INFORMATIK JOURNAL, Volume 21, No 3, 2025

Towards Interpretable Intrusion Detection: A Double-Layer GRU with Feature Fusion Explained by SHAP and LIME

M. Rozikul Wijaya[1*], M. Hanafi[2],

[1] *Department of Computer Science, University of Amikom Yogyakarta, Depok, Sleman, 55283, Indonesia*
[2] *Department of Computer Science, University of Amikom Yogyakarta, Depok, Sleman, 55283, Indonesia*

Article Info

Article history:

Received Aug 20, 2025
Revised Aug 30, 2025
Accepted Sep 01, 2025

Keywords:
Intrusion Detection System
Double Layer GRU
Feature Fusion
Explainable Artificial Intelligence (XAI)
SHAP LIME

ABSTRACT

Computer network security has become increasingly important with the growing complexity of cyberattacks. Deep learning-based Intrusion Detection Systems (IDS) represent a potential solution due to their capability to capture sequential patterns in network traffic. This study proposes a Double-Layer GRU-based IDS with Feature Fusion to enhance the representation of both numerical and categorical data in the NSL-KDD dataset. The training process employs systematic preprocessing techniques, including normalization and one-hot encoding. Experimental results demonstrate high accuracy and generalization with stable performance on both training and testing data, as well as competitive macro F1-scores for multi-class attack detection. Furthermore, interpretability aspects are explored through Explainable Artificial Intelligence (XAI) methods using SHAP and LIME. SHAP provides global insights into the contributions of important features, while LIME explains the influence of features at the local level for individual predictions. The integration of both methods not only enhances transparency and trust in the IDS but also offers deeper insights into dominant attributes in detecting attack patterns. Accordingly, this study contributes to the development of IDS that are accurate, interpretable, and applicable to modern network security.

Corresponding Author:

M. Rozikul Wijaya
Department
University of Amikom Yogyakarta,
Sleman, Yogyakarta, Indonesia
Email: rozikul.wijaya@students.amikom.ac.id

I. INTRODUCTION

Computer network security has become one of the crucial aspects in the digital era, marked by increasingly complex cyberattacks. One of the widely developed mechanisms for detecting such attacks is the Intrusion Detection System (IDS) [1].

In recent years, the application of deep learning in IDS has shown promising results due to its ability to extract complex patterns from network data [2]-[3]. One of the commonly used architectures is the Gated Recurrent Unit (GRU), particularly effective for sequential network traffic data [4]-[5]. However, most studies remain focused on employing a single-layer GRU, while the Double-Layer GRU with Feature Fusion approach has been proven to provide richer representations and improved detection performance.

Although the performance of deep learning-based IDS models continues to improve, significant challenges arise regarding interpretability, as these models are inherently black-box in nature, making it difficult to understand the reasoning behind each prediction [6]. This issue is particularly critical in the context of network security, where administrators need to identify the contribution of specific features to the classification of attacks or normal traffic.

To address this challenge, Explainable Artificial Intelligence (XAI) approaches have been increasingly applied. Two widely used methods are SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations). SHAP enables both global and local analysis based on game theory, while LIME focuses on providing local explanations for model predictions [7]. By combining these methods, a more comprehensive understanding of the behavior of the GRU-based IDS can be achieved.

The NSL-KDD dataset is employed in this study as an enhancement of KDD'99 designed to reduce data duplication, with multi-class labels (Normal, DoS, Probe, R2L, U2R), and a more balanced distribution [4]. These advantages make NSL-KDD remain relevant as a benchmark for evaluating deep learning models and the interpretability of IDS.

This study aims to develop a Double-Layer GRU-based Intrusion Detection System (IDS) with Feature Fusion using the NSL-KDD dataset. In addition, it analyzes the interpretability of the model through SHAP and LIME methods, and compares their effectiveness in explaining feature contributions to attack classification [8].

The main contributions of this research are the design of an IDS architecture that leverages Double GRU with Feature Fusion to strengthen sequential representations, and the interpretability analysis of IDS by combining SHAP and LIME, covering both global and local perspectives. This study also provides insights into the key features of the NSL-KDD dataset that play a significant role in attack detection [9].

Modern IDS research extensively explores sequential architectures such as RNNs and GRUs to capture temporal dependencies in network traffic. Studies on Double-Layer GRU with feature fusion demonstrate that combining statistical and sequential representations can improve multi-class detection on common benchmarks compared to single approaches [10]. Consequently, wrapper/ensemble approaches for feature selection and fusion have also proven effective in reducing dimensionality while retaining relevant features for NIDS, thereby improving both accuracy and efficiency. Furthermore, ensemble methods in IoT ecosystems integrate advanced learning and feature selection techniques, ensuring competitiveness in heterogeneous scenarios such as NSL-KDD and UNSW-NB15 [11].

GRU is often chosen for its efficiency in handling long sequences compared to LSTM, and several studies indicate that double or stacked layers enrich temporal representations, especially when combined with other feature channels (feature fusion). Recent IDS frameworks also propose pipelines that integrate feature extraction, selection/filtering, and sequential classification to achieve a balance between accuracy and computational cost [12].

The growing performance of black-box models has intensified the need for XAI in IDS. Recent surveys emphasize the importance of explaining IDS decisions for SOC analysts and mapping XAI techniques (both global and local) applicable in this domain [11]-[12]. Methodological level, SHAP offers additive consistency based on game theory for local–global feature attribution, while LIME provides local explanations through interpretable models trained around the target instance. Applied studies have shown that SHAP and LIME facilitate decision auditing, feature selection, and threat validation in security scenarios such as IDS and malware analysis [13]-[15]

The NSL-KDD dataset remains an important benchmark due to its improvements over KDD'99 (reduced redundancy and controlled difficulty), its multi-class labels, and its wide availability for experimental replication. Many recent IDS studies continue to report results on NSL-KDD (often alongside CIC/UNSW) to evaluate the generalization of new architectures, including those based on CNN/RNN/GRU, ensembles, and feature fusion .Nevertheless, the literature highlights limitations in distribution gaps compared to modern traffic, thereby encouraging some studies to combine datasets/features and conduct multi-dataset evaluations as well as XAI-based analysis to ensure more reliable findings [12].

Based on the literature review, several research gaps can be identified. First, studies that explicitly implement a Double-Layer GRU architecture with a feature fusion approach and perform a comprehensive evaluation on the NSL-KDD dataset remain very limited [16]. Second, research comparing SHAP and LIME interpretability methods in sequential network-based intrusion detection models is relatively rare, with most studies relying on only one XAI approach [17]-[18].Third, there is a lack of integrated pipelines that provide transparency through both global and local feature analysis [19].

II. METHODOLOGY

A. Dataset NSL-KDD

Data Description and Volume. NSL-KDD is an improved version of KDD'99 that reduces redundancy and controls difficulty levels, making it more representative as an IDS benchmark. The official release provides four files: KDDTrain+, KDDTrain+_20Percent, KDDTest+, and KDDTest-21. The commonly used main subsets consist of 125,973 records for KDDTrain+ and 22,544 for KDDTest+. The dataset contains 41 features (38 numerical and 3 categorical: protocol_type, service, flag) along with class labels (Normal, DoS, Probe, R2L, U2R) [20].

B. Pre-processing

Data Preprocessing. The preprocessing stage in this study was conducted through several systematic steps to ensure dataset quality and consistency. First, data cleaning was performed by removing corrupted rows and irrelevant columns, leaving only 41 features along with the target label. Next, categorical features such as protocol_type, service, and flag were encoded using one hot encoding or embedding techniques, particularly to support the categorical GRU branch in the model architecture. For numerical features, a total of 38 attributes were retained.

In addition, the class imbalance problem in the dataset particularly in minority attack categories such as R2L and U2R was addressed by applying class weights or, alternatively, focal loss, which has been reported in several studies to be effective in feature fusion-based IDS [1]. The data partitioning process followed the official NSL-KDD scheme, using KDDTrain+ as the training set and KDDTest+ as the testing set. From the training data, 10%-20% was set aside as a validation set using stratified sampling to preserve class proportions.

C. *Double Layer GRU with Feature Fusion Architecture*

The proposed Double-Layer GRU with Feature Fusion architecture is designed to enhance the capability of intrusion detection systems in capturing complex patterns within network data. By utilizing two stacked GRU layers, the model strengthens the sequential representation of numerical data, while the categorical pathway is processed through an embedding mechanism to represent symbolic information in a more compact form. These two processing branches are subsequently merged using a feature fusion technique, resulting in a richer and more informative joint representation. This approach not only improves classification accuracy but also provides flexibility in modeling the heterogeneous characteristics of the NSL-KDD dataset.

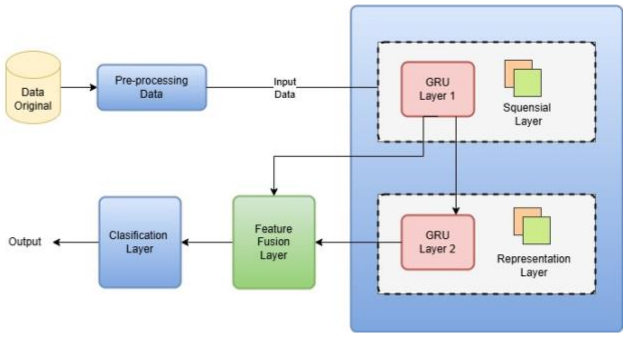


Fig. 1. Double-Layer GRU with Feature Fusion Architecture.

Technically, Fig. 1 illustrates the Double-Layer GRU architecture consisting of two parallel GRU branches:

1. Input Layer: At the initial stage, the preprocessed network traffic data is divided into two main feature groups: the sequential layer and the representation layer. The sequential layer contains numerical features that are continuous or ordinal in nature (e.g., duration, src_bytes, dst_bytes), making them suitable to be treated as sequential data. In contrast, the representation layer consists of categorical features (e.g., protocol_type, service, flag), which have been transformed through embedding to obtain compact representations that can be effectively processed by neural networks. This separation is intended to optimize the role of each feature type according to its characteristics, enabling the numerical branch to capture temporal patterns while the categorical branch enriches the contextual representation of the data.

2. GRU Sequential: This branch is specifically designed to extract temporal information from numerical data processed as sequential series. The input data is structured and then fed into a GRU layer. The first GRU layer (e.g., 128 units) functions as a temporal feature extractor, capturing sequential dynamics of network activities such as normal traffic patterns and intrusion behaviors. The output from the first GRU layer is subsequently passed into the second GRU layer (e.g., 64 units), which further deepens the abstraction of temporal representations. Through the use of these two GRU layers (Double Layer), the model gains a multi-level understanding of complex temporal patterns, thereby becoming more robust in detecting sequence-dependent attacks such as DoS or probing.

3. GRU Representation: In contrast to the sequential branch, the GRU Representation branch focuses on the integration and enrichment of feature representations. This branch can receive two types of input: (a) the output from the first Sequential GRU layer, which contains temporal representations, and (b) the original input data or categorical features that have been embedded. By combining these two sources of information, the branch constructs richer and more comprehensive feature representations. This process ensures that the model does not solely focus on sequential patterns but also incorporates the symbolic context of categorical data. A GRU or MLP layer within this branch serves as an additional processing stage, balancing the contributions of both types of information.

4. Feature Fusion Layer: This stage represents the core of the feature fusion architecture. The outputs from the GRU Sequential branch and the GRU Representation branch are combined using concatenation to form a fused representation. This fused representation incorporates both temporal and symbolic information, thereby providing a holistic view of network traffic data. Subsequently, dense layers with a certain number of neurons are applied to integrate and strengthen feature interactions. These layers are often equipped with dropout as a regularization technique to enhance the model's resilience against overfitting. Through this approach, the fusion results improve the model's generalization capability compared to a single-stream pathway.

5. Output Layer: The fused representation produced by the fusion layer is passed into the classification layer to generate the final prediction. Typically, the last dense layer employs a softmax activation function (for multi-class classification, e.g., Normal, DoS, Probe, U2R, R2L). This function transforms the numerical representation into class probabilities, thereby facilitating the interpretability of results. To improve detection accuracy, the model training process is regulated using regularization strategies such as early stopping based on macro-F1 scores on the validation set, which is particularly relevant given the imbalanced nature of IDS data. Thus, the output layer not only provides predictions but also ensures fairer classification for minority classes.

D. *Training and Evaluation*

The training and evaluation of the model were conducted systematically to ensure optimal and reliable performance. The data partitioning followed the official NSL-KDD scheme, where KDDTrain+ was used as the training set, with 10%-20% stratified sampling reserved as the validation set, and KDDTest+ employed as the final testing set. Training parameters were determined based on preliminary experiments to balance accuracy and computational efficiency. The Adam optimizer was selected for its ability to accelerate convergence through adaptive learning rate updates. Meanwhile, categorical cross-entropy was employed as the loss function, as it is well-suited for multi-class classification scenarios in IDS.

Model performance evaluation was carried out using multiple metrics to provide a comprehensive assessment of classification quality. In addition to accuracy, precision, recall, and F1-score were used to evaluate the model's correctness and sensitivity in attack detection. F1-score, particularly in its macro form, was chosen as the primary metric because it effectively addresses class imbalance commonly found in IDS datasets, especially in minority attack categories such as R2L and U2R. With this combination of metrics, model performance analysis becomes more comprehensive, assessing not only overall accuracy but also the model's effectiveness in detecting various types of attacks, as illustrated in Fig. 2.

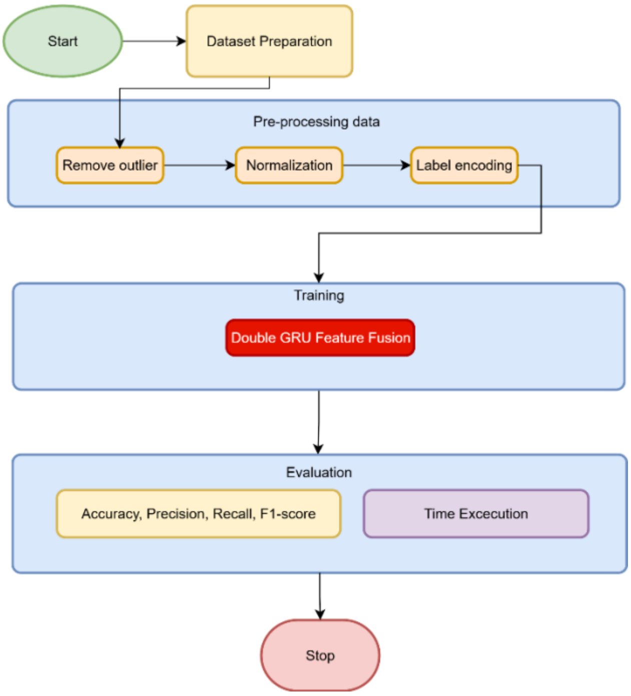


Fig. 2. Experimental Scheme Flow

E. *Interpretability Analysis*

The interpretability analysis in this study was conducted using two primary approaches: SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations), with the main objective of gaining an in-depth understanding of the contribution of each feature to the model's decisions. SHAP was selected due to its strong theoretical foundation in cooperative game theory, where each feature is treated as a "player" that contributes to the overall outcome. In the context of IT security, this perspective allows us to quantify how much each network traffic attribute (e.g., protocol type, connection count, or byte volume) contributes to the detection of an intrusion. Thus, SHAP not only provides an objective measure of feature importance but also translates the abstract principles of game theory into actionable insights for identifying the most critical indicators of malicious behavior.

In the context of global interpretation, SHAP highlights the key features that dominate the classification process across the entire dataset, allowing researchers to identify which features are consistently important in distinguishing between normal and attack traffic. For instance, features such as src_bytes, service, or flag may emerge as primary indicators in differentiating access patterns. At the local level, SHAP provides case-specific explanations regarding how a particular combination of features contributes to the prediction of a single instance. This is particularly important in IDS, as it allows case by case analysis where model decisions can be understood in detail rather than solely based on overall trends. Thus, SHAP serves as a comprehensive interpretability tool, covering both global insights and localized understanding of model behavior.

In addition to SHAP, this study also employs LIME as an interpretability approach focused on the local level. Unlike SHAP, which computes feature contributions based on game theory, LIME works by constructing a simple linear model around a specific instance to approximate the behavior of the complex model in that local region of the data. As such, LIME is able to explain why a given data point is classified into a particular category such as Normal, DoS, Probe, R2L, or U2R. This method is effective because it is model agnostic, meaning it can be applied to various types of models, including complex architectures like the Double Layer GRU with feature fusion.

One of the advantages of LIME lies in its ability to produce explanations that are more intuitive and human readable, particularly for network security practitioners who may not have a deep background in deep learning. In this study, LIME complements SHAP by providing a clearer local perspective on the influence of individual features in specific cases for example, how values of protocol_type or dst_bytes contribute to intrusion detection. By combining these two methods, the study not only evaluates IDS performance in terms of prediction accuracy but also ensures transparency and accountability in decision-making. Such transparency is crucial to increase user trust and to provide deeper insights into the underlying factors that drive network attacks within the NSL-KDD dataset.

III. Result And Discussion

A. *IDS Model Results*

After completing the data preprocessing stage, architectural design, and training process using the NSL-KDD dataset, this study produced a Double-Layer GRU-based Intrusion Detection System (IDS) with feature fusion. To evaluate the model's performance, accuracy and loss were assessed on both training and testing data across varying numbers of epochs. This evaluation aims to examine the consistency of the model's performance in detecting attacks as well as its ability to generalize to previously unseen data. The following table presents the model's accuracy and loss results at several training epoch checkpoints.

Table I. Experimental Results

Epoch	Accuracy Train	Accuracy Test	Loss Train	Loss Test
20	99,46	99,48	1,42	1,35
60	99,51	99,52	1,44	1,39
100	99,58	99,55	1,34	1,47

Based on the training results presented in Table 1, it can be observed that the Double-Layer GRU with feature fusion demonstrates strong consistency between the training and testing datasets. At the 20th epoch, the training accuracy reached 99.46% with a testing accuracy of 99.48%, accompanied by loss values of 1.42 (training) and 1.35 (testing). As the number of epochs increased to 60, the model accuracy improved to 99.51% for training and 99.52% for testing, with loss values of 1.44 (training) and 1.39 (testing).

At the 100th epoch, the accuracy reached 99.58% for training and 99.55% for testing, while the testing loss slightly increased to 1.47. The maximum accuracy gap between training and testing remained below 0.1%, which strongly indicates that the model does not suffer from significant overfitting and maintains good generalization. The slight increase in testing loss at later epochs reflects a normal variation in optimization and does not compromise the stability of the model's performance.

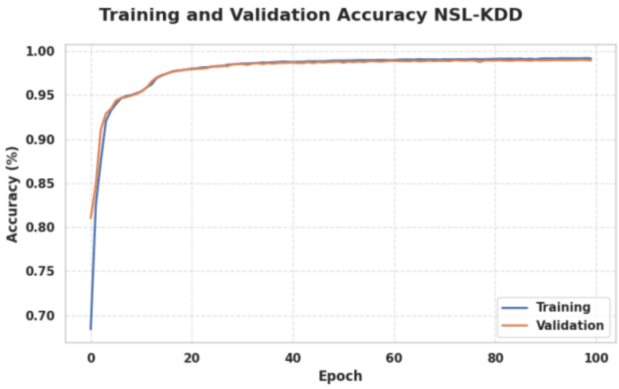


Fig. 3. Training and Validation Accuracy NSL-KDD

Fig. 3 shows the training and validation accuracy curves, which rise sharply in the first 20 epochs and then stabilize above 99% until the end of training. The close alignment of these curves confirms that the risk of overfitting is minimal. Figure 4 presents the training and validation loss curves, which both decrease rapidly in the early epochs and stabilize close to zero. The small gap between the two loss curves further reinforces that the proposed model generalizes well to unseen data. Moreover, regularization techniques such as dropout and early stopping were applied during training to further reduce the risk of overfitting. These findings demonstrate that the proposed Double-Layer GRU with feature fusion effectively captures attack patterns with high accuracy while preserving strong generalization capability.

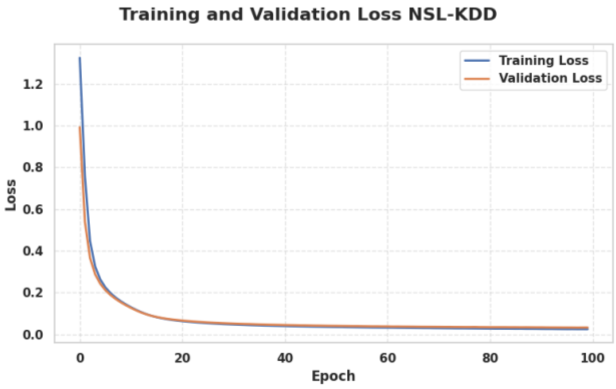


Fig. 4. Training dan Validation Loss NSL-KDD

Fig. 4 presents the training and validation loss of the IDS model on the NSL-KDD dataset over 100 training epochs. It can be observed that the loss values in both curves decrease sharply during the first 20 epochs and then gradually stabilize, approaching zero toward the end of training. The close alignment between the training loss and validation loss curves indicates that the model does not exhibit significant overfitting, but rather is able to learn effectively with good generalization. This result reinforces that the Double-Layer GRU with feature fusion architecture successfully optimizes the learning process for accurate intrusion detection.

B. *SHAP Analysis (Global Interpretability)*

High model accuracy alone is not sufficient, as predictions without transparent explanations are difficult to trust and apply in real-world environments. Explainable AI methods such as SHAP (SHapley Additive exPlanations) provide insights into how each feature contributes to the model's predictions. Beyond describing feature importance, SHAP can guide security practitioners in prioritizing monitoring and response strategies. For example, identifying features with consistently high impact on DoS predictions (e.g., F32, F30, F36, and F26) allows network administrators to focus on those traffic patterns for early detection. Similarly, understanding which features dominate the classification of rare attack types like R2L or U2R can inform the design of tailored detection rules or alert thresholds, improving operational efficiency. Thus, SHAP offers not only transparency but also actionable intelligence, enabling practitioners to interpret model outputs in the context of network security decisions and threat mitigation.

Figure 5–9 below show the SHAP Summary Plots for each attack class in the NSL-KDD dataset. These plots highlight the most influential features per class, revealing patterns that can be monitored or mitigated in real networks. Positive SHAP values indicate features that increase the likelihood of a class prediction, while negative values reduce it. By linking feature impact to concrete security measures.

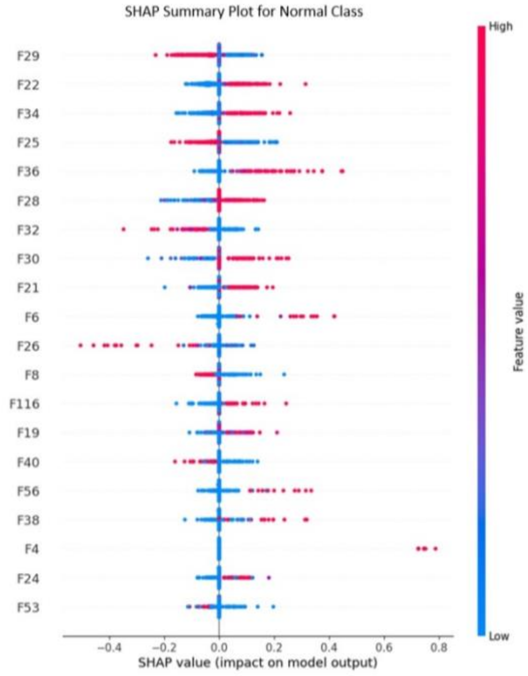


Fig. 5. SHAP Summary Plot for Normal Class

Fig. 5 presents the SHAP Summary Plot for the Normal class in the NSL-KDD dataset. This plot illustrates the contribution of features (such as F21, F22, F25, F29, F34) to the model's prediction in classifying network traffic as normal. Positive SHAP values increase the probability of being classified as Normal, while negative values decrease it. The red color represents high feature values, whereas blue indicates low values. It can be observed that features F29, F22, and F34 have a dominant influence in the classification process.

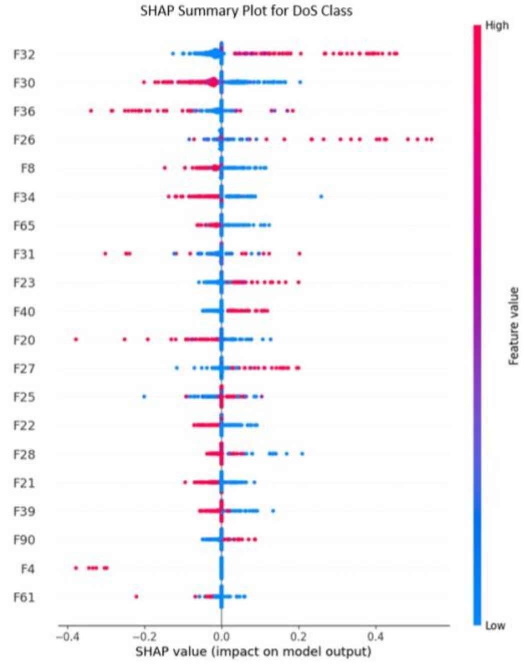


Fig. 6. SHAP Summary Plot for DoS Class

Fig. 6 displays the SHAP Summary Plot for the DoS attack class. The horizontal axis represents the impact of features on the output, where positive SHAP values drive predictions toward DoS, while negative values reduce them. The color of the points indicates feature values (blue = low, red = high). Features F32, F30, F36, and F26 appear dominant due to their wider SHAP value distributions, indicating their significant role in the model's decision-making process.

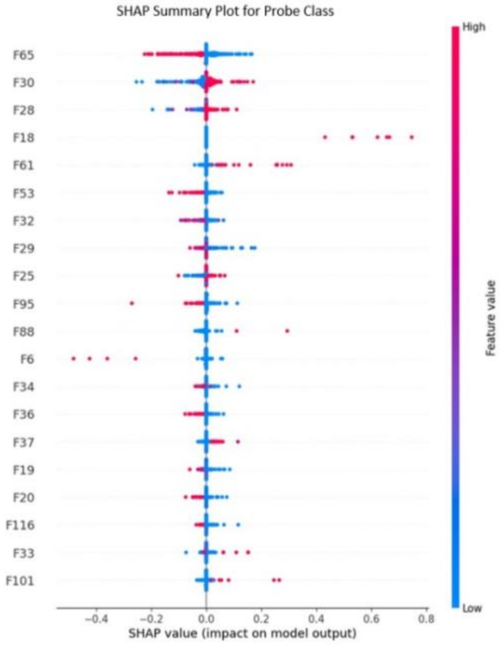


Fig. 7. SHAP Summary Plot for Probe Class

Fig. 7 presents the SHAP Summary Plot for the Probe class. Features such as F65, F30, and F28 appear dominant in influencing predictions. Positive SHAP values drive the model toward the Probe class, while negative values shift predictions toward other classes, thereby facilitating the global interpretation of the model.

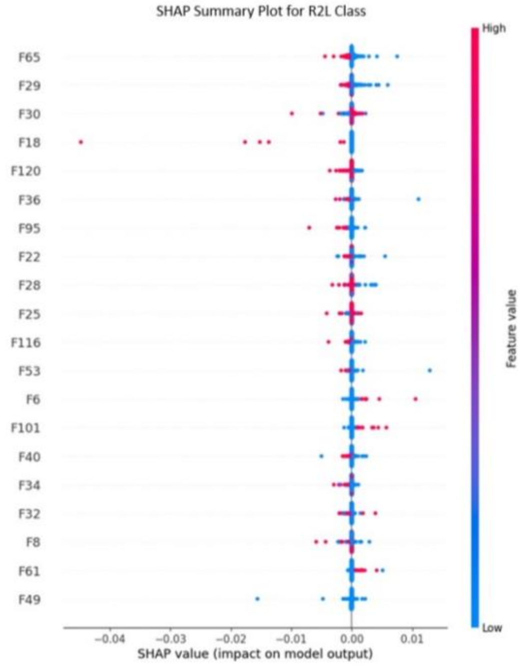


Fig. 8. SHAP Summary Plot for R2L Class

Fig. 8 presents the SHAP Summary Plot for the R2L class, illustrating the influence of features on the model's predictions. Each point represents the SHAP value of an instance, where red indicates high feature values and blue indicates low values. Features such as F65, F29, F30, and F18 appear to have the most significant impact, albeit with a relatively small magnitude. The direction of the SHAP values reflects whether a feature contributes to driving the model toward predicting the R2L class or otherwise.

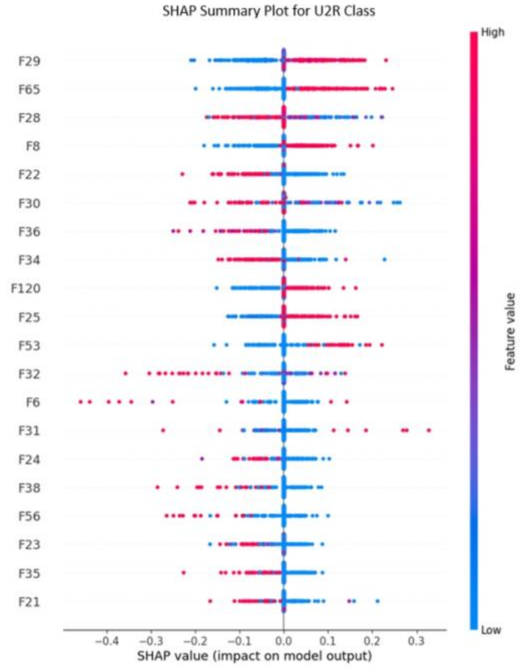


Fig. 9. SHAP Summary Plot for U2R Class

Figure 9 illustrates the SHAP Summary Plot for the U2R class, depicting the contribution of each feature to the model’s predictions. Features F29, F65, F28, and F8 appear to have a dominant influence on the outcomes. Red points indicate high feature values, while blue points indicate low values. Positive or negative SHAP values show whether a feature increases or decreases the likelihood of predicting the U2R class, thereby aiding in understanding the important patterns recognized by the model.

The SHAP Summary Plots presented in Figures 5–9 do not only highlight the most influential features for each class, but also provide actionable insights for security practitioners. By identifying which features have the strongest impact on predictions, network administrators can prioritize monitoring and mitigation strategies.

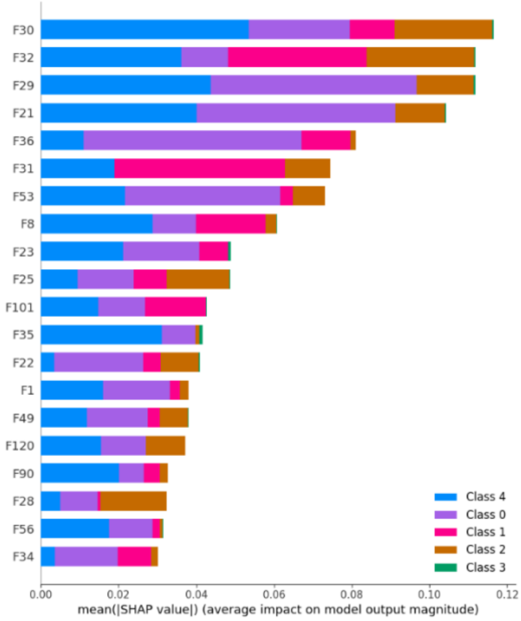


Fig. 5. SHAP feature importance for model IDS

Fig. 10 above illustrates the SHAP feature importance analysis of the Double-Layer GRU-based IDS with Feature Fusion on the NSL-KDD dataset.

1. The horizontal axis represents the mean absolute SHAP values (mean SHAP value), reflecting the magnitude of each feature's contribution to the model output.
2. The vertical axis displays the features ranked according to their highest contribution. It can be observed that F30, F32, F29, and F21 are the dominant features that most influence classification decisions.
3. The color of the bars indicates the distribution of feature contributions across different attack classes, for example, Class 0 (Normal), Class 1 (DoS), Class 2 (Probe), Class 3 (R2L), and Class 4 (U2R).
4. The distribution patterns show that the same feature can contribute differently to multiple classes. For instance, F36 plays a significant role in Class 0 and Class 1, whereas F30 is more influential for Class 4 and Class 2.

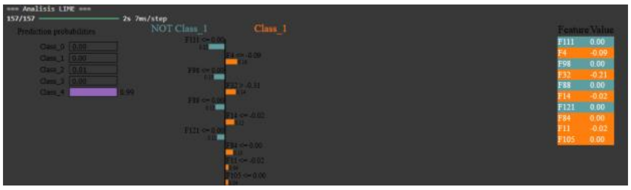
These results confirm that the model does not rely on a single specific feature, but rather utilizes a combination of multi-dimensional features to distinguish attacks. Furthermore, SHAP based interpretations provide global insights into critical features, serving as a reference for network administrators to understand attack patterns and to validate the decisions made by the IDS model.

Overall, the SHAP analysis translates model predictions into practical guidance, allowing security teams to focus resources on the most impactful traffic features, design targeted detection rules, and improve the efficiency of incident response. This ensures that high model accuracy is complemented by actionable intelligence in real-world network security operations.

C. LIME Analysis (Local Interpretability)

In addition to global interpretability, it is also important to understand the reasoning behind the model's prediction for a specific sample. This is particularly relevant in the context of Intrusion Detection Systems (IDS), as each prediction whether traffic is classified as normal or malicious needs to be explainable so that network administrators can comprehend the basis of the model's decision making. One widely used method for this purpose is LIME (Local Interpretable Model-agnostic Explanations).

LIME operates by creating a simpler local approximation of the model around the data being analyzed. Consequently, LIME can highlight which features contribute most to a specific prediction, along with the direction of their influence (positive or negative). This analysis is especially useful for verifying or auditing model predictions, for instance, to ensure that attack classifications are not driven by model bias or errors. Therefore, LIME serves as an important complement to SHAP, providing deeper explanations at the individual level (local interpretability).



```
=== Analisis LIME ===
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 7ms/step
```

Prediction probabilities

Class_0	0.00
Class_1	0.00
Class_2	0.01
Class_3	0.00
Class_4	0.99

Feature	Value
F111	0.00
F4	-0.09
F98	0.00
F32	-0.21
F88	0.00
F14	-0.02
F121	0.00
F84	0.00
F11	-0.02
F105	0.00

Fig. 6. Local Interpretation results using the LIME method

Fig. 11 above illustrates the local interpretation results using the LIME method on a sample from the Double Layer GRU based IDS model.

1. Left section shows the class prediction probabilities. The model assigns the highest prediction to Class 4 (0.99), while probabilities for the other classes are relatively low (Class 0, Class 1, Class 2, Class 3 < 0.01).
2. Middle section illustrates the feature contributions to the prediction, divided as follows.
3. NOT Class 1 (gray) features that decrease the likelihood of the sample being classified as Class 1.
4. Class 1 (orange) features that push the prediction toward Class 1.
5. It can be observed that features such as F111, F74, and F90 have strong negative contributions (shifting the prediction away from Class 1), whereas features F78, F221, and F105 provide small contributions toward Class 1.
6. Right section displays the actual feature values of the analyzed sample, which serve as the basis for calculating feature contributions in the model.

This result indicates that LIME is capable of explaining model decisions at the individual instance/sample level. Such explanations complement SHAP analysis (which is global) by providing insights into why the model produces specific predictions. Consequently, network administrators can understand the precise reasons behind the classification of a packet or network connection, which is crucial for validating IDS decisions and enhancing user trust.

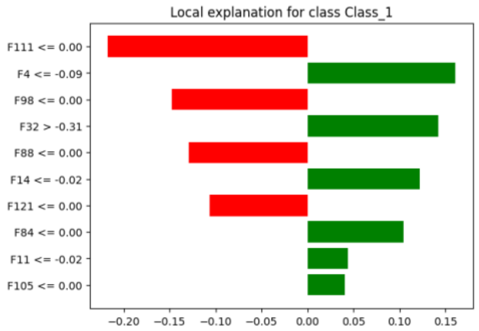


Fig. 7. Local Explanation results for Class 1

Fig. 12 shows the local interpretation results using LIME for an instance predicted as Class 1, where green bars indicate features supporting the classification and red bars indicate opposing features. Features F4, F32, F14, F84, F11, and F105 contribute positively, promoting the prediction, while F111, F98, F88, and F121 contribute negatively, steering the model away from Class 1. The bar lengths represent each feature's relative influence, with F4 (≤ -0.09) and F32 (> -0.31) being the strongest positive factors and F111 (≤ 0.00) the strongest negative factor. This concise visualization allows security analysts to understand the key factors behind specific IDS predictions, enhancing transparency and trust in the system.

The model interpretation results in Figures 10 through 12, both globally using SHAP and locally using LIME, have significant implications for real-world practice, particularly in network security management. By using SHAP, administrators can understand which features consistently influence the model's predictions across the entire dataset, thereby helping to prioritize monitoring or threat mitigation. For example, if SHAP indicates that certain features (e.g., F32 or F74) have a substantial impact on detecting DoS attacks, administrators can focus on monitoring or strengthening protection for those network parameters.

On the other hand, local interpretation using LIME allows for a more detailed understanding of each individual decision made by the IDS. Figures 11 and 12 illustrate how specific features either push or inhibit the classification of a network packet or connection into a particular attack class. In practice, this information is highly useful for auditing and verifying the model's predictions. For instance, if a network packet is classified as an R2L attack (Class 1), administrators can examine the features that contribute most significantly to that decision. This enables them to assess whether the model's prediction is contextually reasonable or potentially influenced by data bias.

IV. Conclusion

A. Conclusion

This study proposes a Gated Recurrent Unit (GRU) based Intrusion Detection System (IDS) integrated with Explainable Artificial Intelligence (XAI) methods, specifically SHAP and LIME, to enhance the interpretability of intrusion detection systems. Experiments on the NSL-KDD dataset demonstrate that the proposed IDS model can achieve competitive performance, as indicated by key metrics such as high accuracy, precision, recall, and F1-score. Furthermore, the integration of XAI methods provides significant added value: SHAP proves effective in delivering a global view of feature importance, facilitating the analysis of attack patterns and the identification of the most influential attributes in classification. Meanwhile, LIME contributes to local interpretability by offering intuitive explanations of the model's decisions at the individual sample level. Consequently, this study not only emphasizes the effectiveness of GRU as a sequence-based IDS model but also highlights the importance of XAI integration in improving transparency, trustworthiness, and providing a foundation for the development of IDS that is both accurate and explainable.

B. Future Directions

Although the results of this study are promising, several avenues remain open to strengthen the contribution of the proposed IDS model. Future work will focus on evaluating the model using modern and more representative datasets, such as CIC-EVSE2024 and CIC-IoV2024 from the Canadian Institute for Cybersecurity, to validate its generalization on contemporary network traffic patterns. In addition, performance assessment will be extended to include computational cost and scalability analysis, particularly in real-time intrusion detection scenarios. The integration of additional XAI techniques, such as Integrated Gradients, DeepLIFT, and Counterfactual Explanations, will also be explored to enrich interpretability and provide multi-perspective insights for practitioners. Furthermore, the development of hybrid IDS that combines deep learning with statistical or rule-based approaches may offer a more balanced trade-off between accuracy, efficiency, and interpretability. Finally, the deployment of the proposed model in real-world environments will be an important step to assess its applicability and practical value in supporting daily cybersecurity operations.

Glossary

count — The number of connections to the same host as the current connection in the past two seconds.

dst_bytes — The number of data bytes sent from the destination to the source in a connection.

duration — The length of the connection in seconds.

flag — The status flag of the connection.

protocol_type — The type of protocol used in the connection (e.g., TCP, UDP, ICMP).

service — The network service on the destination (e.g., http, telnet, ftp).

src_bytes — The number of data bytes sent from the source to the destination in a connection.

srv_count — The number of connections to the same service as the current connection in the past two seconds.

References

[1] V. Z. Mohale and I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability," *Front Comput Sci*, vol. 7, May 2025, doi: 10.3389/fcomp.2025.1520741.

[2] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Comput Commun*, vol. 199, pp. 113–125, Feb. 2023, doi: 10.1016/j.comcom.2022.12.010.

[3] Y. Zeng, "CSAGC-IDS: A Dual-Module Deep Learning Network Intrusion Detection Model for Complex and Imbalanced Data," *arXiv preprint*, May 2025, [Online]. Available: http://arxiv.org/abs/2505.14027

[4] R. Younisse, A. Ahmad, and Q. Abu Al-Haija, "Explaining Intrusion Detection-Based Convolutional Neural Networks Using Shapley Additive Explanations (SHAP)," *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 126, Oct. 2022, doi: 10.3390/bdcc6040126.

[5] M. Al-Imran and S. H. Ripon, "Network Intrusion Detection: An Analytical Assessment Using Deep Learning and State-of-the-Art Machine Learning Models," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, p. 200, Dec. 2021, doi: 10.1007/s44196-021-00047-4.

[6] 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, doi: 10.3390/app15137329.

[7] O. Arreche, T. Guntur, and M. Abdallah, "XAI-IDS: Toward Proposing an Explainable Artificial Intelligence Framework for Enhancing Network Intrusion Detection Systems," *Applied Sciences*, vol. 14, no. 10, p. 4170, May 2024, doi: 10.3390/app14104170.

[8] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in IoT networks," *Inf Sci (N Y)*, vol. 639, p. 119000, Aug. 2023, doi: 10.1016/j.ins.2023.119000.

[9] M. R. Wijaya, "Inovasi Model Intrusion Detection System (IDS) menggunakan Double Layer Gated Recurrent Unit (GRU) dengan Fitur Berbasis Fusion," *Jurnal Ilmiah Edutic : Pendidikan dan Informatika*, vol. 12, no. 1, pp. 10–21, Feb. 2025, doi: 10.21107/edutic.v12i1.28822.

[10] M. S. Al-kahtani, Z. Mehmood, T. Sadad, I. Zada, G. Ali, and M. ElAffendi, "Intrusion Detection in the Internet of Things Using Fusion of GRU-LSTM Deep Learning Model," *Intelligent Automation & Soft Computing*, vol. 37, no. 2, pp. 2279–2290, 2023, doi: 10.32604/iasc.2023.037673.

[11] 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, p. e2289, Sep. 2024, doi: 10.7717/peerj-cs.2289.

[12] S. Neupane *et al.*, "Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities," *IEEE Access*, vol. 10, pp. 112392–112415, 2022, doi: 10.1109/ACCESS.2022.3216617.

[13] S. Nazim, M. M. Alam, S. S. Rizvi, J. C. Mustapha, S. S. Hussain, and M. M. Suud, "Advancing malware imagery classification with explainable deep learning: A state-of-the-art approach using SHAP, LIME and Grad-CAM," *PLoS One*, vol. 20, no. 5, p. e0318542, May 2025, doi: 10.1371/journal.pone.0318542.

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

[15] M. Tawfik, "Optimized intrusion detection in IoT and fog computing using ensemble learning and advanced feature selection," *PLoS One*, vol. 19, no. 8, p. e0304082, Aug. 2024, doi: 10.1371/journal.pone.0304082.

[16] X. Larriva-Novo, C. Sánchez-Zas, V. A. Villagrá, A. Marín-Lopez, and J. Berrocal, "Leveraging Explainable Artificial Intelligence in Real-Time Cyberattack Identification: Intrusion Detection System Approach," *Applied Sciences*, vol. 13, no. 15, p. 8587, Jul. 2023, doi: 10.3390/app13158587.

[17] F. Hassan, J. Yu, Z. S. Syed, A. H. Magsi, and N. Ahmed, "Developing Transparent IDS for VANETs Using LIME and SHAP: An Empirical Study," *Computers, Materials & Continua*, vol. 77, no. 3, pp. 3185–3208, 2023, doi: 10.32604/cmc.2023.044650.

[18] 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.

[19] 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, doi: 10.3390/app15137329.

[20] H.-T. Pai, Y.-H. Kang, and W.-C. Chung, "An Interpretable Generalization Mechanism for Accurately Detecting Anomaly and Identifying Networking Intrusion Techniques," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 10302–10313, 2024, doi: 10.1109/TIFS.2024.3488967.