Volume XI Issue 3 Year 2026 | Page1021-1030 || ISSN: 2527-9866
Received: 15-05-2026 | Revised: 21-06-2026 | Accepted: 10-08-2026

Comparative Analysis of Performance and Interpretability of XGBoost and TabNet Models in IDS Using XAI

Ahmad Jauharul Ilmi[1], Denar Regata Akbi [2]
[1.2] University of Muhammadiyah Malang, Malang, East Java, Indonesia, 65144
e-mail: ahmadjauharul226@webmail.umm.ac.id[1] , dnarregata@umm.ac.id[2]
*Correspondence: dnarregata@umm.ac.id

Abstract: *Cyberattacks are becoming a more apparent danger to modern network traffic. Robust and transparent Intrusion Detection Systems (IDS) are increasingly needed to counter this massive wave. The emergence of Machine Learning (ML) and Deep Learning (DL) offers a huge advantage in detecting this wave with high precision; nevertheless, they also have a downside due to their secretive nature regarding decision-making algorithms, often referred to as the "black-box" problem. Digital forensics is affected by this nature as well. As a result, XGBoost (ML) and TabNet (DL) were selected and compared based on their performance and interpretability using Explainable AI (XAI) to understand the reasoning behind their decisions with the CIC-IDS-2017 dataset in this study. Both models naturally achieved extremely high classification capabilities, with XGBoost slightly outperforming TabNet in overall detection reliability and in minimizing false negatives. The XAI evaluation additionally uncovered equally valid decision-making logic: TabNet prioritizes connection states and temporal variances, while XGBoost primarily relies on rigid volumetric payload statistics. LIME analysis also affirms that both models maintain high consistency with SHAP global explanations. In summary, XGBoost is recommended as the primary real-time detector due to its superior precision and transparent logic, while TabNet serves as a secondary validator.*

Keywords: *Intrusion Detection System, XGBoost, TabNet, Explainable AI, CIC-IDS-2017.*

1. Introduction

The rapid advancement of digital transformation brought the crucial challenge of implementing a proactive and real-time monitoring system due to the increase in the variations and methods of cyberattacks, such as Ransomware and Distributed Denial of Service (DDoS) [1]. An intrusion Detection System (IDS) as the most basic network security system is designed to continuously monitor system activities and network traffic to identify suspicious patterns, malicious activities, or violations of established security policies and to take appropriate action to mitigate the risk of potentially compromised data integrity [4]. These systems also need to be applied in every network security system to help network security administrators analyze, block, or even take preventive measures against the attacks that occur in the systems [15]. However, the traditional IDS relied on signature-based methods by matching flowing network traffic against a database full of previously known threat signatures or attack patterns to identify malicious activities, which means limiting the options to identify only to the threats that have already been identified and being unable to deal with new attacks and unknown variants. This could potentially leave a significant gap in recent organizations' security systems, even though signature-based methods could detect known malicious activity with high precision and a very low error rate [5].

Responding to these weaknesses, signature-based detection is now transitioning to anomaly-based detection in various network security systems. Unlike previous approaches, anomaly-based detection begins by establishing a profile of normal system behavior and then flags any statistical deviation as a potential threat, enabling it to recognize new threats or variants [5]. The shift from signature-based detection to anomaly-based detection is made possible by

integrating Machine Learning (ML) and Deep Learning (DL) algorithms. Specifically, XGBoost, as a Machine Learning (ML) algorithm, has proven to be effective with efficient computation on high-dimensional data for improving anomaly detection accuracy [6]. Meanwhile, TabNet was developed as an advancement in Deep Learning (DL) architectures, utilizing an attention mechanism to mimic the interpretability of decision trees while also maintaining the high performance of neural networks in classifying subtle cyber threats [7].

Although ML and DL models offer high predictive accuracy with its detection capabilities, both models have issues regarding transparency due to its inability to show the parameters that affect the models' decisions, also widely known as the "Black-Box" phenomenon. For security analysts or administrators that responsible for system security, simply knowing that an "attack has occurred" is lacking; security analysts need to know the cause or explanation to understand the decision made by the system to classify particular traffic as an attack [2]. While the explanation for the decision could be essential for maintaining the user trust, assisting the verification of false positives, and facilitating rapid mitigation decision-making in critical structures, it also signifies that the inability to understand the reasoning could mean the opposite [10]. To address this, methods to implementing Explainable AI (XAI), such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) to quantify feature contributions are increasingly applied in various studies [11]. However, while several studies have applied XAI to IDS, significant research gaps remain due to a frequent focus on applying XAI to traditional ML models, neglecting modern tabular-specific architectures in previous research. In contrast, the studies that do compare ML and DL interpretability emphasize severe explanatory inconsistencies in TabNet compared to XGBoost and are constrained by the older UNSW-NB15 dataset, which often fails to capture the complex, high-dimensional polymorphic nature present in modern network traffic's cyberattacks [2].

This study established a more realistic network traffic for assessing and comparing the XGBoost and TabNet models' prediction performance and interpretability with the more fitting CIC-IDS-2017 dataset, which contains modern threats and variations of protocols to overcome the previous older dataset problem [8]. However, a necessary preprocessing steps that eliminate session-identifying attributes, such as timestamps and destination ports still need to be applied to prevent any potential data leakage and ensure the validity during the test [14]. To verify the two models' performance in maintaining consistent interpretability logic during anomaly classification after the test, SHAP and LIME as Explainable AI (XAI) tools, were applied to respectively analyze the dominant features at global and local levels [2].

Three main objectives guide this study to contribute scientific knowledge. First, the intrusion detection performance between XGBoost as a tree-based algorithm and TabNet as an attention-based deep learning architecture was compared using CIC-IDS-2017 as a modern dataset. Second, the models' results were analyzed with SHAP and LIME to investigate the stability of feature explanations at both global and local levels. Finally, by analyzing the black-box reasoning shown by both XAI tools, practical guidance for Security Operation Center (SOC) practitioners and forensics analysts might be established. This allows security teams to select a system that meets the requirements for high detection accuracy while maintaining clear and transparent decision-making in an actual environment.

2. Literature Review

Previous research stated that signature-based IDS remains vulnerable to zero-day and polymorphic attacks, even though a very low error rate was achieved against known attacks [1]. This resulted in anomaly-based detection becoming the new standard approach by identifying new threats through particular statistical pattern monitoring [5]. The massive scale of recent network traffic also requires careful data preprocessing and feature selection [13]. To handle these tasks, Extreme Gradient Boosting or XGBoost as tree-based algorithm performs

exceptionally well for processing large tabular data with minimal computational cost [6]. Deep learning technology has also produced models like TabNet that have an attention-based architecture to replicate the behavior of decision trees. This allows TabNet to maintain both the learning capabilities of neural network and detect threats with an accuracy comparable to XGBoost [7].

In a live IDS environment that a has dynamic flow of traffic, the "black-box" problem is the reason for the difficulty of applying such algorithms. Even though the models offer a high detection rate, a clear and transparent reasoning is needed to confirm whether an anomaly is a genuine threat before taking action [3]. SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) as Explainable AI tools, measure how much each network feature affects the models' output and show the results in a simple manner [10]. However, several missing pieces were revealed in recent studies. Reference [2] tested both XGBoost and TabNet and noticed that the explanations for TabNet model were inconsistent. The main issue was due to the use of an older dataset, namely UNSW-NB15, which fails to capture the complex attacks seen in recent days. Studies that used modern traffic data like CIC-IDS-2017 often focus only on performance scores, while ignoring the model interpretability [12]. Even if the studies do apply SHAP and LIME, both are often tested on older architectures, such as Multi-Layer Perceptrons rather than modern tabular models [11].

To fill these several scientific gaps, SHAP and LIME were applied to both XGBoost and TabNet using much more modern dataset, namely the CIC-IDS-2017 dataset, to additionally measure both detection power and transparency. This evaluates whether the clear and consistent reasoning required for current cybersecurity challenges is succcesfully demonstrated by deep learning models.

3. Methods

A. Research Methods

The current study compares XGBoost as a Machine Learning algorithm and TabNet as a Deep Learning architecture in intrusion detection capabilities with several processes involving data acquisition, data preprocessing, model training, and applying Explainable AI (XAI) tools, namely SHAP and LIME, by examining and analyzing the interpretability behind the models' decisions. The task is considered successful if the models could distinguish between normal network traffic and malicious activity strictly as a binary classification problem and are able to give proper reasoning behind the decisions.

B. Experimental Setup

To simulate a real-world scenario with limited resources, all experiments ran on standard consumer hardware. The system included an AMD Ryzen 5 5600H processor, 16GB of DDR4 RAM, and an NVIDIA GTX 1650 GPU running on Windows 11. An isolated Python v3.11.2 virtual environment was applied to prevent any software conflicts, thus ensuring that reproducibility is possible for future research. The stated environment contained key installed libraries including Skicit-learn for data processing, and XGBoost and Pytorch-Tabnet for the main models. Explanations for the features were also included by applying SHAP and LIME.

C. Data Collection

The primary data source used in the present study was originally from Canadian Institute for Cybersecurity that was extracted using CICFlowMeter tool and arranged to provide modern network traffic logs in CSV format called CIC-IDS-2017 dataset [8]. It contains normal background traffic alongside considerable modern attacks, including DDoS, PortScan, Botnet, and Infiltration. Over 80 statistical features, such as packet length and inter-arrival times, were

also included in the dataset. While the high feature imbalance of 83.34% normal traffic and 16.65% attack instances is present, it also shows that the dataset reflects typical actual network traffic conditions.

D. Preprocessing

The included processes were implemented to prevent any data leakage from typical errors and noise in raw network traffic data, while ensuring model validity and reliability:

1. Data Merging: There was a need to merge the original CIC-IDS-2017 dataset, which comes divided into multiple CSV files into a single, comprehensive dataset before proceeding with further data cleaning.
2. Data Cleaning: Removing all mathematical anomalies, such as Infinity (∞), missing value, inconsistent whitespace, and Not a Number (NaN) values, thus maintaining computational stability and consistensy during the algorithm training.
3. Feature Reduction: To prevent data leakage and ensure the models learn generalizeable behavioral patterns, identity attributes, as such TimeStamp and Destination Port, were permanently removed from the dataset.
4. Imbalance Handling and Data Splitting: To preserve the highly imbalanced class distribution between normal traffic and attacks across both subsets, class weight balancing was applied during the data splitting. The dataset was subsequently partitioned into a Training Set 70% and a Testing Set 30%. Stratified sampling was strictly applied to preserve the original class distribution across all subsets.
5. Data Normalization: Continuous numerical features were scaled using MinMax Scaler to standardize all numerical features into a strictly defined [0, 1] range, fitted exclusively on the training data to prevent data snooping. To maintain methodological integrity and prevent data snooping-where information from the test set inadvertently influences the training phase-the scaler was fitted exclusively on the Training Set before being used to transform both the training and testing data.

E. Model

Two distinct learning architectures were implemented to facilitate the comparative analysis. The first model, Extreme Gradient Boosting (XGBoost), was selected as the representative Machine Learning approach due to its state-of-the-art efficiency and performance on high-dimensional tabular data [4]. It was configured as a binary classifier utilizing the binary:logistic objective function, max_depth=8, learning_rate=0.08, n_estimator=300, and scale_pos_weight to manage imbalance, all o fit was carefully calculated to prevent overfitting while maximizing detection capability. The second model, TabNet, was selected as the representative Deep Learning approach because of its novel sequential attention mechanism that inherently mimics the interpretability of decision trees while retaining deep representational power [5]. The model was optimized using Adam optimizer with hyperparameter set to: n_d=8, n_a=8 n_steps=3. An early stopping mechanism was implemented, monitoring validation loss with a patience set to 10 epoch to prevent overfitting.

F. Explainable AI (XAI) and Evaluation Metrics

To audit the autonomous security models and address the black-box dilemma [7], two post-hoc interpretability frameworks were applied to the testing predictions. For the SHAP implementation, the TreeExplainer was utilized for XGBoost, and the KernelExplainer was used for TabNet, extracting global feature importance through Summary Plots and local instance explanations via Force Plots based on Shapley values [6]. Concurrently, the LIME framework was applied using the LimeTabularExplainer to construct localized, surrogate linear models around specific attack instances to observe local feature weights [6]. The reliability of these explanations was mathematically evaluated using the Jaccard Similarity Index on the top-k extracted features, calculated as the intersection over the union of the feature sets deemed most important by SHAP and LIME:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

This index objectively indicates the level of interpretability consistency between the methods [6]. Through a confusion matrix, Recall and F1-score were primarily measured for evaluating the performance of both models. Preventing real attacks from bypassing security by maximizing Recall to keep false negatives low is crucial for any intrusion detection system [1]. Rather than standard accuracy, the F1-Score served as a better indicator of success due to the highly imbalanced data. Reproducibility was also ensured in the experiment by applying a fixed random seed in all algorithms.

4. Results and Discussion

The final results of threat detection accuracy for both XGBoost and TabNet models are broken down in this section, followed by applying Explainable AI (XAI) tools for an explanation of their decision-making transparency.

A. Predictive Performance Analysis and Metric Evaluation

The performance results of both architectures are presented in Table 1 to assess both models' ability to classify network traffic accurately.

Table 1. Model Performance Comparison on Testing Data

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	0.9998	0.9998	0.9996	0.9997
TabNet	0.9984	0.9972	0.9958	0.9965

Table 1 confirms the quality of both models in detection capabilities, with XGBoost maintaining a slight advantage across the measured parameters over TabNet, while also managing to balance Precision and Recall in its high F1-Score, as shown in Table 1, even though the network data has heavily imbalanced normal traffic and attack instances. Additionally, XGBoost produces a very few false negatives as shown by the high Recall rate. By definition, XGBoost handles the requirement exceptionally well for blocking every threat before it enters the network, and it is highly regarded in an actual Intrusion Detection System (IDS).

Scores above 99% often raise concerns about whether data leakage occurred during the test. Previous research stated that near-perfect accuracy may occur simply because identity features, such as Source/Destination IPs, ports, and timestamps, were memorized by the algorithms. However, the preprocessing phase of this study permanently removed all identifiers to prevent any possible data leakage. As a result, the models had to rely entirely on network behaviors, such as payload variance and inter-arrival times. This eliminates any doubt that the accuracy

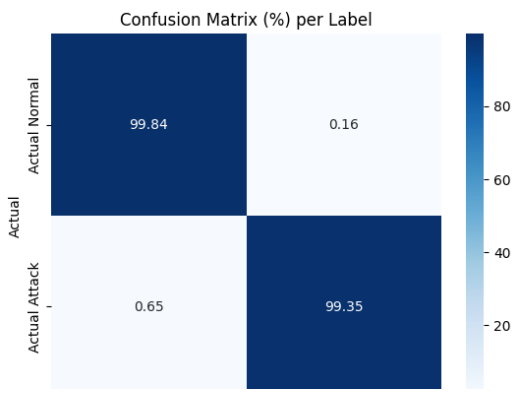

Figure 1. XGBoost Confusion Matrix

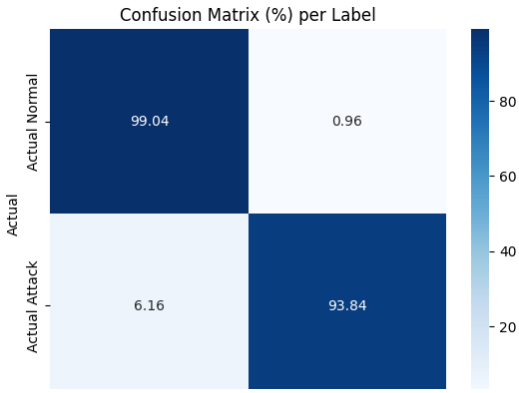

Figure 2. Tabnet Confusion Matrix

results were an accidental byproduct of leaked features, proving instead that they reflect genuine threat detection capabilities. The confusion matrix distribution displayed in Figure 1 demonstrates that XGBoost, as a tree-based Machine Learning model, is highly reliable for operational IDS deployment with effective interception of real threats with a 99.35% success rate, while keeping false negatives to a minimum 0.65% scores. It is also worth to mentioning that XGBoost achieved the intended result by showing 99.84% of normal traffic and a low false positive rate of 0.16%.

While the confusion matrix distribution for TabNet as deep learning model also demonstrated high detection capabilities in Figure 2, its accuracy was slightly lower compared to XGBoost. To calculate the difference, the TabNet model achieved 0.8% less on normal traffic and 5.51% less on actual attacks than of XGBoost. It also means that TabNet achieved a notable false negative rate of 6.16% that raises the risk in cybersecurity, as the system was bypassed by a considerable amount of malicious traffic more so than XGBoost. Based on the shown results, a tree-based based ensemble approach like XGBoost currently has an advantage, both in raw precision and operational safety for high-dimensional log over TabNet, which proven to be slightly prone to missing nuanced network intrusion despite its advanced sequential attention mechanism.

B. Global Interpretability Analysis Using SHAP

The models' performance evaluation was then applied by SHAP to understand specific network features that carried most weight in their classification with analyzing top global decision-making patterns. The illustrated Figure 3 and Figure 4 respectively present SHAP bar plots for XGBoost and TabNet.

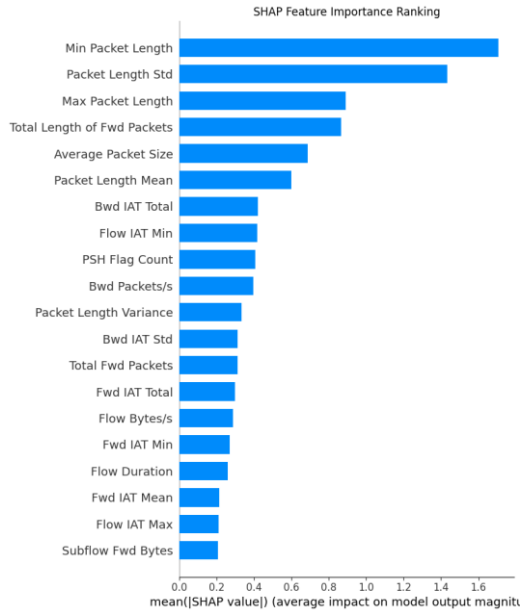

Figure 3. XGBoost SHAP Bar Plot

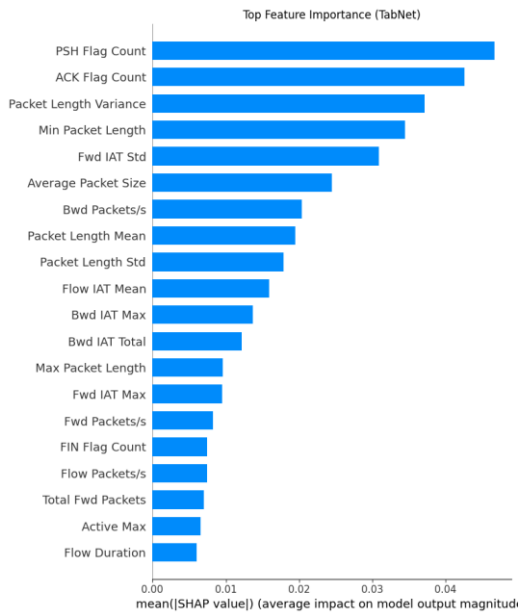

Figure 4. TabNet SHAP Bar Plot

As shown in Figure 3 and Figure 4, distinct strategies for feature prioritization could be seen in both architectures. Following Figure 3, the primary decision driver for XGBoost is Min Packet Length which is shown to be substantially higher than other variables, such as Packet Length Std, Max Packet Length, and Total Length of Fwd Packets. This could be explained by the fact that to classify malicious traffic, the extreme packet sizes and their statistical variance are the most important for XGBoost's decision choices.

Although packet length variables remain relevant, TabNet evaluates network features fairly differently from XGBoost. As shown in Figure 4, PSH Flag Count ranks as the most significant, followed by ACK Flag Count, Packet Length Variance, and Min Packet Length. This result implies that the specific states and control signals of network connections are more important features for the deep learning model, contrary to XGBoost, which focuses predominantly on payload sizes.

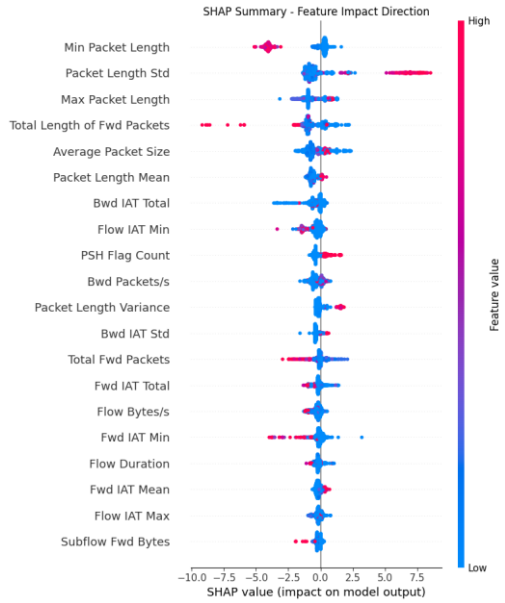

Figure 5. XGBoost SHAP Beeswarm Plot

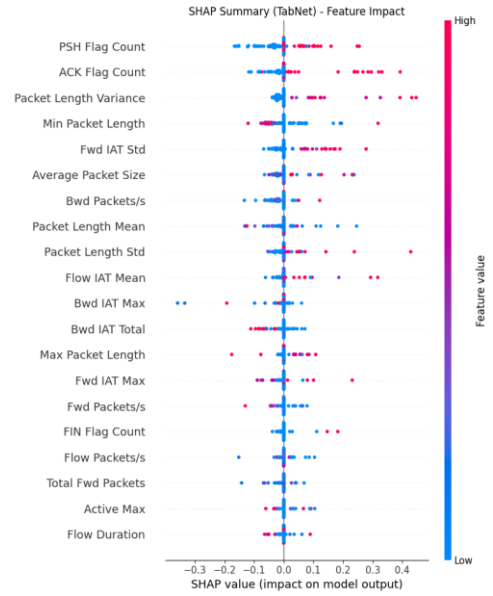

Figure 6. TabNet SHAP Beeswarm Plot

The beeswarm plots illustrated in Figure 5 and Figure 6 further explain the specific features that affect the models' predictions. For instance, Figure 5 shows that the red dots for the Packet Length Std feature are consistently generate positive SHAP values or increasingly go to the right side; this indicates that higher values of Packet Length are more likely to be classified as attack instances. Another instance, shown in the blue dots indicated by Min Packet Length and Total Length of Fwd Packets being pulled in the center, suggests that lower values of both features are more likely to be classified as normal traffic. XGBoost additionally confirms that high variance in packet sizes is the most important features in network intrusions.

A common trait of deep learning architectures to handle non-linear interactions is shown by the more scattered feature impacts of TabNet in Figure 6. The plot shows that higher values of PSH Flag Count are generally classified as an attack instance, while the opposites remain neutral or slightly negative and the ACK Flag Count feature shares a similar result. The much lesser effect of the packet length variance feature compared to XGBoost in TabNet's plot demonstrates that the distinct mechanisms for threat detection are applied in both models, even though they share similar accuracy results. The SHAP global interpretation shows that high variance in packet size is more important for XGBoost's decision-making, while TabNet evaluates a mix of connection flags followed by packet size variations.

C. Local Interpretability Analysis Using LIME

To illustrate feature importance in the models, LIME, focusing on the exact rules and thresholds in specific classifications by the models, was applied to highlight the top features that SHAP showed in the previous section. The illustrations for XGBoost and TabNet at the local are shown in Figure 7 and Figure 8.

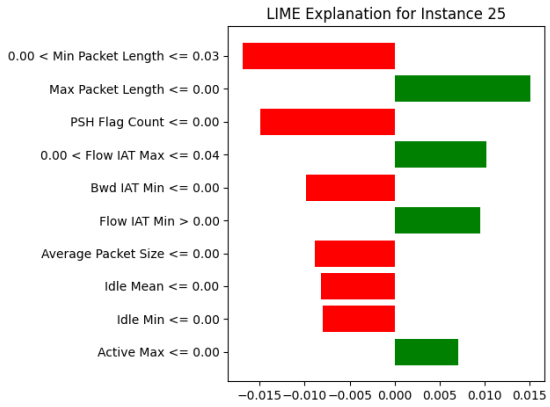

Figure 7. XGBoost LIME

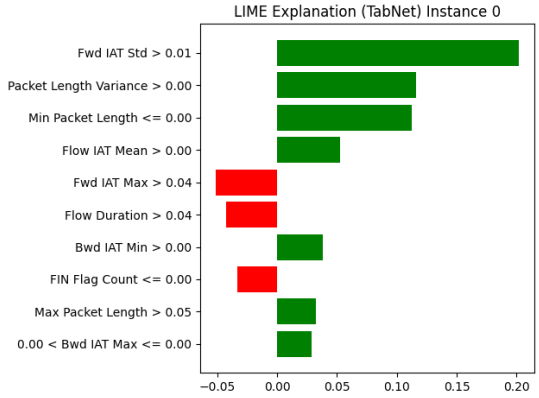

Figure 8. Tabnet LIME

The main factor pushing the prediction for XGBoost in Figure 7 is the use of strict packet length thresholds, such as 0.00 < Min Packet Length <= 0.03 to classify toward normal traffic, while the Max Packet Length <= 0.00 drives the classification toward attack instances. The result confirms that high variance of packet sizes is the most important factor for XGBoost's decision algorithm, while also matching the SHAP's global findings.

The different approach for TabNet is shown in Figure 8 by prioritizing the variance in packet timing and size, such as FWD IAT Std > 0.01 as the strongest indicator for an attack, followed by Packet Length Variance > 0.00 and Min Packet Length <= 0.00. However, the prediction was pulled toward the normal traffic due to the timing condition factor, such as FWD IAT Max > 0.04 and flow Duration > 0.04. Through the local explanations, both methods provide enough transparency for security analysts to understand the technical logic behind any detected anomaly, with XGBoost focusing on the minimum and maximum of packet sizes, and TabNet mixing timing variance and packet size variance.

D. Explanatory Consistency Measurement (Jaccard Similarity)

A score of 42.86% on the Jaccard Similarity Index was achieved after measuring the overlapping features in the top 10 SHAP feature importances in both architectures. The moderate score indicates that network traffic was analyzed from different angles rather than duplicating the same logic, even though the models share the similar accuracy. The difference between both models creates a possibility to implement a highly effective, dual-engine IDS by combining both architectures' perspectives.

E. Discussion and Research Implications

This section is dedicated to addressing several problems that may occur during and after the research, such as:

1. Imbalance Handling: Both models have avoided the common issue to inflating their accuracy during the test, which may happen due to natural network data typically having higher normal traffic than attacks, by learning the patterns of the attack instances as proven by the high F1-Scores and Recall rates.
2. Dataset Bias and Overfitting: The near-perfect scores that both models achieved might have happened by memorizing specific patterns in the dataset, because the CIC-IDS-2017 dataset was produced in a controlled, laboratory environment. This type of problem is a vulnerability for TabNet as a deep learning architecture that relies on rigid numerical boundaries as shown in LIME outputs, even though applying an early stopping mechanism might have helped. If attackers slightly alter their transmission timing and packet size using Adversarial Machine Learning (AML), detection by the system could be avoided.

3. Real-World Generalization: The high experimental scores shown in this study face difficulties in actual implementation because of the natural dynamic flow of network traffic in an actual Security Operation Center (SOC). Standard updates that change normal payloads could trigger a wave of false positives for XGBoost, which relies on minimum and maximum packet sizes, making it necessary to apply dynamic feature thresholds before implementing the model in the system.

5. Conclusions

This study compared the network detection capabilities of XGBoost and TabNet, while applying SHAP and LIME as Explainable AI (XAI) tools to understand the reasoning behind the decision made by the models. The network detection results confirm that both models offer highly reliable classification, with XGBoost performing slightly better across the main metrics with an impressive accuracy of 99.98% and F1-Score of 99.97%, while keeping the false negative rate at a minimal 0.65%.

To understand the underlying logic behind the models' decisions, SHAP and LIME were applied and achieved the result that both architectures relied on distinctly different reasoning to reach a decision. The XGBoost was shown to rely on a high variance of packet length, such as its minimum and maximum to decide whether the traffic is normal or an attack instance, contrary to TabNet, which shows greater consideration toward the variations in protocol states and transmission timing, followed by packet sizes. This difference in perspective is further proven by the moderate Jaccard Similarity score of 42.86%. Additionally, TabNet as a deep learning architecture was proven not to be a strictly opaque "black box", due to clear explanations at the local level shown by LIME.

Despite the successful outcomes, being exclusively dependent on the CIC-IDS-2017 dataset and the lack of external validation using more recent alternatives, such as the CIC-Bell-DNS dataset resulted in uncertainty in applicability to different network environments, due to the possibility of the models memorizing patterns in this specific dataset, rather than universally recognizable attacks. Future research should focus on validating the operational viability of these models and verifying the XAI explanations with different or newer datasets. There is also a need to expand the scope of the detection from binary to multiclass categorization to reveal the models' effectiveness in differentiating various attack types. In conclusion, implementing the systems in a live environment, such as a Security Operation Center (SOC) and subjecting them to Adversarial Machine Learning (AML) would be the only way to know the true value of these models.

References

[1] Diana, L., Dini, P., & Paolini, D. (2025). Overview on intrusion detection systems for computers networking security. *Computers*, *14*(3), 87. https://doi.org/10.3390/computers14030087

[2] Hermosilla, P., Díaz, M., Berríos, S., & Allende-Cid, H. (2025). Use of explainable artificial intelligence for analyzing and explaining intrusion detection systems. *Computers*, *14*(5), 160. https://doi.org/10.3390/computers14050160

[3] Khan, N., Ahmad, K., Al Tamimi, A., Alani, M. M., Bermak, A., & Khalil, I. (2025). Explainable AI-based intrusion detection systems for industry 5.0 and adversarial XAI: A systematic review. *Information*, *16*(12), 1036. https://doi.org/10.3390/info16121036

[4] Deski Ari Sandi and Agus Tedyyana, "Implementasi dan Analisa Sistem Pencegahan Intrusi pada Aplikasi Web Menggunakan Web Application Firewall", *Repeater*, vol. 2, no. 4, pp. 16–26, Aug. 2024.

[5] Ravindran, V. K., Ojha, S. S., & Kamboj, A. (2025). A comparative analysis of signature-based and anomaly-based intrusion detection systems. *International Journal of Latest Technology in Engineering, Management & Applied Science*, *14*(5). https://doi.org/10.51583/IJLTEMAS.2025.140500026

[6] Rudianto, A. M. A., Pramukantoro, E. S., & Kurnianingtyas, D. (2025). Implementasi Sistem Deteksi Anomali pada Jaringan Komputer dengan Pendekatan XGBoost dan Data SNMP. (2025). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *9*(2). https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14446

[7] Tedyyana, A., and O. Ghazali. "W. Purbo, O.(2024). Enhancing intrusion detection system using rectified linear unit function in pigeon inspired optimization algorithm." *IAES International Journal of Artificial Intelligence (IJ-AI)* 13.2: 1526-1534.

[8] Lumban Gaol, A. P., Tedyyana, A., & Hidayasari, N. (2025). Penerapan sistem keamanan jaringan menggunakan intrusion prevention system dengan machine learning. *Sistemasi: Jurnal Sistem Informasi*, *14*(4). https://doi.org/10.32520/stmsi.v14i6.5460

[9] Hossain, M. A. (2025). Deep learning-based intrusion detection for IoT networks: a scalable and efficient approach. *EURASIP Journal on Information Security*, *2025*(1), 28. https://doi.org/10.1186/s13635-025-00202-w

[10] Mane, S., & Rao, D. (2024). *Explaining network intrusion detection system using explainable AI framework.* Persistent Systems Limited. https://doi.org/10.48550/arXiv.2103.07110

[11] Gaspar, D., Silva, P., & Silva, C. (2024). Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron. *IEEE Access*, *12*, 31448-31462. https://doi.org/10.1109/ACCESS.2024.3368377

[12] Chavan, P. V., & Alone, N. V. (2025). Optimizing intrusion detection with random forest: A high-accuracy approach using CIC-IDS 2017. *International Journal of Computer Applications*, *187*(3), 17-22. https://doi.org/10.5120/ijca2025924816

[13] Devia, A., & Soewito, B. (2023). Analisis perbandingan metode seleksi fitur untuk mendeteksi anomali pada dataset CIC-IDS-2018. *Jurnal Teknologi dan Sistem Informasi Bisnis*, *5*(4), 572-578. https://doi.org/10.47233/jteksis.v5i4.1069

[14] Teddyana, A. (2021). Monitoring Jaringan Internet Menggunakan Notifikasi Bot APITelegram. SATIN-Sains dan Teknologi Informasi, 7(1), 144-152.

[15] Rishika, R. V., Pratomo, B. A., & Hidayati, S. C. (2025). Network intrusion detection system with time-based sequential cluster models using LSTM and GRU. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *23*(1), 1-12. https://doi.org/10.12962/j24068535.v23i1.a1241