Volume XI Issue 3 Year 2026 | Page 969-979 | ISSN: 2527-9866
Received: ***15-05-2026*** | *Revised:* ***29-05-2026*** | *Accepted:* ***15-08-2026***

SHAP and LIME Analysis on CNN-GRU Deep Learning Models for IoT Network Intrusion Detection

Satria Purfie Purnama Putra[1], Denar Regata Akbi[2]
[1,2]University of Muhammadiyah Malang, Malang, Indonesia
e-mail: satria_purfie@webmail.umm.ac.id[1], dnarregata@umm.ac.id[2]
*Correspondence: dnarregata@umm.ac.id

Abstract: *As digital devices evolve quickly and cyberattacks grow more diverse, the Internet of Things (IoT) ecosystem faces a much higher security risk than before. An AI-based security system is commonly used for post-attack mitigation. However, this approach has several issues, such as high computational load, data imbalance, and a black-box that does not explain the attack patterns. This study addresses a research gap by implementing a hybrid CNN-GRU architecture. With XAI, this architecture remains lightweight and robust while mapping out malicious attack patterns quite clearly. The technique breaks down these security incidents by providing both global and local explanations. This study uses two IoT datasets, BoT-IoT with 72 million records and IoTID20 625,783 records. The SMOTE post-split technique was performed on 80% of the total data to address data imbalance and avoid data leakage. To validate the results, stratified holdout is used to evaluate the training results. This research has very satisfying results with 99% accuracy, and 99% F1-Score so that it can minimize errors in both datasets. The contributions made by this research are: (1) adapting a stable model for anomaly classification, (2) handling data imbalance and avoiding data leakage, and (3) integrating SHAP and LIME to overcome black-box.*

Keywords: *Intrusion Detection System, CNN-GRU, Bot-IoT, IoTID20, SHAP, Lime.*

1. Introduction

The Development of IoT devices presents interconnected networks, thereby providing various benefits to the industrial sector and the household sector, enabling extensive automation and device connectivity However, as the adoption of the Internet of Things grows, there are weaknesses in security features; frequently, IoT devices are often hit by intrusion attacks such as DDoS which can cause infrastructure damage and service disruptions for users[2]. Conventional network security systems rely on firewalls and signature-based IDS, which are increasingly considered ineffective in facing continuously evolving cyber threats. They face difficulties in identifying new attacks that are not stored in their memory. This weakness has led to the development of IDS systems to identify and detect new security threats [3].

The operational limitations of signature-based IDSs in threat detection have driven the integration of more adaptive anomaly-based systems into modern security frameworks [4]. Highlighting recent advancements in AI for anomaly detection, deep learning-based hybrid models have demonstrated the capability to achieve high classification accuracy while remaining highly adaptive within large-scale network environments [5]. An effective IDS must reduce the margin of error to a minimum [4]. For this reason, implementing a high-performance hybrid deep learning model makes it possible to create an IDS that is optimal for detecting attacks.

The results of applying deep learning show that it is effective in performing efficient feature extraction even on large datasets [6]. The combination of two deep learning models outperforms a single model. The hybrid CapsNet + BiLSTM successfully achieved satisfactory accuracy results on three datasets: on UNSW-NB15 it obtained an accuracy of 97%, on KDD CUP 99 it achieved

98%, and finally on the CIC-IDS2017 dataset it obtained the highest result at 99%. [7]. in this architecture, CapsNet is used to extract spatial features while BiLSTM is used to learn temporal features [7]. So that each component can address its detection focus in a structured manner. The CNN-GRU hybrid architecture operates on the same principle; this model combines the strength of CNNs in extracting spatial features or local patterns with the GRU's ability to capture long-term temporal dependencies in network traffic sequences [6]. The results of this hybrid model are very high, where on the Bot-IoT dataset it obtained an accuracy of 99.01% and on the IoTID20 dataset it obtained an equally high accuracy of 99.83%. [6].

With the results of the application of hybrid Deep Learning that separates model workloads into spatial and temporal dimensions for anomaly recognition [6], this study proposes the use of an efficient hybrid CNN-GRU architecture tailored for IoT network environments. In this architecture CNN is used to capture local spatial features and GRU is used to learn temporal features in IoT network packets [6]. Although this model demonstrates high classification performance and can even outperform the hybrid CapsNet + BiLSTM model, it has limitations in the domain of digital forensics. The black-box problem poses a significant challenge in understanding how the model makes decisions based on the available features [6]. The integration of XAI into this hybrid model certainly increases the reliability of the system [4]. The role of SHAP and LIME in this study aims to help explain the model when making decisions, so that cyber operators can understand how the model recognizes anomalies and can carry out optimal mitigation [8].

This study was conducted to address the research gap, which exists in primary reference journals, which are still black-box [6]. The following are the contributions of this study:

- Adapting a stable CNN-GRU hybrid model for anomaly classification, especially in the ever-evolving IoT environment.
- Handling data imbalance and preventing data leakage with the post-split SMOTE technique.
- Integrating SHAP and LIME to explain how the model is able to transparently dissect cyberattacks, both locally and globally.

2. Literature Review

The development of AI-based intrusion detection systems within IoT networks has been widely implemented, proving successful in identifying threats accurately, reliably, and efficiently for cybersecurity purposes. Studies conducted by Adefemi et al. [6] and Diana L. et al. [4] demonstrate that hybrid Deep Learning models excel in determining attacks by extracting both spatial and temporal features. Additionally, research by Ogunseyi & Thiyagarajan [9] and Le T. et al. [10] integrates Explainable AI (XAI) into artificial intelligence frameworks to clarify their decision-making mechanisms, thereby achieving transparent information evaluation. However, to systematically deconstruct the opaque 'black-box' nature of these models, it is imperative to identify the specific limitations inherent in each previous study [6]. To find out what gaps there are in each previous study, Table 1 compares the models, datasets, use of XAI and research gaps so that it can provide novelty and the contribution that this study makes.

Table 1. Summary of gaps in previous research

No	Researcher & Year	Method/ Model	Dataset	Performance	XAI usage	Research Gaps
1	Adefemi et al (2025) [6]	CNN-GRU	BoT-IoT IoTID20	Accuracy: 99.01% Accuracy: 99.83%	-	The model is black-box

2	Diana L et al (2025) [4]	CapsNet + BiLSTM	CIC-IDS2017 UNSW-NB15 KDD CUP 99	Accuracy: 99% Accuracy: 97% Accuracy: 98%	-	High architectural complexity resulting in high computational load
3	Ogunseyi & Thiyagarajan (2025) [9]	LSTM and BP-LSTM optimized using the Firefly Algorithm	NF-BoT-IoT-v2 IoTID20	Accuracy: 98.42% Accuracy: 89.54%	SHAP & LIME	The model still needs improvement in accuracy or using deep learning with the proposed architecture.
4	Le T et al (2022) [10]	Decision Tree & Random Forest	IoTID20 NF-BoT-IoT-v2 NF-ToN-IoT-v2	Accuracy: 100% Accuracy: 100% Accuracy: 100%	SHAP	In the NF-ToN NF-ToN-v2 dataset, there is still data imbalance

Based on table 1, it can be concluded that previous research still focuses on high performance, these results are indeed extraordinary, but with these high results it is necessary to explain how the model makes decisions and the limitations on IoT performance need to be considered so that the computational load is not high.[6], [4]. Then, research that has explained how the model for making decisions by applying SHAP and LIME has good results, but there are shortcomings in classification performance and data balance [9], [10]. Based on the research gaps mentioned above, this study offers the following novelty and contributions: (1) Adaptation of a model with high performance and lower computational load, (2) Handling data imbalance in each dataset to prevent data leakage, and finally, (3) Integrating SHAP and LIME to explain the high-performance model for transparency.

3.Methods

The general steps of this research methodology are illustrated in Figure 1, starting with data input, data preprocessing (using post-split SMOTE), application of the hybrid model (CNN-GRU), and concluding with the implementation of XAI (SHAP and LIME).

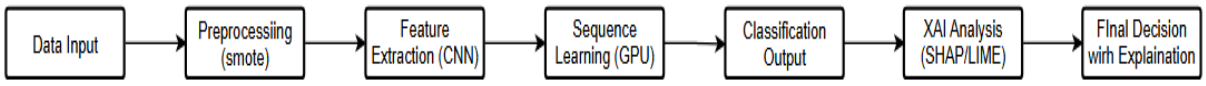

Figure 1. Prototyping Method Flow

A. Experimental Setup

This study used a personal computer with an Intel i7-13650HX processor, 24 GB of DDR5 RAM, and an NVIDIA RTX 4050 graphics card, running on the Windows 11 operating system. To maintain the validity of the experiment, this study utilized a Python 3.10.0 virtual environment. This strategy was chosen to keep the system clean and prevent the libraries from being mixed with others. The libraries used include Keras 2.10.0, Scikit-learn 1.1.3, Imbalanced-learn 1.1.3, TensorFlow 2.10.0, LIME 0.2.0.1, and SHAP 0.41.0.

B. Data Acquisition

This study used two intrusion detection datasets from the Internet of Things (IoT) environment. The IoTID20 dataset, developed by researchers Ullah and Mahmoud, consists of internet traffic logs from IoT smart homes that contain various types of intrusion attacks and scanning attempts [11]. The second dataset is BoT-IoT, which contains a massive record of IoT traffic, including various flooding, scanning, and data theft attacks, developed by Koroniotis et al. [12]. According to the study by Adefemi et al. [6], the dataset used represents 10% of the original dataset, achieved by sampling from an overall total of 72 million data instances. To understand the size of the dataset and the class distribution, the summary is provided in Table 2 and Table 3.

Table 2. Size of the BoT-IoT dataset

Category	Subcategory	Number of Instances
Normal	Normal	9,543
Reconnaissance	Service Scan	1,463,364
DoS	TCP	12,315,997
DoS	UDP	20,659,491
Reconnaissance	OS Finger	358,275
DoS	HTTP	29,706
Thef	Data Exfiltration	118
Thef	Keylogging	1,469
DDoS	HTTP	19,771
DDoS	TCP	195,476,03
DDoS	UDP	18,965,106
Total		73,370,443

Table 3. Size of the IoTID20 dataset

Category	Subcategory	Number of Instances
Normal	Normal	40,073
Mirai	UDP Flooding	183,554
	Host Brute Force	121,181
	HTTP Flooding	55,818
	Ack Flooding	55,124
Scan	Scan Port OS	53,073
	Scan Host Port	22,192
MITM ARP Spoofing	MITM ARP Spoofing	35,377
DoS	Syn Flooding	59,391
Total		625,783

C. Preprocessing

Because the dataset was very large and imbalanced, preprocessing was essential. This step was performed to ensure that the data used as input for the CNN-GRU model is appropriate and of high quality.

1) Feature Cleaning

The first step involved removing irrelevant attributes, those that are purely identifying, or those that have the potential to cause data leakage, thereby reducing noise. Features removed included Source Port, Source IP, Destination IP, Flow ID, and Timestamp. Additionally, low-variance features-those with nearly identical values-were removed. Identifiers and descriptors are also removed because they reflect the identity of network communications-rather than the characteristics of normal traffic or attacks-which can interfere with the model during training and cause overfitting [6]. The final features used by the model to make decisions are as follows:

- BoT-IoT: Mean, State_REQ, State_RST, Flgs_e s, min, stddev, max, Flgs_eg, Proto_tcp, State_INT.

- IoTID20: Dst_Port, Init_Bwd_Win_Byts, Protocol, Bwd_Seg_Size_Avg, Flow_Duration, Pkt_Size_Avg, Ack_Flag_Cnt, Idle_Mean, Fwd_Act_Data_Pkts, Idle_Max.

2) Label Encoding & Data Normalization

Categorical features were converted into numerical values for binary classification, where 0 represents Normal and 1 represents an attack (anomaly); this represented the use of the Label Encoding Method [6]. Additionally, data normalization was performed Min-Max to map the original data to the range [0,1] with the aim of accelerating model convergence and maintaining stability during the training process.

3) Data Splitting

The data must be split for performance purposes and to evaluate the results of model training. Specifically, 80% of the total data was allocated for training and 20% was used for evaluating the results of the training [6]. This process is crucial for training; if the dataset is not split, the model's classification will be disrupted, finally causing bias and reducing evaluation performance [13].

4) SMOTE

Because both datasets have very large volumes of traffic, the potential for data imbalance is a major concern in model training, as this issue leads to evaluation bias. SMOTE addresses this by generating synthetic data that mimics the anomaly or normal patterns in the dataset. Models that do not apply this technique can see their evaluation scores drop by 46.939% to 56.560% [14]. Therefore, to avoid this issue, post-split SMOTE must be applied to address the large class ratio disparity.

D. Hybrid CNN-GRU Architecture

To understand the characteristics of heavy network traffic, this study adapted a combined deep learning model consisting of a CNN and a GRU, developed using the TensorFlow and Keras frameworks [6]. These two models were specifically designed to identify how anomalies behave on the network. They worked separately to distribute the workload evenly; the CNN focused on extracting spatial features, while the GRU learned temporal features, resulting in greater efficiency and more optimal classification results. The first step was to use Conv1D by applying 64 filters, setting the kernel size to 3, and enabling ReLU. In this first stage, the model extracted local features from network traffic data, which helped it identify the specific characteristics of the dataset [15]. In the second step, because the dataset was very large, data dimension reduction was required to minimize computational complexity by applying a MaxPooling1D layer with a pool size of 2. The function of MaxPooling1D is to produce a more efficient training process, which improves robustness, feature selectivity, and parameter efficiency [16].

The results of the convolutional layer were fed into the GRU layer, which used 64 units to model long-term temporal dependencies in IoT traffic. Overfitting is a problem in a model where it memorizes too much rather than recognizing the characteristics of a pattern; therefore, a Dropout layer was applied with a rate of 0.5. The final part of this architecture was a Dense layer that utilized 32 units and applied a single output layer with the Sigmoid activation function, mapping the input to the range [0, 1] to effectively represent binary classification probabilities [17]. The Adam algorithm was used to optimize the model with a learning rate of 0.001 and used the Binary Cross-Entropy loss function. The Adam algorithm adaptively calculates the learning rate for each parameter to optimize convergence performance [18]. In this study, 50 epochs were used for the training process with a batch size of 128 to ensure the stability of model convergence.

E. Explainability Framework (XAI)

1) SHAP

The implementation of SHAP in this study serves to evaluate the contribution of features to the predictions of the hybrid CNN-GRU model. This method is applied post-hoc to maintain the validity of the model's integration. SHAP works by providing consistent mathematical calculations to allocate the contribution of each feature to a specific prediction [19]. To ensure the validity of the SHAP application, this method was implemented using KernelExplainer with 100 randomly selected background data samples and a random seed of 42.

2) LIME

In addition to a comprehensive feature contribution analysis, this study applies the LIME to provide local and specific explanations at the individual instance level. LIME is a post-hoc interpretation method that works by generating synthetic data around a single instance and building a simple model to explain the prediction locally [8]; specifically, it generates perturbation samples around the analyzed data and trains a simple linear model as a surrogate model to simulate the actions of the complex deep learning model within that local area. The implementation of this method has used LimeTabularExpainer from the Python library.

F. Validation Strategy

In this study, a validation strategy was used to validate the training results, with 20% of the total dataset aside after splitting process for hold-out validation to prevent overfitting of the model. A random seed with a value of 42 is used to ensure that when the program is rerun, the "random" results remain the same, thus ensuring reliability.

G. Evaluation Metrics

CNN-GRU architecture processes large volumes of IoT network logs, this study uses a standard set of metrics: accuracy, precision, recall, and F1 score [6]. In addition to these measures, we need to establish how well the model learns using the AUC, which ranges from 0 to 1.

3. Results and Discussion

A. Evaluation Metrics on Two Dataset

In this section, Table 4 and Figure 2 will show a comparison of the model evaluation results from the previous study by Adefemi et al. [6] with the replication model developed in this study.

Table 4. Evaluation Metrics Result

Evaluation Metric	IoTID20 [6]	BoT-IoT [6]	IoTID20	Bot-IoT
Accuracy	99.83%	99.01%	99.36%	99.33%
Precision	99.83%	98.94%	99.79%	99.99%
Recall	99.82%	98.53%	99.51%	99.33%
F1-Score	99.83%	98.73%	99.65%	99.66%
AUC	1	0.99	0.99	0.99

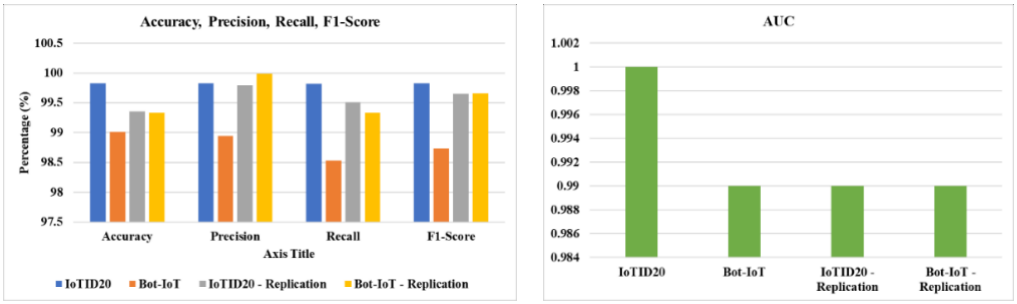

Figure 2. (a) Accuracy, Precision, Recall, and F1-Score (b) AUC

Based on Tables 4 and 2, the results of this study's replication demonstrate satisfactory performance and exhibit similarities to the previous study by Adefemi et al. [6], where the accuracy metric reached 99%, proving that both models can accurately detect anomalies in BoT-IoT and IoTID20. Supported by advanced classification abilities with an AUC value above 0.99, it can be concluded that the Hybrid CNN-GRU model is not only versatile in the learning process but also accurate in detecting anomalies, making it robust and reliable.

B. Accuracy and Loss on the Two Datasets

In this section, Figure 3 shows the model's training history over 50 epochs, visualizing the model's progression from initially learning to recognize attack patterns to accurately identify anomalies and reduce detection loss.

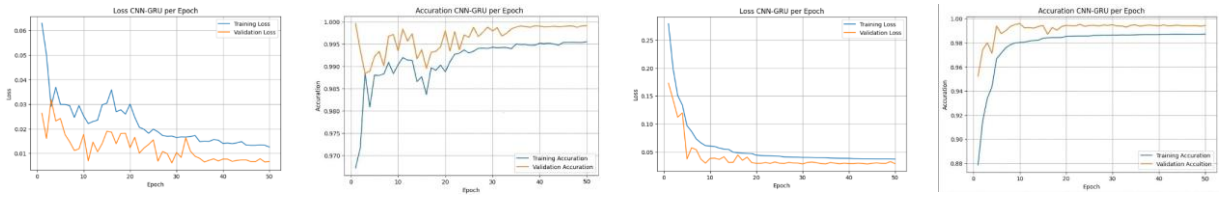

Figure 3. (a) Loss BoT-IoT (b) Accuracy BoT-IoT (c) Loss IoTID20 (d) Accuracy IoTID20

In this study, we can observe that the CNN-GRU training process achieved impressive results, with high accuracy reaching 99% on both datasets. Figures 3a and 3b present the epoch history of BoT-IoT, which shows a slight spike in the loss and accuracy curves; however, the model gradually stabilized its performance, eventually reaching the expected results. This demonstrates that the model can adaptively reduce loss and improve accuracy.

Furthermore, different from the previous results, the IoTID20 training history plots shown in Figures 3c and 3d reveal a more stable trend; by the 20th epoch, the model achieved an accuracy of over 98% and gradually reduced the loss. This indicates that the model can handle the training process on this dataset with ease, without experiencing spikes as encountered in the BoT-IoT dataset.

C. Model Performance Evaluation on the Two Datasets

In this section, the model will be explained using a confusion matrix. A confusion matrix is used to assess how well the model performs in classifying traffic as either an attack or normal traffic; sometimes a model incorrectly assigns or identifies categories based on their class. Figure 4 will show the distribution of classifications by class.

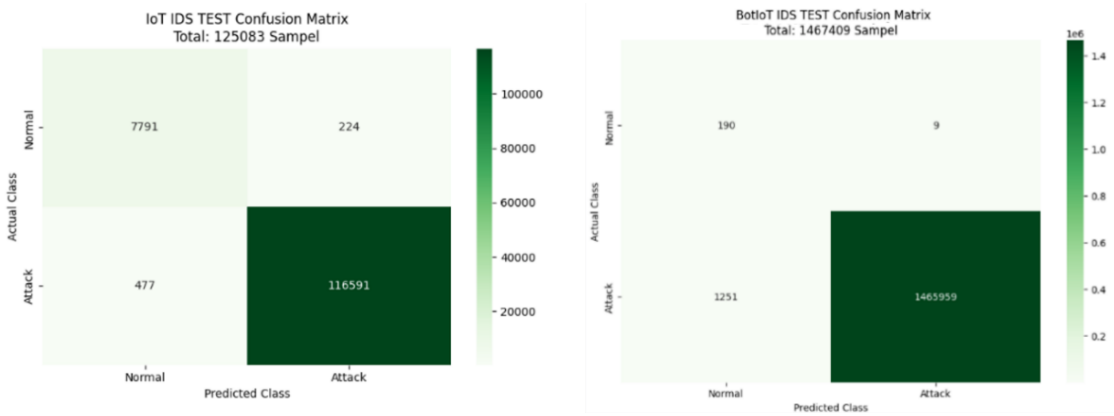

Figure 4. (a) Confusion Matrix IoTID20 (b) Confusion Matrix BoT-IoT

Based on Figure 4a, the confusion matrix analysis for IoTID20 achieved favorable results, with the model successfully classifying samples into their correct categories with a low error margin. From a total of 125,083 samples, the model successfully identified attacks (True Positives) in 116,591 samples (99.81%) and correctly identified normal traffic (True Negatives) in 7,791 samples (94.23%), demonstrating that the Hybrid CNN-GRU effectively minimizes both False Positives and False Negatives. It can be concluded that this model can reduce the risk of false alarms and breaches that are disadvantageous to the cybersecurity team.

Meanwhile, in Figure 4b, the results of the confusion matrix analysis for BoT-IoT differ from the previous ones. The difference lies in the False Positive and True Negative rates, where the model incorrectly classified normal traffic as an anomaly in 1,251 samples (86.81%), compared to 190 samples (13.19%) that were correctly identified as normal. This is not a flaw in the model's architecture, but rather due to bias in the dataset resulting from the post-split process, which has sharply different ratios of attacks and normal traffic. On the positive side, out of 1.46 million samples in the BoT-IoT dataset, successfully identified attacks (True Positives) reached 1.46 million (100%) samples, and only 9 anomalous samples (False Negatives) escaped detection. It can be concluded that this model is capable of detecting attacks in extremely dense network traffic, thereby minimizing errors and saving the cybersecurity team from constant alert fatigue.

D. Global Interpretability Analysis Using SHAP

SHAP highlights the features most frequently used to determine traffic classification. To maintain the validity of the SHAP analysis, this study employed a fixed random seed of 42 to ensure consistency of results.

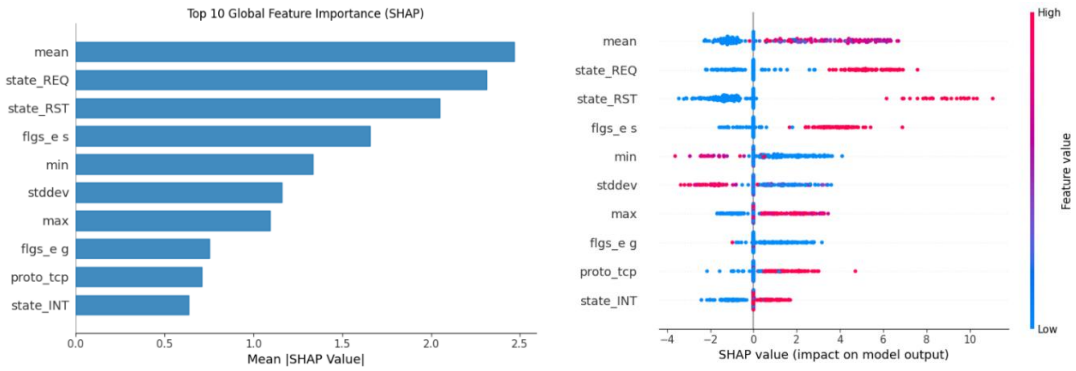

Figure 5. (a) Important features in the BoT-IoT dataset (b) Distribution of Bot-IoT Features

Based on Figure 5a, the important feature analysis of the BoT-IoT dataset reveals that the mean feature is the most critical in determining traffic classification decisions. This feature represents the average traffic duration and serves as an indicator of botnet attacks such as DDoS or Keylogging. It is followed by State_REQ as the second most important feature, where the REQ status indicates the presence of high-frequency connection requests. State_RST ranks as the third most important feature in the dataset, which in network forensics is associated with TCP reset attacks or botnet activity. These features do not illustrate how the model makes decisions, but rather indicate how much each feature contributes to the decision-making process. Figure 5b shows how the model makes decisions. The state_RST and state_REQ features show the most noticeable anomalies among all features, with red points trending to the right; from this behavior, it can be concluded that there is continuous manipulation of the TCP handshake and connection requests. Analysis of the anomaly characteristics of the most influential features proves that the model makes precise decisions based on intrusion activity in IoT.

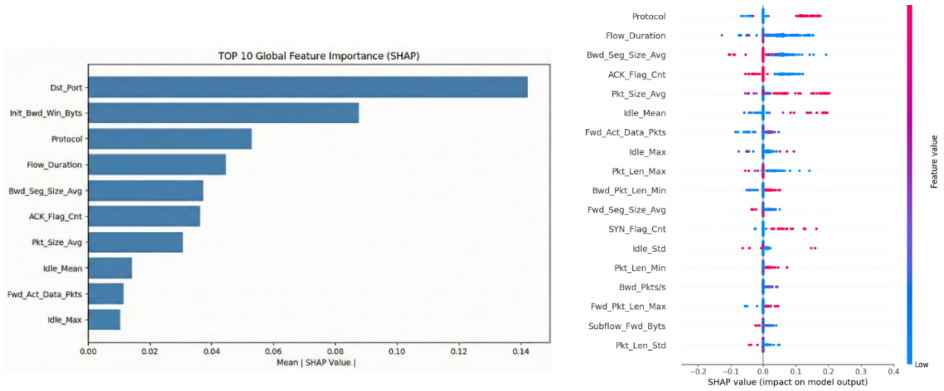

Figure 6. (a) Important features in the IoTID20 dataset (b) Distribution of IoTID20 Features

Based on Figure 6a, an analysis of key features in the IoTID20 dataset shows that the Dst_Port feature makes the greatest contribution to decision-making; it contains port services on the system and is often the target of cyberattacks. Next is Init_Bwd_Win_Byts, which relates to the window size when sending packets to a device. Finally, the Protocol is crucial in the IoT world, as devices communicate using a protocol to exchange information, and intrusions often involve the manipulation of the TCP handshake. Figure 6b illustrates how the model makes decisions. As can be seen in the Dst_Port feature, many values cluster there, but there are values indicating that this feature plays a significant role in detecting anomalies. Predicted attacks on Dst_Port typically involve scanning or brute force, which is supported by anomaly values in the protocol feature indicating the type of targeted attack corresponding to IoT traffic. Based on the above analysis, it is found that the model can identify suspicious activity critically and enable optimal mitigation.

E. Local Interpretability Analysis Using LIME

LIME unpacks how the model makes local-level decisions. To keep this analysis valid, we applied the LimeTabularExplainer in a post-hoc fashion right after the training phase

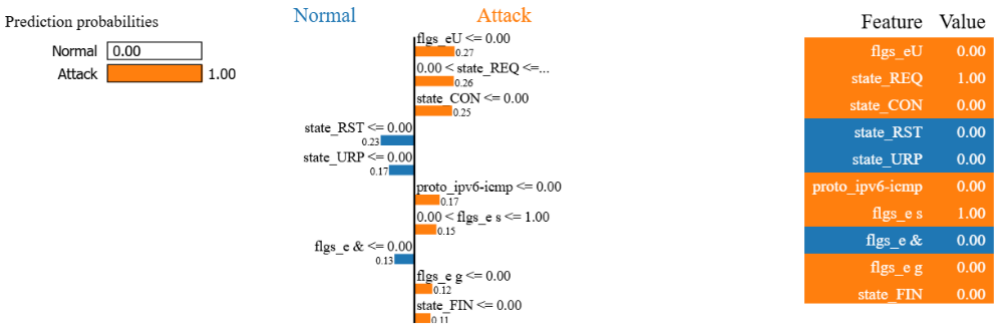

Figure 7. LIME's Explanation of Attack on BoT-IoT

Based on Figure 7, the model confirms that one of these samples is an attack with a 100% probability. The model indicates the presence of a DDoS attack, based on logical reasoning. The timeline begins with state_REQ receiving non-stop requests due to a spike in flgs_eU packet traffic indicating abnormal activity, and is further supported by state_FIN, which never received a stop request from the attacker. This proves that the model can detect anomalies in network traffic based on the actual DDoS chronology.

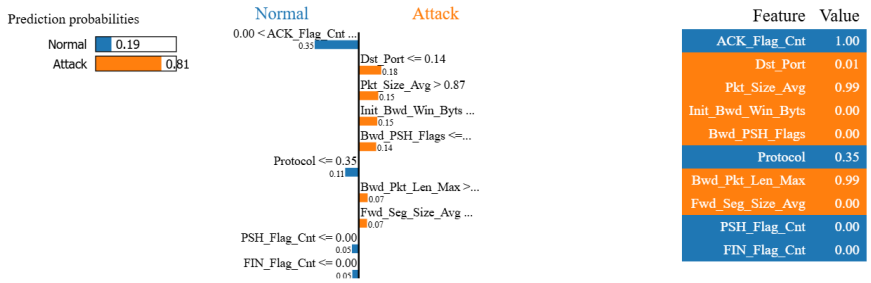

Figure 8. LIME's Explanation of Attack on IoTID20

Based on Figure 8, the model confirms the presence of a DoS attack on one of the samples from the IoTID20 dataset with an 81% probability even though it is not as high as in the previous LIME analysis, CNN-GRU can accurately explain that it is an anomaly. This is indicated by Pkt_Size_Avg reporting the presence of abnormally sized packets, and this is supported by other features related to packet size. These packets are directed to a specific port service, causing the Dst_Port feature to show an increase in attack values, proving that there is intrusion-related activity aimed at incapacitating the system. This demonstrates that the model can logically detect abnormalities in IoT network traffic based on DoS activity.

4. Conclusions

Based on the results of the research conducted, the success of the CNN-GRU hybrid model in detecting attacks on IoT networks is very satisfactory. The presence of SHAP and LIME significantly assists in model analysis for decision-making at both global and local levels. From the XAI results mentioned above, the research can answer the gaps in previous research which were still black-box.

The results of the analysis indicate that the model has forensic validity for analyzing network intrusions. The following conclusions can be drawn:

- In the BoT-IoT dataset, the model tends to focus on detecting statistical features related to duration (mean) and transaction status (state_RST and state_REQ), indicating that the model is highly sensitive to attack types involving forced disconnections and flooding.
- In the IoTID20 dataset, the adaptive model identifies network identities (Dst_port) and protocols that reflect detection patterns for targeted intrusions and scanning.
- LIME's analysis confirms the presence of anomalies based on valid criteria. LIME also determines whether a network is normal based on features that operate linearly, where no anomalies manipulating the TCP handshake or suspicious data volumes.

This study successfully addressed the black-box problem, which represented a gap in previous research. Future researchers may explore the use of newer IoT datasets as well as more advanced models to handle emerging attack variations. The findings of this study are highly valuable for mitigation planning and in-depth analysis of network behavior following an attack, making it well-suited for post-attack network forensics. As a recommendation for future research, the integration of the Hybrid CNN-GRU model with XAI methods into real-time IDS represents a strategic direction with significant research potential. Such implementation would enable devices to adaptively detect attacks and comprehend attack characteristics, thereby allowing for optimal threat mitigation in IoT environments.

References

[1] A. A. Almuqren, "Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions," *J. Cyber Secur. Risk Audit.*, vol. 1, no. 1, pp. 1–11, Jan. 2025, doi: 10.63180/jcsra.thestap.2025.1.1.

[2] A. A. Alahmadi *et al.*, "DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions," *Electronics*, vol. 12, no. 14, p. 3103, Jul. 2023, doi: 10.3390/electronics12143103.

[3] M. L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos, "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study," *Appl. Sci.*, vol. 15, no. 4, p. 1903, Feb. 2025, doi: 10.3390/app15041903.

[4] L. Diana, P. Dini, and D. Paolini, "Overview on Intrusion Detection Systems for Computers Networking Security," *Computers*, vol. 14, no. 3, p. 87, Mar. 2025, doi: 10.3390/computers14030087.

[5] R. Almuhanna and S. Dardouri, "A deep learning/machine learning approach for anomaly based network intrusion detection," *Front. Artif. Intell.*, vol. 8, Sep. 2025, doi: 10.3389/frai.2025.1625891.

[6] K. O. Adefemi, M. B. Mutanga, and O. A. Alimi, "A Hybrid CNN–GRU Deep Learning Model for IoT Network Intrusion Detection," *J. Sens. Actuator Networks*, vol. 14, no. 5, p. 96, Sep. 2025, doi: 10.3390/jsan14050096.

[7] V. Sharma and M. Kumar, "Improving Intrusion Detection with Hybrid Deep Learning Models: A Study on CIC-IDS2017, UNSW-NB15, and KDD CUP 99," 2024. [Online]. Available: https://www.jisem-journal.com/

[8] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron," *IEEE Access*, vol. 12, pp. 30164–30175, 2024, doi: 10.1109/ACCESS.2024.3368377.

[9] T. B. Ogunseyi and G. Thiyagarajan, "An Explainable LSTM-Based Intrusion Detection System Optimized by Firefly Algorithm for IoT Networks," *Sensors*, vol. 25, no. 7, p. 2288, Apr. 2025, doi: 10.3390/s25072288.

[10] T.-T.-H. Le, H. Kim, H. Kang, and H. Kim, "Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method," *Sensors*, vol. 22, no. 3, p. 1154, Feb. 2022, doi: 10.3390/s22031154.

[11] ITedyyana, A., & Ghazali, O. 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.

[12] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards Developing Network forensic mechanism for Botnet Activities in the IoT based on Machine Learning Techniques."

[13] V. R. Joseph and A. Vakayil, "SPlit: An Optimal Method for Data Splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, 2022, doi: 10.1080/00401706.2021.1921037.

[14] Ratnawati, F., & Tedyyana, A. (2026). Multiclass IoT Intrusion Detection Based on Particle Swarm Optimization-Tuned Light Gradient Boosting Machine. *Journal of Embedded Systems, Security and Intelligent Systems*, 49-63..

[15] M. Cheng, J. Yang, T. Pan, Q. Liu, and Z. Li, "ConvTimeNet: A Deep Hierarchical Fully Convolutional Model for Multivariate Time Series Analysis," Dec. 2024, [Online]. Available: http://arxiv.org/abs/2403.01493

[16] A. Sun, W. Hong, J. Li, and J. Mao, "An Arrhythmia Classification Model Based on a CNN-LSTM-SE Algorithm," *Sensors*, vol. 24, no. 19, Oct. 2024, doi: 10.3390/s24196306.

[17] V. Shatravin, D. Shashev, and S. Shidlovskiy, "Sigmoid Activation Implementation for Neural Networks Hardware Accelerators Based on Reconfigurable Computing Environments for Low-Power Intelligent Systems," *Appl. Sci.*, vol. 12, no. 10, May 2022, doi: 10.3390/app12105216.

[18] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," *Neural Comput. Appl.*, vol. 35, no. 23, pp. 17095–17112, Aug. 2023, doi: 10.1007/s00521-023-08568-z.

[19] C. E. Ben Ncir, M. A. Ben HajKacem, and M. Alattas, "Enhancing intrusion detection performance using explainable ensemble deep learning," *PeerJ Comput. Sci.*, vol. 10, 2024, doi: 10.7717/PEERJ-CS.2289.