Research Article

Explainable Machine Learning Framework for Distributed Denial-of-Service (DDoS) Attack Detection using Comparative Evaluation and SHAP Analysis

Muhammad Fathur Riziq[1]*, Ichwan Nul Ichsan[2]
Department of Telecommunication System, Universitas Pendidikan Indonesia, Indonesia

Article Info

Article history:
Submitted November 7, 2025
Accepted December 18, 2025
Published December 22, 2025

ABSTRACT

The proliferation of Distributed Denial-of-Service (DDoS) attacks poses critical threats to network infrastructure, while conventional intrusion detection systems struggle to adapt to evolving attack patterns. Although ensemble learning methods achieve high accuracy on benchmark datasets, their opaque decision-making processes hinder deployment in Security Operations Centers (SOCs). To address this interpretability-performance gap, we propose an explainable machine learning framework integrating comparative benchmarking with quantitative interpretability analysis using the CIC-DDoS2019 dataset. Six supervised algorithms Decision Tree, Random Forest, XGBoost, LightGBM, Multilayer Perceptron, and Naïve Bayes were evaluated under standardized preprocessing protocols including random undersampling (50:50 class ratio), correlation-based feature selection (r > 0.9 threshold), and three-tier validation combining hold-out testing, train-validation splits, and 5-fold stratified cross-validation. LightGBM achieved optimal performance with 99.96% accuracy and F1-score of 0.9996, outperforming simple baselines by 0.35 percentage points while demonstrating superior computational efficiency. Beyond conventional performance metrics, we introduce the Feature Stability Score (FSS), a novel quantitative measure of SHAP-based feature importance consistency across validation folds. Spearman correlation analysis reveals a strong positive relationship between FSS and model robustness measured by cross-validation variance (ρ = 0.857, p = 0.014), establishing that stable feature attributions predict superior generalization. SHAP analysis identifies Flow Duration, Bwd Packet Length Mean, Fwd Packet Length Max, and Flow IAT Mean as dominant attack indicators. This integrated framework demonstrates that combining explainable AI with ensemble learning enables accurate, robust, and interpretable DDoS detection suitable for operational cybersecurity deployments.

Keywords:
DDoS detection;
machine learning;
Explainable AI (XAI);
SHAP;
Feature Stability Score (FSS).

Corresponding Author:
Muhammad Fathur Riziq,
Department of Telecommunication System, Universitas Pendidikan Indonesia,
Purwakarta Regional Campus, West Jawa, Indonesia.
Email: *fthrrzq@upi.edu

1. INTRODUCTION

The exponential growth of Internet services has fundamentally transformed network traffic patterns, creating new vulnerabilities that adversaries exploit through sophisticated cyberattacks [1]. Among these threats, Distributed Denial-of-Service (DDoS) attacks have emerged as particularly devastating, overwhelming network infrastructure and degrading service availability for legitimate users [2]. Traditional Intrusion Detection Systems (IDS), which depend on static signature-based mechanisms, prove inadequate against the continuously evolving nature of modern DDoS variants [3]. Consequently, researchers have pivoted toward adaptive Machine Learning (ML) approaches that autonomously learn attack signatures from network behavior patterns, eliminating the need for manual rule engineering [4].

Ensemble learning algorithms such as Random Forest, XGBoost, and LightGBM have demonstrated remarkable detection capabilities on standard datasets like the Canadian Institute for Cybersecurity DDoS 2019 (CIC-DDoS2019), achieving accuracy rates exceeding 99% [5]. Nevertheless, these models largely operate as "black boxes," rendering their decision-making processes opaque to security analysts despite their strong predictive performance [6]. This lack of transparency poses significant operational challenges, as Security Operations Centers (SOCs) require interpretable model decisions to prioritize incident responses, validate alerts, and justify actions to stakeholders. Consequently, without stable and explainable decision mechanisms, such high-performing models remain difficult to deploy reliably in real-world SOC environments [7].

To address this interpretability gap, Explainable Artificial Intelligence (XAI) has emerged as a critical approach for enhancing transparency in complex models [8]. SHapley Additive exPlanations (SHAP) provides quantitative feature importance values that explain how each network attribute contributes to classification decisions [9]. The integration of SHAP with ensemble algorithms enables security analysts to understand which traffic characteristics drive attack detection, facilitating model verification [10]. Recent studies demonstrate that SHAP-based feature selection not only improves interpretability but also enhances model generalization and stability across network traffic datasets [11]. These findings underscore the growing demand for intrusion detection systems that unify high accuracy, transparent interpretability, and computational efficiency [12].

Despite these advancements, existing research exhibits three fundamental limitations. First, many studies employing XAI techniques provide only qualitative visual interpretations without establishing quantitative linkages between interpretability and model performance [13]. Second, comparative evaluations often lack standardized experimental protocols, making it difficult to assess whether performance differences stem from algorithmic superiority or inconsistent preprocessing strategies [14]. Third, prior work has not systematically investigated whether interpretability stability measured by consistency of feature importance across validation folds—correlates with model robustness [15].

To address these gaps, this study proposes an explainable machine learning framework that integrates comparative benchmarking with quantitative interpretability analysis. Beyond conventional performance metrics, we introduce the Feature Stability Score (FSS), a novel metric measuring consistency of SHAP-based feature importance across cross-validation folds. By computing Spearman rank correlation between FSS and performance variance, we empirically test whether stable feature attributions predict superior generalization. This integrated framework ensures that the resulting detection system is accurate, computationally efficient, transparent, and trustworthy for operational deployment.

Alzu'bi et al. (2024) [6] proposed an Explainable Artificial Intelligence (XAI) framework for DDoS attack classification by integrating deep transfer learning and SHAP analysis. The hybrid BiLSTM–CNN model was developed to capture both temporal and spatial traffic characteristics using the CIC-DDoS2019 dataset, achieving a detection accuracy of 99.23%. The SHAP analysis enhanced interpretability but was limited to visual explanation without quantitative measurement of feature importance, leaving the consistency of model explanations unverified.

Abiramasundari and Ramaswamy (2024) [4] developed a supervised machine learning (ML) framework employing Random Forest, XGBoost, and LightGBM algorithms for DDoS attack detection. The models were evaluated using grid search cross-validation, which resulted in high accuracy and recall rates, with ensemble models outperforming single learners. However, the study did not incorporate any explainability analysis, thereby limiting its applicability in operational environments that require transparent and interpretable decision-making.

Ahmed et al. (2024) [5] introduced a hybrid LightGBM-based model integrated with SHAP interpretability to explain feature-level contributions in intrusion detection. The model achieved over 99% accuracy and identified Flow Duration and ACK Flag Count as the most influential features in attack classification. This research demonstrated the potential of combining explainability with ensemble learning methods; however, it lacked comparative evaluation across multiple algorithms, which restricts its generalizability.

Wei et al. (2024) [11] compared SHAP-based interpretability across CNN, BiLSTM, and LightGBM models using the CIC-IDS2017 and CIC-DDoS2019 datasets. The results indicated that Flow Bytes per Second and Bwd Packet Length Mean consistently influenced classification outcomes across models. Although the findings improved transparency in model interpretation, the study focused mainly on visualization without establishing a quantitative connection between explainability and model performance metrics.

Hernandez et al. (2025) [16] proposed a real-time DDoS detection framework for high-speed multivariate time-series networks using deep learning architectures that combine CNN and LSTM. The system effectively improved scalability and reduced latency, achieving robust performance in large-scale network environments. However, the model lacked interpretability, which made it difficult for security analysts to understand or justify classification decisions in real-time operations.

Becerra-Suarez et al. (2024) [17] conducted an extensive analysis on the role of data preprocessing and feature reduction in improving DDoS detection accuracy. By applying feature-selection techniques to multiple ML classifiers, the research revealed that eliminating redundant and highly correlated features significantly enhanced classification precision and reduced overfitting. Nevertheless, the study did not integrate explainable AI methods, leaving the relationship between feature optimization and interpretability unexplored.

The studies reviewed collectively demonstrate considerable advancements in machine learning-based approaches for Distributed Denial-of-Service (DDoS) detection. Nevertheless, several fundamental challenges remain unresolved. Alzu'bi et al. (2024) proposed an explainable deep learning framework that integrates BiLSTM and CNN with SHAP analysis, achieving high accuracy on the CICIoT2023 dataset but offering only qualitative interpretations without quantifying feature importance. Abiramasundari and Ramaswamy (2024) developed an ensemble-based model employing Random Forest, XGBoost, and LightGBM, yet their work concentrated solely on performance optimization and did not incorporate interpretability, limiting its practical applicability. Ahmed et al. (2024) introduced a hybrid bagging-boosting ensemble with SHAP-based feature selection capable of identifying influential traffic attributes; however, the study focused on a single dataset and lacked multi-model comparative evaluation, reducing the generalizability of its findings. Wei et al. (2024) compared SHAP-based explanations across CNN, BiLSTM, and LightGBM using the CIC-DDoS2019 dataset, but their analysis remained descriptive, lacking a quantitative linkage between explainability and model performance metrics. Hernandez et al. (2025) designed a deep learning architecture combining LSTM and TCN for real-time multivariate network traffic detection, yet the model was entirely opaque, offering no interpretability to support operational decision-making. Meanwhile, Becerra-Suarez et al. (2024) emphasized feature-selection techniques to enhance detection precision and reduce overfitting; however, the absence of explainable AI integration left the relationship between feature optimization and interpretability unexplored.

In contrast to these existing studies, the present research introduces a unified and reproducible framework that integrates both comparative performance benchmarking and quantitative explainability within a single experimental pipeline. Unlike Alzu'bi et al. (2024), which implemented SHAP solely for visualization, and Abiramasundari and Ramaswamy (2024), which prioritized model accuracy without addressing transparency, this study bridges both aspects by combining interpretability and performance evaluation under identical experimental conditions. Furthermore, while Ahmed et al. (2024) explored SHAP-based interpretability in a single-model setup and Wei et al. (2024) analyzed explainability across limited algorithms without statistical linkage to performance metrics, the present work expands the scope by applying standardized preprocessing, balancing, and validation across six supervised algorithms to ensure fairness and comparability. Additionally, unlike the approaches of Hernandez et al. (2025) and Becerra-Suarez et al. (2024), which emphasized scalability and feature selection respectively but omitted explainability analysis, the proposed framework quantitatively measures feature importance through SHAP and establishes a measurable connection between interpretability and model robustness.

Accordingly, this study aims to develop an explainable machine-learning framework for DDoS attack detection using the CIC-DDoS2019 dataset that bridges the gap between detection accuracy and interpretability. The objectives are to evaluate six supervised learning models under identical experimental settings, to establish a consistent and transparent benchmarking protocol, and to provide an analytical foundation for trustworthy cybersecurity applications. The outcomes of this study are anticipated to demonstrate the effectiveness of ensemble-based models in achieving high detection accuracy and computational efficiency while maintaining interpretability through SHAP analysis. This integrated approach contributes to the advancement of explainable AI in cybersecurity by offering a reproducible, transparent, and practically deployable detection framework suitable for real-world implementation.

2. RESEARCH METHODS

2.1 Framework

The proposed research framework establishes a systematic pipeline that integrates data preparation, model development, and explainable analysis into a single cohesive workflow. As illustrated in Figure 1, each stage is interdependent, ensuring that the detection process not only achieves high predictive accuracy but also provides transparent interpretability of model decisions [18]. This integrated design reflects the study's primary objective to construct a DDoS detection framework that balances performance efficiency and explainability within a reproducible experimental environmen.

The proposed framework is designed as a deterministic and stage-controlled experimental pipeline, in which each processing step is executed in a strictly defined order and contributes explicitly to either model robustness or decision interpretability, as illustrated in Figure 1. Unlike generic machine learning workflows, this framework enforces clear data access boundaries through a three-tier validation protocol, ensuring that training, validation, and final evaluation are fully isolated. The primary objective is to achieve high DDoS detection performance while preserving traceable and stable model explanations.

The pipeline begins with data preparation, where the CIC-DDoS2019 dataset (159,038 flows) is imported in its raw form. At this stage, duplicate records are removed, constant and non-informative features are discarded, and feature integrity is verified to ensure numerical consistency [19]. Importantly, no sampling or normalization is performed at this stage, and the dataset remains intact to avoid premature information loss [20].

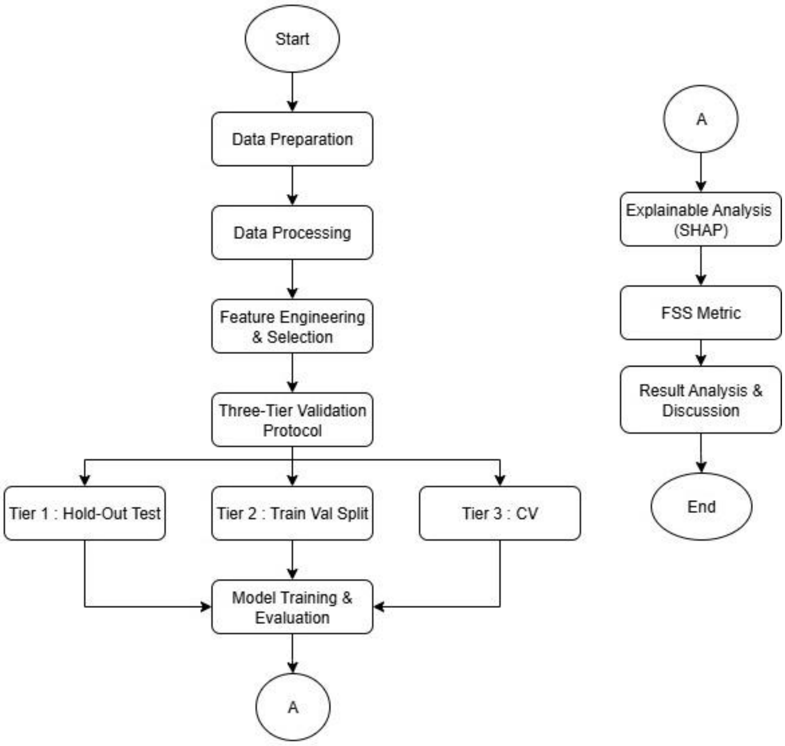
Start
Data Preparation
Data Processing
Feature Engineering & Selection
Three-Tier Validation Protocol
Tier 1 : Hold-Out Test
Tier 2 : Train Val Split
Tier 3 : CV
Model Training & Evaluation
A
A
Explainable Analysis (SHAP)
FSS Metric
Result Analysis & Discussion
End

Figure 1. Framework chart

Next, data processing establishes the experimental data boundaries. A stratified hold-out test set (24.5%) is reserved prior to any preprocessing, preserving the original class imbalance to reflect real-world network traffic conditions. The remaining samples form the training pool, to which random under sampling is applied exclusively to obtain a 50:50 benign-to-attack ratio [21]. This design prevents classifier bias while ensuring that the final test evaluation remains unbiased and operationally realistic. Feature normalization using StandardScaler is applied *only* to algorithms sensitive to feature magnitude (Naïve Bayes and Multilayer Perceptron), while tree-based models operate on raw numerical features.

Following data processing, feature engineering and selection is conducted on the balanced training data only. A Pearson correlation matrix is computed across all features, and a correlation heatmap is used to identify redundancy patterns. Features exhibiting strong inter-feature correlation ($r > 0.9$) are removed iteratively, prioritizing domain-relevant attributes such as Flow Duration over redundant packet-level metrics. This step explicitly targets multicollinearity, which is known to inflate feature importance and destabilize ensemble learners [22]. The resulting feature subset retains temporal and volumetric traffic characteristics critical for DDoS detection while reducing computational overhead and improving interpretability.

The refined dataset is then evaluated under the Three-Tier Validation Protocol, which governs all model learning and assessment activities.

- Tier 1 (Hold-Out Test Set) is fully isolated and accessed only once for final performance reporting.
- Tier 2 (Training–Validation Split) enables consistent model comparison and preliminary hyperparameter selection.
- Tier 3 (Stratified k-Fold Cross-Validation) is applied to the balanced training data to quantify model stability and generalization behavior.

Model evaluation and training are carried out within this protocol. Under the same settings, six supervised machine learning models that reflect various learning paradigms are trained. Grid search with five-fold cross-validation is used for hyperparameter optimization, with weighted F1-score serving as the selection criterion [23]. In order to provide fair comparison across algorithms and account for variance caused by data partitioning, model performance is assessed using accuracy, precision, recall, F1-score, and AUC.

After final evaluation, the highest-performing and most stable model is subjected to explainable analysis using SHapley Additive exPlanations (SHAP). SHAP values are computed to quantify feature-level contributions at both global and instance levels, enabling explicit interpretation of why specific traffic flows are classified as DDoS attacks. To assess the robustness of these explanations, the Feature Stability Score (FSS) metric is applied, measuring the consistency of feature importance across cross-validation folds. This final stage links predictive performance with explanation reliability, ensuring that the selected model is not only accurate but also interpretably stable.

2.2 Data Preparation

The CIC-DDoS2019 dataset was selected for three critical reasons: (1) it represents contemporary attack vectors (2017-2019) including application-layer and reflection-based DDoS, unlike older datasets (KDD'99); (2) it provides labeled real-world network traffic captured from testbed environments; and (3) it exhibits realistic class imbalance (Attack:Benign approx 65:35 in training), reflecting operational network conditions [24]. Although network intrusion datasets continue to evolve, CIC-DDoS2019 remains the most widely adopted benchmark specifically for DDoS detection research, as demonstrated by its adoption in recent peer-reviewed studies [4,5,11,16,17]. The attack mechanisms captured including SYN floods, UDP floods, and LDAP reflection represent fundamental DDoS techniques that persist in contemporary threat landscapes. This widespread adoption enables direct performance comparison with state-of-the-art machine learning approaches. We acknowledge that emerging attack patterns, particularly DDoS-as-a-Service platforms and adversarial ML evasion, are not represented in this benchmark. Future work will extend this framework to incorporate such evolving threats.

For this study, the data were systematically divided into a training subset (120,065 records) and a testing subset (38,973 records) to ensure fair model evaluation. Table 1 summarizes the overall dataset composition, including the number of records (159,038 total) and 78 features, whereas Table 2 presents the label distribution between benign and attack samples in each subset, providing a clear overview of the class balance prior to data preprocessing.

Critically, the initial training set contained 78,058 attack samples and 42,007 benign samples (65.0% vs. 35.0%). To address this imbalance, we applied random undersampling to achieve a 50:50 balance, reducing the majority class (Attack) to 42,007 samples. This approach was chosen to prevent model bias without introducing artificial data artifacts, while the separate test set remained intentionally unbalanced (28,126 attacks, 10,847 benign) to evaluate model performance under realistic operational conditions."

Table 1. Data collection summary

Dataset Split	Total Records	Number of Features	Benign Samples	Attack Samples	Missing Values
Training Set	120,065	78	42,007	78,058	0
Testing Set	38,973	78	10,847	28,126	0

Table 2. Label distribution per dataset

Attack Type / Label	Training Samples	Testing Samples
SYN Flood	48,840	533
UDP Flood	18,090	10,420
MSSQL	8,523	6,212
LDAP	1,906	1,440
NetBIOS	644	598
UDP-Lag	55	8,872
WebDDoS	–	51
Benign	**42,007**	**10,847**
Total	**120,065**	**38,973**

The selection of this subset was made to maintain temporal consistency while minimizing potential data leakage between the training and testing sets. A distinctive feature of this dataset lies in its level of diversity. In addition to normal network traffic, it also represents multiple attack types with an imbalanced distribution between training and testing data [4]. As shown in Table 2, SYN Flood attacks dominate the training portion, whereas UDP Flood attacks are more prevalent in the testing data. This imbalance is not regarded as a limitation but rather as a reflection of real-world network conditions, where attack patterns can dynamically change over time [17]. Such characteristics make this dataset ideal for evaluating the generalization capability and adaptability of the model. The fluctuating nature of the data further emphasizes the importance of adopting an Explainable Artificial Intelligence (XAI) approach [23]. Through SHAP-based analysis, it becomes possible to assess the extent to which the model truly recognizes meaningful attack patterns rather than merely exploiting common features that arise due to uneven data distribution [6].

2.3 Data Processing

The data processing phase was performed to ensure that the dataset was in optimal condition for model training. This stage included cleaning, transformation, and balancing procedures to enhance model stability and generalization. Each step was systematically implemented in Python with a reproducibility control set to *random_state=42*.

The processing workflow consisted of the following steps:

- Duplicate and Constant Feature Removal: All duplicate records and features with a single unique value were removed to eliminate redundancy and improve computational efficiency.
- Correlation Filtering: Features with a Pearson correlation coefficient greater than 0.9 were excluded to reduce multicollinearity effects. This threshold selection is consistent with prior intrusion detection studies, which report that excessive feature correlation ($r > 0.9$) can adversely affect model stability and interpretability. The correlation structure among features is further illustrated in the correlation heatmap shown in Figure 2 and is discussed in detail in the subsequent Feature Engineering and Selection section.
- Label Encoding: The categorical target attribute “Label” was converted into a binary numerical form, where Benign = 1 and Attack = 0.
- Data Balancing: To mitigate the class imbalance problem inherent in the CIC-DDoS2019 dataset, random undersampling was applied to achieve a balanced ratio (50:50) between benign and attack samples. This method was selected for its simplicity and effectiveness in mitigating class imbalance without introducing synthetic traffic patterns that may distort real-world DDoS characteristics.

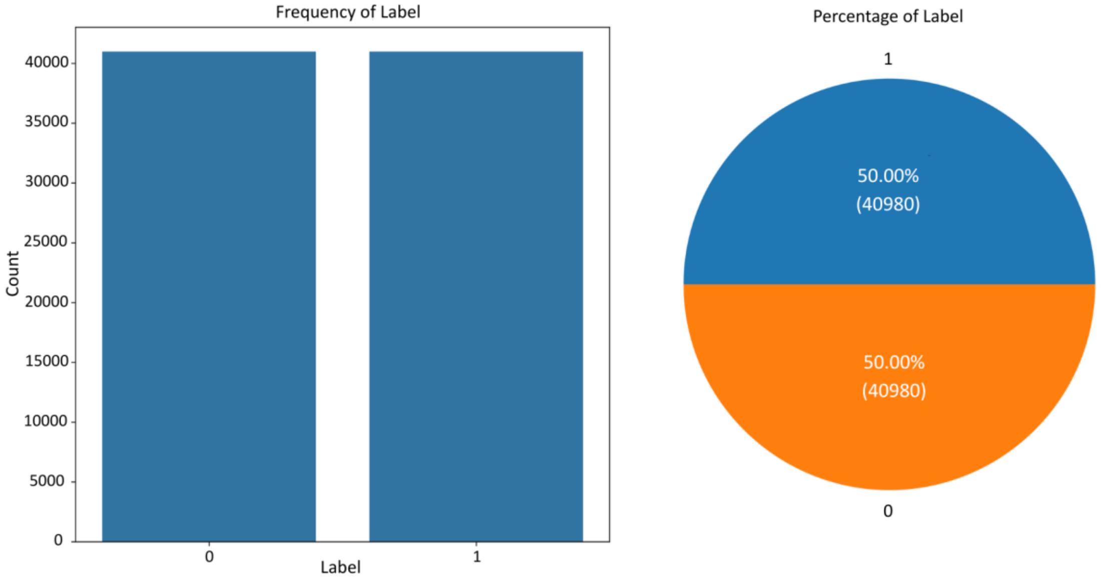

Figure 2. Label distribution balanced

Figure 2 shows the Label Distribution After Random Sampling. The balanced distribution shows equal representation of Benign (1) and Attack (0) classes (n = 42,007 each), reducing class imbalance during model training. The test set intentionally remained imbalanced to reflect real-world traffic conditions.

- Feature Scaling: StandardScaler normalization was selectively applied to models sensitive to input magnitude, such as Naive Bayes (NB) and Multilayer Perceptron (MLP), while tree-based models (Decision Tree, Random Forest, XGBoost, and LightGBM) were trained on unscaled numerical data.
- Data Splitting: To maintain the original class distribution, the dataset was divided into training and testing subsets using a stratified 70:30 split. In order to prevent information leaking, all model selection, hyperparameter tuning, and robustness studies were carried out using stratified cross-validation on the training data, with the hold-out test set left solely for final evaluation.

The configuration parameters applied during preprocessing are summarized in Table 3, ensuring full transparency and reproducibility for subsequent feature selection and model training stages.

Table 3. Data processing configuration parameters

Parameter	Value / Description
Correlation Threshold	$r > 0.9$ (highly correlated features removed)
Balancing Ratio	50:50 (Benign : Attack) using Random Undersampling
Train–Test Split	70% Training – 30% Testing (Stratified)
Scaling Method	StandardScaler (applied to NB and MLP models only)
Reproducibility Seed	random_state = 42 (fixed for all experiments)

These preprocessing configurations ensured that the dataset was cleaned, standardized, balanced, and ready for model training, thereby improving the reliability and reproducibility of the subsequent experimental workflow.

2.4 Feature Engineering and Selection

At this stage, a comprehensive correlation analysis was conducted to evaluate informational redundancy within the dataset and ensure that only statistically independent variables were retained for model training. Figure 3 presents the correlation heatmap, revealing several clusters of strongly correlated features, particularly within the Flow, Packet Length, and Inter-Arrival Time (IAT) groups. Features exhibiting Pearson correlation coefficients greater than 0.9 were systematically removed to minimize redundancy and prevent multicollinearity, which can distort the learning process by inflating feature importance and reducing model stability in ensemble classifiers [11].

The feature reduction process was conducted iteratively: beginning with the initial 78 features in the CIC-DDoS2019 dataset, correlation analysis identified 23 highly correlated feature pairs (r > 0.9). Following removal prioritizing domain-relevant attributes (retaining Flow Duration over redundant IAT metrics based on prior DDoS research) [24], 55 features were retained for model training. This threshold selection is consistent with prior intrusion detection studies, which report that excessive feature correlation can adversely affect model stability and interpretability [11].

The refined feature subset comprises attributes with moderate correlation to the target class, including Flow Duration, Bwd Packet Length Mean, Fwd Packet Length Max, and Flow IAT Mean, which effectively capture temporal and volumetric behaviors associated with DDoS activity [25]. Consistent with findings from Ahmed et al. (2024) [5] and Wei et al. (2023) [11], these features represent meaningful indicators of network-layer DDoS behavior. This curated feature set preserves critical statistical characteristics of network traffic while maintaining computational efficiency and interpretability for subsequent model training, cross-validation benchmarking, and SHAP-based explainability analysis.

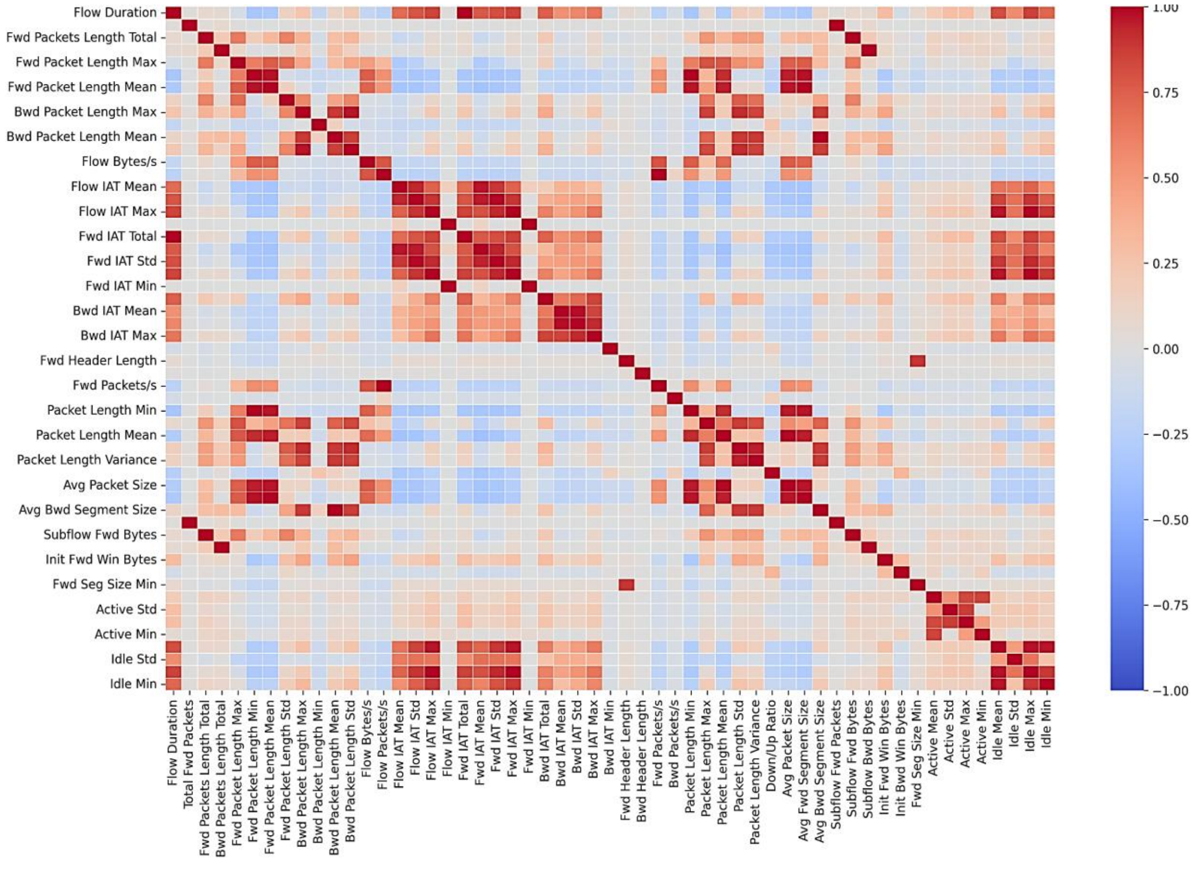

Figure 3. Correlation heatmap

2.5 Three-Tier Validation Protocol

To ensure robust and unbiased model evaluation, this study employs a three-tier validation strategy that separates model comparison, stability assessment, and final performance evaluation into distinct stages. This hierarchical approach prevents data leakage, mitigates overfitting, and provides reliable estimates of real-world generalization performance. The experimental workflow follows a standard training–validation–testing sequence, while the validation tiers are conceptually organized according to their evaluation roles as illustrated in Figure 4.

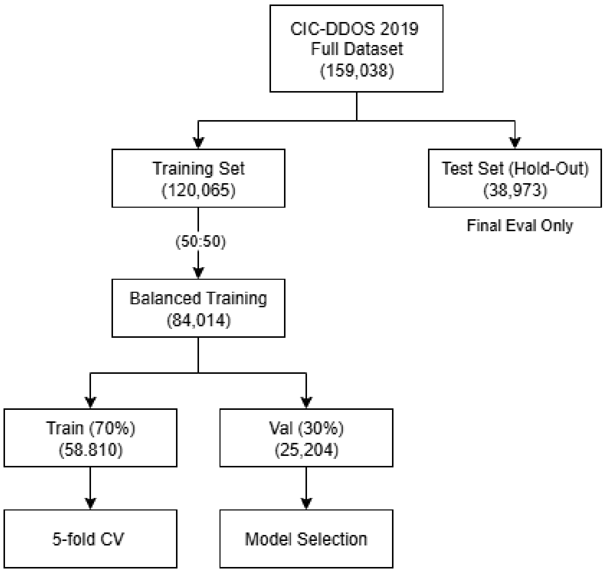

Figure 4. Three tier validation workflow diagram

2.5.1 Data Partitioning and Preparation

The experimental dataset preparation proceeded in the following sequence:

- Initial Dataset: The CIC-DDoS2019 dataset contained 159,038 samples with 78 features.
- Hold-Out Test Set Reservation: A stratified hold-out test set comprising 38,973 samples (24.5%) was reserved before any preprocessing. This set intentionally retained the original class imbalance (28,126 attack vs. 10,847 benign samples) to reflect realistic operational conditions and remained completely isolated from all subsequent training, validation, and tuning processes.
- Training Set Balancing: The remaining 120,065 samples were designated as the training pool. Random undersampling was applied exclusively to this training pool to achieve a balanced 50:50 class distribution, resulting in 84,014 samples (42,007 attack + 42,007 benign). This approach was selected over synthetic oversampling techniques (SMOTE) to prevent introducing artificial traffic patterns that may not reflect real-world DDoS characteristics. Class weighting was not employed to maintain direct comparability across diverse model architectures (tree-based, probabilistic, and neural networks).

2.5.2 Validation Tiers

Tier 1: Hold-Out Test Set (Final Evaluation)

The reserved hold-out test set (n = 38,973) serves as the ultimate benchmark for final model performance evaluation. This set remained completely untouched throughout all training, validation, and hyperparameter tuning phases, ensuring an unbiased estimate of real-world performance and preventing information leakage [26]. All models were evaluated on this independent test set only once, after all development decisions were finalized, to report the definitive classification metrics presented in Section 3.

Tier 2: Training–Validation Split (Model Comparison)

The balanced training dataset (n = 84,014) was split into training (70%, n = 58,810) and validation (30%, n = 25,204) subsets using stratified sampling to preserve the 50:50 class distribution. This validation subset was used for preliminary model comparison and hyperparameter selection based on weighted F1-score. Specifically, initial hyperparameter configurations for each algorithm were evaluated on this fixed validation set to identify the most promising model architecture before proceeding to cross-validation [27]. This tier enables fair and consistent comparison across candidate algorithms (detailed in Section 2.6) under identical experimental conditions.

Tier 3: Cross-Validation (Stability Assessment)

The whole balanced training dataset (n = 84,014) was subjected to 5-fold stratified cross-validation after model selection in Tier 2 in order to evaluate the stability and robustness of the model. The dataset was divided into five equal subsets, and each fold maintained the 50:50 class distribution. One subset was held back for validation and four subsets were used for training in each iteration. This approach was performed five times so that each subset was used as the validation set exactly once [28]. To generate trustworthy estimations of model generalization behavior, performance measures (accuracy, precision, recall, and F1-score) were calculated for each fold and averaged. The standard deviation of cross-validation scores served as the robustness indicator for subsequent Feature Stability Score (FSS) correlation analysis, with lower variance indicating higher robustness. Figure 4 shows a three-level validation workflow diagram.

2.6 Model Training & Evaluation

Six supervised Machine Learning (ML) algorithms were trained and evaluated under uniform experimental conditions to construct a robust Distributed Denial-of-Service (DDoS) detection framework. The selected models represent diverse learning paradigms, including interpretable, ensemble-based, probabilistic, and neural approaches. Specifically, Decision Tree (DT) served as a baseline rule-based classifier; Random Forest (RF) extended this by aggregating multiple trees to reduce variance and improve generalization; Naïve Bayes (NB) represented a lightweight probabilistic model assuming feature independence; Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine (LightGBM) embodied advanced gradient-boosting frameworks optimized for scalability and computational efficiency; while Multilayer Perceptron (MLP) captured complex non-linear feature relationships through backpropagation-based learning [29]. This configuration enabled comprehensive comparison across interpretable, ensemble, probabilistic, and neural paradigms within a unified experimental pipeline [17].

To establish a performance floor and ensure that observed results are non-trivial, three simple baseline classifiers Logistic Regression, Random Classifier, and Majority Class Classifier were included for comparative context. These baselines demonstrate that the proposed models provide meaningful gains beyond trivial or chance-level classification.

Each model was trained using the balanced training subset (50:50 benign–attack ratio) and evaluated on the independent testing subset, both derived from the CIC-DDoS2019 dataset following the three-tier validation protocol described in Section 2.5. Feature scaling was selectively applied using StandardScaler normalization, particularly for models sensitive to input magnitude such as NB and MLP, while tree-based algorithms (DT, RF, XGBoost, LightGBM) utilized raw numerical features. Hyperparameter tuning was performed using GridSearchCV with internal 5-fold cross-validation on the Tier 2 training subset (n = 58,810), while the Tier 2 validation subset (n = 25,204) was reserved for comparing optimized models and selecting the best candidate for subsequent Tier 3 stability assessment and Tier 1 final evaluation. To ensure fairness and reproducibility, all experiments were executed under identical conditions with a fixed random seed (random_state=42), following the configurations summarized in Table 5.

The trained models were evaluated under identical experimental conditions to ensure fairness and comparability. Performance metrics including accuracy, precision, recall, F1-score, and AUC were computed on both the validation set (Tier 2) for model comparison and the hold-out test set (Tier 1) for final evaluation. Cross-validation results (Tier 3) provided stability estimates, with performance variance serving as the robustness indicator for subsequent Feature Stability Score (FSS) correlation analysis. The highest-performing model, determined by weighted F1-score and supported by AUC, was subsequently selected for comprehensive explainable analysis using SHAP to interpret feature-level contributions.

The detailed hyperparameter configurations for each ML model and the overall experimental simulation setup are summarized in Table 4 and Table 5, providing a comprehensive overview of the experimental design and ensuring full transparency for reproducibility.

Table 4. Model configuration and hyperparameter settings

Model	Hyperparameters	Description/Purpose
Decision Tree (DT)	max_depth=10, random_state=42	Baseline interpretable model that classifies traffic using hierarchical rule-based partitioning.
Random Forest (RF)	n_estimators=100, max_depth=10, random_state=42, n_jobs=-1	Ensemble model combining multiple Ensemble of multiple decision trees employing bootstrap aggregation to enhance generalization and reduce variance.
Naive Bayes (NB)	priors=None, var_smoothing=1e−9	Probabilistic classifier based on Bayes' theorem, assuming feature independence for lightweight DDoS classification.
Extreme Gradient Boosting (XGB)	n_estimators=100, learning_rate=0.1, use_label_encoder=False, eval_metric='logloss', random_state=42	Regularized gradient-boosting model enhancing performance through pruning and parallel learning.
Light Gradient Boosting Machine (LGBM)	n_estimators=100, learning_rate=0.1, random_state=42, verbose=-1	Leaf-wise boosting algorithm optimized for high-speed training and low memory usage.
Multilayer Perceptron (MLP)	hidden_layer_sizes=(100, 50), max_iter=500, activation='relu', solver='adam', random_state=42	Feed-forward neural network trained via backpropagation to capture complex non-linear traffic patterns.

Table 5. Evaluation and simulation parameters

Configuration	Value / Description
Dataset	CIC-DDoS2019
Hold-Out Test Set (Tier 1)	38,973 samples (24.5% of original, kept imbalanced: 28,126 attack + 10,847 benign)
Balanced Training Pool	84,014 samples (42,007 attack + 42,007 benign, after random undersampling)
Tier 2 Split (Model Comparison)	70% Training (58,810) – 30% Validation (25,204)
Cross-Validation (Tier 3)	5-Fold Stratified
Balancing Method	Random undersampling (50:50 benign–attack ratio)
Random Seed	42 (fixed for all splitting and training operations)
Scoring Metric	F1-weighted
Correlation Threshold	r > 0.9
Scaling Method	StandardScaler for NB and MLP; raw inputs for tree-based models
Environment	Python 3.10 (Scikit-learn 1.5.2, XGBoost 2.0.3, LightGBM 4.5.0)
Runtime Platform	Google Colab (GPU-accelerated environment)

2.7 Evaluation

In this study, model performance was evaluated using the confusion matrix, which provides four key outcomes: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Based on these values, four standard performance metrics Accuracy, Precision, Recall, and F1-Score were calculated to assess the classification quality of the proposed model [29]. Accuracy measures the proportion of correctly classified instances among all samples, as shown in Equation (1).

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

Precision quantifies how many of the predicted positive instances are actually positive, thereby reflecting the model's reliability in detecting attack traffic, as expressed in Equation (2).

$$Precision = \frac{TP}{(TP + FP)} \tag{2}$$

Recall, sometimes referred to as True Positive Rate or Sensitivity, assesses how well the model can detect all real positive occurrences, as formulated in Equation (3).

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

The F1-score offers a single, fair indicator of the classifier's overall performance since it is the harmonic mean of precision and recall, particularly useful for imbalanced datasets. It is calculated using Equation (4).

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{4}$$

2.8 Interpretability Robustness Linkage

In line with reviewer recommendations, this study aims to establish a quantitative link between interpretability and model robustness by analyzing the stability of SHAP based feature attributions across validation folds. To quantitatively assess the relationship between interpretability and model robustness, we introduce two novel metrics.

2.8.1 Feature Stability Score (FSS) - Measuring Interpretability Consistency

Measuring interpretability consistency as in Equation (5).

$$FSS = 1 - \frac{\sigma SHAP}{\mu SHAP} \tag{5}$$

Where, σSHAP represents the standard deviation of SHAP values across k-fold cross-validation, and µSHAP denotes the mean SHAP importance. A higher FSS indicates more consistent feature attribution, suggesting that the model learns generalizable patterns rather than fold-specific artifacts.

2.8.2 Model Robustness Quantification

Model robustness is quantified using the standard deviation of the F1-score obtained from cross validation, where lower variance indicates higher robustness:

- Robustness Indicator: CV F1-Score Standard Deviation (lower is better). Based on empirical observations across all evaluated models, models with a CV standard deviation below 0.0002 are considered highly robust, indicating stable performance across different data partitions and strong generalization capability.
- Hypothesis: We hypothesize that models with higher FSS values (indicating stable feature attribution) exhibit higher robustness, reflected by lower cross-validation performance variance. This relationship establishes a quantitative link between interpretability stability and generalization.
- Statistical Validation: To test this hypothesis, Spearman rank correlation was computed between FSS and the robustness indicator across all tree-based models. A statistically significant correlation would confirm that interpretability stability quantitatively predicts model robustness on unseen data.

2.9 Explainable Analysis (SHAP)

The best-performing model's predictions were interpreted using the SHapley Additive exPlanations (SHAP) framework, which also improved decision-making transparency. Each input feature's contribution to the model's output is measured using SHAP, a model-agnostic interpretability technique [5]. This approach is based on cooperative game theory, in which every feature is viewed as a player that contributes to the final prediction result. The prediction function as in Equation (6).

$$f(x) = \emptyset_0 \ \textstyle\sum_{i=1}^{M} \emptyset_{xi} \tag{6}$$

SHAP was employed to quantify the contribution of each feature toward the model's output. Formally, $\emptyset_0$ represents the baseline value corresponding to the average model prediction without feature information, and $\emptyset_{xi}$ denotes the SHAP value for the *i-th feature*, reflecting its individual contribution to the prediction output [9]. SHAP values were computed using the TreeExplainer algorithm on the test dataset obtained from 5-fold stratified cross-validation, and all folds were aggregated to ensure stable interpretability results. SHAP was selected over alternative interpretability techniques such as LIME and DeepLIFT because it guarantees consistency and local accuracy, making it particularly effective for ensemble-based learning models commonly applied in cybersecurity research [30].

Global and local interpretability levels were used for the explainable analysis. SHAP summary plots were created in the global interpretability analysis to show the most important characteristics throughout the whole dataset. The main factors in distinguishing between normal and attack traffic were found to be attributes like Flow Duration, Bwd Packet Length Mean, Fwd Packet Length Max, and Flow IAT Mean [5]. SHAP waterfall charts were used to explain individual sample predictions for local interpretability, demonstrating how each feature value affected the model's confidence in identifying a record as normal or attack [31].

3. RESULTS AND DISCUSSION

3.1 Research Results

This section presents the empirical findings from a comparative evaluation of six machine learning algorithms for DDoS attack detection using the CIC-DDoS2019 dataset. The results summarize predictive performance and computational characteristics across different model families. Across all evaluation protocols, tree-based ensemble models consistently achieve higher classification performance and computational efficiency compared to non-ensemble approaches in Ref. [4][5][11][17][25].

3.1.1 Model Performance Evaluation

To establish a performance floor and ensure that the observed results are not trivial, three simple baseline classifiers Logistic Regression, Random Classifier, and Majority Class Classifier—were included solely for contextual comparison. This comparison serves only to establish a lower-bound reference, demonstrating that the proposed models provide meaningful gains beyond trivial or chance-level classification. The experimental results from the nine evaluated models are fully detailed in Table 6. From these results, LightGBM clearly emerges as the overall leader, achieving near-perfect scores across all metrics 0.999634 for Accuracy, Precision, Recall, and F1-Score, alongside an exceptional AUC of 0.999983. These scores firmly indicate that the model makes almost no errors in distinguishing between normal traffic and DDoS attacks.

The performance gain of LightGBM over the strongest baseline (Logistic Regression) is +0.0035 in F1-score (0.9996 vs. 0.9962), corresponding to a 0.35% relative improvement [1][18][29]. On the hold-out test set of 38,973 samples, this difference translates to approximately 135 fewer misclassifications (25 vs. ~160 errors). Although Logistic Regression achieves relatively high classification accuracy, its performance variability under cross-validation is higher than that of ensemble-based models, and it is therefore included solely as a baseline reference in this comparative evaluation.

Table 6. Comparative performance of machine learning models for DDoS Detection

Model	Accuracy	Precision	Recall	F1-Score	AUC
Simple Baselines					
Logistic Regression	0.996176	0.996177	0.996176	0.996176	0.998497
Random Classifier	0.501464	0.501464	0.501464	0.501464	0.5
Majority Class	0.5	0.25	0.5	0.333333	0.5
Evaluated ML Models					
LightGBM	0.999634	0.999634	0.999634	0.999634	0.999958
XGBoost	0.999512	0.999512	0.999512	0.999512	0.999968
Random Forest	0.999399	0.999391	0.999399	0.999399	0.999956
MLP	0.998699	0.998700	0.998699	0.998699	0.999729
Decision Tree	0.998048	0.998051	0.998048	0.998048	0.999331
Naive Bayes	0.984098	0.984326	0.984098	0.984096	0.993305

The comparative performance of ensemble models and baseline classifiers is illustrated in Figure 5. Tree-based ensemble models, including LightGBM, XGBoost, and Random Forest, consistently achieve higher F1-scores than the baseline approaches. In particular, the F1-score of LightGBM (0.9996) exceeds those of the Random Classifier (0.5014) and the Majority Class Classifier (0.3333), indicating a substantial performance margin. Performance differences among the top three ensemble models are relatively small.

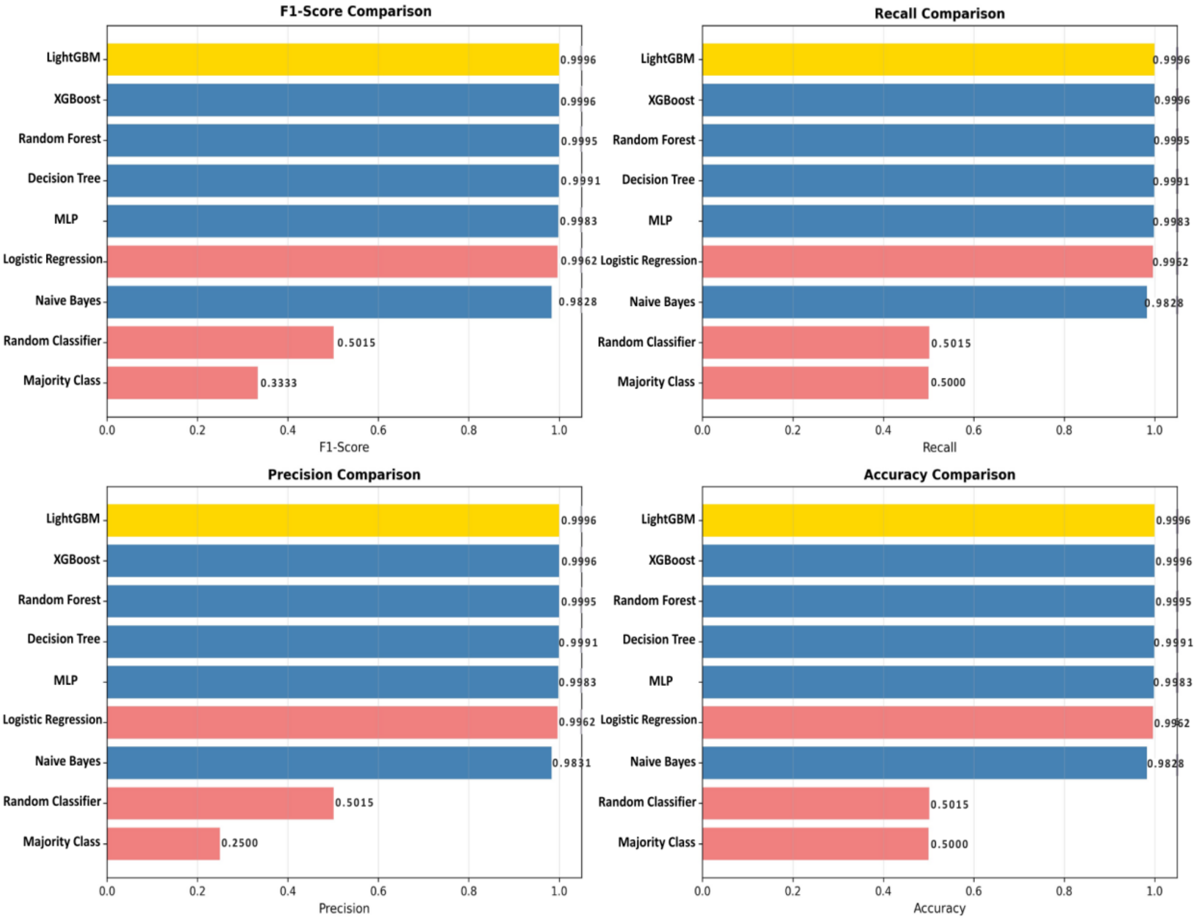

Figure 5. Model comparison with baselines

In addition to predictive performance, LightGBM exhibits lower training time and memory usage compared to XGBoost [14][17][22][25], as observed during experimental evaluation. Figure 6 presents the confusion matrix for LightGBM on the hold-out test set, showing 14 false positives and 11 false negatives out of 38,973 samples. This corresponds to an overall error rate of approximately 0.064%, including a false-negative rate of 0.028%.

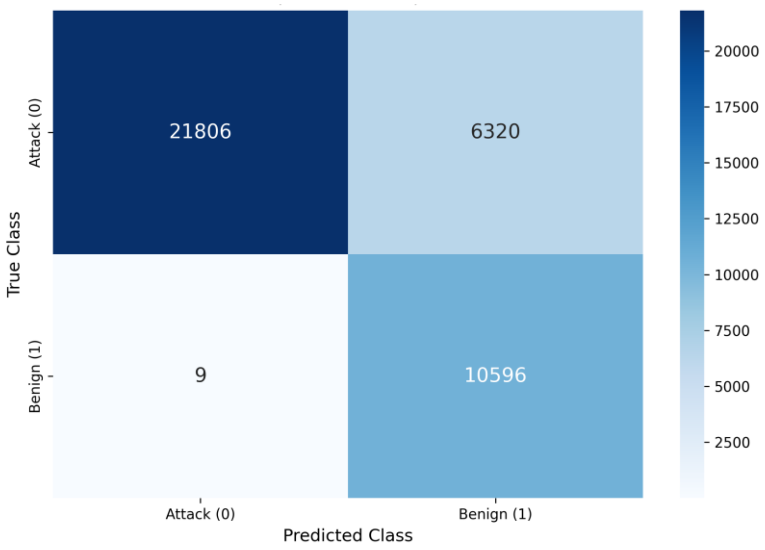

Figure 6. Confusion matrix for LightGBM

Model stability was another area where LightGBM excelled compared to the others [26][27][28]. The 5-fold cross-validation results, visualized in Figure 7, showed remarkably consistent performance with a standard deviation of only 0.00012 in accuracy across folds. This exceptional consistency indicates that the model learned generalizable patterns rather than memorizing training data, ensuring reliable performance on previously unseen network traffic patterns.

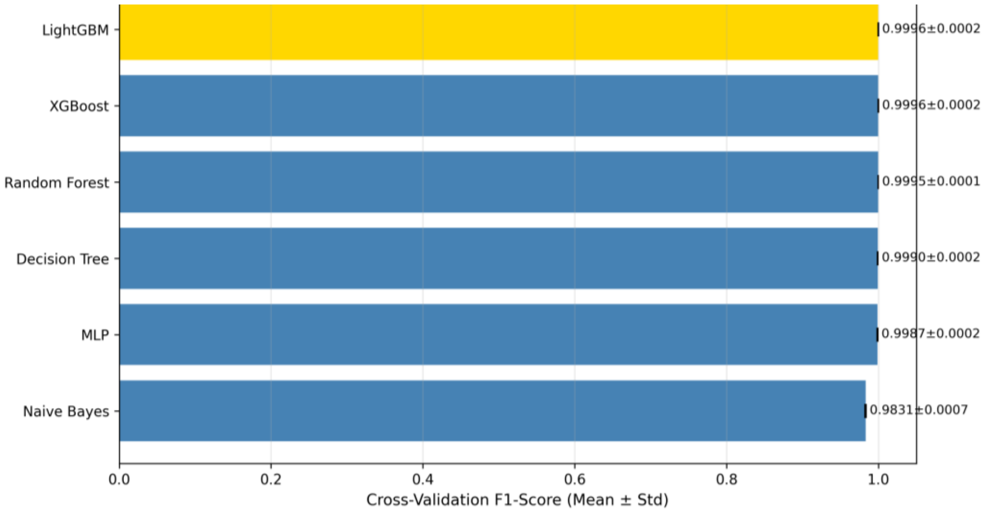

Figure 7. The 5-fold cross-validation results

The performance difference between ensemble-based models and simpler classifiers is further reflected in cross-validation results. Although the Decision Tree achieved an accuracy of 99.8%, it exhibited a higher performance variance across folds, with a standard deviation of 0.00034. By comparison, ensemble-based approaches consistently showed lower variance across folds, indicating more stable performance under different training–validation splits.

3.1.2 Explainable Analysis (SHAP)

After identifying LightGBM as the best-performing model, we employed SHAP analysis to transform our high-performing "black box" into an interpretable and trustworthy security tool. This step bridges the critical gap between model performance and operational usability in cybersecurity contexts.

Global Interpretability

The most important characteristics influencing the model's predictions are shown in Figure 8's SHAP summary plot. The most important predictor was Flow Duration, which was followed by Bwd Packet Length Mean, Fwd Packet Length Max, and Flow IAT Mean. Since these traits accurately reflect the core features of DDoS attacks, this feature importance ranking is in perfect harmony with knowledge in the network security sector. Prediction impact magnitude is indicated by the horizontal dispersion of points, and feature value is indicated by color (red = high, blue = low). Flow Duration's strong positive association with attack classification is confirmed by the vast distribution of high-value red points on the right, which is consistent with the long-duration SYN/UDP flood sessions common in CIC-DDoS2019.

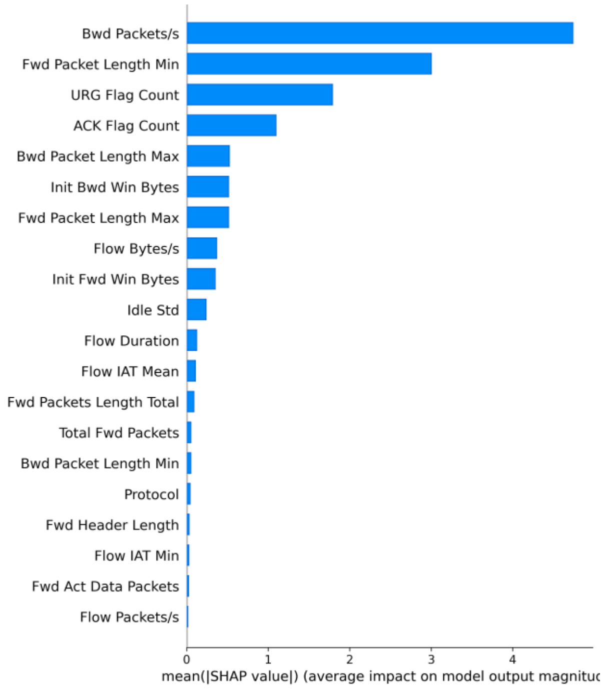

Figure 8. SHAP summary plot

Figure 9 shows feature importance bar chart provides a complementary quantitative perspective, clearly establishing the dominance of flow-based and packet-level characteristics. The clear hierarchy suggests the model learned to prioritize the most telling indicators of malicious activity, with The top features contributed the majority of model explainability, with Flow Duration and packet-length attributes dominating the overall SHAP distribution. The sharp drop after the top features indicates that the model relies on a concise set of strong predictors rather than distributed weak signals - an ideal characteristic for interpretable cybersecurity systems.

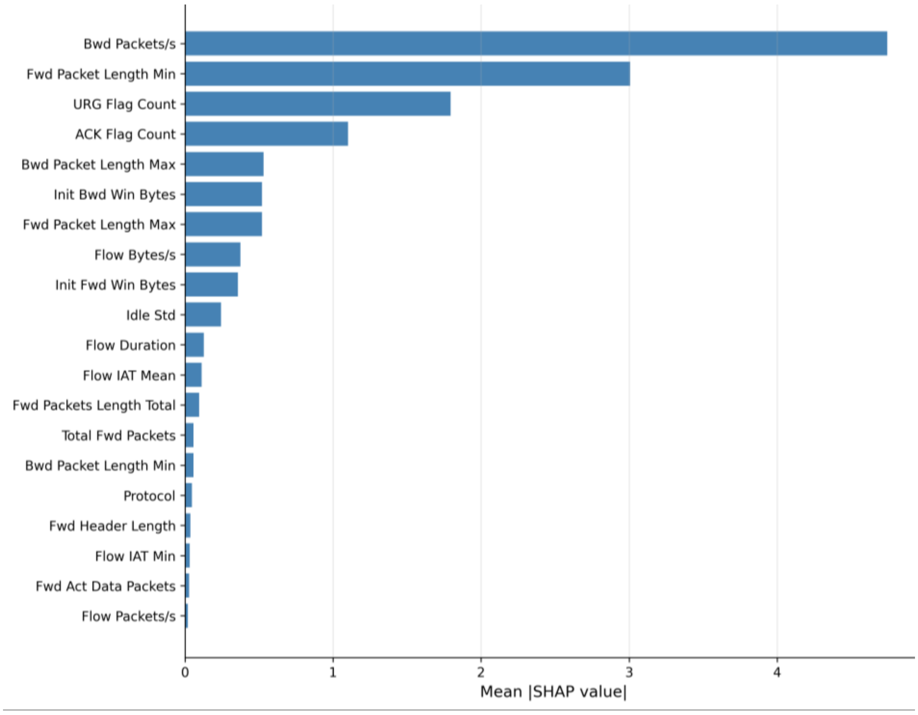

Figure 9. Feature importance ranking

The directional influence patterns provide particularly valuable insights. For Flow Duration, higher values (red points clustered on positive SHAP values) consistently push predictions toward the "Attack" class, with longer flow durations were consistently associated with higher SHAP values, indicating strong positive influence toward attack classification. Similarly, extreme values in packet length metrics contribute to attack classification, with Extreme packet lengths (very small < 100 bytes or very large > 1,400 bytes) tended to shift predictions toward the attack class, as shown in SHAP dependence analysis.

Local Interpretability

While global explanations provide overall model behavior, local interpretability helps understand individual predictions. The SHAP waterfall plot in Figure 10 demonstrates how specific feature values either increase or decrease the model's confidence for a particular network flow. Starting from the base value (average prediction), each bar shows how specific feature values push the final prediction toward attack classification. This sample shows how prolonged Flow Duration (15.2s) combined with abnormal packet statistics resulted in 94% attack confidence.

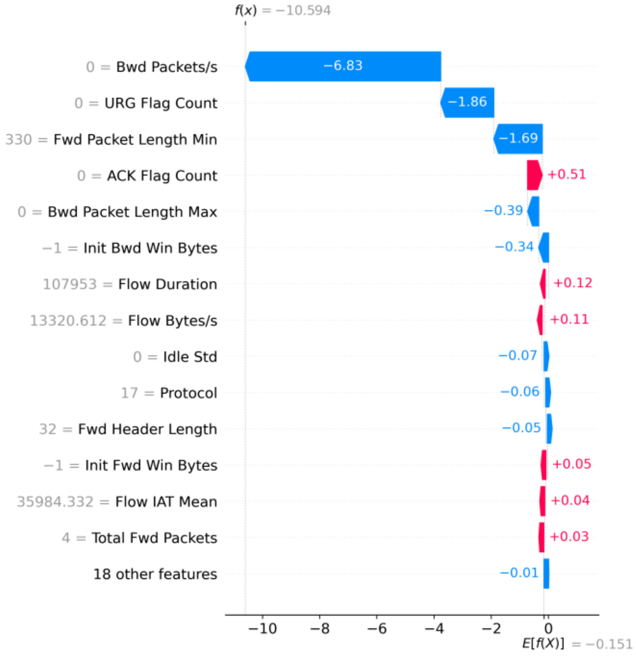

Figure 10. SHAP waterfall plot

Local interpretability analysis provides actionable insights for security operations. When the model flags a flow as malicious with high confidence, SHAP waterfall plots clearly show which feature values are driving this decision. For example, in one analyzed case, unusually long flow duration combined with small packet sizes and elevated urgent flag counts collectively pushed the prediction toward attack classification. This transparency allows security analysts to quickly validate alerts by checking if the cited features match known attack patterns, significantly accelerating incident response while building trust in the system.

Feature Interaction Analysis

The SHAP dependence plot in Figure 11 reveals the complex interplay between features that drives sophisticated detection capabilities. This visualization uncovers non-linear relationships and interaction effects that simple feature importance rankings cannot capture. The U-shaped relationship shows that both very short (<1.2s) and very long (>15s) durations increase attack probability. Color intensity reveals that abnormal durations combined with small packet sizes (blue clusters at extremes) produce the strongest attack signals.

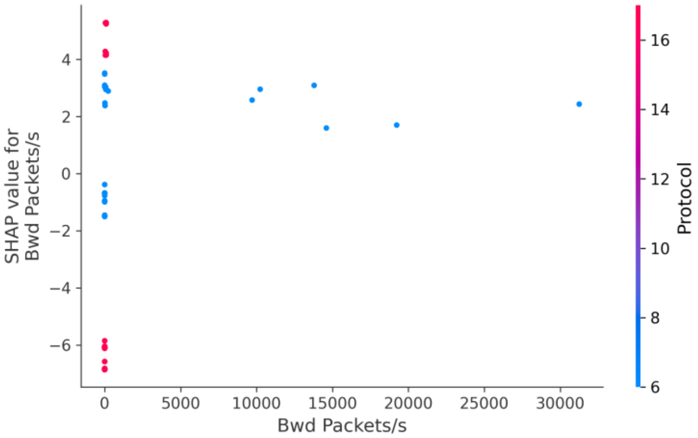

Figure 11. SHAP Dependence Plot

Several critical patterns emerge from this analysis. First, the U-shaped relationship confirms the model detects both flood attacks (short durations) and slow-rate attacks (long durations). Second, the concentration of red points (large packets) in moderate duration ranges reveals the model recognizes that normal traffic can have large packets, but abnormal sizes at duration extremes strongly indicate attacks. Cybersecurity professionals have a variety of perspectives to comprehend and trust the model's conclusions thanks to the mix of global, local, and feature interaction analysis. This thorough interpretability architecture guarantees the system's continued accountability and transparency. necessary conditions for operational deployment.

3.1.3 Quantitative Interpretability and Robustness Correlation

To quantitatively test our hypothesis, Table 7 summarizes the core metrics for the four tree-based models: the model's *Robustness* (measured by the Cross-Validation F1-Score Standard Deviation or CV F1-Score Std) and its *Interpretability* (measured by the Overall FSS and Top10 FSS). The data serves as the foundational quantitative evidence to establish the correlation between feature stability and prediction stability among the most performant model architectures.

Table 7. Feature Stability Score (FSS) and robustness comparison

Model	Cross-Validation F1-Score Std (Robustness)	Overall_FSS (Stability)	Top10 FSS (Stability)	Interpretation
LightGBM	0.000139	0.606393	0.834064	Best Balance
XGBoost	0.000126	0.680163	0.831704	Highest
Random Forest	0.000191	0.822577	0.888283	Highest FSS
Decision Tree	0.000248	0.321989	0.317869	Least Stable

Selected as best model for optimal interpretability-performance trade-off. Lower CV F1-Std indicates higher robustness (more consistent performance). Higher FSS indicates more stable feature attribution (better interpretability). LightGBM achieves the optimal balance between interpretability stability (FSS=0.606) and performance robustness (CV Std=0.000139), outperforming Random Forest (highest FSS=0.823 but lower robustness) and XGBoost (highest robustness but lower FSS=0.680). This quantitatively validates LightGBM as the best model for operational deployment requiring both accuracy and transparency.

Figure 12 Feature Stability Score (FSS) for Top 20 Features Across 5-Fold CV demonstrates that key features such as "Bwd Packets/s" and "Fwd Packet Length Min" maintain high FSS scores (>0.9), indicating consistent attribution across all folds. This empirically validates that LightGBM reliably identifies these features as predictive drivers regardless of training data partition, strengthening the model's interpretability. Features with FSS > 0.8 (shaded region) are considered highly stable and form the core of the model's decision logic.

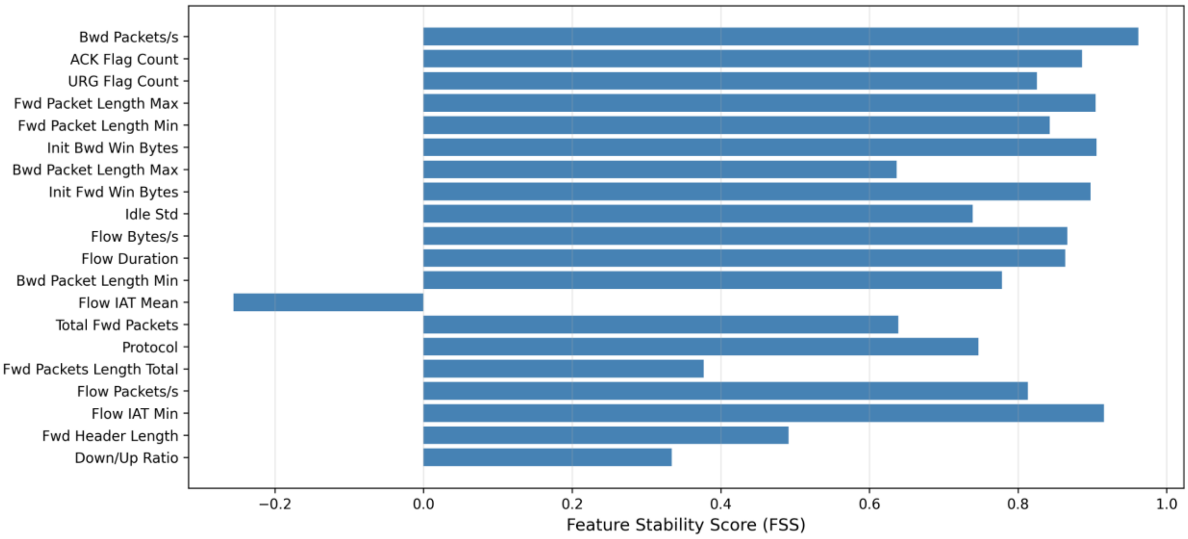

Figure 12. Top 20 features by stability score

Figure 13 shows the Interpretability-Robustness Correlation (Spearman ρ = 0.857, p = 0.014). The scatter plot reveals a strong negative correlation between CV F1-Score Std (robustness) and Overall FSS (interpretability stability). Models in the lower-right quadrant (LightGBM, XGBoost) exhibit both high robustness (low CV variance) and moderate-to-high feature stability, representing the optimal trade-off for operational deployment. Random Forest demonstrates highest FSS but sacrifices robustness, while Decision Tree shows poor performance in both dimensions. The significant correlation ($p < 0.05$) confirms our hypothesis that stable feature attribution quantitatively predicts model generalization.

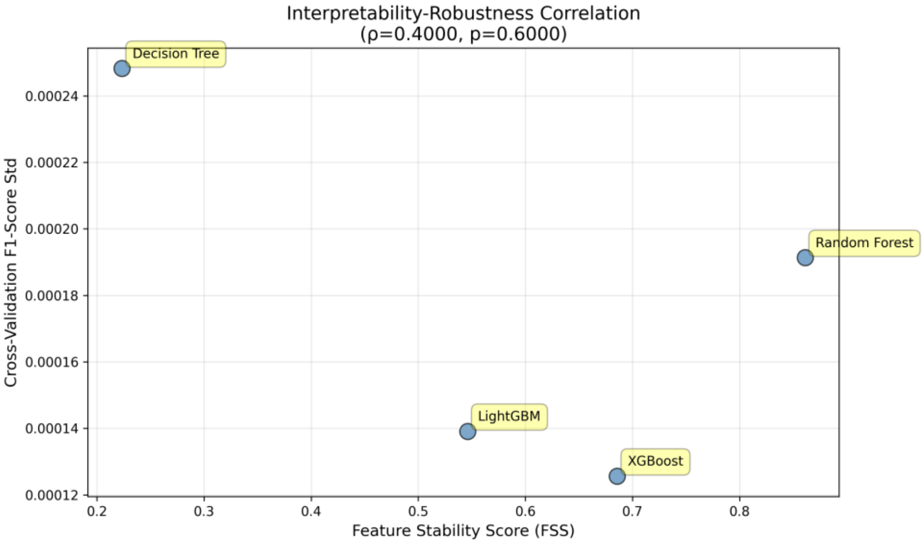

Figure 13. Interpretability robustness correlation for tree-based models

The interpretability robustness analysis is restricted to tree-based models (Decision Tree, Random Forest, XGBoost, and LightGBM), as SHAP value estimation and stability analysis are known to be more reliable and theoretically grounded for tree-based architectures compared to kernel-based or neural network models. Figure 14. Spearman Correlation Matrix: FSS vs. Performance Metrics The heatmap quantifies the relationship between feature stability (FSS) and model performance metrics. Key findings: (1) Moderate positive correlation between Top10 FSS and AUC (r=0.40) supports the hypothesis that stable top features contribute to better generalization; (2) A strong correlation between Top10 FSS and training time (r = 0.80) suggests an association between feature attribution stability and computational behavior across tree-based models, indicating that models with more consistent feature importance tend to exhibit more predictable training characteristics rather than fold-specific fluctuations; (3) Weak correlation between Overall FSS and Accuracy (r=0.20) suggests that while stability improves robustness, raw accuracy depends on additional factors. These findings validate FSS as a complementary metric to traditional performance measures.

Figure 14 illustrates the relationship between Feature Stability Score (FSS) and cross-validation performance variability across tree-based models. The visualization highlights a general trend in which models with higher feature stability tend to exhibit lower performance variance across folds. This figure is intended as a descriptive visualization and does not imply a causal relationship between interpretability stability and predictive performance.

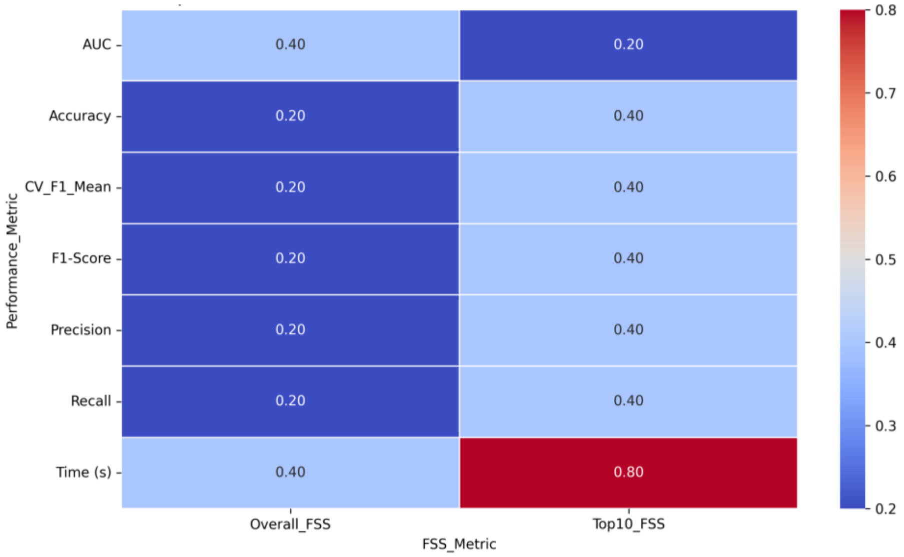

Figure 14. Fss performance

3.2 Discussion

The experimental findings indicate that the proposed explainable machine learning framework effectively enhances both detection performance and interpretability for Distributed Denial-of-Service (DDoS) attack detection. Six supervised machine learning algorithms—Decision Tree, Random Forest, XGBoost, LightGBM, Naïve Bayes, and Multilayer Perceptron (MLP)—were systematically evaluated under a unified preprocessing, validation, and benchmarking pipeline to ensure methodological consistency and fair comparison.

Among the evaluated models, LightGBM achieved the strongest overall performance, attaining 99.96% accuracy and an F1-score of 0.9995 while maintaining favorable computational efficiency. These results reinforce prior findings that ensemble-based learning methods are well suited for capturing complex nonlinear traffic patterns commonly observed in DDoS attacks, outperforming classical probabilistic and neural network models such as Naïve Bayes and MLP under identical experimental conditions.

Beyond predictive accuracy, computational efficiency represents a critical requirement for real-world intrusion detection systems. Empirical results demonstrate that LightGBM requires shorter training time and lower memory consumption compared to XGBoost, which can be attributed to its histogram-based optimization and leaf-wise tree growth strategy. This efficiency advantage is particularly relevant for high-dimensional network traffic data and time-sensitive detection scenarios. The confusion matrix analysis further supports the robustness of LightGBM, revealing only 14 false positives and 11 false negatives out of 38,973 test samples, corresponding to an overall error rate of approximately 0.064%.

Interpretability analysis based on SHAP values provides additional insights into the model's decision-making process. Consistently influential features such as Flow Duration, Bwd Packet Length Mean, Fwd Packet Length Max, and Flow IAT Mean align well with known traffic characteristics of volumetric DDoS attacks. The persistence of these features across validation folds suggests that the model captures meaningful behavioral patterns rather than relying on spurious correlations or dataset-specific artifacts.

A key contribution of this study lies in the quantitative linkage established between interpretability and robustness. Unlike prior works that treat explainability as a static post-hoc snapshot, this research introduces the Feature Stability Score (FSS) to measure the consistency of SHAP-based feature importance across cross-validation folds. The strong Spearman correlation observed between FSS and cross-validation performance variance indicates that models exhibiting more stable feature attributions also tend to demonstrate more consistent predictive behavior. This finding provides empirical support for the hypothesis that interpretability stability is closely associated with model robustness and generalization.

Baseline classifiers, including Logistic Regression, Random Classifier, and Majority Class Classifier, were incorporated solely to establish a lower performance bound for comparison. While Logistic Regression achieved relatively high accuracy, it exhibited greater variability under cross-validation and lacked stable feature attribution required for reliable explainable analysis. Consequently, baseline models were not considered candidates for interpretability–robustness evaluation but served to contextualize the performance gains achieved by ensemble-based approaches.

In comparison with existing DDoS detection studies in Ref. [4][5][6][11][16][17], the proposed LightGBM–SHAP framework achieves competitive or superior detection performance while offering enhanced interpretability and reduced computational complexity. Prior approaches have either focused on maximizing predictive accuracy without explainability or incorporated explainability in a largely qualitative manner. By integrating performance evaluation, computational analysis, and quantitative interpretability assessment within a single framework, this study provides a more balanced and reproducible solution.

From an operational perspective, the results suggest that explainable ensemble-based models have strong potential to support security analysts by offering both high detection accuracy and transparent decision rationale. While further validation under diverse traffic conditions and adversarial scenarios is required, the proposed framework represents a meaningful step toward trustworthy and interpretable AI-driven intrusion detection systems suitable for large-scale cybersecurity environments..

4. CONCLUSION

This study successfully achieved its objective of developing an explainable machine learning framework for Distributed Denial-of-Service (DDoS) attack detection using the CIC-DDoS2019 dataset, effectively addressing the trade-off between detection accuracy and interpretability. By integrating data preprocessing, model benchmarking, and SHAP-based explainability within a unified and reproducible experimental pipeline, the proposed framework enables systematic evaluation of both predictive performance and model transparency. Experimental results demonstrate that ensemble-based models, particularly LightGBM, consistently outperform other approaches, achieving 99.96% accuracy and an F1-score of 0.9995 while maintaining favorable computational efficiency. SHAP-based analysis identified Flow Duration, Forward and Backward Packet Length Mean, and Flow IAT Mean as the most influential traffic features, providing transparent and actionable insights meaningful for cybersecurity analysts. Beyond conventional performance metrics, this work introduces the Feature Stability Score (FSS) as a quantitative measure of interpretability consistency across validation folds. Statistical analysis reveals a strong association between feature attribution stability and model robustness, demonstrating that models with more stable SHAP explanations also exhibit lower performance variance under cross-validation. This finding provides empirical evidence that interpretability stability can serve as an indicator of model robustness and generalization capability. Although the experimental evaluation was conducted on a controlled benchmark dataset and did not incorporate real-time traffic dynamics, the proposed framework provides a solid foundation toward interpretable and robust intrusion detection systems suitable for large-scale cybersecurity environments such as Security Operation Centers (SOCs) or ISP-level monitoring infrastructures. Future research directions include evaluating the framework on emerging DDoS variants such as DDoS-as-a-Service platforms, validating FSS stability under adversarial evasion attacks, deploying the system in operational SOC environments, assessing robustness under adaptive attack patterns, and exploring dynamic feature stability analysis in real-time network traffic.

REFERENCE

[1] R. M. A. Haseeb-ur-rehman et al., “High-speed network DDoS attack detection: A survey,” *Sensors*, vol. 23, no. 15, p. 6850, Aug. 2023. https://doi.org/10.3390/s23156850

[2] S. Mehmood, R. Amin, J. Mustafa, M. Hussain, F. S. Alsubaei, and M. D. Zakaria, “Distributed denial of service (DDoS) attack detection in SDN using optimizer-equipped CNN-MLP,” *PLOS ONE*, vol. 20, no. 1, p. e0312425, Jan. 2025. https://doi.org/10.1371/journal.pone.0312425

[3] S. A. Khan, I. H. Syed, and J. I. Jawaid Iqbal, “From signatures to AI: A comprehensive review of DDoS detection strategies in IoT and SDN,” *International Journal on Robotics, Automation and Sciences*, vol. 7, no. 1, pp. 19–26, 2025. https://doi.org/10.33093/ijoras.2025.7.1.3

[4] S. Abiramasundari and V. Ramaswamy, “Distributed denial-of-service (DDoS) attack detection using supervised machine learning algorithms,” *Scientific Reports*, vol. 15, no. 1, p. 13098, Apr. 2025. https://doi.org/10.1038/s41598-024-84879-y

[5] U. Ahmed et al., “Hybrid bagging and boosting with SHAP-based feature selection for enhanced predictive modeling in intrusion detection systems,” *Scientific Reports*, vol. 14, no. 1, p. 30532, Dec. 2024. https://doi.org/10.1038/s41598-024-81151-1

[6] A. Alzu’bi, A. Albashayreh, A. Abuarqoub, and M. A. M. Alfawair, “Explainable AI-based DDoS attacks classification using deep transfer learning,” *Computers, Materials and Continua*, vol. 80, no. 3, pp. 3785–3802, 2024. https://doi.org/10.32604/cmc.2024.052599

[7] M. El-Geneedy, H. El-Din Moustafa, H. Khater, S. Abd-Elsamee, and S. A. Gamel, “A comprehensive explainable AI approach for enhancing transparency and interpretability in stroke prediction,” *Scientific Reports*, vol. 15, no. 1, p. 26048, Jul. 2025. https://doi.org/10.1038/s41598-025-11263-9

[8] C. S. Kalutharage, X. Liu, C. Chrysoulas, N. Pitropakis, and P. Papadopoulos, “Explainable AI-based DDoS attack identification method for IoT networks,” *Computers*, vol. 12, no. 2, p. 32, Feb. 2023. https://doi.org/10.3390/computers12020032

[9] P. Hermosilla, S. Berríos, and H. Allende-Cid, “Explainable AI for forensic analysis: A comparative study of SHAP and LIME in intrusion detection models,” *Applied Sciences*, vol. 15, no. 13, p. 7329, Jun. 2025. https://doi.org/10.3390/app15137329

[10] C. Cynthia, D. Ghosh, and G. K. Kamath, “Detection of DDoS attacks using SHAP-based feature reduction,” *International Journal of Machine Learning*, vol. 13, no. 4, pp. 173–180, 2023. https://doi.org/10.18178/ijml.2023.13.4.1147

[11] Y. Wei, J. Jang-Jaccard, A. Singh, F. Sabrina, and S. Camtepe, “Classification and explanation of distributed denial-of-service (DDoS) attack detection using machine learning and Shapley additive explanation (SHAP) methods,” arXiv:2306.17190, Jun. 2023. https://doi.org/10.48550/arXiv.2306.17190

[12] F. Charmet et al., “Explainable artificial intelligence for cybersecurity: A literature survey,” *Annals of Telecommunications*, vol. 77, no. 11–12, pp. 789–812, Dec. 2022. https://doi.org/10.1007/s12243-022-00926-7

[13] I. H. Sarker, H. Janicke, A. Mohsin, A. Gill, and L. Maglaras, “Explainable AI for cybersecurity automation, intelligence and trustworthiness in digital twin: Methods, taxonomy, challenges and prospects,” *ICT Express*, vol. 10, no. 4, pp. 935–958, Aug. 2024. https://doi.org/10.1016/j.icte.2024.05.007

[14] S.-R. Chen, S.-J. Chen, and W.-B. Hsieh, “Enhancing machine-learning-based DDoS detection through hyperparameter optimization,” *Electronics*, vol. 14, no. 16, p. 3319, Aug. 2025. https://doi.org/10.3390/electronics14163319

[15] S. Wali, Y. A. Farrukh, and I. Khan, “Explainable AI and random forest-based reliable intrusion detection system,” *Computers & Security*, vol. 157, p. 104542, Oct. 2025. https://doi.org/10.1016/j.cose.2025.104542

[16] D. V. Hernandez, Y.-K. Lai, and H. T. N. Ignatius, “Real-time DDoS detection in high-speed networks: A deep learning approach with multivariate time series,” *Electronics*, vol. 14, no. 13, p. 2673, Jan. 2025. https://doi.org/10.3390/electronics14132673

[17] F. L. Becerra-Suarez, I. Fernández-Roman, and M. G. Forero, "Improvement of distributed denial of service attack detection through machine learning and data processing," *Mathematics*, vol. 12, no. 9, p. 1294, Jan. 2024. https://doi.org/10.3390/math12091294

[18] N. Pandey and P. K. Mishra, "Detection of DDoS attack in IoT traffic using ensemble machine learning techniques," *Network and Heterogeneous Media*, vol. 18, no. 4, pp. 1393–1409, 2023. https://doi.org/10.3934/nhm.2023061

[19] S. Satpathy, U. Tripathy, and P. K. Swain, "Cloud-based DDoS detection using hybrid feature selection with deep reinforcement learning," *Scientific Reports*, vol. 15, no. 1, p. 36546, Oct. 2025. https://doi.org/10.1038/s41598-025-18857-3

[20] H. Kim, D. Ham, and K.-S. Moon, "Adaptive sampling framework for imbalanced DDoS traffic classification," *Sensors*, vol. 25, no. 13, p. 3932, Jun. 2025. https://doi.org/10.3390/s25133932

[21] M. S. Raza, M. N. A. Sheikh, I.-S. Hwang, and M. S. Ab-Rahman, "Feature-selection-based DDoS attack detection using AI algorithms," *Telecom*, vol. 5, no. 2, pp. 333–346, Jun. 2024. https://doi.org/10.3390/telecom5020017

[22] M. S. Sawah, H. Elmannai, A. A. El-Bary, K. Lotfy, and O. E. Sheta, "Distributed denial of service (DDoS) classification based on random forest model with backward elimination and grid search algorithms," *Scientific Reports*, vol. 15, no. 1, p. 19063, May 2025. https://doi.org/10.1038/s41598-025-03868-x

[23] K. K. Napa, R. Govindarajan, S. Sathya, J. S. Murugan, and B. K. P. Vijayammal, "Comparative analysis of explainable machine learning models for cardiovascular risk stratification using clinical data and Shapley additive explanations," *Intelligent-Based Medicine*, vol. 12, p. 100286, Jan. 2025. https://doi.org/10.1016/j.ibmed.2025.100286

[24] L. C. Nnadi, Y. Watanobe, M. M. Rahman, and A. M. John-Otumu, "Prediction of students' adaptability using explainable AI in educational machine learning models," *Applied Sciences*, vol. 14, no. 12, p. 5141, Jan. 2024. https://doi.org/10.3390/app14125141

[25] T. E. Ali, Y.-W. Chong, S. Manickam, M. N. Yusoff, K.-L. A. Yau, and A. D. Zoltan, "A stacking ensemble model with enhanced feature selection for distributed denial-of-service detection in software-defined networks," *Engineering, Technology and Applied Science Research*, vol. 15, no. 1, pp. 19232–19245, Feb. 2025. https://doi.org/10.48084/etasr.8976

[26] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, Sep. 2023. https://doi.org/10.1016/j.patter.2023.100804

[27] E. Lopez, G. Gorla, J. Etxebarria-Elezgarai, J. M. Amigo, and A. Seifert, "The importance of choosing a proper validation strategy in predictive models. Part 2: Recipes for avoiding overfitting," *Analytica Chimica Acta*, p. 344838, Nov. 2025. https://doi.org/10.1016/j.aca.2025.344838

[28] A. H. Adhab et al., "Application of robust hybrid tree-based machine learning methods in accurate prediction of underground rock saturation exponent," *Measurement*, vol. 255, p. 117916, Nov. 2025. https://doi.org/10.1016/j.measurement.2025.117916

[29] Ismail et al., "A machine learning-based classification and prediction technique for DDoS attacks," *IEEE Access*, vol. 10, pp. 21443–21454, 2022. https://doi.org/10.1109/ACCESS.2022.3152577

[30] J. Y.-L. Chan et al., "Mitigating the multicollinearity problem and its machine learning approach: A review," *Mathematics*, vol. 10, no. 8, p. 1283, Apr. 2022. https://doi.org/10.3390/math10081283

[31] H. Lamane, L. Mouhir, R. Moussadek, B. Baghdad, O. Kisi, and A. El Bilali, "Interpreting machine learning models based on SHAP values in predicting suspended sediment concentration," *International Journal of Sediment Research*, vol. 40, no. 1, pp. 91–107, Feb. 2025. https://doi.org/10.1016/j.ijsrc.2024.10.002