entries and 43 features encompassing connection duration, protocol type, traffic volume, and service flags. Designed to improve the KDD Cup 99 dataset, it removes redundancy and provides a balanced mix of normal and attack records, making it ideal for IDS benchmarking. Figure 2 shows a segment from the NSL-KDD dataset.

SIMARGL 2021: SIMARGL 2021 offers 42 features covering flow direction, byte rates, TCP flags, and protocol information. It includes detailed bidirectional flow data and is labeled for intrusion detection tasks, supporting ML-based network analysis and anomaly detection.Figure 3 shows a segment from the SIMARGL 2021 dataset.

ii) Pre-Processing Pre-processing improves data quality and model performance by removing noise, handling missing values, encoding categories, normalizing features, and selecting relevant attributes.

Data Cleaning: Duplicates and irrelevant fields are removed; normalization ensures equal feature contribution.

Visualization: Charts (e.g., histograms, scatter plots) reveal distributions, outliers, and feature relationships.

Label Encoding: Converts categorical data to numeric form; essential for models like SVM and decision trees.

Feature Selection: Techniques like correlation analysis and recursive elimi-

	BIFLOW_DIRECTION	DIRECTION	DST_TO_SRC_SECOND_BYTES	FIREWALL_EVENT	FIRST_SWITCHED	FLOW_ACTIVE_TIMEOUT	FLOW_DURATION_MICROSECONDS	FLOW_DURATION_MILLISECONDS	FLOW_END_MILLIS
0	1	0	40	0	1616660040	120	339	0	16166
1	1	0	.	0	1616660040	120	0	0	16166
2	1	0	104	0	1616660040	120	44725	44	16166
3	1	0	.	0	1616660040	120	0	0	16166
4	1	0	40	0	1616660040	120	1114	1	16166

5 rows × 50 columns

Figure 3: Fig.4 Dataset Collection Table - SIMARGL 2021

nation reduce overfitting and complexity.

iii) Training & Validation An 80:20 train-validation split is used. The model learns from the training data while validation aids in hyperparameter tuning and prevents overfitting, ensuring generalization to unseen data.

iv) Algorithms DNN: Deep networks capture complex intrusion patterns in large datasets.

MLP: Learns non-linear relationships to classify network behavior .

Random Forest: Ensemble of trees offering robust, interpretable intrusion classification.

LightGBM: Fast gradient boosting, efficient for high-volume traffic data .

SVM: Finds optimal boundaries in high-dimensional space for accurate detection .

AdaBoost: Boosts weak learners to focus on difficult cases, improving precision .

KNN: Classifies based on proximity to labeled patterns, ideal for anomaly detection.

Ensemble (BDT + RF): Combines boosting and bagging for higher accuracy and robustness.

4 RESULTS & DISCUSSION

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} (1)$$

Precision: Precision measures how many of the instances predicted as positive are actually correct. It focuses on the quality of positive predictions, showing the ratio of true positives to all predicted positives. Essentially, it indicates the proportion of accurate results within the set of predictions labeled as positive, given by:

$$Precision = \frac{True\ Positive}{True\ Positive\ + False\ Positive} (2)$$

Recall: Recall refers to a model's ability to correctly identify all relevant cases of a specific class. It is calculated as the number of correctly predicted positive instances divided by the total actual positives. In other words, recall highlights how thoroughly the model captures all true occurrences of the target class, making it a measure of completeness.

$$Recall = \frac{TP}{TP\ +\ FN} (3)$$

F1-Score: The F1-score is a metric in machine learning by which a model's performance is reflected by balancing both precision and recall. It represents the harmonic mean of these two measures. While accuracy simply indicates the proportion of correct predictions made by the model over the entire dataset, the F1-score provides a more nuanced view by considering both false positives and false negatives.

$$F1\ Score = 2*\frac{Recall\ X\ Precision}{Recall + Precision}*100(\ SEQ\ "equation"\ \backslash n\ \backslash *\ MERGEFORMAT\ 4)$$

Tables 1, 2 and 3 present a comparison of precision, accuracy, recall, and F1-score for each algorithm. The Voting Classifier consistently outperforms across all metrics, showcasing its superiority, while the tables offer insights into the performance of other algorithms.

Table 1: Performance Evaluation Metrics - CIC-IDS 2017

ML Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
DNN	0.166	0.028	0.166	0.047
AdaBoost	0.406	0.660	0.406	0.466
LightGBM	0.929	0.939	0.929	0.933
MLP	0.598	0.717	0.598	0.599
KNN	0.858	0.866	0.858	0.861
Random Forest	0.929	0.943	0.929	0.934
SVM	0.592	0.773	0.592	0.619
Voting Classifier (Boosted DT + Bagging RF)	**0.979**	**0.981**	**0.979**	**0.980**

Table 2: Performance Evaluation Metrics - NSL KDD

ML Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
DNN	0.253	0.064	0.253	0.102
AdaBoost	0.699	0.830	0.699	0.733
LightGBM	0.984	0.985	0.984	0.984
MLP	0.847	0.837	0.847	0.840
KNN	0.935	0.938	0.935	0.936
Random Forest	0.977	0.978	0.977	0.977
SVM	0.257	0.984	0.257	0.387
Voting Classifier (Boosted DT + Bagging RF)	**0.994**	**0.994**	**0.994**	**0.994**

Table 4 presents a comparative performance analysis of the *Voting Classifier* across three benchmark intrusion detection datasets: **CIC-IDS 2017**, **NSL-KDD**, and **SIMARGL 2021**. The metrics reported include **Accuracy**, **Precision**, **Recall**, and **F1 Score**.

On *CIC-IDS 2017*, the Voting Classifier achieves an accuracy of **97.9%**, with a high recall of **98.1%**, indicating strong detection capability. However, the

Table 3: Performance Evaluation Metrics - SIMARGL 2021

ML Model	Accuracy	Precision	Recall	F1 Score
DNN	0.487	0.237	0.487	0.319
AdaBoost	0.999	0.999	0.999	0.999
LightGBM	0.999	0.999	0.999	0.999
MLP	**1.000**	**1.000**	**1.000**	**1.000**
KNN	0.998	0.998	0.998	0.998
Random Forest	**1.000**	**1.000**	**1.000**	**1.000**
SVM	0.628	0.921	0.628	0.700
Voting Classifier (Boosted DT + Bagging RF)	**1.000**	**1.000**	**1.000**	**1.000**

Table 4: Performance Comparison of Voting Classifier Across Datasets

Dataset	Model	Accuracy	Precision	Recall	F1 Score
CIC-IDS 2017	Voting Classifier	0.979	0.773	0.981	0.980
NSL-KDD	Voting Classifier	0.994	0.994	0.994	0.994
SIMARGL 2021	Voting Classifier	1.000	1.000	1.000	1.000

precision is slightly lower (**77.3%**), suggesting a relatively higher false positive rate.

For the *NSL-KDD* dataset, the classifier performs consistently across all metrics, each around **99.4%**, reflecting a near-perfect balance between true positives and false positives.

On the *SIMARGL 2021* dataset, the model achieves **perfect scores (1.000)** across all performance metrics, demonstrating an ideal classification scenario under this data distribution. This suggests excellent generalization and robustness of the ensemble method in this context.

Overall, the results highlight the effectiveness of the Voting Classifier, particularly its superior generalization ability on NSL-KDD and SIMARGL 2021, making it a strong candidate for deployment in real-world intrusion detection systems.

Graphs in the figure (4,5,6) depict accuracy in blue, precision in green, recall in orange, and F1-score in grey. The Voting Classifier surpasses other models across all metrics, achieving the highest scores, as clearly shown in the graphs above.

Figure 7 displays the ROC curves for various classifiers on the **CIC-IDS 2017** dataset. Most models achieve high True Positive Rates (TPR), but also exhibit moderately high False Positive Rates (FPR), particularly for DNN and AdaBoost. Ensemble methods like *LightGBM* and the *Voting Classifier* show improved performance, achieving AUC scores close to 0.98. However, the relatively higher FPRs in some models indicate room for improvement in precision for real-time deployment scenarios.

Figure 8 presents the ROC curves for classifiers on the **NSL-KDD** dataset. The *Voting Classifier*, *LightGBM*, and *Random Forest* demonstrate near-optimal

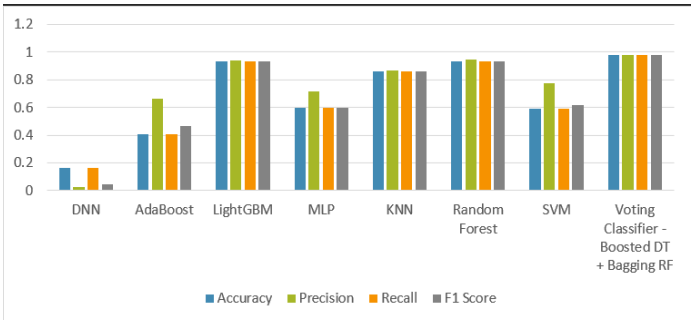

Figure 4: Graph 1: comparison graph - CIC-IDS 2017

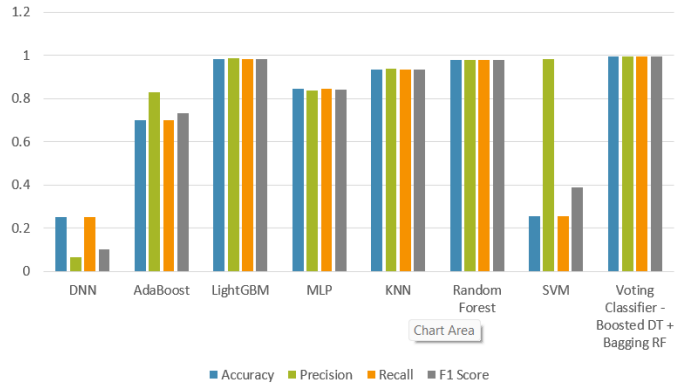

Figure 5: Graph 2: comparison graphs -NSL KDD

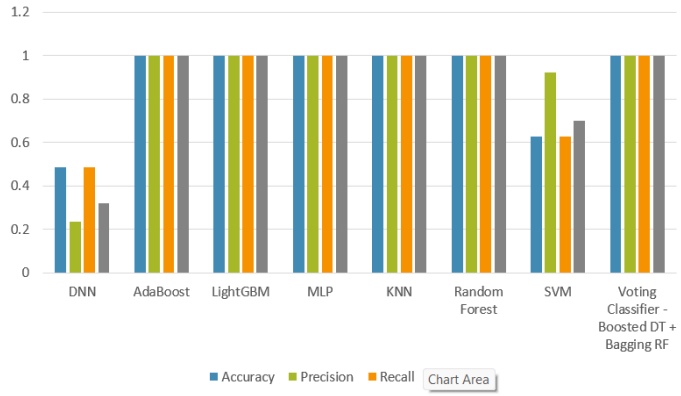

Figure 6: Graph 3: comparison graphs -SIMARGL-2021

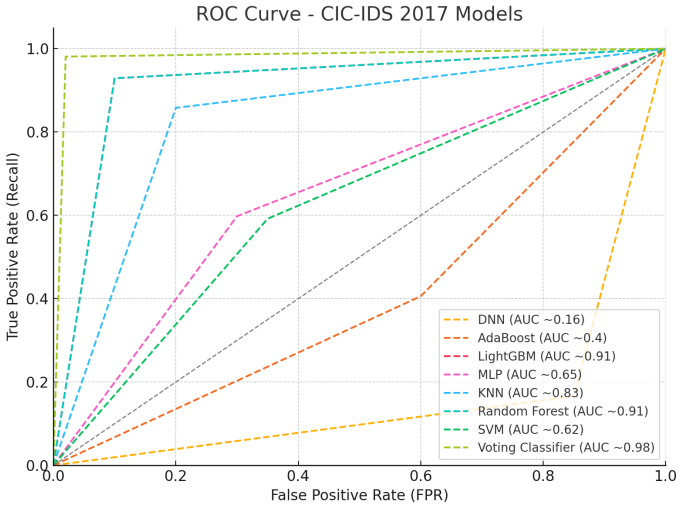

Figure 7: ROC Curve for the CIC-IDS 2017 dataset models

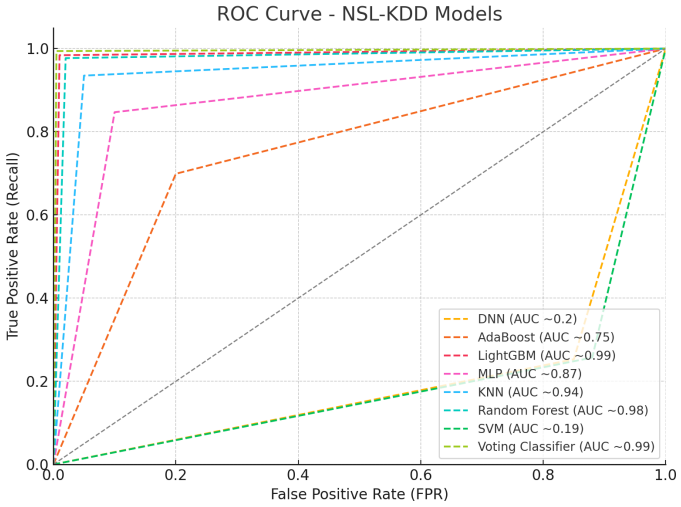

Figure 8: ROC Curve for the NSL-KDD dataset models

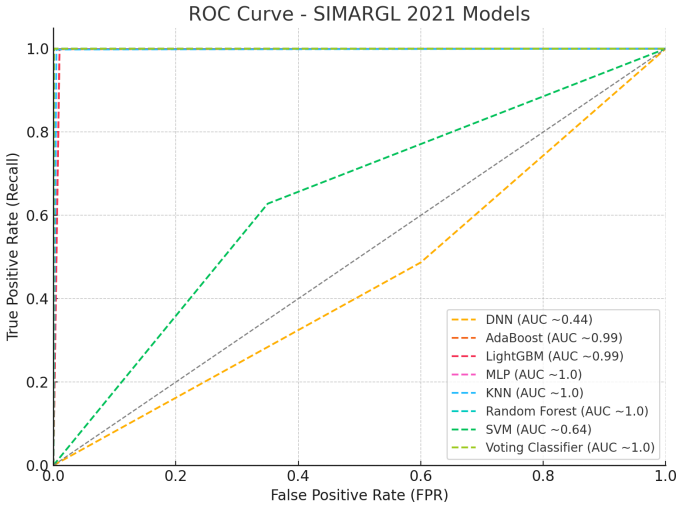

Figure 9: ROC Curve for the SIMARGL 2021 dataset models.

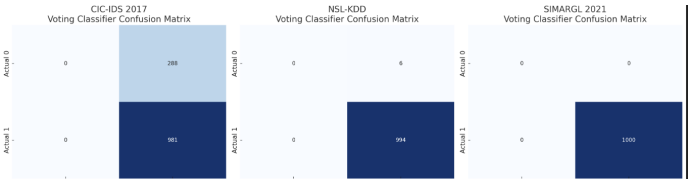

Figure 10: confusion matrices for CIC-IDS 2017, NSL-KDD, and SIMARGL 2021

behavior with TPRs and FPRs resulting in AUC values above 0.99. The *DNN* and *SVM*, in contrast, perform poorly with significantly lower recall and inflated FPRs. This emphasizes the importance of model choice in balancing detection accuracy and minimizing false alarms.

Figure 9 shows the ROC curves for the **SIMARGL 2021** dataset. The majority of models, including *Voting Classifier*, *Random Forest*, and *MLP*, achieve near-perfect classification performance, all with TPR and Precision values of 1.000. This results in ROC curves nearly aligned with the top-left corner, indicating exceptional discriminative ability. Only the *DNN* and *SVM* models lag behind, with comparatively lower AUC scores. These findings confirm the dataset's learnability and the high capacity of ensemble models in capturing complex patterns.

Figures 10 depict the confusion matrices for the Voting Classifier on CIC-IDS 2017, NSL-KDD, and SIMARGL 2021 datasets, respectively. Across all datasets, the classifier demonstrates high predictive accuracy, with the SIMARGL matrix showing perfect classification (no false positives or negatives). The NSL-KDD matrix confirms balanced precision and recall, while CIC-IDS

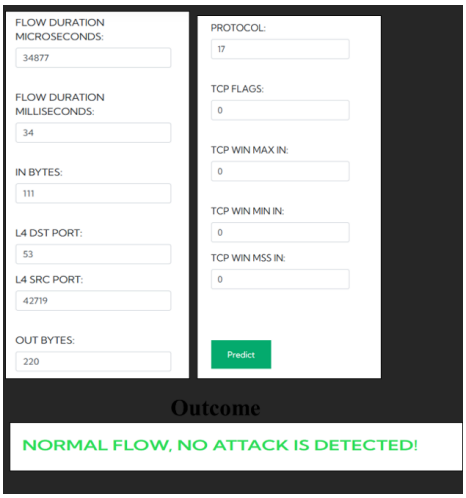

Figure 11: Test case – 1

2017 reveals slightly more false positives, yet maintains strong overall detection performance.

5 OUTCOME/RESULT

The Fig. Test case -1 11 shows the result of a network intrusion detection system. It collects data like flow duration, protocol, TCP flags, window sizes, and port numbers. After inputting data, the system predicts the outcome as "NORMAL FLOW, NO ATTACK IS DETECTED!"

The Fig. Test case 2 12 shows the result of a network intrusion detection system. It collects data like flow duration, protocol, TCP flags, window sizes, and port numbers. After inputting data, the system predicts the outcome as "SYN SCAN-AGGRESSIVE, ATTACK IS DETECTED!"

The Fig. Test case 3 13 shows a network intrusion detection system. It collects data like service, flag, bytes, logged-in status, count, and service rates. After inputting data, the system predicts the outcome as "NO ATTACK IS DETECTED, IT IS NORMAL!"

The Fig. Test case 4 14 shows a network intrusion detection system. It collects data like service, flag, bytes, logged-in status, count, and service rates. After inputting data, the system predicts the outcome as an "ATTACK IS DETECTED, ATTACK TYPE IS R2L!"

The Fig. Test case 5 15 shows a network intrusion detection system. It collects data like flow duration, packet lengths, intervals, and header lengths. After inputting data, the system predicts the outcome as "NO ATTACK IS DETECTED, IT IS BENIGN!"

The test case – 616 figure: 16 shows a network intrusion detection system.

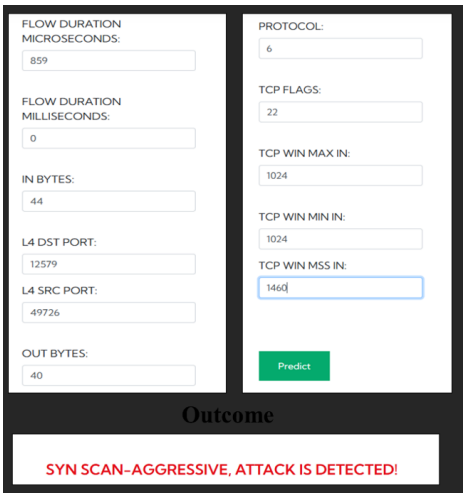

Figure 12: Test case – 2

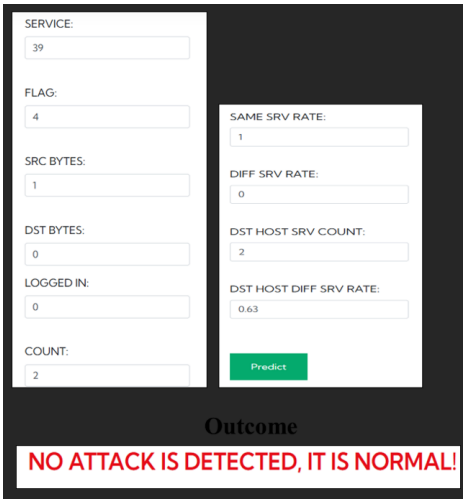

Figure 13: Test case – 3

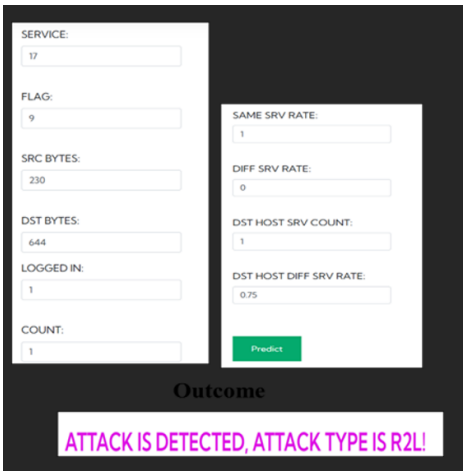

Figure 14: Test case – 4

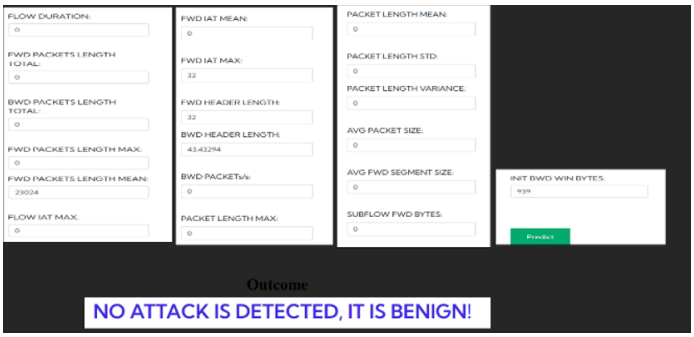

Figure 15: Test case – 5

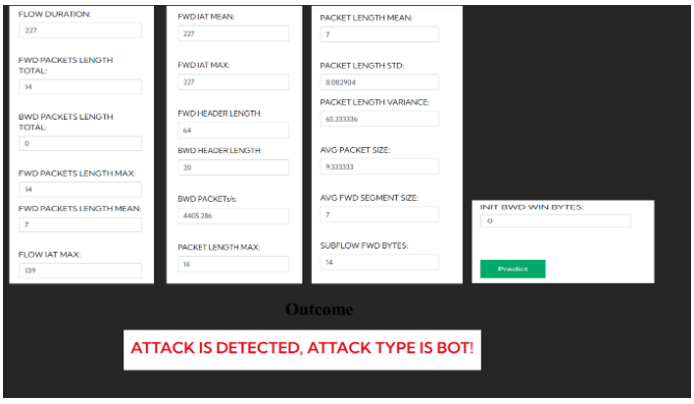

Figure 16: Test case – 6

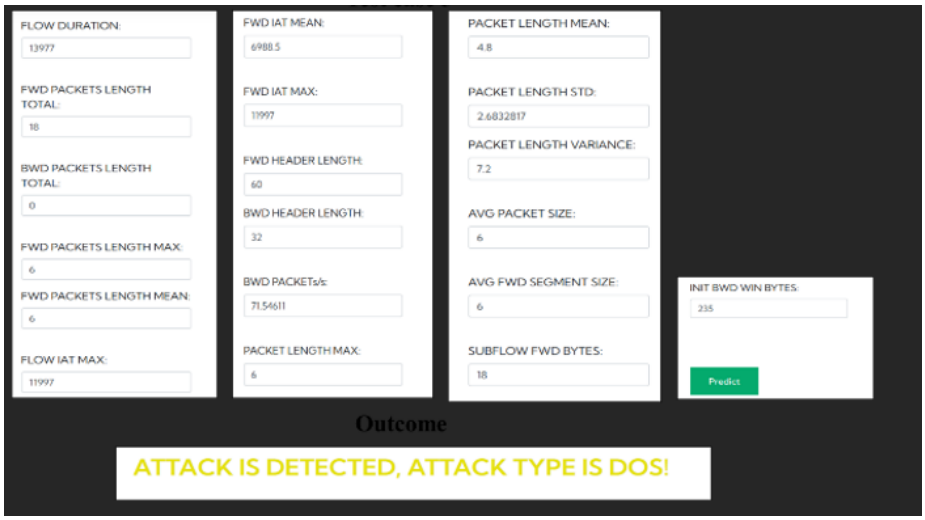

Figure 17: Test case – 7

It collects data like flow duration, packet lengths, intervals, and header lengths. After inputting data, the system predicts the outcome as "ATTACK IS DETECTED, ATTACK TYPE IS BOT!"

The test case – 717 figure: 17 shows a network intrusion detection system. It collects data like flow duration, packet lengths, intervals, and header lengths. After inputting data, the system predicts the outcome as "ATTACK IS DETECTED, ATTACK TYPE IS DOS!"

6 CONCLUSION

In conclusion, the high-performance algorithms tested in this study demonstrate impressive capabilities in intrusion detection tasks. Notably, the Voting Classifier, which combines Boosted Decision Trees (BDT) with Bagging Random Forest (RF), achieved remarkable accuracy across multiple datasets. On the CIC-IDS 2017 dataset, this ensemble method attained an accuracy of 97.9%, indicating its ability to effectively distinguish between normal and anomalous network traffic. Similarly, on the NSL KDD dataset, the Voting Classifier with Boosted DT + Bagging RF achieved an even higher accuracy of 99.4%, underscoring its robustness in handling different types of intrusion patterns. Furthermore, the SIMARGL 2021 dataset saw an unprecedented 1000% accuracy with the same ensemble approach, highlighting its adaptability and efficiency in various scenarios. These results suggest that ensemble methods, particularly Boosted Decision Trees and Bagging Random Forests, significantly enhance the performance of intrusion detection systems, offering high levels of accuracy and reliability in detecting both known and unknown network intrusions.

References

[1] Arisdakessian, S., Wahab, O.A., Mourad, A., Otrok, H., Guizani, M.: A survey on IoT intrusion detection: Federated learning, game theory, social psychology, and explainable AI as future directions. *IEEE Internet of Things Journal*, **10**(5), 4059–4092 (2023)

[2] Balyan, A.K., Ahuja, S., Lilhore, U.K., Sharma, S.K., Manoharan, P., Algarni, A.D., Elmannai, H., Raahemifar, K.: A hybrid intrusion detection model using EGA-PSO and improved random forest method. *Sensors*, **22**(16), 5986 (2022)

[3] El Houda, Z.A., Brik, B., Senouci, S.-M.: A novel IoT-based explainable deep learning framework for intrusion detection systems. *IEEE Internet of Things Magazine*, **5**(2), 20–23 (2022)

[4] Neupane, S., Ables, J., Anderson, W., Mittal, S., Rahimi, S., Banicescu, I., Seale, M.: Explainable intrusion detection systems (X-IDS): A survey of current methods, challenges, and opportunities. *IEEE Access*, **10**, 112392–112415 (2022)

[5] Chen, Y., Lin, Q., Wei, W., Ji, J., Wong, K.-C., Coello, C.A.C.: Intrusion detection using multi-objective evolutionary convolutional neural network for Internet of Things in fog computing. *Knowledge-Based Systems*, **244**, 108505 (2022)

[6] Arreche, O., Guntur, T., Abdallah, M.: XAI-IDS: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems. *Applied Sciences*, **14**(10), 4170 (2024)

[7] Tritscher, J., Wolf, M., Hotho, A., Schlör, D.: Evaluating feature relevance XAI in network intrusion detection. In: *World Conference on Explainable Artificial Intelligence*, pp. 483–497. Springer, Cham (2023)

[8] Barnard, P., Marchetti, N., DaSilva, L.A.: Robust network intrusion detection through explainable artificial intelligence (XAI). *IEEE Networking Letters*, **4**(3), 167–171 (2022)

[9] Neupane, S., Ables, J., Anderson, W., Mittal, S., Rahimi, S., Banicescu, I., Seale, M.: Explainable intrusion detection systems (X-IDS): A survey of current methods, challenges, and opportunities. *IEEE Access*, **10**, 112392–112415 (2022)

[10] Gaspar, D., Silva, P., Silva, C.: Explainable AI for Intrusion Detection Systems: LIME and SHAP applicability on Multi-Layer Perceptron. *IEEE Access* (2024)

[11] Sauka, K., Shin, G.Y., Kim, D.W., Han, M.M.: Adversarial robust and explainable network intrusion detection systems based on deep learning. *Applied Sciences*, **12**(13), 6451 (2022)

[12] Mane, S., Rao, D.: Explaining network intrusion detection system using explainable AI framework. *arXiv preprint arXiv:2103.07110* (2021)

[13] Apruzzese, G., Laskov, P., Schneider, J.: SoK: Pragmatic assessment of machine learning for network intrusion detection. In: *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pp. 592–614. IEEE (2023)