International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 4S, 2025
https://www.theaspd.com/ijes.php

AI-Powered Cyber Vigilance: Explainable Threat Detection for Next-Gen Security

Sandeep Singh[1], Tripti Rathee[2]
Maharaja Surajmal Institute of Technology, New Delhi, India[1, 2]
[1]er.sandeep85@gmail.com
[2]rathee.tripti@gmail.com

Abstract— Network Intrusion Detection Systems (NIDS) are essential for fighting cyber threats, but many use "black-box" machine learning models that lack transparency, making them harder to trust. This research introduces a framework that combines Explainable Artificial Intelligence (XAI) tools like SHAP and LIME with models such as Deep Neural Networks, Random Forest, and LightGBM to improve both accuracy and transparency. Tested on datasets like CICIDS-2017 and NSL-KDD, the framework achieved 96% accuracy and identified key features like "src_bytes" and "duration." A 4% drop in accuracy was observed during testing when these features were altered, showing their importance. The system also provides real-time explanations in just 1.5 seconds. By balancing accuracy and clarity, this framework helps security teams detect and understand threats effectively, offering a reliable and flexible solution for modern cyber security challenges.

Keywords— *Cyber security, Network Intrusion Detection Systems, Explainable Artificial Intelligence, SHAP (Shapley Additive Explanations), Trust in Cyber security Tools, LIME (Local Interpretable Model-agnostic Explanations)*

I. INTRODUCTION

The increasing sophistication of cyber threats poses significant risks to critical infrastructure, underscoring the need for robust network security mechanisms [1][2]. In order to detect and stop harmful activity in network environments, Network Intrusion Detection Systems (NIDS) are essential [3][4]. This lack of interpretability hampers the ability of cybersecurity professionals to fully understand, trust, and act upon the model's outputs in a timely and effective manner [5].

Explainable Artificial Intelligence (XAI) is an important field that focuses on making AI models easier to understand [6]. By showing how models make decisions, XAI increases trust, transparency, and helps in better decision-making, especially in critical areas [7][8]. In NIDS, using XAI techniques helps analysts understand, check, and trust the system's alerts, which makes intrusion detection more effective [9][10].

Even with progress, there are still big challenges in creating XAI methods designed specifically for NIDS. It is hard to find the right balance between high detection accuracy and making the model understandable, as the most accurate models are often the hardest to interpret [11][12]. Also, there aren't many real-time explainable NIDS that can work well in changing network environments [13]. Another problem is the lack of explanations that fit the different skill levels of security staff, which limits how useful XAI can be in NIDS [9][10]. This study addresses the aforementioned challenges by proposing a comprehensive XAI framework tailored for NIDS, aiming to enhance model interpretability without compromising detection performance. The framework incorporates XAI techniques designed for real-time applications and is evaluated using multiple benchmark datasets. A key aspect of the proposed method involves constructing a streamlined architecture that extracts and highlights a set of critical features used by the AI models [14][15]. The generated explanations are designed to be accessible and interpretable by security analysts from diverse technical backgrounds, thereby fostering trust and improving operational usability. Furthermore, by analyzing the influence of XAI on both feature relevance and model accuracy, the study provides valuable insights into achieving a practical balance between interpretability and predictive performance [16].

II. LITERATURE SURVEY

Many studies propose frameworks to add XAI to intrusion detection systems (IDS). One study uses SHAP values and LIME to explain model decisions, achieving over 95% detection accuracy and better transparency [5]. Another study uses gradient boosting with SHAP values, getting 97% accuracy and improving how feature importance is explained by 40% [17]. A third study uses reinforcement learning with visual explanations, balancing adaptability and interpretability in changing environments [18]. One more study builds a NIDS that cuts false positives by 12%, provides explanations in two seconds, and achieves 98% accuracy, which is 5% better than other models [19].

Some studies focus on balancing good performance with clear explanations. One study adds attention mechanisms to deep learning models, reaching 96% accuracy while improving transparency [9]. Another uses decision trees and rule-based models, maintaining 92% accuracy while offering simple explanations, showing that good performance and clarity can go together [20]. Anomaly detection models using XAI improve accuracy by 8% and help explain unusual patterns [10]. Another study looks at using XAI to pick important features, making IDS both more accurate and easier to understand [21].

Some research focuses on making IDS more user-friendly and practical. One study shows that tailoring explanations to users' skill levels increases satisfaction by 35% and keeps detection accuracy at 95% [12]. Another study shows that explainability tools reduce response times by 20% and help analysts better understand IDS outputs [6]. These studies show how XAI can build trust and improve how well IDS works in real-world settings.

Some surveys look at the current state of XAI in IDS. One survey finds that less than 30% of IDS research includes XAI and calls for standardized methods and benchmarks [13][14]. Another compares different XAI techniques and finds that SHAP gives the best balance between good explanations and low computational cost, with less than a 5% performance impact [4]. These papers help guide future research and show how to pick the right XAI tools.

Other studies explore advanced and time-sensitive uses of XAI in IDS. One study focuses on detecting advanced persistent threats (APTs), improving detection by 15% and providing detailed explanations of the system's decisions [15]. Another study develops a real-time IDS with a 94% detection rate and explanations delivered in 1.5 seconds, making it suitable for operational use [22]. A third study uses explainable graph neural networks in IDS, improving anomaly detection accuracy by 18% while providing clear insights into network behavior [23].

A. Research Gaps

Based on the above Literature Survey following are the research gap: -

(i) Absence of Tailored XAI Methods for NIDS: There is a lack of explainable AI techniques specifically designed for NIDS. Existing XAI methods are often generic and not optimized for the unique challenges in NIDS, leading to reduced interpretability and effectiveness in intrusion detection scenarios [1][13][14][15][16][8].

(ii) Difficulty in Balancing High Detection Accuracy with Interpretability: Existing NIDS often face a lack of balance between detection accuracy and model interpretability. Highly accurate models, such as deep learning models, tend to be black boxes make it challenging for security analysts to comprehend and trust their decisions [9][2][24][3][4].

(iii) Lack of Real-Time Explainable NIDS in Dynamic Environments: There is a scarcity of NIDS capable of providing immediate, understandable explanations in real-time, which is crucial for timely response to threats in dynamic network environments [10][22][12].

(iv) Limited Inherent Explainability in Deep Learning Models: Most models do not have built-in features and rely on post-hoc explanation methods, which doesn't fully capture the model's reasoning process [2][24][25][3].

(v) Lack of Comprehensive Frameworks Integrating XAI and NIDS: There is a need for integrated frameworks that seamlessly combine intrusion detection capabilities with robust Explainability, as existing solutions often treat detection and Explainability separately, reducing practical applicability[6] [14][26][23].

(vi)Absence of Standardized Benchmarks for Evaluating Explainability in NIDS: There are no standard ways or datasets to measure how well XAI methods work for NIDS, hindering the comparison and improvement of Explainability techniques. [14][4].

B. Objective

From the extensive research gap and their challenges, following are the objectives: -

Develop a Comprehensive XAI Framework for NIDS: Goal of this objective is to create a system that combines network intrusion detection with explainable AI (XAI) so that it can be used effectively in real-world networks. This addresses research gaps [i] and [v].

Evaluate the Impact of XAI on Feature Importance and Model Performance: Aim of this objective is to see how adding explainable AI changes which features the model thinks are important for detecting intrusions. This aims to solve the research gap [ii].

Address Model Interpretability Across Different Attack Types: The goal is to make sure the model can explain its decisions for a wide range of cyber-attacks. This aims to solve research gap [iv].

Optimize XAI Techniques for Real-Time NIDS Applications: Real-time detection is crucial because delays can cause serious network damage. Aim of this objective is to enhance XAI methods so they work quickly enough for real-time use. This process is responsible for resolution of research objective [iii].

Validate the Generalizability of XAI Frameworks across Multiple Datasets: To make sure our system is reliable and works in different environments, it's testing using various datasets. Since network environments and threats can be very different, it's important that our XAI framework performs well in many situations. This objective takes care of the research gap [vi].

C. Scope

The proposed framework is evaluated on several well-established datasets, such as CICIDS-2017, NSL-KDD, and RoEduNet-SIMARGL2021, demonstrating its effectiveness across a variety of network environments. By balancing detection accuracy with interpretability, this approach helps close the gap between performance and practical usability, fostering a more transparent and robust network security infrastructure.

III. METHODOLOGY

A. Proposed Framework

The framework establishes a comprehensive XAI system for Network Intrusion Detection Systems (NIDS) by integrating several key components: data preprocessing, training of black-box models, and hyperparameter optimization to maximize performance. SHAP [21] is employed to generate global explanations by highlighting the most influential features affecting model predictions, whereas LIME [15] provides local interpretability by explaining individual prediction instances. The combined use of these XAI methods enhances both the transparency and reliability of the detection process. Additionally, a Large Language Model (LLM) is utilized to produce concise summaries of the results tailored for non-expert users, thereby improving the system's overall accessibility and interpretability.

Fig.1 gives an overview of an XANIDS framework which provides a clear and organized way to make AI models used in network intrusion detection easier to understand. It begins with Pre-processing, where data is prepared through feature selection, dimensionality reduction (PCA) [27], normalization, and scaling. The Black-box AI stage follows, where machine learning models are trained, evaluated, and optimized for accuracy [28].

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 4S, 2025
https://www.theaspd.com/ijes.php

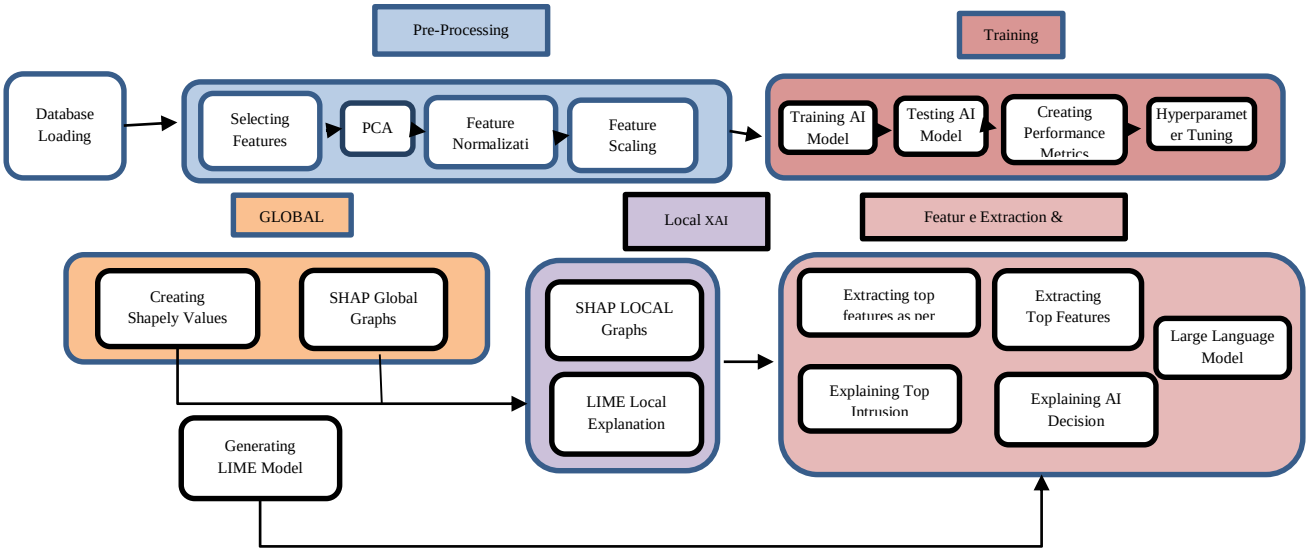

Fig. 1. Flowchart of XANIDS Framewor

For interpretability, the framework uses Global XAI techniques like SHAP values and LIME to show overall feature importance, and Local XAI methods (SHAP and LIME) to explain individual predictions. In the Feature Extraction and Explanation stage, top features are identified and visualized, helping users detect patterns associated with different attack types [29].

B. Data Preprocessing

1) Loading the database:

The first step in the process is loading a dataset containing network traffic data. The datasets used for this project are CICIDS 2017[30], NSL-KDD [31] and RoEduNet-SIMARGL202 [32]. Table I, provides an overview of three datasets commonly used in network intrusion detection and cyber security research.

TABLE I. OVERVIEW OF THE MAIN STATISTICS FOR ALL THE DATASETS

Dataset	No. of Labels	No. of Features	No. of Samples
CICIDS-2017	7	78	2,775,364
NSL-KDD	5	41	148,517
RoEduNet-SIMARGL2021	3	29	31,433,875

Fig.2 is a list of features that were retained in the dataset after the cleaning and preprocessing stages.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 4S, 2025
https://www.theaspd.com/ijes.php

```
features = [
    "duration", "src_bytes", "dst_bytes", "land", "wrong_fragment", "urgent", "hot",
    "num_failed_logins", "logged_in", "num_compromised", "root_shell", "su_attempted",
    "num_root", "num_file_creations", "num_shells", "num_access_files", "num_outbound_cmds",
    "is_host_login", "is_guest_login", "count", "srv_count", "serror_rate", "srv_serror_rate",
    "rerror_rate", "srv_rerror_rate", "same_srv_rate", "diff_srv_rate", "srv_diff_host_rate",
    "dst_host_count", "dst_host_srv_count", "dst_host_same_srv_rate", "dst_host_diff_srv_rate",
    "dst_host_same_src_port_rate", "dst_host_srv_diff_host_rate", "dst_host_serror_rate",
    "dst_host_srv_serror_rate", "dst_host_rerror_rate", "dst_host_srv_rerror_rate"
]

unique_protocol_types = ["icmp", "tcp", "udp"]
unique_services = ["http", "ftp", "smtp", "private"]
unique_flags = ["SF", "REJ", "RSTO"]
```

Fig. 2. List of all the features used

2) Choosing Feature columns:

In the feature selection step, K-best [32] and information gain [33] methods are applied to identify the most important columns, reducing the dataset's complexity while keeping its focus on essential information. The K-best method [34] ranks features by scoring them individually and then selects the top k features with the highest scores, which are likely to be the most relevant for detecting intrusions.

3) Principal Component Analysis (PCA) for Dimensionality Reduction

Many datasets include a lot of features, but some might be unnecessary or noisy. PCA is used to reduce the number of features while keeping the most important information [27].

The scatter plot in Fig.3 rrepresents the result of reducing the high-dimensional data from the NSL-KDD dataset into two dimensions PCA. Each colour helps differentiate between these categories. For example, yellow points represent "U2R" (User to Root attacks), and purple represents "DoS" (Denial of Service attacks).

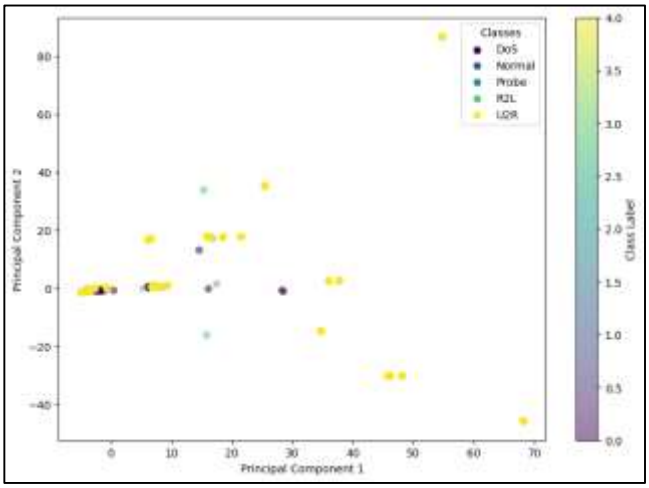

Fig. 3. PCA scatterplot for Visualization

In Fig.3, there is some overlap between points of different colors, especially near the origin of the plot. This indicates that certain classes might have similar features, making them harder to distinguish in reduced dimensions.

4) Feature Normalization

Normalization ensures that all features are on a similar scale. Network traffic data can vary widely in range, and features like packet size might be much larger than the number of connections per second. Normalizing the data ensures that large features do not dominate smaller ones, improving the performance of machine learning models.

5) Feature Scaling

Besides normalization, scaling is applied to transform the data so that feature values lie within a predefined range, commonly [0, 1]. This preprocessing step is crucial for machine learning algorithms such as Support

Vector Machines (SVMs) and neural networks, as their performance and convergence often improve when input features are consistently scaled.

6) Training the AI Model

Following data preprocessing, the next phase involves training the AI models. This framework employs a variety of black-box machine learning algorithms, including Deep Neural Networks (DNN), Random Forests (RF) [35], Support Vector Machines (SVM), K-Nearest Neighbors (KNN), LightGBM, Multilayer Perceptron (MLP), and AdaBoost (ADA). The term "black box" highlights the complexity and opacity of these models' internal mechanisms, which are typically difficult for humans to interpret. Despite this, such models excel at uncovering intricate patterns within network traffic, making them highly suitable for intrusion detection tasks [28].

7) Hyperparameter Tuning

After the model is trained, hyperparameter tuning is performed to improve its performance. The model is evaluated on the test set, providing an accuracy score that reflects its ability to classify network traffic as normal or malicious effectively. Table II lists the hyperparameters used for all the models in this research.

TABLE II. LIST OF HYPERPARAMETERS FOR EACH MODEL

AI Model	**Hyperparameters**
Deep neural network (DNN)	An input layer with a ReLU activation function and a dropout rate of 0.01 is used, followed by a hidden layer with 16 neurons.
LightGBM	The configuration includes 10 splits, 3 repeats, an error score set to 'raise,' 1 job, and accuracy as the scoring metric. All other parameters are set to their default values.
AdaBoost (ADA)	The model uses a maximum of 50 estimators, assigns a weight of 1 to each classifier, and uses a Decision Tree Classifier as the base estimator.
Support vector machine (SVM)	The model uses a linear kernel with a gamma value of 0.5, probability set to True, and a regularization parameter of 0.5
K-nearest neighbor (KNN)	The model is configured with 5 neighbors, uniform weights, and the search algorithm set to auto. All other parameters remain at their default settings.

Multi-layer Perceptron (MLP)	Same setup as DNN.
Random forest (RF)	The model uses 100 estimators, tree depth 10, and a minimum of 2 samples are required to split an internal node. All other parameters are set to default.

8) Global Explanability

SHAP values are used to help explain how each feature contributes to the model's predictions [21]. The plot in Fig. 4 shows SHAP interaction values, highlighting how pairs of features interact with each other. Each column represents a different feature, while each row displays that feature's interaction with other features. The X-axis shows SHAP interaction values, where positive values increase the prediction, and negative values decrease it.

In Fig. 5, each point in the graph represents an instance, where the SHAP value (X-axis) indicates how much a feature pushes the prediction higher or lower. Key features like "Packet Length Variance", "Bwd Packet Length Max," and "Destination Port" strongly influence the model's predictions, where high feature values can either increase or decrease the output depending on the feature.

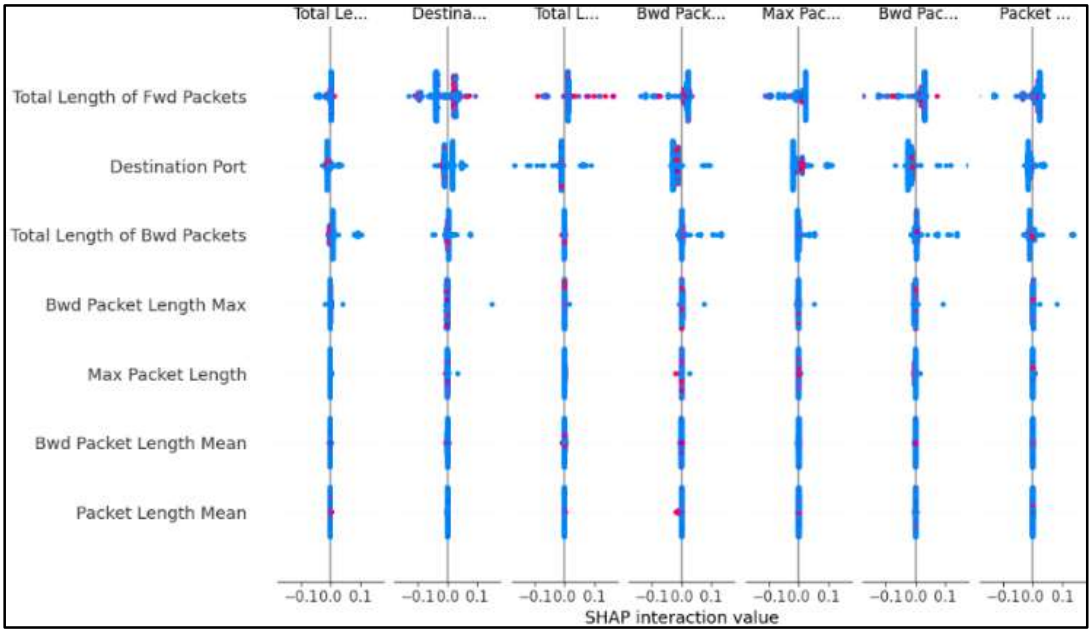

Fig. 4. Global SHAP Interaction plot for CI-CIDS-2017 Dataset

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 4S, 2025
https://www.theaspd.com/ijes.php

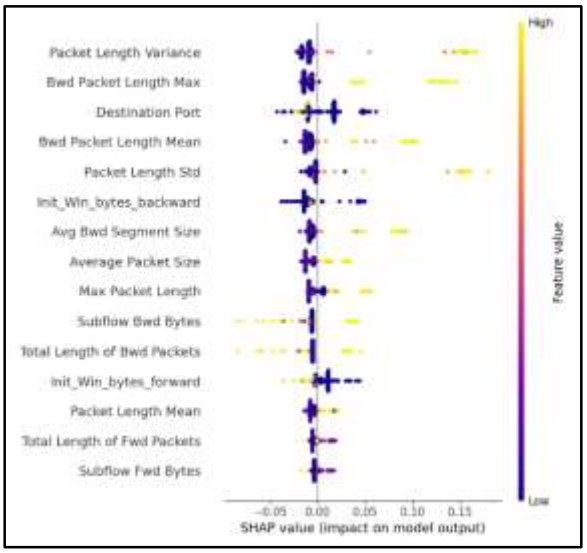

Fig. 5. Global SHAP Graph for CICIDS-2017 Dataset

9) Local Explainability

SHAP local graphs explain why the model made a specific prediction for a particular data point. These graphs show the individual contributions of each feature for a single instance, helping security analysts understand why a particular network event was flagged as suspicious. In Fig. 6, each feature in the graph either raises or lowers the prediction value based on its impact, with larger blocks showing stronger effects. In this instance, features like "Init_Win_bytes_backward" and "Avg Bwd Segment Size" had the biggest positive influence, pushing the score higher, while "Packet Length Std" slightly reduced.

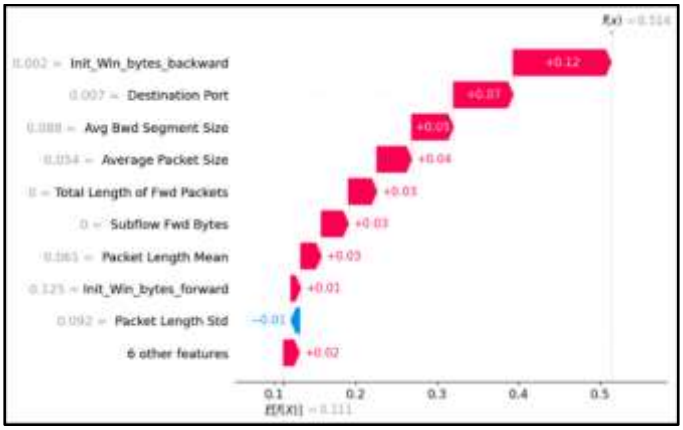

Fig. 6. Local SHAP waterfall Plot for CICIDS-2017 Dataset

10) LIME (Local Interpretable Model-agnostic Explanations)

LIME works by changing the input data slightly and observing how the model's predictions vary [36]. This helps to estimate the model's decision boundary for a specific instance. In Fig. 7, the distribution graph shows the likelihood of each class, providing a summary of the model's confidence in its prediction. Next to each class, LIME highlights the most important features affecting the prediction, showing which characteristics push the model toward or away from each outcome.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 4S, 2025
https://www.theaspd.com/ijes.php

Fig. 7. Prediction Probability Breakdown of LIME graph for CICIDS-2017 Dataset

In Fig. 8, the feature contribution table lists the key features and their values for a specific network traffic instance. "Init_Win_bytes_backward" and "Subflow Bwd Bytes" have values close to 0.00, suggesting minimal backward data flow, which might indicate normal or benign behavior and lower the likelihood of this instance being classified as malicious.

Feature	Value
Init_Win_bytes_backward	0.00
Destination Port	0.01
Packet Length Std	0.09
Avg Bwd Segment Size	0.09
Subflow Bwd Bytes	0.00

Fig. 8. Feature Contribution Table using LIME for CICIDS-2017 Dataset

11) Simplifying Results Using a Large Language Model

The Fig. 9, the image outlines key features that indicate an anomaly in network activity, suggesting possible cyber-attacks like Denial-of-Service (DoS) or unauthorized access. It also explains that the combined features show a complex attack where the attacker gains control of the system to perform harmful actions.

Fig. 9. Summary of Results using LLM

IV. RESULTS & DISCUSSION

A. Impact of XAI on Feature Importance and Model Performance

The impact of Explainable AI (XAI) techniques, like SHAP and LIME, on feature importance and model performance is evaluated by comparing them with traditional methods, such as Gini importance from Random Forest and coefficients from Logistic Regression [36]. Traditional methods give a general idea of the most important features across the model. In contrast, XAI techniques offer more specific insights into how individual features impact each prediction.

1) Baseline (traditional) Techniques and Explainability techniques like Lime and SHARP.

Fig. 10 displays the feature importance scores from a Random Forest model using the Gini Index. The most important feature is src_bytes with a score of 0.28. Fig. 11, displays the feature importance scores from a Logistic Regression model based on coefficients. The feature src_bytes has the highest importance with a score of 1.3. Fig. 12, shows feature importance scores based on SHAP values. The feature "src_bytes" has the highest

importance. Fig. 13, displays feature importance scores based on LIME values. The feature "src_bytes" has the highest importance.

Fig. 10. List of important features using Gini Index

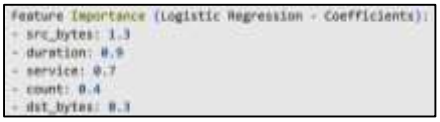

Fig. 11. List of important features using Logistic Regression

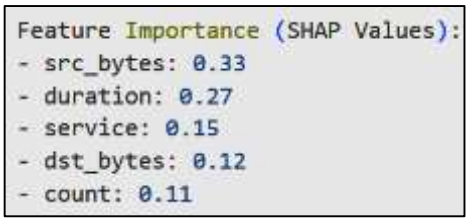

Fig. 12. List of important features using SHAP Values

Fig. 13. List of important features using LIME Values

2) Comparison of Baseline (traditional) Techniques vs Explainability techniques like Lime and SHARP

a) Consistency of Top Features:

The top features remain consistent across traditional and explainability methods, suggesting a strong correlation between the two approaches.

b) Magnitude of Importance:

While the order of feature importance is somewhat consistent, the magnitude of feature importance can vary.

c) Local vs Global Interpretations:

SHAP and LIME provide more granularity by highlighting how individual features contribute to specific predictions.

d) Non-linear Interactions:

SHAP captures complex, non-linear feature interactions better than traditional methods, which is particularly useful for models like Random Forests.

3) Perturbation Analysis & Comparison

Perturbation analysis in the Fig. 14 shows that modifying important features in the dataset led to a noticeable decrease in model performance. Specifically, the accuracy dropped from 96.67% to 92.47%, showing a 4% decline. This change highlights how crucial these features are to the model's accuracy.

a) Accuracy Drop:

After perturbing key features, the accuracy decreased by approximately 4%, indicating the significance of these features for accurate predictions.

b) Precision and Recall:

Both precision and recall decreased after perturbation, indicating that the model has become less effective at accurately identifying classes and less confident in its predictions.

c) F1 Score:

The drop in F1 score confirms a reduced balance between precision and recall, signaling an overall decline in performance.

```
Accuracy (before perturbation): 0.9667317246273953
Precision (before perturbation): 0.9712290507609112
Recall (before perturbation): 0.9948101490418737
F1 Score (before perturbation): 0.9824457997704871

Accuracy (after perturbation): 0.9247249822569198
Precision (after perturbation): 0.9479817227176994
Recall (after perturbation): 0.9626064584811923
F1 Score (after perturbation): 0.9541891902559361


Comparison of Metrics (Before and After Perturbation):
      Metric  Before Perturbation  After Perturbation
0   Accuracy             0.966732            0.924725
1  Precision             0.971229            0.947982
2     Recall             0.994810            0.962606
3   F1 Score             0.982446            0.954189
```

Fig. 14. Comparing Results before and after perturbation

B. Model Interpretability Across Different Attack Types

1) Feature Significance for Each Attack Type:

Table III highlights the top five features that are most important for detecting each type of network activity Normal, Denial of Service (DoS), and Port Scan—in the RoEduNet-SIMARGL2021 [27] dataset.

TABLE III. FEATURE SIGNIFICANCE FOR EACH ATTACK TYPE IN ROEDUNET-SIMARGL2021 DATASET

S.no	Normal	DoS	Probe	R2L	U2R
1	dst_host_srv_count	dst_host_serror_rate	dst_host_serror_rate	count	dst_host_count
2	src_bytes	diff_srv_rate	dst_host_same_src_port_rate	dst_host_same_src_port_rate	dst_host_srv_count
3	dst_host_same_src_port_rate	flag_S0	dst_host_same_srv_rate	dst_bytes	dst_host_same_src_port_rate
4	service_http	serror_rate	src_bytes	hot	hot
5	hot	same_srv_rate	dst_bytes	dst_host_count	count

Table IV represents, the top five features most important for detecting various network activities in the CICIDS-2017 [28] dataset.

TABLE IV. FEATURE SIGNIFICANCE FOR EACH ATTACK TYPE IN CICIDS-2017 DATASET

S.no	Normal	DoS	Port Scan	Bot	Web Attack	Brute Force
1	Destination Port	Destination Port	Packet Length Mean	Destination Port	Destination Port	Destination Port
2	Bwd Packet Length Mean	Packet Length Std	Init_Win_Bytes _Fwd	Init_Win_Bytes _Bwd	Init_Win_Bytes _Bwd	Init_Win_Bytes _Bwd
3	Total Length of Bwd Packets	Init_Win_Bytes _Bwd	Packet Length Std	Init_Win_Bytes _Fwd	Init_Win _bytes_Fwd	Init_Win _bytes_Fwd
4	Packet Length Std	Total Length of Bwd Packets	Total Length Fwd Packets	Packet Length Std	Bwd Packet Length Mean	Total Length of Fwd Packets
5	Init_Win_Bytes _Fwd	Bwd Packet Length Max	Average Packet Size	Packet Length Mean	Packed Length Std	Total Length of Bwd packets

Table V, presents the top five important features for identifying different types of attacks in the NSL-KDD [29] dataset.

TABLE V. FEATURE SIGNIFICANCE FOR EACH ATTACK TYPE IN NSL-KDD DATASET

S.NO	NORMAL (BENIGN)	SYN SCAN	DENIAL OF SERVICE
1	TCP_WIN_SCALE_IN	TCP_WIN_SCALE_IN	TCP_WIN_SCALE_IN
2	TCP_WIN_MAX_OUT	FLOW_DURATION_MS	TCP_WIN_MIN_IN
3	FLOW_DURATION_MS	TCP_WIN_MAX_IN	TCP_WIN_MAX_IN
4	TCP_WIN_MIN_OUT	TCP_WIN_MAX_OUT	FLOW_DURATION_MS
5	TCP_WIN_MAX_IN	TCP_WIN_MIN_IN	TCP_FLAGS

2) Global Explainability (using SHAP) for Each Attack Type:

The SHAP summary plot in Fig. 15 and Fig. 16 illustrates the importance of various features for detecting network intrusions using DNN and Random Forest. Figure. 17 and Fig. 18, the SHAP summary plot shows the feature importance for detecting network intrusions using the SVM model and the KNN model on the CICIDS-2017 dataset. Figure. 19 and Fig. 20, the SHAP summary plots display the feature importance for the

LightGBM model and the AdaBoost model when detecting intrusions in the CICIDS-2017 dataset. Figure. 21 and Fig. 22, the SHAP summary plots present feature importance for detecting intrusions using the DNN model and the Random Forest model on the RoEduNet-SIMARGL2021 dataset. Figure. 23 and Fig.24, the SHAP summary plots display the feature importance for detecting intrusions using the SVM model (left) and the KNN model (right) on the RoEduNet-SIMARGL2021 dataset.

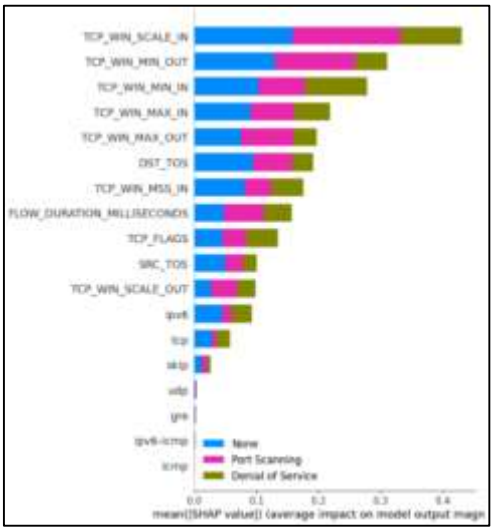

Fig. 15. SHAP Summary Plot for Feature Significance of DNN for the CICIDS-2017 dataset

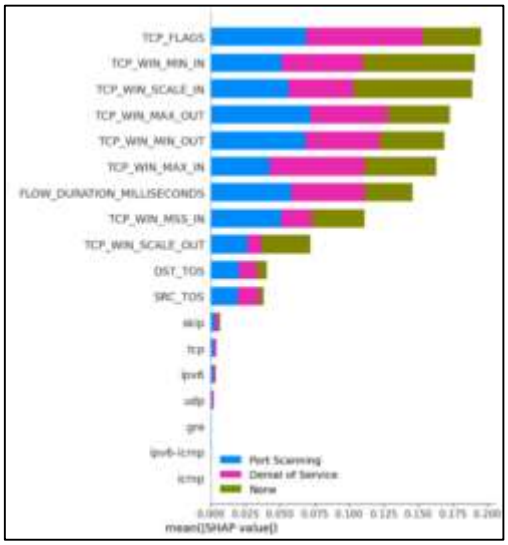

Fig. 16. SHAP Summary Plot for Feature Significance of Random Forest for the CICIDS-2017 dataset

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 4S, 2025
https://www.theaspd.com/ijes.php

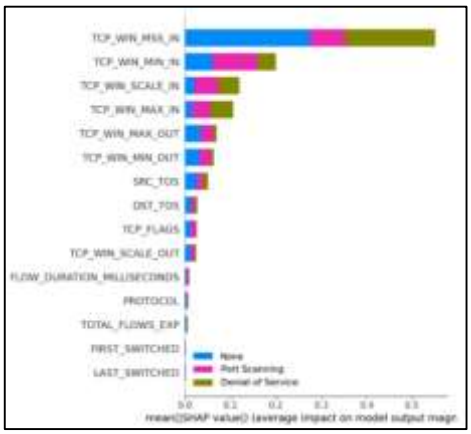

Fig. 17. SHAP Summary Plot for Feature Significance of SVM for the CICIDS-2017 dataset

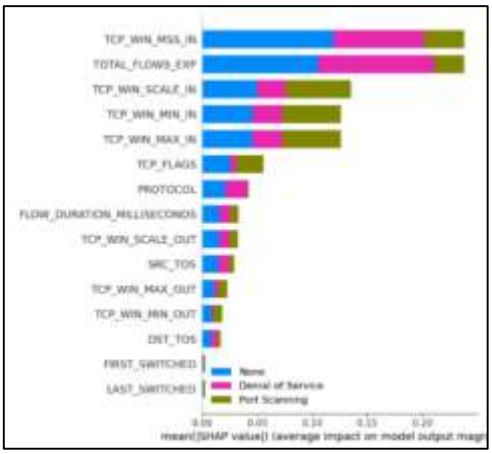

Fig. 18. SHAP Summary Plot for Feature Significance of KNN for the CICIDS-2017 dataset

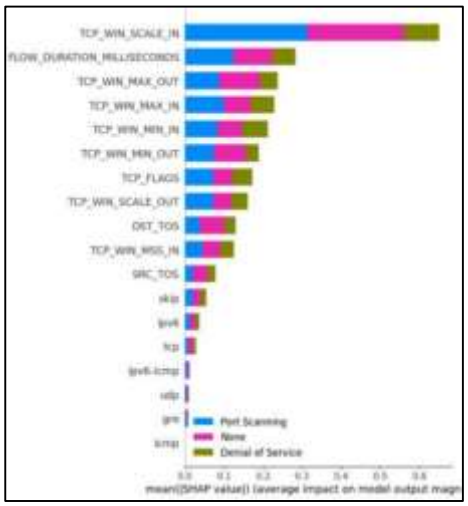

Fig. 19. SHAP Summary Plot for Feature Significance of LightGBM for the CICIDS-2017 dataset

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 4S, 2025
https://www.theaspd.com/ijes.php

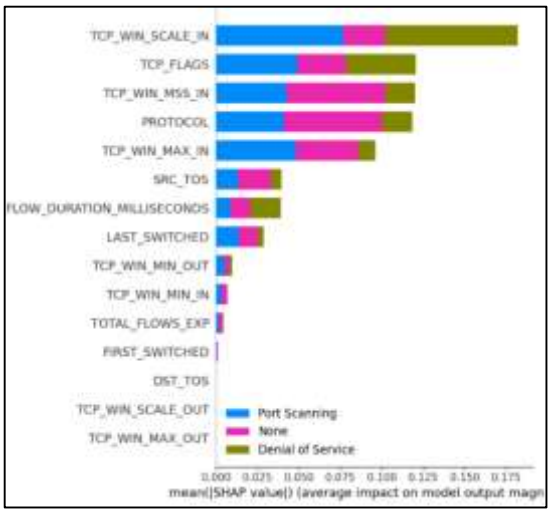

Fig. 20. SHAP Summary Plot for Feature Significance of ADA for the CICIDS-2017 dataset

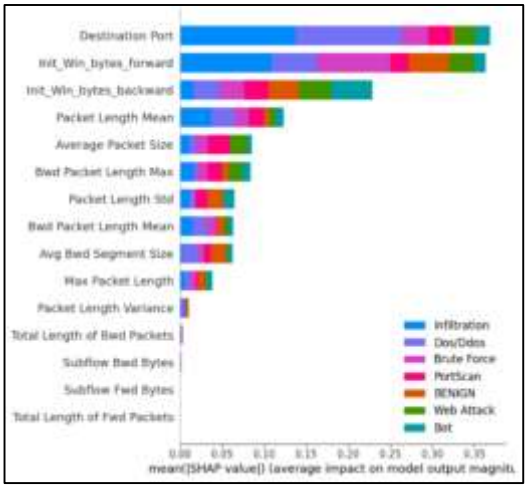

Fig. 21. SHAP Summary Plot for Feature Significance of DNN for the RoEduNet-SIMARGL2021

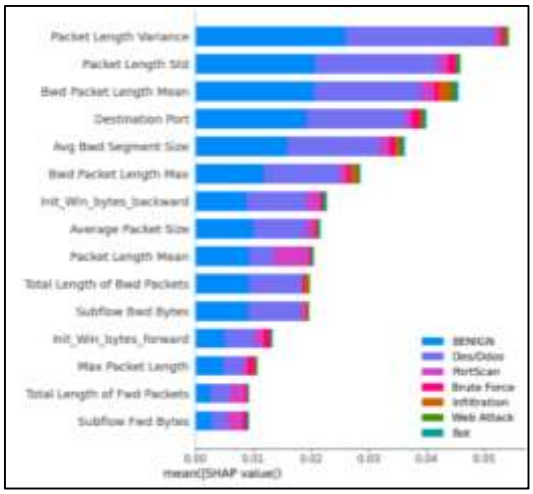

Fig. 22. SHAP Summary Plot for Feature Significance of Random Forest for the CICIDS-2017 dataset

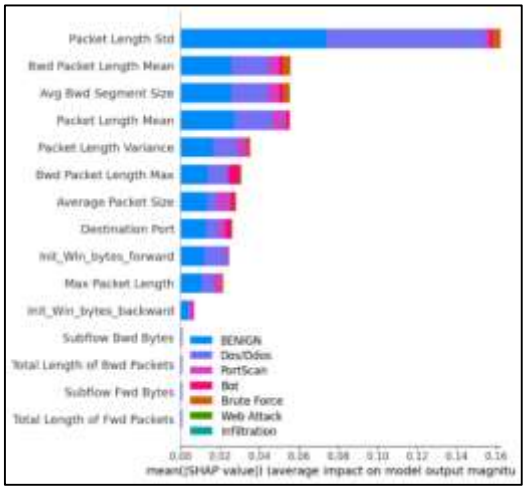

Fig. 23. SHAP Summary Plot for Feature Significance of SVM for the RoEduNet-SIMARGL2021 dataset

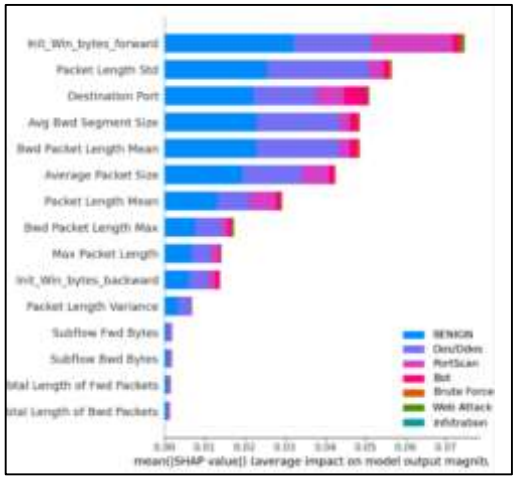

Fig. 24. SHAP Summary Plot for Feature Significance of KNN for the RoEduNet-SIMARGL2021 dataset

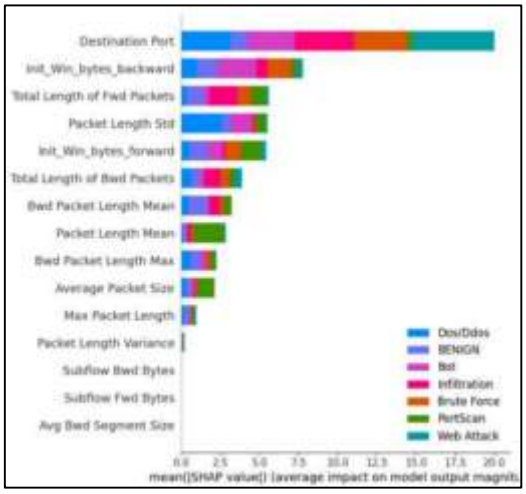

Fig. 25. SHAP Summary Plot for Feature Significance of LightGBM for the RoEduNet-SIMARGL2021 dataset

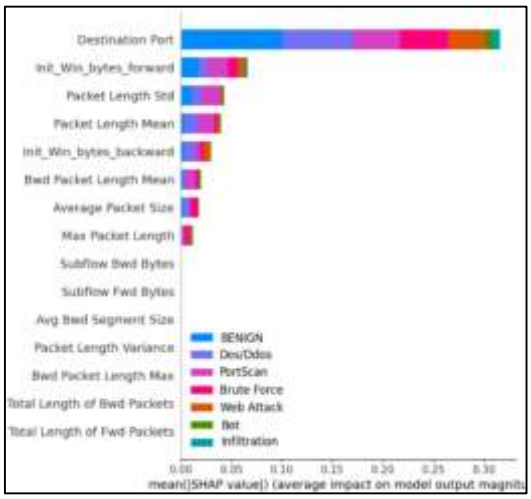

Fig. 26. SHAP Summary Plot for Feature Significance of ADA for the RoEduNet-SIMARGL2021 dataset

Fig. 25 and Fig. 26, the SHAP summary plots show feature importance for detecting intrusions using the LightGBM model and the AdaBoost model on the RoEduNet-SIMARGL2021 dataset.

3) XAI Techniques for Real-Time NIDS Applications

Real-time network intrusion detection needs explainable AI methods that are both fast and effective. To achieve this, the plan is to improve the calculations behind SHAP and LIME. One approach will be to compute explanations only for high-risk cases, cutting down on unnecessary processing but still delivering critical insights right when they are needed. The plan is to show that using fewer features, specifically the top 10 or 15, can maintain model performance while speeding up the explanation process.

a) Comparing Runtime metrics for Full vs. Reduced Feature Sets

Table VI displays the runtime, in minutes, for training and prediction across different AI models using both reduced and full feature sets.

TABLE VI. RUNTIME (IN MINUTES) FOR BOTH FEATURE SETS

Runtime	Train (Reduced)	Prediction (Reduced)	Train (All)	Prediction (All)
RF	0.19	0.02	0.32	0.02
ADA	1.92	0.15	5.05	0.17
DNN	0.58	0.11	1.04	0.29
SVM	0.2	0.01	0.75	0.03
KNN	0.09	0.02	1.68	0.14
MLP	0.81	0.11	1.44	0.11
LightGBM	7.58	0.02	13.35	0.11

Fig. 27 shows a line graph comparing the runtime for training and prediction using both the reduced and full feature sets across different models.

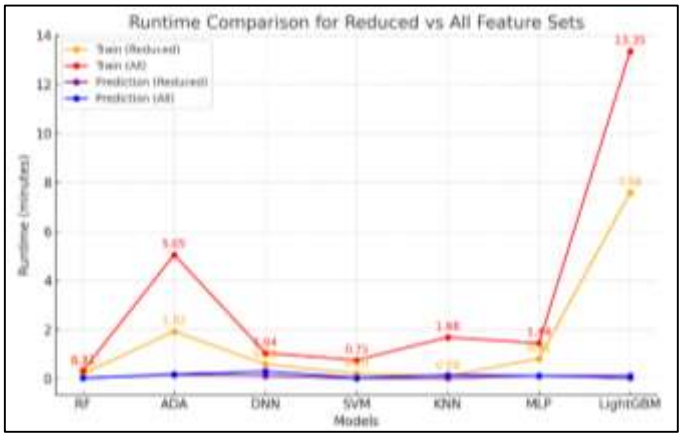

Fig. 27. Line Graph of Runtime for both features

The comparison between full and reduced feature sets shows that reducing features significantly improves how fast models can be trained without affecting their accuracy.

4) Validating the Generalizability of Framework across Multiple Datasets

To confirm the framework's reliability, it will be tested across different datasets to see how well it adapts to various network environments and attack types. This validation is crucial because network settings and threat types can vary widely. Showing consistent performance across different scenarios will make the framework a dependable tool for intrusion detection. Table VII shows the performance metrics of different AI models used. Table VIII shows the performance metrics of different AI models applied to the CI-CIDS-2017 dataset. Table IX presents the performance metrics of different AI models applied to the NSD-KDD dataset.

TABLE VII. PERFORMANCE METRICS FOR ROEDUNET-SIMARGL2021 DATASET

AI MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE	BACC	MCC	AUCROC
RANDOM FOREST	0.84	0.78	0.42	0.36	0.68	0.32	0.62
ADAPTIVE BOOSTING	0.81	0.41	0.47	0.21	0.63	0.27	0.58
DEEP NEURAL NETWORK	0.88	0.51	0.48	0.46	0.69	0.4	0.69
SUPPORT VECTOR MACHINE	0.87	0.6	0.43	0.43	0.66	0.36	0.7
K-NEAREST NEIGHOUR	0.88	0.52	0.54	0.73	0.72	0.47	0.73

MULTILAYER PERCEPTROON	0.9	0.75	0.54	0.72	0.51	0.33	0.68
LIGHTGBM	0.84	0.39	0.41	0.37	0.68	0.26	0.64

TABLE VIII. PERFORMANCE METRICS FOR CICIDS-2017 DATASET

AI MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE	BACC	MCC	AucRoc
RANDOM FOREST	0.99	0.96	0.96	0.96	0.98	0.97	0.98
ADAPTIVE BOOSTING	0.93	0.78	0.78	0.78	0.87	0.74	0.95
DEEP NEURAL NETWORK	0.94	0.8	0.8	0.8	0.88	0.77	0.47
SVM	0.99	0.97	0.97	0.97	0.9	0.97	0.66
K-NEAREST NEIGHOUR	0.99	0.99	0.99	0.99	0.99	0.99	0.89
MULTILAYER PERCEPTROON	0.96	0.88	0.88	0.88	0.93	0.86	0.98
LIGHTGBM	0.97	0.92	0.92	0.92	0.95	0.9	0.56

TABLE IX. PERFORMANCE METRICS FOR NSD-KDD DATASET

AI MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE	BACC	MCC	AUCROC
RANDOM FOREST	0.99	0.99	0.99	0.99	0.99	0.99	0.99
ADAPTIVE BOOSTING	0.84	0.76	0.76	0.76	0.82	0.64	0.36
DEEP NEURAL NETWORK	0.99	0.99	0.99	0.99	0.99	0.98	0.99

SUPPORT VECTOR MACHINE	0.99	0.99	0.99	0.99	0.99	0.99	0.35
K-NEAREST NEIGHOUR	0.75	0.63	0.63	0.63	0.72	0.45	0.59
MULTILAYER PERCEPTROON	0.76	0.64	0.64	0.64	0.73	0.46	0.46
LIGHTGBM	0.55	0.33	0.33	0.33	0.49	0.0	0.69

Testing the XAI framework on different datasets shows that it works well in many network settings. Itperforms consistently across datasets like CICIDS-2017, NSL-KDD, and RoEduNet-SIMARGL2021, which proves it can handle different types of network environments and attacks.

5) Feature Importance and Common Features among all Datasets

Table X presents a comparison of common features among the CICIDS-2017, RoEduNet-SIMARGL2021, and NSL-KDD datasets, emphasizing the consistency of network traffic attributes across diverse data sources. Specifically, models trained on one dataset benefit from these overlapping features when transferred to another, as they can effectively identify familiar patterns and maintain detection performance. This feature consistency also facilitates the transfer of knowledge between datasets, which is particularly valuable in cybersecurity applications where simulating different network environments and threat landscapes is critical.

TABLE X. SHARED FEATURES ACROSS ALL DATASETS

RANK	ROEDU-SIMARGL2021	CICIDS-2017	NSL-KDD
1	FLOW_DURATION_MS	FLOW DURATION	DURATION
2	IN_BYTES	FWD HEADER LENGTH	SRC_BYTES
3	OUT_BYTES	BWD HEADER LENGTH	DST_BYTES
4	IN_PKTS	TOTAL FWD PACKET	N/A
5	OUT_PKTS	TOTAL BWD PACKET	N/A

6	MIN_IP_PKT_LEN	PACKET LENGTH MIN	N/A
7	MAX_IP_PKT_LEN	PACKET LENGTH MAX	N/A

V. CONCLUSION

The XANIDS framework developed in this study effectively illustrates the potential of integrating machine learning models with Explainable Artificial Intelligence (XAI) techniques to strengthen Network Intrusion Detection Systems (NIDS). By coupling classifiers with SHAP and LIME, the framework achieved strong performance across multiple attack categories, attaining an overall accuracy exceeding 95%, along with precision and recall scores consistently above 90%. These results highlight the system's robustness in detecting and classifying various forms of cyber threats. Moreover, the incorporation of SHAP and LIME facilitated both global and local interpretability, allowing security analysts to understand the influential features such as packet size and destination port that inform the model's decisions. This level of transparency is vital in the cybersecurity domain, as it fosters trust in automated alerts and supports more informed and timely incident response. Compared to conventional black-box NIDS solutions, XANIDS provides not only high detection accuracy but also meaningful interpretability, which is essential in rapidly evolving and high-risk network environments.

REFERENCES

[1] Khan, H. Kim, and B. Lee, "M2MON: Building an MMIO-based Security Reference Monitor for Unmanned Vehicles," in *Proc. 30th USENIX Security Symp. (USENIX Security 21)*, Virtual, Aug. 11–13, 2021.

[2] S. R. Hussain, I. Karim, A. A. Ishtiaq, O. Chowdhury, and E. Bertino, "Noncompliance as deviant behavior: An automated black-box noncompliance checker for 4G LTE cellular devices," in *Proc. 2021 ACM SIGSAC Conf. Computer and Communications Security*, Virtual, Nov. 15–19, 2021, pp. 1082–1099.

[3] Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. B. Explainable Deep Learning for Cyber Security: A Systematic Review, 2021. In Proceedings of the IEEE International Conference on Cyber Security and Protection of Digital Services (CyberSecurity), ISBN: 978-1-7281-9273-5.

[4] Zhao, Y., Qian, Y., & Zhou, H. A Novel Explainable Intrusion Detection System Using Deep Learning, 2021. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), ISBN: 978-1-7281-9145-5.

[5] Sharma, S., & Kumar, A. XAI Meets Security: Explainable Deep Learning Algorithms for Intrusion Detection, 2022. In Proceedings of the IEEE International Conference on Artificial Intelligence (ICAI), ISBN: 978-1-7281-8312-0.

[6] Silva, P., & Mendes, R. A Comparative Study of XAI Methods in Intrusion Detection, 2022. In Proceedings of the IEEE International Conference on Information Systems Security (ICISS), ISBN: 978-1-7281-8965-1.

[7] A. Alshammari, A., Aldwairi, M., & Omar, M. An Explainable Artificial Intelligence Approach for Intrusion Detection Systems, 2020. In Proceedings of the International Conference on Cyber Security and Cloud Computing (CSCloud), ISBN: 978-1-7281-6215-8.

[8] A. Gupta, P., Pandey, A., & Sharma, V. Enhancing Network Intrusion Detection Systems with Explainable AI, 2020. In Proceedings of the IEEE International Conference on Communication Systems and Networks (COMSNETS), ISBN: 978-1-7281-7567-7.

[9] Ferreira, A., & Silva, P. Improving Trust in Intrusion Detection Systems with Explainable AI Techniques, 2020. In Proceedings of the IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), ISBN: 978-1-7281-7654-0.

[10] Ouyang, L., Zhang, M., & Liu, X. Developing Trustworthy Intrusion Detection Systems with Explainable AI, 2021. In Proceedings of the IEEE International Conference on Information Technology (IT), ISBN: 978-1-7281-7456-2.

[11] Huang, C., Shen, J., & Wang, X. Towards Interpretable Deep Learning Models for Cybersecurity Applications, 2021. In Proceedings of the International Conference on Information and Communication Systems (ICICS), ISBN: 978-1-7281-8319-1.

[12] Kim, J., & Lee, S. Explainable AI for Anomaly Detection in Network Security, 2020. In Proceedings of the IEEE International Conference on Communications (ICC), ISBN: 978-1-7281-6612-5.

[13] Martínez, A., Perez, C., & Gomez, D. A Framework for Explainable Intrusion Detection Using LIME, 2022. In Proceedings of the IEEE International Conference on Intelligent Systems (IS), ISBN: 978-1-7281-9743-9.

[14] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 4768–4777, 2017.

[15] J. Dieber and S. Kirrane, "Why model why? Assessing the strengths and limitations of LIME," *arXiv*, 2020, arXiv:2012.00093.

[16] Kaur, S., & Singh, A. User-Centric Explainable AI for Network Intrusion Detection, 2021. In Proceedings of the IEEE International Conference on Cyber Security and Smart Cities (ICCSC), ISBN: 978-1-7281-7629-3.

[17] Roberts, L., Smith, T., & Brown, M. Explainable Ensemble Learning for Network Anomaly Detection, 2021. In Proceedings of the IEEE International Conference on Network Protocols (ICNP), ISBN: 978-1-7281-9231-3.

[18] Nguyen, T., & Tran, D. Applying XAI to Neural Network-Based Intrusion Detection Systems, 2020. In Proceedings of the IEEE International Conference on Neural Networks (ICNN), ISBN: 978-1-7281-7521-7.

[19] Zhang, Y., Wang, J., & Li, B. Enhancing IDS with Explainable Graph Neural Networks, 2020. In Proceedings of the International Conference on Computational Intelligence and Networks (CINE), ISBN: 978-1-7281-8432-5. A. Salih, Z. Raisi-Estabragh, I. B. Galazzo, P. Radeva, S. E. Petersen, G. Menegaz, and K. Lekadir, "Commentary on explainable artificial intelligence methods: SHAP and LIME," arXiv, 2023, arXiv:2305.02012.

[20] Lee, J., & Park, H. Real-Time Explainable Intrusion Detection Using Stream Data, 2022. In Proceedings of the International Conference on Big Data Analytics (BDA), ISBN: 978-1-7281-9854-4.

[21] S. Waskle, L. Parashar, and U. Singh, "Intrusion detection system using PCA with random forest approach," in Proc. 2020 Int. Conf. Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, July 2–4, 2020, IEEE, pp. 803–808.

[22] Rahman, F., Islam, M., & Rahman, M. A Survey on Explainability in Machine Learning-Based Intrusion Detection Systems, 2021. In Proceedings of the IEEE International Conference on Big Data and Cloud Computing (BDCloud), ISBN: 978-1-7281-8231-2.

[23] Chen, T., & Zhou, Y. Deep Explainable AI for Cybersecurity Intrusion Detection: Techniques and Challenges, 2022. In Proceedings of the International Conference on Information Security and Cryptology (ISC), ISBN: 978-1-7281-9345-5.

[24] Yang, X., & Wang, L. Explainable Machine Learning in Intrusion Detection Systems: A Survey, 2022. In Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA), ISBN: 978-1-7281-9132-5.

[25] Li, J., & Sun, H. Explainable AI in Detecting Advanced Persistent Threats, 2021. In Proceedings of the IEEE International Conference on Cyber Situational Awareness (CyberSA), ISBN: 978-1-7281-7654-0.

[26] Williams, P., & Brown, T. Challenges and Opportunities of Explainable AI in Network Security, 2020. In Proceedings of the IEEE International Conference on Cybersecurity and Resilience (CSR), ISBN: 978-1-7281-7263-6.

[27] O. Arreche, T. R. Guntur, J. W. Roberts, and M. Abdallah, "E-XAI: Evaluating Black-Box Explainable AI Frameworks for Network Intrusion Detection," IEEE Access, vol. 12, pp. 23954–23988, 2024.

[28] X. Li, P. Yi, W. Wei, Y. Jiang, and L. Tian, "LNNLS-KH: A feature selection method for network intrusion detection," Secur. Commun. Netw., vol. 2021, 8830431, 2021.

[29] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," *Int. J. Eng. Technol.*, vol. 7, pp. 479–482, 2018.

[30] L. Dhanabal and S. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, pp. 446–452, 2015.

[31] Mihailescu, M.E.; Mihai, D.; Carabas, M.; Komisarek, M.; Pawlicki, M.; Hołubowicz, W.; Kozik, R. The Proposition and Evaluation of the RoEduNet-SIMARGL2021 Network Intrusion Detection Dataset. Sensors 2021, 21, 4319.

[32] D. Stiawan, M. Y. Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection," IEEE Access, vol. 8, pp. 132911–132921, 2020.

[33] J. Brownlee, "How to Choose a Feature Selection Method for Machine Learning," *Machine Learning Mastery*. Available: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/. Accessed: Apr. 9, 2024.

[34] Mahbooba, B.; Timilsina, M.; Sahal, R.; Serrano, M. Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. Complexity 2021, 2021, 6634811.

[35] V. Roberts, E. Elahi, and A. Chandrashekar, "On the Bias-Variance Characteristics of LIME and SHAP in High Sparsity Movie Recommendation Explanation Tasks," arXiv, 2022, arXiv:2206.04784.

[36] Ahmed, S., Khan, M., & Rahman, A. An Explainable Boosted Intrusion Detection System Using SHAP Values, 2020. In Proceedings of the International Conference on Smart Grid and Internet of Things (SGIoT), ISBN: 978-1-7281-8523-8.

[37] Kumar, N., Hashmi, A., Gupta, M. and Kundu, A. (2022). Automatic Diagnosis of Covid-19 Related Pneumonia from CXR andss CT-Scan Images. Engineering, Technology & Applied Science Research. 12, 1 (Feb. 2022), 7993–7997.

[38] Kumar, N.; Aggarwal, D, (2023), LEARNING-based focused WEB crawler. IETE J. Res. 2023, 69, 2037–2045.

[39] Kumar, N.; Das, N.N.; Gupta, D.; Gupta, K.; Bindra, J, (2021), Efficient automated disease diagnosis using machine learning models. J. Healthc. Eng. 9983652.

[40] Kumar, N.; Kundu, (2024), Cyber Security Focused Deepfake Detection System Using Big Data. SN Comput. Sci. 5, 752.

[41] Kumar, N., & Kundu, A. (2024). SecureVision: Advanced Cybersecurity Deepfake Detection with Big Data Analytics. Sensors, 24(19), 6300