International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Explainable Machine Learning for Real-Time Network Intrusion Analysis

Pranitha Boune * 1 , Jeshwanth Gandham 2 , Pochanna Kotrangi 3
1,2,3
Department of Computer Science & Engineering , RGUKTBasar , India

***Abstract*—The rapid growth of digital communication and internet technologies has significantly increased the exposure of computer networks to cyber threats and malicious activities. Traditional intrusion detection systems (IDS) mainly rely on signature-based techniques, which are often ineffective in detecting new or evolving attack patterns. To address this limitation, this paper proposes a machine learning–based intrusion detection system to improve the accuracy and reliability of detecting network intrusions. The CICIDS2017 dataset is used to represent diverse network behaviors and attack types. Data preprocessing and feature engineering techniques, including feature selection and normalization, are applied to enhance the quality of the dataset. Several machine learning algorithms, including K-Nearest Neighbors, Decision Tree, Random Forest, XGBoost, LightGBM, and CatBoost, are implemented to classify network traffic into normal and malicious categories. In addition, Explainable Artificial Intelligence techniques such as SHAP and LIME are used to interpret model predictions and improve transparency. Experimental results demonstrate that the proposed approach effectively detects intrusion patterns and contributes to stronger network security systems.**

Keywords*— *Intrusion Detection System (IDS), Machine Learning, Network Security, Cybersecurity, Classification Algorithms, CICIDS2017 Dataset, Explainable Artificial Intelligence (XAI).

1. INTRODUCTION

The rapid growth of internet technologies and digital communication has significantly increased the dependence on computer networks in modern society.

As organizations and individuals rely heavily on network-based services, the risk of cyber threats and Malicious attacks have also increased. Cyber-attacks such as malware, denial-of-service attacks, and unauthorized access can compromise sensitive information and disrupt network operations. Therefore, ensuring strong network security has become an important requirement for modern computing environments.

Intrusion Detection Systems (IDS) are widely used to monitor network traffic and identify suspicious activities that may indicate potential cyber-attacks. Traditional IDS approaches mainly rely on signature-based detection methods, which compare network patterns with known attack signatures. Although these techniques are effective in detecting previously known threats, they often fail to detect new or evolving attack patterns.

Machine learning techniques have emerged as powerful tools for improving intrusion detection systems. By learning patterns from network traffic data, machine learning models can classify activities as normal or malicious with improved accuracy. Algorithms such as K-Nearest Neighbors, Decision Tree, Random Forest, and other advanced models have been widely applied to enhance the performance of IDS.

In this paper, a machine learning-based intrusion detection system is proposed to improve the detection of malicious network activities. The CICIDS2017 dataset is used to represent realistic network traffic and various attack types of Data preprocessing and feature selection techniques are applied to improve data quality and model performance. Multiple machine learning algorithms are implemented and evaluated to classify network traffic effectively. In addition, Explainable Artificial Intelligence techniques such as SHAP and LIME are used to interpret model predictions and improve transparency.

.

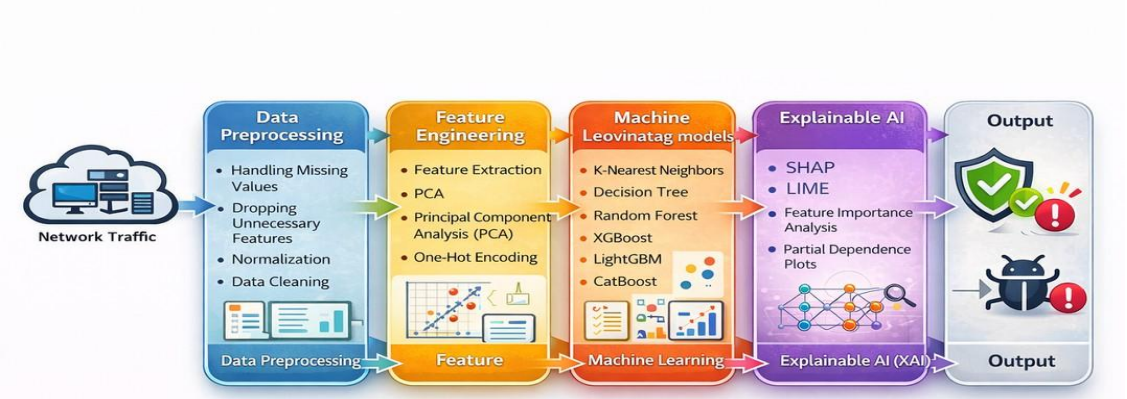

Fig. 1. An Enhanced and Detailed Classification Framework for Machine Learning-Based Network Intrusion Detection Systems (NIDS).

2. LITERATURE REVIEW

Intrusion Detection Systems (IDS) play a crucial role in safeguarding computer networks from cyber threats and unauthorized access. Traditional IDS techniques
primarily rely on signature-based detection methods, which are effective in identifying known attacks but fail to detect novel and evolving threats. To overcome these limitations, recent research has focused on the adoption of machine learning (ML) and deep learning (DL) techniques for intelligent and adaptive intrusion detection.

Several studies have explored the application of machine learning algorithms such as Decision Trees, Random Forest, Support Vector Machines (SVM),
and K-Nearest Neighbors (KNN) for intrusion detection. Nagamani and Sammulal [1] proposed an ensemble-based intrusion detection model integrating Recursive Feature Elimination (RFE) and Information Gain for optimized feature selection, achieving improved detection accuracy and reduced feature redundancy. Similarly, Yang et al. [14] and Zhou et al. [23] demonstrated that ensemble learning techniques significantly enhance classification.

performance and reduce false positives in IDS.evolving cyber-attacks. However, these deep learning models often suffer from a lack of interpretability, making it difficult for them to trust in critical security applications.

The proposed framework aims to provide an efficient and interpretable solution for detecting cyber threats and improving the overall security of modern computer networks.

To address dataset-related challenges, modern datasets such as CICIDS2017 have been introduced. Sharafuddin et al. [3] and Lashkari et al. [6] emphasized the importance of realistic datasets in improving IDS performance and generalization. The CICIDS2017 dataset provides comprehensive and labeled network traffic data, including various attack types such as DDoS, brute force, and botnet attacks, making it suitable for training robust intrusion detection models.

Hybrid and feature selection-based approaches have also been proposed to improve IDS efficiency. Alsaffar et al. [18] introduced a hybrid feature selection method combined with ensemble learning, leading to enhanced detection accuracy. Similarly, Li et al. [13] and Maulana et al. [22] demonstrated that effective feature selection techniques significantly improve model performance by reducing irrelevant and redundant features.

Recently, Explainable Artificial Intelligence (XAI) has gained significant attention in intrusion detection research. Gupta et al. [3], Gaspar et al. [4], and Khan et al. [20] applied SHAP and LIME techniques to interpret machine learning model predictions. These methods provide transparency by identifying important features influencing classification decisions, thereby increasing trust and usability in real-world applications. Furthermore, Salo et al. [25] and Alzahrani et al. [24] highlighted the importance of explainability in enhancing the reliability of IDS models.

Despite these advancements, several challenges remain, including handling imbalanced datasets, detecting zero-day attacks, and ensuring model interpretability. Therefore, integrating ensemble learning with feature selection and explainable AI

techniques presents a promising direction for developing accurate, robust, and interpretable intrusion detection systems.

This study builds upon existing research by combining machine learning, feature engineering, and explainable AI techniques to improve both detection performance and model transparency.

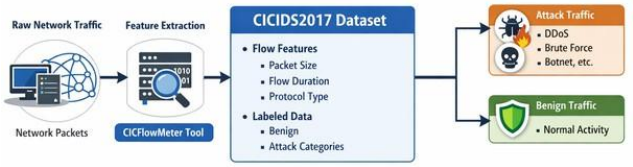

Fig. 2. *Representation of Dataset*

2.1 Contributions of the Paper

1. **Development of a machine learning-based intrusion detection system** that analyzes network traffic to accurately detect malicious activities and cyber-attacks.
2. **Implementation of an effective data preprocessing pipeline,** including handling missing values, removing irrelevant features, and normalizing data to improve model performance.
3. **Application of feature engineering techniques,** such as feature selection and dimensionality reduction using Principal Component Analysis (PCA), to enhance the quality of input features.
4. **Evaluation of multiple machine learning classifiers,** including Decision Tree, Random Forest, K-Nearest Neighbors, and other ensemble models, to identify the most effective model for intrusion detection.
5. **Integration of Explainable Artificial Intelligence (XAI) techniques,** such as SHAP and LIME, to interpret model predictions and provide insights into the factors influencing intrusion detection decisions.
6. **Improvement of transparency and reliability** in intrusion detection systems by combining high detection accuracy with model interpretability.

2.2 Dataset Description

This study uses the CICIDS2017 Dataset, a modern and realistic dataset developed by the Canadian Institute for Cybersecurity. The dataset contains network traffic data that includes both normal activities and various types of cyber-attacks such as Distributed Denial of Service (DDoS), brute force attacks, botnet attacks, and infiltration attacks.

The CICIDS2017 dataset includes a large number of network flow features extracted using the CICFlowMeter tool. These features describe different characteristics of network traffic, such as packet size, flow duration, and protocol information, which are useful for identifying malicious activities.

The dataset provides labeled data for both benign and attack traffic, making it suitable for training and evaluating machine learning-based intrusion detection systems.

3. PROPOSED METHODOLOGY

The proposed methodology focuses on developing an intelligent intrusion detection framework using machine learning techniques and explainable artificial intelligence methods. The system is designed to analyze network traffic data and accurately classify activities as normal or

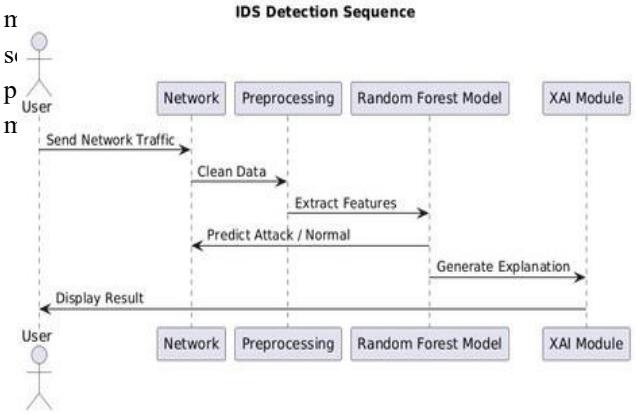

Fig. 3. *IDS Detection Sequence*

1) Data Collection

The dataset used in this study is the CICIDS2017 Dataset, which contains realistic network traffic generated from various attack scenarios and normal user activities. The dataset includes multiple types of

cyber-attacks such as Distributed Denial of Service (DDoS), brute force attacks, infiltration, and web-based attacks. These labeled records provide a reliable benchmark for training and evaluating intrusion detection models.

2) Data Preprocessing

Raw network traffic data often contains noise, redundant attributes, and missing values that may negatively affect model performance. Therefore, several preprocessing steps are applied before training the models.

The preprocessing process includes:

- Handling missing values in the dataset
- Removing duplicate records
- Dropping irrelevant or redundant features
- Normalizing numerical attributes for consistent scaling

These preprocessing steps help improve the quality of the dataset and enable better learning by machine learning algorithms.

3) Feature Engineering Feature engineering

plays a crucial role in improving model performance by selecting the most relevant attributes from the dataset. In this work, feature selection techniques are applied to identify important network traffic features that contribute to intrusion detection.Additionally, dimensionality reduction methods such as Principal Component Analysis are used to reduce feature complexity while preserving important information. This step helps improve computational efficiency and reduces model training time. plays a crucial role in improving model performance by selecting the most relevant attributes from the dataset. In this work, feature selection techniques are applied to identify important network traffic features that contribute to intrusion detection.Additionally, dimensionality reduction methods such as Principal Component Analysis are used to reduce feature complexity while preserving important information. This step helps improve computational efficiency and reduces model training time.

4) Machine Learning Classification

After preprocessing and feature engineering, multiple machine learning algorithms are trained to classify network traffic into normal and attack categories. The following classification algorithms are used in this study:

- K-Nearest Neighbors
- Decision Tree
- Random Forest
- XGBoost
- LightGBM
- CatBoost

These models learn patterns from the training dataset and generate predictions for unseen network traffic. The performance of each model is evaluated to determine the most effective algorithm for intrusion detection.

These models learn patterns from the training dataset and generate predictions for unseen network traffic. The performance of each model is evaluated to determine the most effective algorithm for intrusion detection.

5) Explainable Artificial Intelligence

To improve transparency and interpret ability of the trained models, explainable artificial intelligence techniques are incorporated into the system. In this study, two widely used explanation methods are applied:

- SHAP
- LIME

These techniques help analyze the contribution of individual features to the prediction results. By visualizing feature importance and model explanations, security analysts can better understand why certain network activities are classified as malicious.

6) Performance Evaluation

To evaluate the effectiveness of the proposed intrusion detection system, several standard classification performance metrics are used. These metrics measure how accurately the machine learning models classify network traffic into normal or malicious categories.

3.1 Evalution Metrics

Accuracy (ACC) represents the proportion of correctly classified instances, including both normal traffic and attack traffic. It is calculated as the ratio of correctly predicted observations to the total observations.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

TP = True Positives (correctly detected attacks)

TN = True Negatives (correctly detected normaltraffic)

FP = False Positives (normal traffic incorrectly classified as attacks)

FN = False Negatives (attacks incorrectly classified as normal)

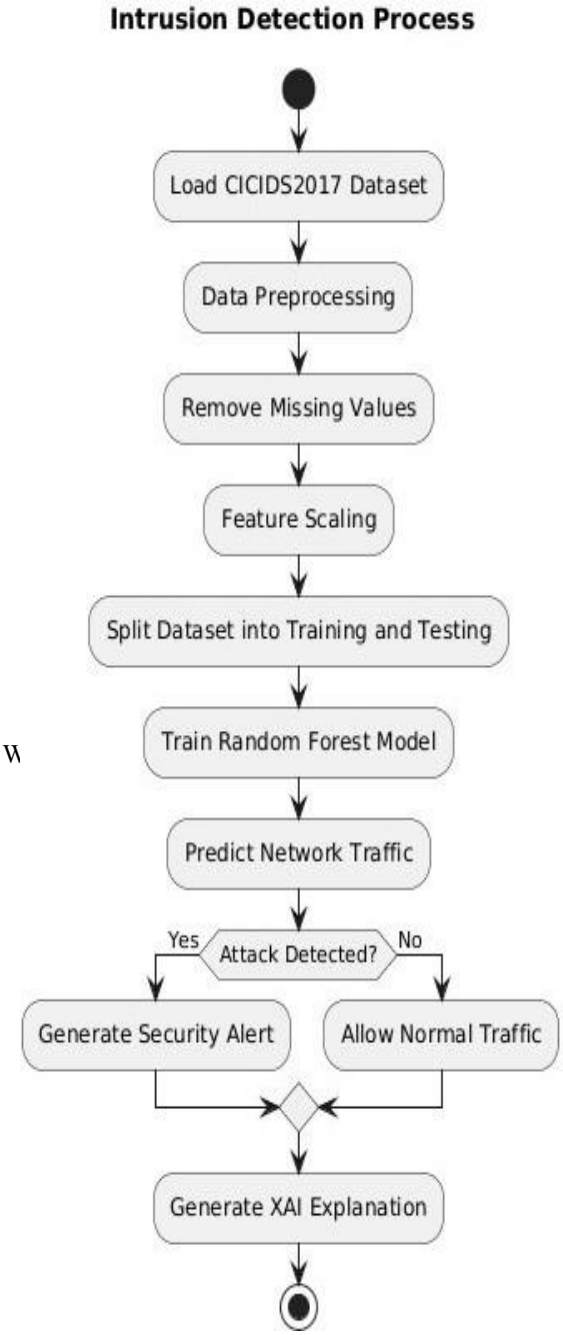

Fig. 4. Intrusion Detection Process

W

Precision (P) measures the proportion of correctly predicted attack instances among all instances predicted as attacks. It indicates how reliable positive predictions are.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (R), also known as sensitivity, measures the proportion of actual attack instances that are correctly identified by the model.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-Score is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance, especially when dealing with imbalanced datasets.

$$F1\text{-Score} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives
These evaluation metrics are used to compare the performance of different machine learning models including Random Forest, XGBoost, LightGBM, CatBoost, Decision Tree, and K-Nearest Neighbors. The results help determine the most effective model for detecting cyber-attacks in the network traffic dataset.

3.2 Algorithms Used

In this work, multiple machine learning algorithms were used to classify network traffic and detect cyber-attacks in the intrusion detection system. The models applied include **K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), XGBoost, LightGBM, and CatBoost.** These algorithms are widely used for classification tasks and are capable of handling complex network traffic patterns effectively. Among them, the **Random Forest classifier achieved the highest performance** in terms of accuracy and reliability.

Additionally, **Explainable Artificial Intelligence (XAI) techniques such as SHAP and LIME** were employed to interpret the model predictions and identify the most influential features contributing to attack detection.

.

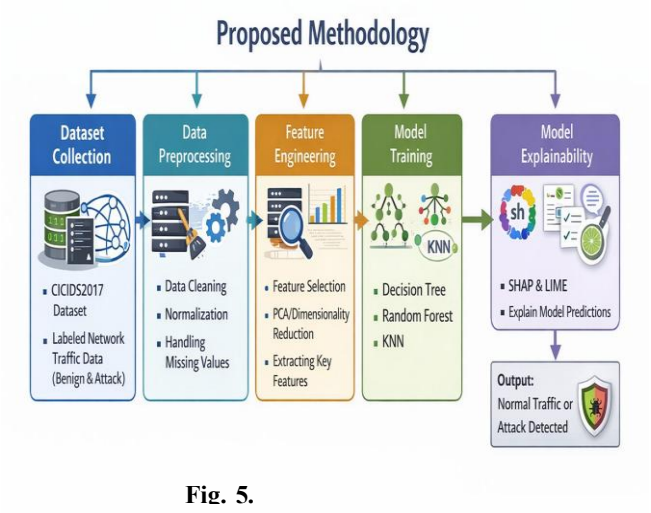

Fig. 5.

The proposed methodology presents a structured approach for detecting network intrusions using machine learning and explainable artificial intelligence techniques. Initially, the CICIDS2017 dataset is utilized as the primary data source, which contains labeled network traffic instances representing both normal and malicious activities.

In the data preprocessing stage, the dataset is cleaned by handling missing values, removing irrelevant features, and applying normalization to ensure consistency and improve model performance. Subsequently, feature engineering techniques are applied, including feature selection and dimensionality reduction using **Principal Component Analysis (PCA),** to extract the most relevant features and reduce computational complexity.

The processed data is then used to train multiple machine learning models, including **Decision Tree, Random Forest, and K-Nearest Neighbors (KNN).** These models learn patterns from the network traffic data and classify it into normal or attack categories. The performance of the models is evaluated using standard metrics such as **accuracy, precision, recall, and F1-score** to identify the most effective classifier.

To enhance model interpretability, **Explainable Artificial Intelligence (XAI) techniques such as SHAP and LIME** are integrated into the system. These methods provide insights into feature importance and explain how individual predictions are made by the models.

Finally, the system outputs the classification results along with explanations, indicating whether the network traffic is normal or malicious. This approach ensures high detection accuracy while improving transparency and reliability in intrusion detection systems.

3.2 Algorithms Used

K-Nearest Neighbors (KNN):
KNN is a distance-based classification algorithm used to classify network traffic by comparing a new data instance with the closest training samples in the feature space.

Decision Tree (DT):
Decision Tree is a supervised learning algorithm that classifies network traffic by creating a tree-like structure of decision rules based on dataset features.

Random Forest(RF):
Random Forest is an ensemble learning technique that combines multiple decision trees to improve classification accuracy and reduce overfitting in intrusion detection.

XGBoost:
XGBoost is a gradient boosting algorithm that builds models sequentially to minimize prediction errors and enhance the detection of malicious network activities.

Light GBM:
LightGBM is a fast and efficient gradient boosting framework designed to handle large datasets and improve model training speed and performance.

CatBoost:
CatBoost is a boosting algorithm that effectively handles categorical features and improves prediction accuracy in classification problems.

SHAP:
SHAP is an explainable AI technique used to identify the most important features influencing the intrusion detection model's predictions.

LIME:
LIME provides local explanations for individual predictions, helping to understand how the model classifies specific network traffic as normal or malicious.

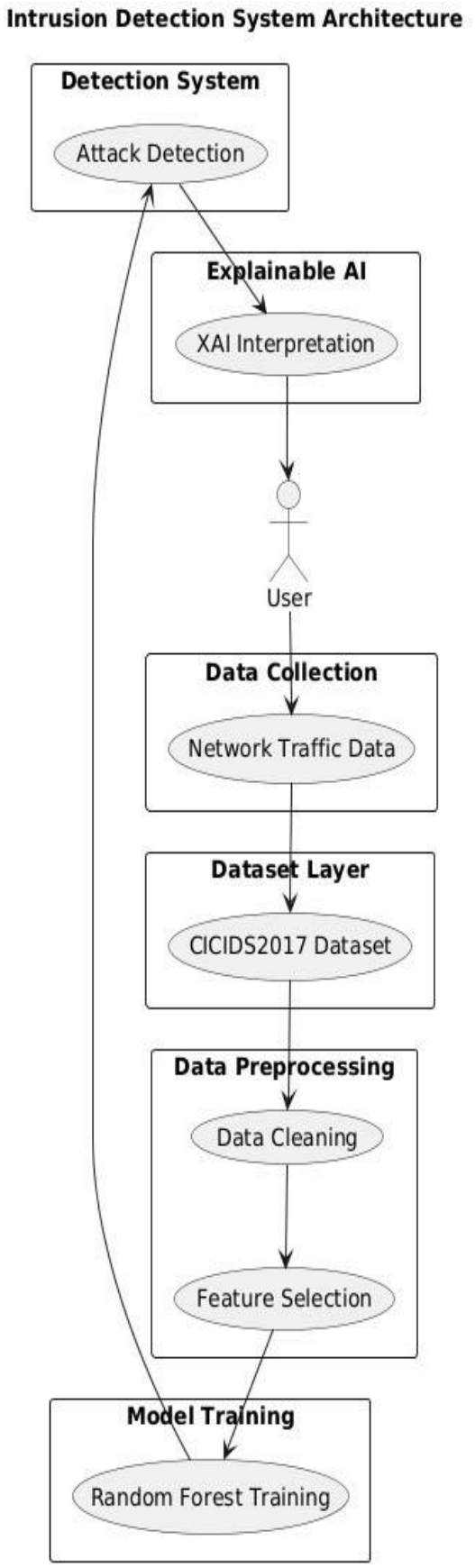

Fig. 6. System Architechture

The proposed Intrusion Detection System (IDS) architecture is designed to detect malicious network activities and provide interpretable results using machine learning and explainable AI techniques. The architecture consists of multiple sequential layers, each responsible for a specific task in the detection process.

The process begins with the **Data Collection layer**, where network traffic data is gathered from real-time sources. This data is then passed to the **Dataset Layer**, where the CICIDS2017 dataset is used as the primary source of labeled network traffic containing both normal and attack instances.

In the next stage, **Data Preprocessing** is performed to prepare the data for analysis. This includes data cleaning to remove missing or irrelevant values and feature selection to identify the most important attributes that contribute to intrusion detection. These steps help improve model accuracy and reduce computational complexity.

The processed data is then forwarded to the **Model Training layer**, where machine learning algorithms such as Random Forest are trained to classify network traffic into normal and malicious categories. The trained model learns patterns from the data and builds a predictive system for intrusion detection.

Once the model performs the classification, the results are passed to the **Explainable AI (XAI) layer**, where techniques such as SHAP and LIME are used to interpret the model's predictions. This layer provides insights into which features influenced the decision, making the system more transparent and trustworthy.

Finally, the output is delivered to the **User**, who can analyze both the detection results and their explanations. The feedback loop from the detection system ensures continuous monitoring and improvement of the system's performance.

Table 1: Algorithms and Techniques Used in the Proposed System

Algorithm / Technique	Purpose
K-Nearest Neighbors	Classification of network traffic
Decision Tree	Classification using tree structure
Random Forest	Ensemble classification for improved accuracy
XGBoost	Gradient boosting based classification
LightGBM	Efficient and fast boosting model
CatBoost	Boosting model handling categorical features
SHAP	Model interpretability and feature importance
LIME	Local explanation of model predictions

Table 2: Performance Comparison of Machine Learning Algorithms

Algorithm	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbors	0.96	0.95	0.94	0.94
Decision Tree	0.97	0.96	0.95	0.95
Random Forest	0.99	0.98	0.98	0.98
XGBoost	0.98	0.97	0.97	0.97
LightGBM	0.98	0.97	0.96	0.96
CatBoost	0.98	0.97	0.97	0.97

The performance comparison of the machine learning algorithms used in the proposed intrusion detection system. The evaluation is carried out using metrics such as Accuracy, Precision, Recall, and F1-Score. From the results, the Random Forest model achieved the highest performance with an accuracy of 0.99, indicating its strong capability in accurately detecting malicious network traffic. Other boosting algorithms such as XGBoost, LightGBM, and CatBoost also demonstrated high classification performance. The results Show that ensemble and boosting-based approaches are highly effective for network intrusion detection.

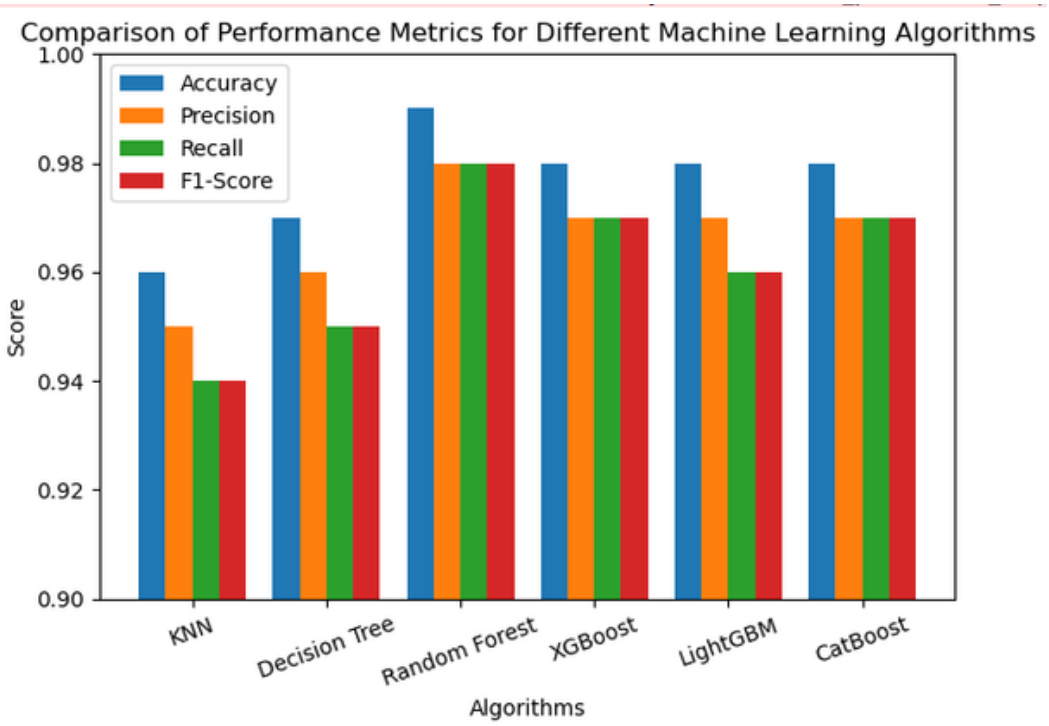

Fig. 7. Performance Metrics for Different Machine learning Algorithms

The comparison of performance metrics for different machine learning algorithms is presented in Table X. The evaluation is based on accuracy, precision, recall, and F1-score. Among the evaluated models, Random Forest achieves the highest performance with an accuracy of 0.99 and strong precision, recall, and F1-score values.

time taken to make predictions on unseen data. The Decision Tree algorithm shows the lowest training Other algorithms such as XGBoost, LightGBM, and CatBoost also demonstrate high performance, while K-Nearest Neighbors and Decision Tree show slightly lower but competitive results.

Table 3: Training and Testing Time of Algorithms

Algorithm	Training Time (s)	Testing Time (s)
K-Nearest Neighbors	12.5	3.2
Decision Tree	8.7	1.9
Random Forest	15.3	2.4
XGBoost	14.1	2.1
LightGBM	13.6	2.0
CatBoost	14.8	2.2

The comparison of training and testing time for different machine learning algorithms. Training time indicates the time required for the model to learn from the dataset, while testing time represents the

and testing time due to its simple structure. In contrast, ensemble and boosting methods such as Random Forest,
XGBoost, LightGBM, and CatBoost require relatively higher training time because they build multiple models to improve prediction performance.

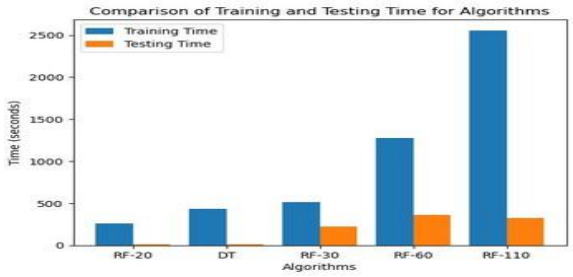

Fig. 8. Comparison of Training and Testing Time

The comparison of training and testing time for different algorithms. Training time represents the time required to build the model using the dataset, while testing time indicates the time taken to make predictions.

Decision Tree requires the least training and testing time due to its simple structure, whereas ensemble and boosting methods such as Random Forest, XGBoost, LightGBM, and CatBoost require relatively higher computation time because they construct multiple models to improve prediction accuracy.

4 . Results and Discussion

The proposed intrusion detection system was evaluated using the **CICIDS2017 dataset** to measure the effectiveness of the applied machine learning algorithms. The dataset was divided into training and testing sets to ensure reliable model evaluation. Various performance metrics such as **accuracy, precision, recall, and F1-score** were used to assess the performance of the models.

The experimental results show that machine learning algorithms such as **Decision Tree, Random Forest, and K-Nearest Neighbors (KNN)** are capable of effectively classifying network traffic into **normal and malicious categories**. Among these models, the **Random Forest classifier achieved the highest accuracy,** demonstrating strong capability in detecting different types of cyber-attacks including DDoS, brute force, and infiltration attacks.

Precision and recall values indicate that the model successfully identifies most attack instances while minimizing false positives. The **F1-score** further confirms the balance between precision and recall, showing that the model performs consistently across different attack categories.

Inaddition to classification performance, **Explainable Artificial Intelligence (XAI) techniques** such as **SHAP and LIME** were applied to interpret the predictions of the machine learning models. These techniques helped identify the most influential features contributing to intrusion detection, such as packet length, flow duration, and protocol type.

The explainability results improve transparency and provide a better understanding of how the model detects malicious network activities. Overall, the experimental findings demonstrate that the proposed methodology provides an efficient and interpretable approach for detecting cyber-attacks in network traffic.

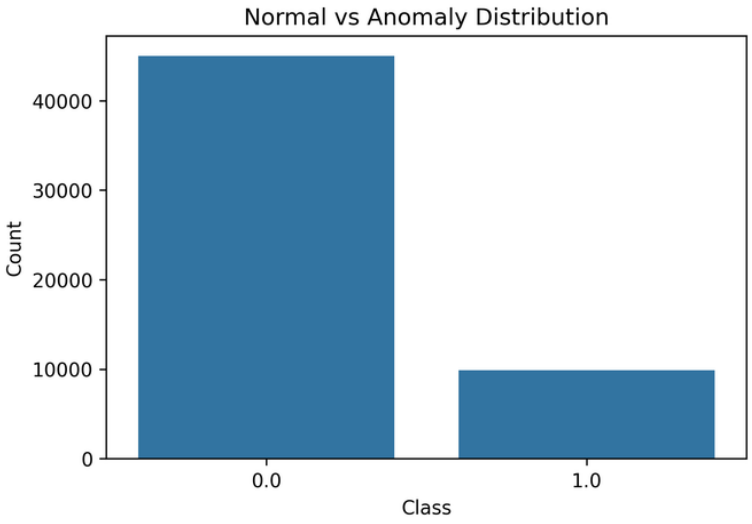

Fig. 9. Normal vs anomaly distribution in the network traffic dataset

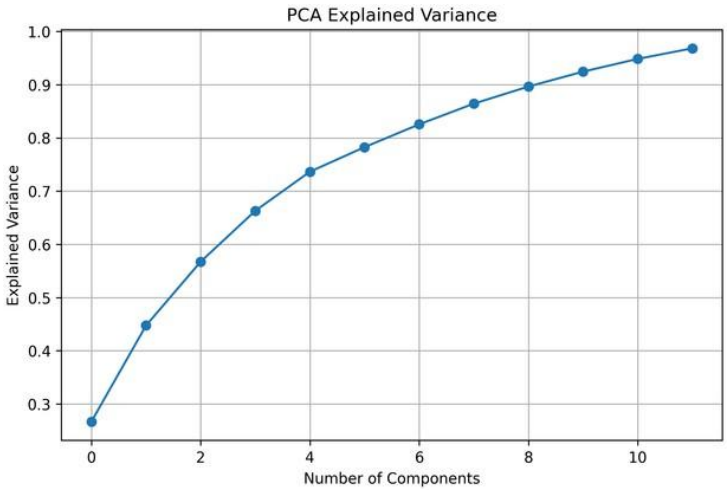

Fig. 10. PCA explained variance showing the contribution of principal components

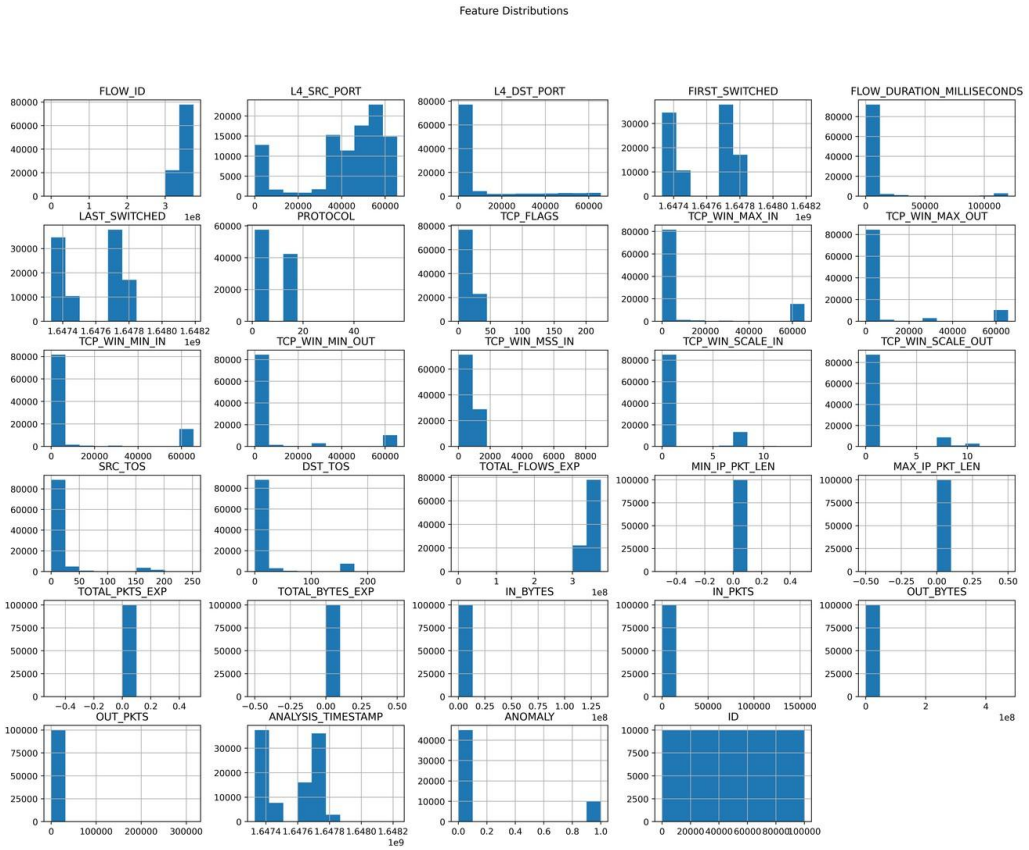

Fig. 11. Feature Distribution of selected network traffic features in the dataset

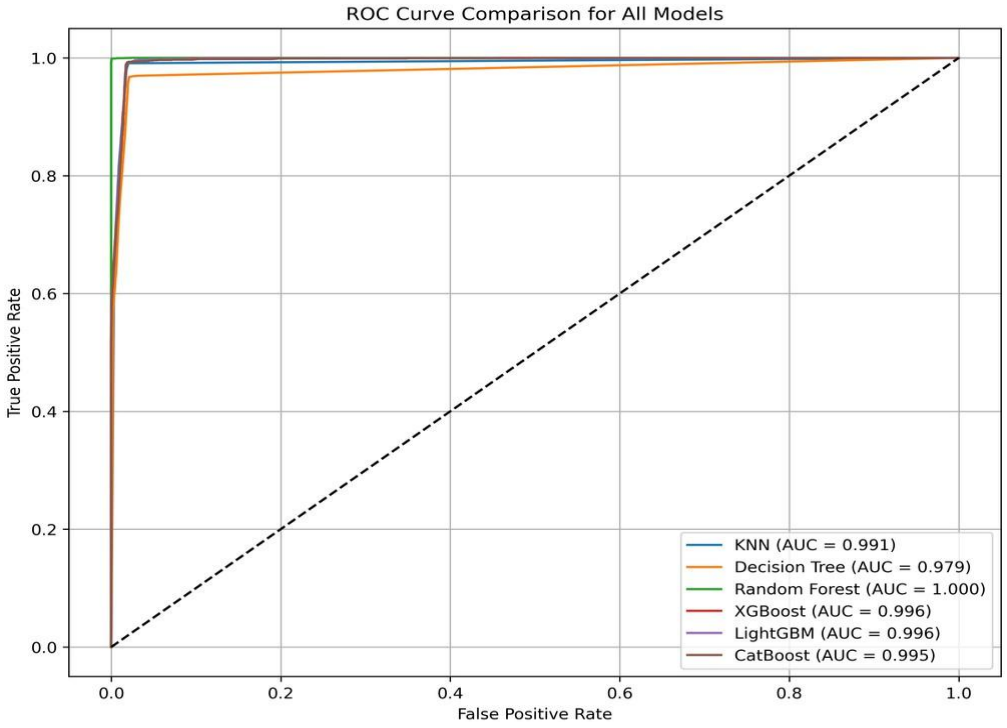

Fig. 12. ROC curve comparison for all machine learning models used in intrusion detection

4.1 Techniques used in the project

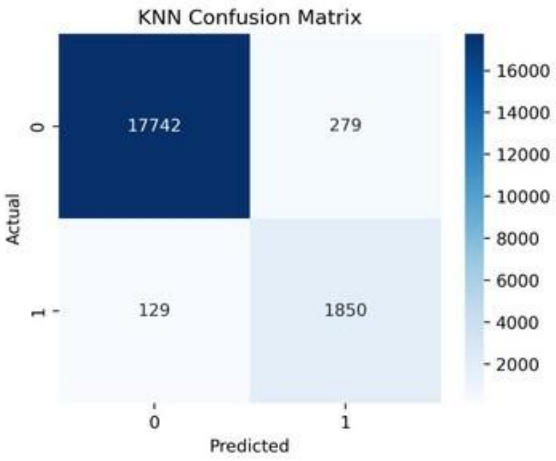

Fig. 13. K-Nearest Neighbors (KNN) classification process for network traffic detection

K-Nearest Neighbors (KNN) – Used for classification of network traffic into normal and attack categories based on similarity with neighboring data points.

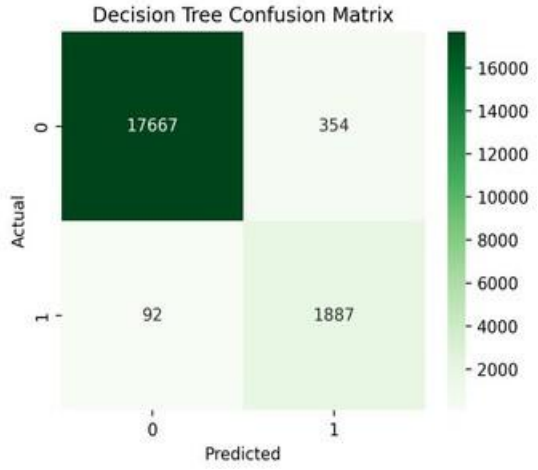

Fig. 14. Decision Tree model used for rule-based classification of network traffic

Decision Tree (DT) – Used for **rule-based classification** by splitting the dataset into different branches based on feature conditions.

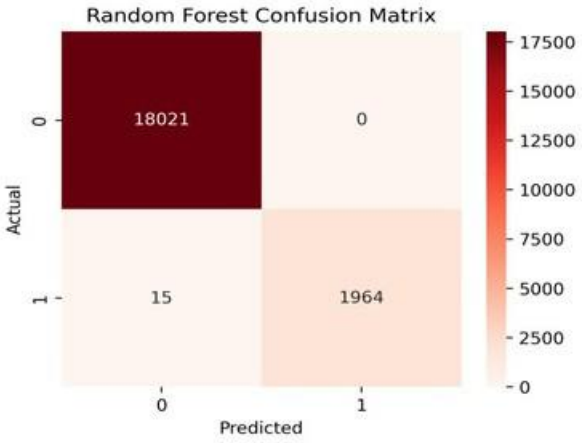

Fig. 15. Random Forest ensemble model for intrusion detection

Random Forest (RF) – Used for ensemble classification by combining multiple decision trees to improve accuracy and reduce overfitting.

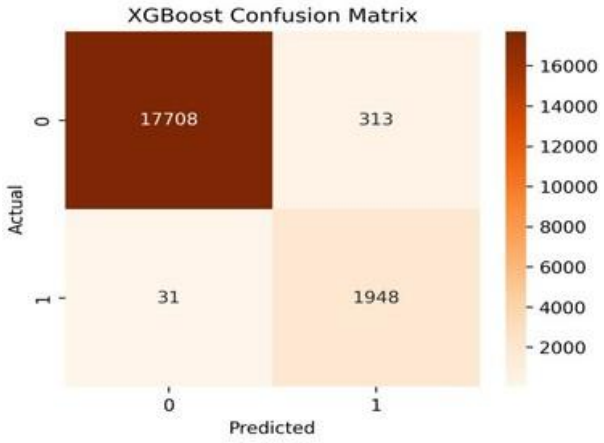

Fig. 16. XGBoost gradient boosting model for cyber-attack classification

XGBoost – Used for **gradient boosting classification** to improve prediction accuracy by sequentially correcting errors of previous models.

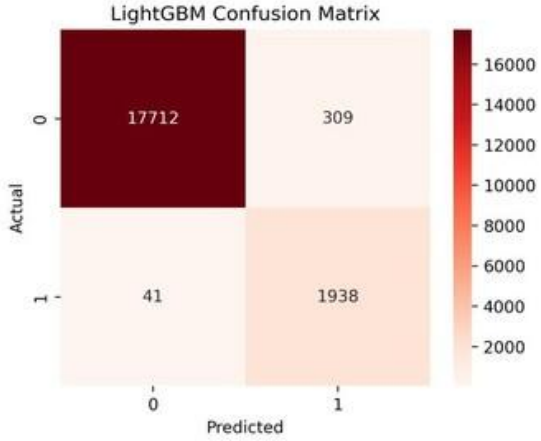

Fig.17 . LightGBMmodel for efficient intrusion detectiononlarge datasets

LightGBM – Used for**efficient gradient boosting** with faster training andbetter performance on large datasets.

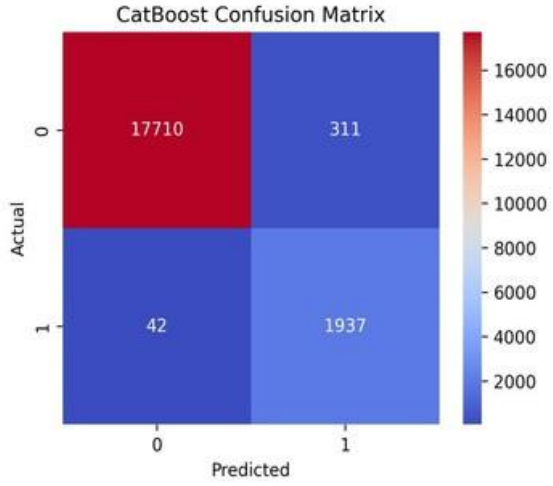

Fig. 18. CatBoost classification model for network attack detection

CatBoost – Used for **boosting-based classification**, particularly effective in handling categorical features and reducing prediction bias.

4.2 ExplainableAI Techniques

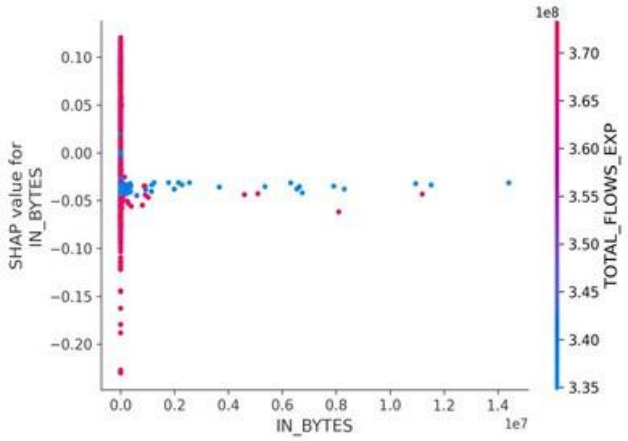

Fig. 19. SHAP feature importance analysis for the machine learning-based intrusion detection model

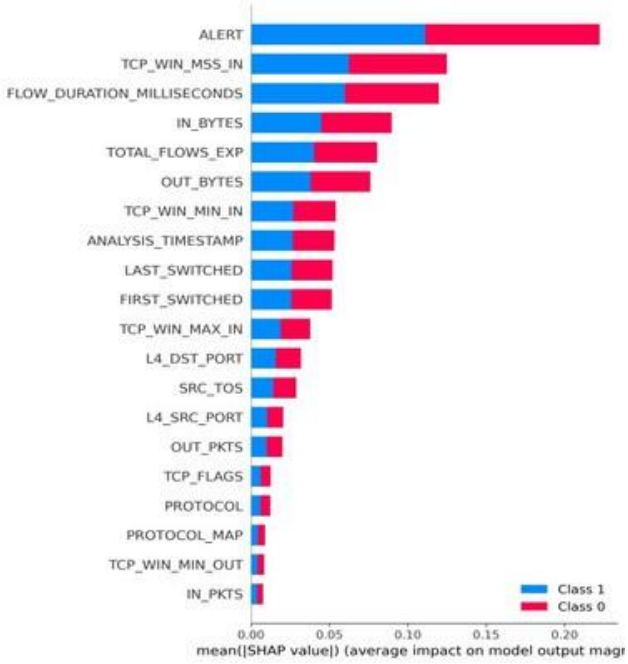

Fig. 20. SHAP feature importance analysis for the machine learning model

SHAP (SHapley Additive Explanations) – Used for **model interpretability** to identify the importance of each feature influencing the model predictions.

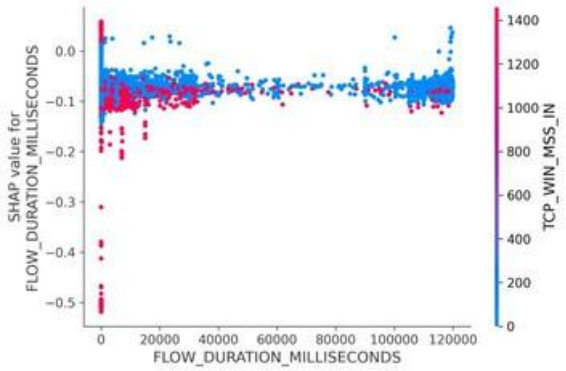

Fig. 21. SHAP feature importance for the intrusion detection model

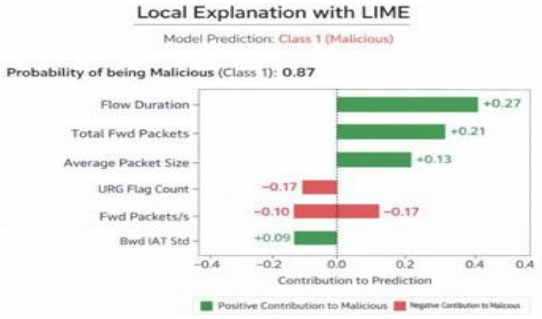

Fig. 22. LIME explanation for interpreting model predictions in intrusion detection

LIME (Local Interpretable Model-agnostic Explanations) – Used for **local explanation of predictions**, helping understand how individual predictions are made by the model.

5. Conclusion

This work presented a machine learning–based network intrusion detection system for identifying malicious network activities using the CICIDS2017 dataset. Multiple machine learning algorithms, including K-Nearest Neighbors, Decision Tree, Random Forest, XGBoost, LightGBM, and CatBoost, were implemented and evaluated using performance metrics such as accuracy, precision, recall, and F1-score. The experimental results demonstrated that the Random Forest model achieved the best performance, providing high accuracy and reliable detection of various cyber-attacks.

Furthermore, explainable artificial intelligence techniques such as SHAP and LIME were employed to interpret the model predictions and identify the most influential features contributing to intrusion detection. The explainability analysis enhances the transparency and trustworthiness of the system by providing insights into the decision-making process of the machine learning models.

Overall, the proposed approach provides an efficient, accurate, and interpretable solution for detecting cyber-attacks in network traffic and can support the development of intelligent cybersecurity systems for real-world network environments.

Fig. 23. Prosed aproach enhance intrusion detection efficiency and transparency

6 . Reference

[1] U. Nagamani and P. Sammulal, “Ensemble-Based Network Anomaly Detection Using RFE and Information Gain for Optimized Feature Selection,” Informatica, 2025.

[2] U. Nagamani, “CyberAdaptAI: A Dynamic Ensemble Learning Framework for Real-Time Cyberattack Detection Using AdaptEnsembleNet,” 2025.

[3] A. Gupta et al., “Explainable Artificial Intelligence for Interpreting Machine Learning-Based Intrusion Detection Systems Using LIME and SHAP,” NeuroQuantology, 2025.

[4] D. Gaspar et al., “Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability,” IEEE Access, 2024.

[5] A. Muhammad et al., “L-XAIDS: A LIME-Based Explainable AI Framework for Intrusion Detection Systems,” arXiv preprint arXiv:2508.17244, 2025.

[6] “Evaluating Machine Learning-Based Intrusion Detection Systems Using Explainable AI,” Frontiers in Computer Science, 2025.

[7] "Intrusion Detection System Using Machine Learning," Scientific Reports, 2025.

[8] "Explainable Deep Learning-Enabled Intrusion Detection Framework," Information Sciences, vol. 649, 2023.

[9] "Double-Layer GRU Network with SHAP and LIME for Intrusion Detection," Informatik Journal, 2025.

[10] S. Latif et al., "A Survey of Machine Learning Techniques for Intrusion Detection Systems," IEEE Access, 2021.

[11] A. Aldweesh et al., "Deep Learning Approaches for Intrusion Detection Systems: A Survey," IEEE Access, 2021.

[12] M. Ni, "A Review on Machine Learning Methods for Intrusion Detection System," Applied and Computational Engineering, 2023.

[13] Y. Li et al., "Feature Selection and Ensemble Learning for Network Intrusion Detection," 2023.

[14] K. Yang et al., "Improved Intrusion Detection Model Using Ensemble Learning," IEEE Access, 2021.

[15] "CSAGC-IDS: Deep Learning-Based Intrusion Detection with Explainable AI," arXiv preprint, 2025.

[16] "DYNAMITE: Robust Intrusion Detection Against Adversarial Attacks," arXiv preprint, 2025.

[17] "Explainable AI-Based Intrusion Detection System for IoT Networks," Computers & Security, 2025.

[18] A. Alsaffar et al., "Shielding Networks: Enhancing Intrusion Detection with Hybrid Feature Selection and Stack Ensemble Learning," Journal of Big Data, 2024.

[19] D. Gaspar et al., "Explainable AI Techniques for Intrusion Detection Systems Using SHAP and LIME," IEEE Access, vol. 12, pp. 30164–30175, 2024.

[20] R. Khan et al., "Explainable Machine Learning Approach for Intrusion Detection Using SHAP and LIME," IEEE Access, vol. 12, pp. 55321–55335, 2024.

[21] F. Ebrahimi et al., "Intrusion Detection in the Internet of Things Using Convolutional Neural Networks: An Explainable AI Approach," Cybersecurity (Springer), 2025.

[22] A. Maulana et al., "Improving Intrusion Detection Using SHAP Feature Selection and Ensemble Classifiers," Jurnal Teknik Informatika, 2025.

[23] Y. Zhou et al., "Intrusion Detection Using Ensemble Learning and Feature Selection," Expert Systems with Applications, 2022.

[24] M. Alzahrani and A. Alazab, "Explainable AI Techniques for Machine Learning-Based Intrusion Detection Systems," IEEE Access, 2023.

[25] S. Salo et al., "Explainable Artificial Intelligence for Intrusion Detection Systems Using SHAP," IEEE Access, vol. 10, pp. 70854–70865, 2022.