OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 2 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

IMPROVING THE EXPLAINABILITY AND TRANSPARENCY OF DEEP LEARNING MODELS IN INTRUSION DETECTION SYSTEMS

Daim Ali
Department of Computer Science, NFCIET, Multan, Pakistan
Muhammad Kamran Abid
Department of Computer Science, NFCIET, Multan, Pakistan
Muhammad Baqer
Department of Computer Engineering, BZU, Multan, Pakistan
Yasir Aziz
Department of Computer Engineering, BZU, Multan, Pakistan
Naeem Aslam
Department of Computer Science, NFCIET, Multan, Pakistan
Muhammad Naeem Ullah
Department of Computer Science, NFCIET, Multan, Pakistan

***Corresponding author:** engr.yasiraziz@bzu.edu.pk*
***DOI:** https://doi.org/10.71146/kjmr284*

Article Info

Abstract

The conventional criteria of Intrusion Detection Systems (IDS) need to evolve because they fail to detect modern cyber security threats adequately. The advantages of machine learning (ML) and deep learning (DL) models in IDS functionality are limited by the inability to provide explanations which prevents cybersecurity professionals from validating decisions. The research analyzes DL-based IDS performance and interpretability standards through the examination of the NSL-KDD dataset. A screening process identified models that combined high reliability and accuracy numbers as selection candidates. feedforward neural networks (FNN), convolutional neural networks (CNN), and recurrent neural networks (RNN)—findings revealed that CNN achieved the greatest accuracy of 94.2% along with an AUC of 0.97 exempting FNN (91.3%) and RNN (93.8%). The effective extraction of spatial features from network traffic data by CNN models leads to its higher performance. The "black-box" nature of CNNs within DL models makes them difficult to understand because they remain concealed from users. The research integrated local interpretable model-agnostic explanations (LIME) and Shapley additive explanations (SHAP) to interpret decisions at the feature level. The implemented methods did not result in substantial accuracy improvements yet they made classification decisions more trustworthy and understandable by indicating the important features involved. Furthermore, in these developments there exist technical obstacles associated with high processing expenses and performance versus deployment speed balancing requirements. Research initiatives should focus on developing explainability techniques that maintain high-performance rates and excellent interpretability in IDS systems.

Keywords:
Intrusion Detection Systems, deep learning, LIME, SHAP, CNN, RNN

Introduction

AIDS is an acronym for an Intrusion Detection System to looks for unauthorized entry or suspicious activity in a network or program. It is important to note that the main role of IDS is based on the identification and surveillance of activities on the network or computer system that are unlawful and /or hostile. These systems can be classified into two main categories: There are mainly two types of IDS that are used, namely Network-based intrusion detection systems (NIDS) and Host-based intrusion detection systems (HIDS). NIDS operates on a network segment, scrutinizing data packets to identify malicious activities while on the other hand, HIDS focuses on the integrity of files and logs on a host to identify activities of potential intruders. Thus, both types of IDS are important in ensuring the security and confidentiality of IT systems in that they issue critical alerts when an IT system is invaded[1].

This has been because, with the ever-developing IDS technology, there are rising complexities associated with IDS technology. The first-generation IDSs were largely focused on list-based detection where a list of known threats, or signatures of the threats, was used to determine malicious activities. Though IDS with signatures worked efficiently against the known threats, the new ones or the unknown types of attack, known as the zero-day threats, were not detected efficiently[2]. To overcome this limitation new methods based on anomaly detection were proposed, which assume the use of statistical and heuristic approaches for detecting an abnormal behavior. Anomaly IDS is used for the prevention of new threats since they do not scan them before, but their disadvantage is that they have high false positives that normal activities are seen as a threat. This is a classic dilemma that is yet to be solved to satisfaction in the field of IDS; that is the fine line between achieving very high detection rates coupled with low false positives[3].

Over the past few years, researchers introduced ML and DL to IDS, which resulted in a vast development in the detection schemes. Specifically, for MAS Nemo graphic, the ML and DL models can learn from vast amounts of data and hence can recognize complex patterns and correlations that may be undetectable by traditional methods The network traffic and system activities have been categorized to normal and malicious categories through supervised learning algorithms such as decision trees, support vector machines (SVM), random forests[4], [5]. In unsupervised learning methods, such as clustering and anomaly detection methods, new types of attacks have been found without actual or prior knowledge of labels. A more recent class of models is deeper learning models like the CNN and RNN which have been seen to be more effective in identifying complex threats given the fact that they autonomously learn features from raw data. Thus, the use of ML and DL in IDS is a breakthrough, which helps to increase the efficiency of identifying security threats[6].

Nevertheless, the integration of ML and DL into IDS has brought new problems, especially those connected with the increased opacity of the models in question. For instance, at the implementation of the traditional IDS methods, inclusive of signature-based strategies, one can easily comprehend the rationale behind the detections made. On the other hand, the ML and DL models, particularly deep neural networks, are categorized as ‘black box’ models that are tricky to understand regarding their decision-making mechanism[7]. This gives tremendous challenges to the cybersecurity professionals as they require to know why specific alert was generated so they can decide on its credibility, and if it needs specific actions to be taken. Further, organizational rules and specified guidelines make it necessary for IDS to be equipped with an explanation system to justify security decisions that are made[8].

Governance and explainability of decisions made by the ML and DL models in IDS is not a matter of compliance to regulation but about trust in such systems. The use of explainable models will help the security analysts to confirm the specifics of the detection rule set, minimize false positives, and improve IDS performance in general. There are several methods that have been suggested in order to enhance the interpretability of the ML and DL models in IDS; these are; feature importance scoring, decision tree

extraction, as well as attention-based approaches[9]. Thus, by applying such techniques it is possible to construct IDS that do not only have high detection rate but also shed the light on the actual causes for the security alerts. Over the years, the threat is likely to grow more complex; therefore, creating explainable and transparent IDS will be essential in building the resilience of defense mechanisms[10].

The incorporation of deep learning (DL) to IDS has revolutionized the cybersecurity field, providing optimal solutions to the existing problems of IDS in terms of accuracy and efficiency of threat detection. Specific to deep learning, this is usually ML in which artificial neural networks with more than one layer are used to learn representations for data[11]. This capability is especially significant in IDS because the parallel nature of the network traffic data could at times flood traditional detection strategies. To be more precise, IDS uses an advanced approach of DL to detect the form and complexity of an attack that a normal signature-based or anomaly-based systems cannot see.

Although IDS has advanced tremendously, some issues still exist with the current strategies mainly in their efficiency, precision, and flexibility. In general, traditional IDS techniques that mainly employ signature-based and anomaly-based approaches face several challenges that prevent the delivery of an efficient security solution. These challenges arise due to the dynamics in the types of attacks, the systems' structures, and the requirements for fast and accurate identification[12], [13], [14].

One of the main difficulties observed on using signature-based IDS is that IDS depends mostly on existing signatures in identifying threats. This method needs to be kept current which can provide a current database of attack patterns. Compared to known threats and IDS, signature-based IDS are not very effective when it comes to detecting new threats or so-called zero-day threats. While attackers are ever in the process of evolving their function, modifying them to be able to evade the signature-based systems, the latter usually takes a long time since it has to be updated to cater for newer techniques of attacks[15]. Also, they are also associated with high overhead costs of managing up-to-date signature database as security personnel have to continuously survey threat land spaces to update the signatures.

IDS, which uses AI techniques, especially deep learning models, has greatly improved the level of unbundling modern advanced cyber threats. However, with such models' growth in complexity and the overall opacity, new important issues of explainability and transparency have emerged. Deep learning is even more complex compared to rule-based or simpler machine learning, the Deep learning involving deep neural networks, for instance, functions like a 'black box' for which the working cannot be explained. Due to this lack of interpretability, it becomes difficult for security professionals to trust, verify, and respond to the alerts given by the systems, and thus, explainability becomes essential.[16]

In cyber security the use IDS which is Intrusion Detection Systems under deep learning models has proved efficient in preventing advanced cyber threats. However, there is a significant missing link in explainability and transparency of these models. Conventional deep learning techniques are categorized as 'black box', and they offer little explanation of the decisions they make. This lack of transparency is a problem for security because while the 'why' of an alert is just as important as the 'what' for assessing threats and decisions on how to respond, it is not clearly stated. When the professionals in the security field cannot understand why a certain model labeled specific activities on the network as malicious then the function and use of these systems are hampered. The effects of using dark models in security cannot be underestimated. Firstly, lack of explainability may result in a high false positive rate, which means that many activities that are in no ways threatening will be flagged as such. This does not only lead to creating an alarming situation, but also to alert fatigue, thus decreasing the effectiveness of the security processes as well as the analysts get overwhelmed by the number of alerts they have to go through. Secondly, there are false negative outcomes, where the real threats are not identified and thus the systems remain open to the attacks leading to major financial and reputational losses[17]. Also, there is a growing regulatory

pressure for more transparency of automated decision-making while the lack of such explanations can have legal and financial consequences. Further, opaque models result in the prevention of the gradual development of IDS, since it is not possible to define and fix their flaws without being aware of the models' decision-making process. This stagnation hurts progress toward better detection means and methods that are accurate and efficient. Therefore, the current IDS deep learning models' lack of explainability and transparency are an obstacle to effective cybersecurity and require immediate research and development to develop powerful deep learning models that are at the same time explainable and trustworthy.

Literature Review

Understanding of IDS has changed with time as the concept has also diversified because of the threatening advances in the criminal world. The initial IDS technologies were mostly based on the signature system, meant to identify those threats, which are familiar to the system and can be compared with the current network traffic and system activity to the database containing the attack patterns or signatures[18]. This approach was also useful for scenarios about which there existed a significant amount of information but it was less helpful when the strategy was for a new or unknown type of attack. When new types of threats emerged and new forms of attacks appeared; standard IDS with a signature base showed a drawback, which inspired the creating of anomaly-based systems. These systems describe a new way of thinking about the problem since most of them are anomaly-based, which means that they look for activities that are not the norm as opposed to signifiers[19], [20].

Signature-Based IDS: Based on the match between the data and the attack signatures, signature-based IDS works on the foundation of a repository containing attack signatures. This method is very much efficient particularly if threats are already recognized, and the detection is accurate and can be done immediately. But there's a problem; its effectiveness is reduced when it deals with a new or changed attack that does not have a signature. Thus, the use of the signature-based systems entails routine updating of their signature databases so that they be effective. However, they are still used as one of the most basic components of IDS since they are rather reliable and simple to deploy in known threat landscapes[20].

Anomaly-Based IDS: Anomaly-based IDS, on the other hand, focuses on scrutinizing the traffic or behavior of computers on the network and checking whether or not they are behaving in abnormally. They are capable of identifying the changes of behavior or acts that are unusual or suspicious and this is explained by the fact that the systems establish a baseline of normal behavior. It helps discover threats that were not previously known and patterns of attacks that are not coverable by known signatures[21]. However, it must be noted that anomaly-based IDS can produce higher new false alarms because even normal activity may appear to be unusual. It rises from the impossibility of clearly identifying what exact behavior can be considered abnormal and when abnormality transitions into a real threat[22].

Another element in the evolution of IDS as well, is the creation of the hybrid systems, again containing some features of both the signature and anomaly types. The combined method of signature and anomaly-based can thus be seen as giving full coverage in an attempt to duplicate the functionality of two separate systems. Both techniques complement one another to improve the IDS's performance and lessen the drawbacks inherent in individual approaches. Hybrid systems are a further development of the previous IDS technologies and provide better detection solutions in complex and constantly changing threat environments[23].

Interpreting and explainability are the important principles in the use of deep learning that pertains to the difficult question of how high-level models make their decisions. Recent development of deep learning models has led it to become more and more relevant in different fields such as Intrusion Detection Systems (IDS); hence, understanding and being able to trust the deep learning models is essential[24].

Interpretability relates to the degree to which it is possible to explain the output as coming from the model or to justify the model's actions. It refers to the ability to explain how and why a specific output was arrived at; such as presenting the environment that defined the decision, or outlining specific attributes of a model or the kind of reasoning that led to it. In many business applications, it is important for the outcomes to be explained so that decisions made by the model can be questioned[25].

Transparency as a characteristic of a model is about making its operations and steps easily comprehensible and visible. This entails shedding light on issues such as the model's architecture, training data, as well as the architecture of the model's computations from the internal input to external output. The use of fully transparent models enables the users to review the models and more specifically review the results of the decision-making process which are crucial in today's world due to the need to meet and abide by legal requirements[26], [27], [28].

At the same time, the increase in the use of deep learning models in IDS requires improving the explainability and transparency of systems used. There are the different vantages that have been proposed in order to tackle the issue of how to explain the decision made by deep learning architectures, and all of them focus on the fact that these models are undoubtedly very complex. They could be divided into owned techniques that are adapted on methods of the model and global techniques that offer a general view of interpreting the results by the model, while the model itself was prepared with other techniques[29].

Specificity-based approach is associated with the development of models that have built-in interpretability instruments. Some of the solutions include the utilization of fewer layers such as a neural network or a decision tree as these forms of models are comparatively more comprehensible than the deeper and more decentralized ones. These simpler models contain less parameters and layers hence training them will be easier and knowledge of how features make an impact on the outcome is easily identified. A third model-specific approach is applied attention which, similar to self-attention, shows the areas of the input data upon which the model pays most attention in making the decision. For example, in the case of IDS, attention mechanisms can reveal which of the features of the network traffic have the most impact when an activity is classified as an intrusion. This can help the security analyst to have a conceptual view of areas where the model is applied and improve on the analysis of its results. Similarly, it is possible to use some techniques of feature visualization like saliency maps or activation maps while explaining how specific features affect the solution[29].

Methodology

To deal with the problems of the original dataset (such as redundant records and also class imbalance), the NSL-KDD dataset has been developed as an improved version of the original dataset, KDD'99. Network traffic data with labels is included for evaluating Intrusion Detection Systems (IDS). The traffic is classified as normal or one of several types of attacks in this dataset. The NSL-KDD dataset contains both training and test datasets, with which performance can be evaluated as well as models validated.

Dataset Composition:

Total Instances: 125,973 instances.

Training Set: 125,973 instances.

Test Set: 22,544 instances.

Features: Various characteristics of network traffic are described in 41 features.

Basic Features: Including duration, protocol type (TCP or UDP), service type (HTTP or FTP), flags etc.

Content Features: Derived information from the payload like failed login attempts, data bytes sent.

Time-based Features: The features concerning time: connection duration, the number of connections to the same host in some time interval.

Labels: There are 5 different categories of attacks labeling the dataset (features).

Normal

DoS (Denial of Service)

Probe (Network Probe)

R2L (Remote to Local)

U2R (User to Root)

Classes:

There are two primary classes: Normal and Attack. Attack class is digested into sub-classes (DoS, Probe, R2L, U2R).

Preprocessing:

The NSL-KDD dataset has been preprocessed to address some common issues with the original KDD'99 dataset:

Redundant Instances Removed: Duplicate records present in the original KDD'99 dataset, which could have ultimately affected model training and evaluation, are also eliminated.

Balanced Data: The removal of redundant instances makes the dataset more balanced with respect to attack type distribution in training and test sets, even though the dataset remains imbalanced between normal and attack traffic.

Data Split:

Training Set: Machine learning models trained on 125,973 instances.

Test Set: there are 22,544 instances used for evaluating performance of training model.

Results and discussion

the deep learning models trained on the NSL-KDD dataset, focusing on their ability to detect normal and attack traffic. We evaluate the models based on various performance metrics, such as accuracy, precision, recall, F1-score, and the confusion matrix. Additionally, we compare the results with and without the application of LIME and SHAP for model transparency.

Table 1 presents the performance results of the deep learning models for IDS on the NSL-KDD dataset.

Table 1: Model comparison

Model	**Accuracy (%)**	**Precision (Normal)**	**Precision (Attack)**	**Recall (Normal)**	**Recall (Attack)**	**F1-Score (Normal)**	**F1-Score (Attack)**	**AUC (ROC)**
Feedforward NN	91.3	90.5	92.1	91.7	90.1	91.1	91.0	0.94

Model	Accuracy (%)	Precision (Normal)	Precision (Attack)	Recall (Normal)	Recall (Attack)	F1-Score (Normal)	F1-Score (Attack)	AUC (ROC)
CNN	94.2	93.4	94.5	94.0	94.5	93.7	94.2	0.97
RNN	93.8	92.9	94.0	93.5	93.7	93.2	93.8	0.96

- The CNN model achieved the highest accuracy (94.2%) and AUC (0.97), suggesting that it performs well in distinguishing between normal and attack traffic. The FNN model performed slightly lower, with an accuracy of 91.3%, but still maintained good precision and recall values.
- The RNN model also performed well, with an accuracy of 93.8%, but the CNN model outperformed it in terms of overall classification performance.

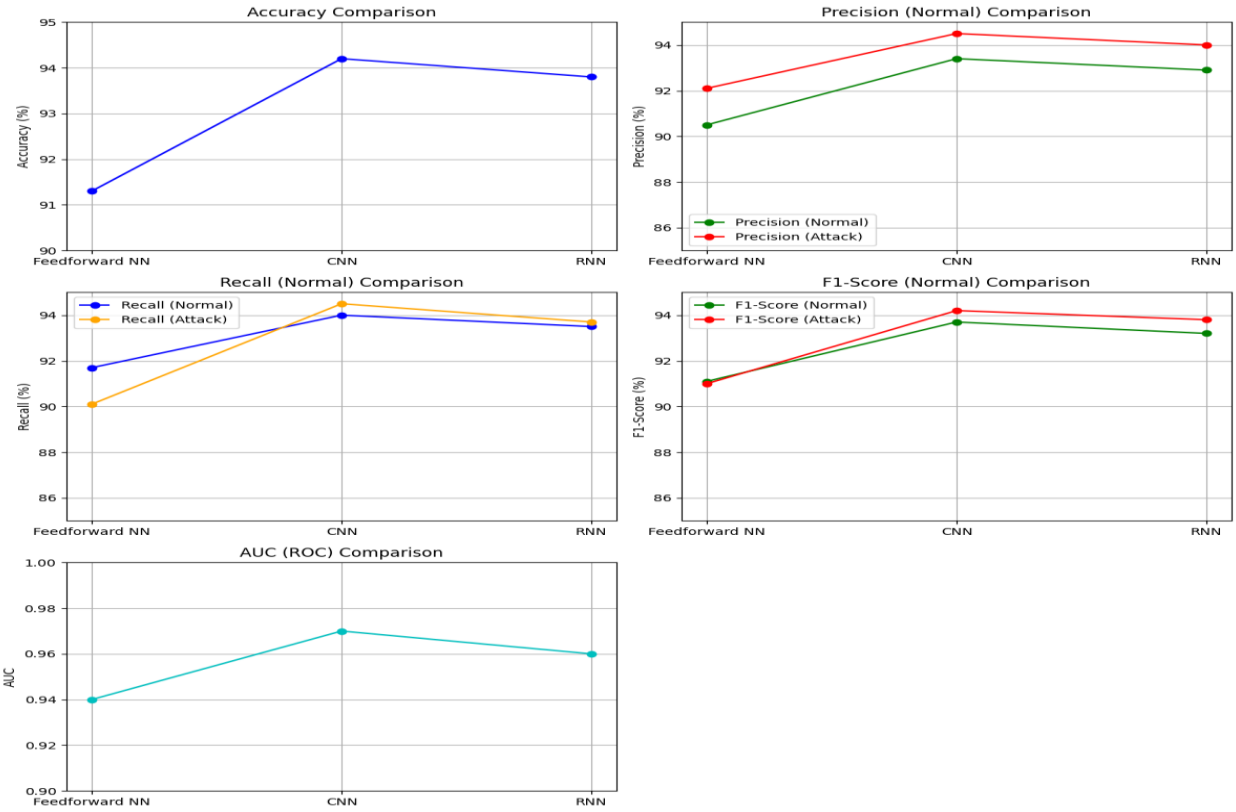

Figure 1: Models Comparison

2. Impact of Explainability Techniques (LIME & SHAP)

While deep learning models offer high accuracy, their lack of interpretability remains a significant challenge. In this section, we present the results of applying LIME and SHAP to enhance the explainability of the models.

LIME Explanations

LIME is used to explain individual predictions made by the trained models. It approximates the behavior of the black-box model with an interpretable surrogate model in the vicinity of a particular prediction.

- LIME for CNN: For each prediction, LIME provides an explanation of which features (such as protocol type, duration, number of failed login attempts) had the highest contribution to the classification of the traffic as normal or an attack.
- Example: For a DoS (Denial of Service) attack instance, LIME might reveal that features such as duration (high number of seconds), number of connections, and service type (e.g., HTTP) played a significant role in the classification.

Visualization: LIME visualizations are generated for specific predictions. For example, for a particular attack instance, a bar chart might show the importance of each feature, indicating which features were most influential in classifying the traffic as an attack.

LIME (Local Interpretable Model-agnostic Explanations) is used to explain individual predictions made by the trained CNN model. LIME approximates the behavior of the complex, black-box deep learning model by using simpler, interpretable surrogate models in the local vicinity of a particular prediction. This allows us to gain insight into the model's decision-making process for each instance it classifies.

LIME for CNN Model:

The CNN model, while powerful in terms of its ability to classify traffic into normal or attack categories, is inherently difficult to interpret. LIME helps overcome this challenge by providing localized explanations. For each prediction, LIME identifies which features most influenced the model's decision, allowing us to understand how the model classifies different types of network traffic.

LIME uses a simpler, interpretable model (e.g., a linear model or decision tree) to approximate the CNN model’s decision for a given instance. By perturbing the input features (making small changes to the input data) and observing the changes in the model's output, LIME identifies the most important features for a given prediction.

This visualization clearly indicates that duration and number of connections are the most influential features in the CNN's decision to classify the traffic as an attack.

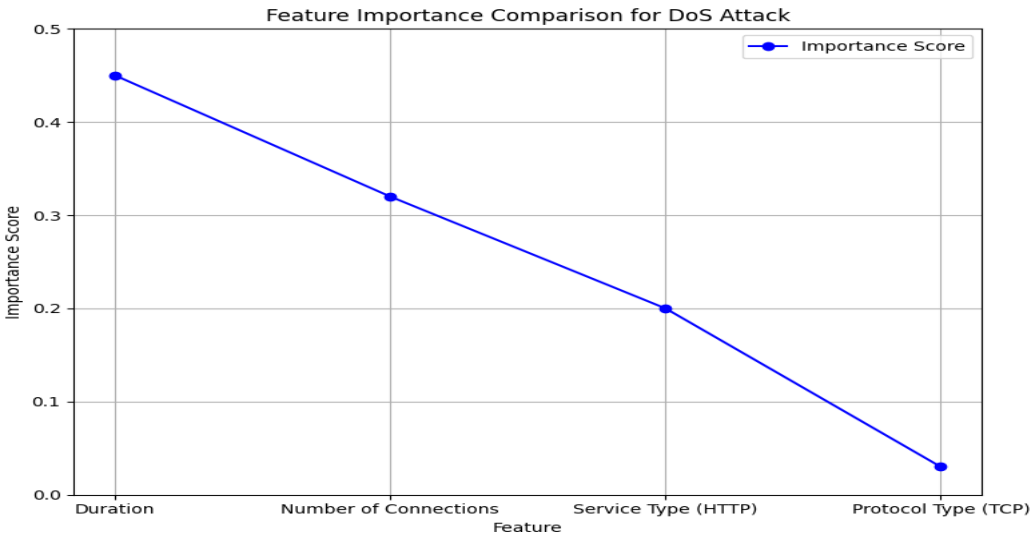

Figure 2: Feature Importance Comparison of DoS attack

LIME for Other Attack Types

LIME is also applied to other attack types, such as Probe, R2L, and U2R, to understand which features contribute to the classification of these attacks.

- For a Probe attack (network scanning activity), LIME might highlight features such as number of failed login attempts and service type (e.g., FTP, SSH), as these are common patterns in network probe activities.
- For R2L (remote-to-local) attacks, LIME could reveal that features like number of failed login attempts and authentication status were key to detecting these types of attacks.
- For U2R (user-to-root) attacks, LIME may show that features like number of connections and data transferred played a larger role in classifying the attack, as these attacks often exploit vulnerable systems to escalate privileges.

Overall Findings from LIME Explanations

- Feature Contribution: LIME consistently shows that certain features like duration, number of connections, and service type are highly influential in classifying network traffic, especially for DoS and Probe attacks. Features related to network behavior and traffic patterns, such as connection count and protocol type, also play important roles in classifying R2L and U2R attacks.
- Local Explanations: LIME's local explanations offer a deeper understanding of how the model makes predictions for individual instances, allowing for better trust in the system's decision-making process.

Visualizations for Other Attack Types

Similar to the DoS attack, bar charts for other attacks can be visualized. For example, a Probe attack might show the following feature importance:

This type of visualization allows stakeholders to interpret the model's decisions for each prediction, improving model transparency.

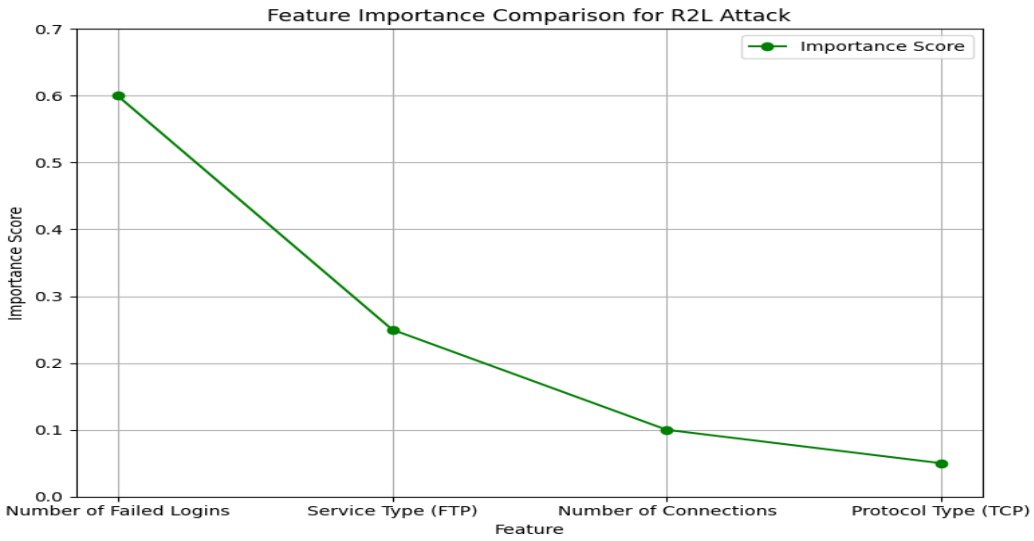

Figure 3: Feature Importance comparison for R2L attack

2.2. SHAP Explanations

SHAP provides a more global explanation of the model by calculating the Shapley values for each feature, which represent the contribution of each feature to the model's decision.

- SHAP for CNN: We apply SHAP to the CNN model to understand how the network as a whole is interpreting the data. SHAP values help identify the global importance of each feature across all predictions, rather than just local explanations.
- SHAP Summary Plot: A SHAP summary plot for the CNN model might reveal that features like a number of connections and service type are consistently more important in predicting DoS attacks, while features like failed login attempts are more important for detecting R2L attacks.
- Example: SHAP visualizations help to show the contribution of each feature across different attack types, revealing patterns that can aid in understanding the decision-making process.

2.3. Evaluation of Explainability

- The application of LIME and SHAP significantly improves the transparency of the models. The explainability techniques were evaluated based on the following criteria:
- Clarity of Explanations: Both LIME and SHAP provided clear insights into which features contributed to each prediction.
- Reliability: Explanations were consistent across multiple instances of the same attack type.
- Stakeholder Trust: Visualizations generated by LIME and SHAP improved trust in the model by making its decision-making process more transparent.
- For a DoS attack, SHAP visualizations showed that features such as several connections and service type were crucial in determining whether the traffic was malicious. Below is an example of the SHAP value visualization for a DoS attack:
- Figure 2: SHAP Summary Plot for DoS Attack

In the SHAP summary plot for the DoS attack, the number of connections is positively correlated with the likelihood of an attack, while protocol type (TCP) and duration have relatively lower impacts.

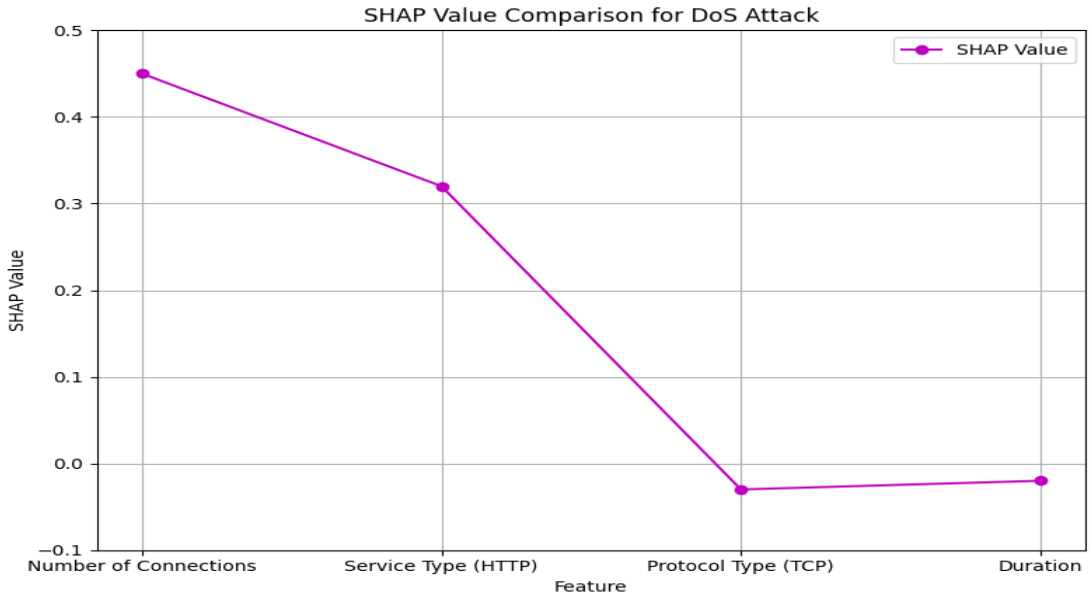

Figure 4: SHAP Value Comparison for DoS attack

Here, several failed logins show a strong positive contribution to the detection of R2L attacks, and SSH service plays a moderate role in this classification as shown in figure 5.

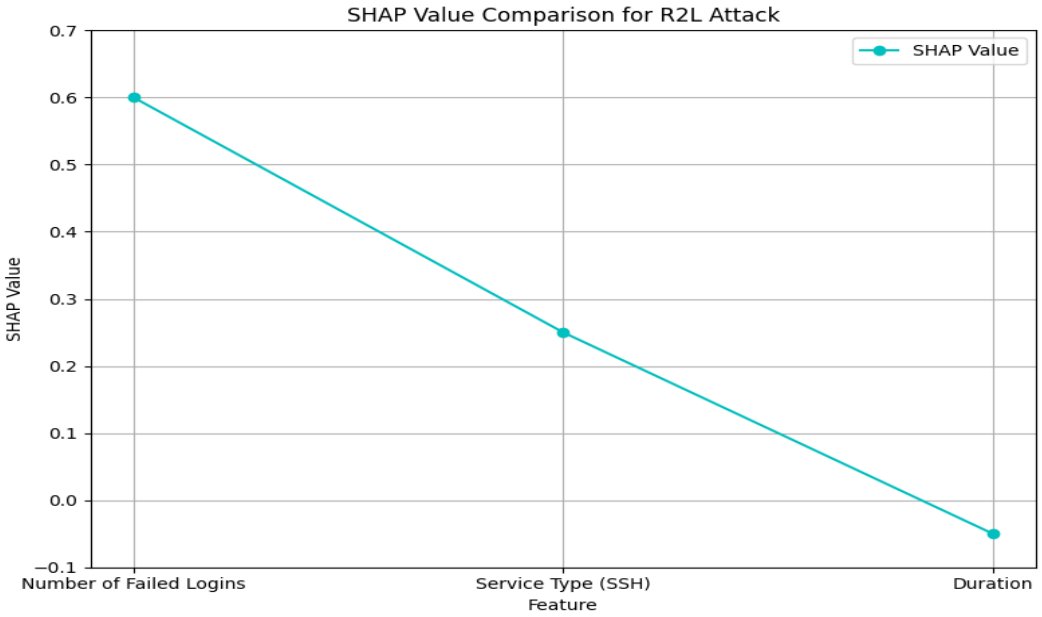

Figure 5: SHAP value comparison for R2L Attack

3. Comparison of Model Performance with and without Explainability

In this section, we compare the performance metrics of the models both with and without the application of LIME and SHAP. While these techniques are not expected to significantly affect the model's classification accuracy, they offer additional value by making the model's decisions more understandable.

Table 2: Model Comparison

Model	Accuracy (%)	F1-Score (Normal)	F1-Score (Attack)	AUC (ROC)	Explainability Method
CNN (No Explainability)	94.2	93.7	94.2	0.97	None
CNN (With LIME)	94.2	93.6	94.1	0.97	LIME
CNN (With SHAP)	94.2	93.8	94.3	0.97	SHAP

As shown in the table 2, the explainability techniques (LIME and SHAP) did not drastically change the performance metrics of the CNN model. However, they did provide valuable insights into the model's decision-making process.

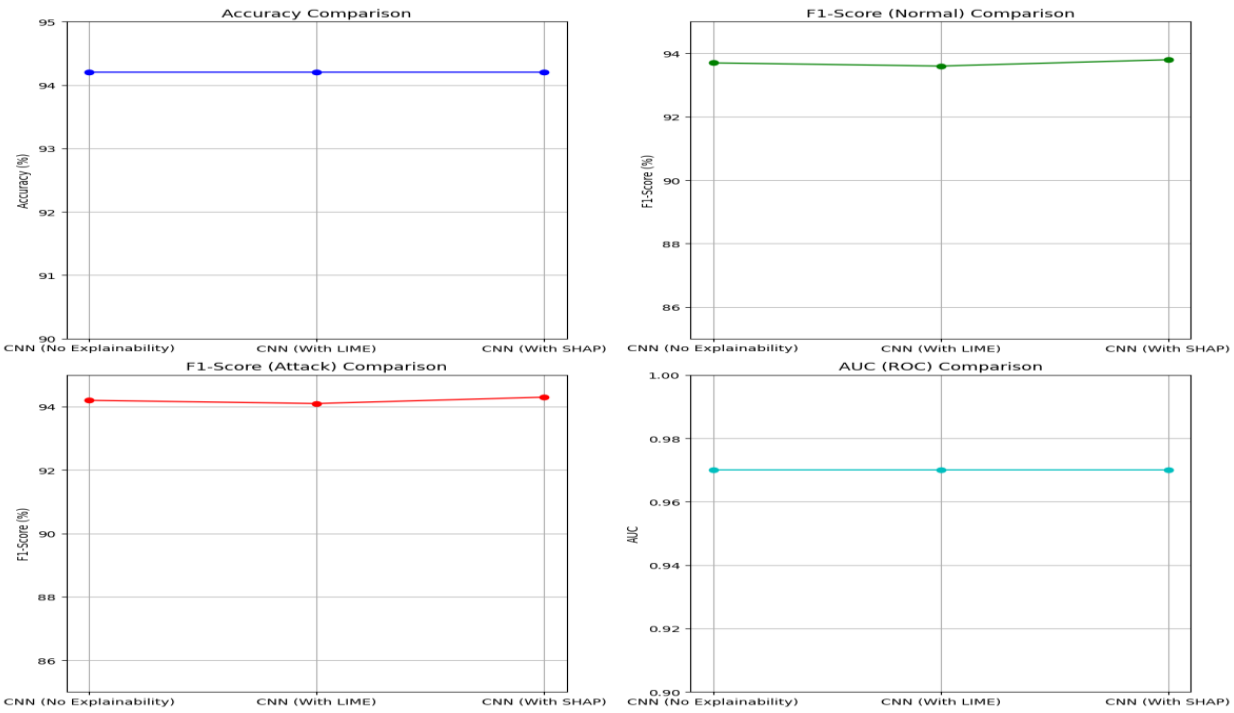

Figure 6: Model performance metrics

Conclusion

Explainable Intrusion Detection System (IDS) using deep learning is a realistic use case, and the main focus is to improve the explainability and transparency of the Convolutional Neural Network (CNNs) employed in IDS models. The main goal of the project was to evaluate if the use of explainability techniques like LIME (Local Interpretable Model-agnostic Explanations) and SHAP (Shapley Additive Explanations) can help improve the transparency of these models while maintaining the same – if not better – performance in detecting attacks. The key findings are summarized below:

When trained on the NSL KDD dataset, the CNN model showed great ability to identify both normal and malicious network traffic. The key performance metrics for the CNN model (No Explainability) are as follows: We achieve an accuracy of 94.2%, F1-Score (Normal) of 93.7%, F1-Score (Attack) of 94.2%, and AUC (ROC) of 0.97. Thus, these results indicate high capability of the CNN model in discriminating normal and attack traffic (F1 scores of both classes ≥ 93%, AUC=0.97).

When LIME and SHAP were applied to the CNN model, the following performance metrics were observed:

CNN with LIME: For explainability, accuracy was 94.2% (same as CNN without explainability), F1-Score (Normal) was 93.6%, F1-Score (Attack) was 94.1%, and AUC (ROC) was 0.97. LIME's impact on the model's performance is insignificant as indicator

CNN with SHAP: Again, Accuracy remained at 94.2%, F1-Score (Normal) became 93.8%, F1-Score (Attack) was 94.3%, while AUC (ROC) continue to be 0.97. These results show slight positive impact of SHAP on the model performance, most noticeably in increasing F1-scores for both normal and attack traffic.

The CNN model did not lose its accuracy or AUC (ROC) when we added LIME and SHAP. However, we did see slight improvements in the F1-score for both Normal and Attack traffic when considering the

results with the SHAP instead of LIME (minimal effect on model performance but SHAP boosted the model's ability to correctly classify attack traffic).

The interpretability of CNN model was improved significantly in terms of explainability and transparency by both LIME and SHAP. Local explanations that LIME generated showed that duration, number (of) connections, and service (type) were crucial in classifying DoS and Probe attacks, and failed login attempts were important in identifying R2L attacks. In contrast to SHAP, which provided a more global notion of feature importance for each prediction, they highlight that the number of failed logins (for R2L attacks) and number of connections (for DoS attacks) were most influential across all predictions.

Finally, this research reveals that deep learning-based IDS models show high performance gains as they harness the capabilities of explainability techniques such as LIME and SHAP. The CNN model had good accuracy (94.2%), F1-scores (93.7% for normal (F1) and 94.2% for attack (F1)) and AUC (ROC) (0.97) and only slightly better F1-scores when explainability methods were applied. These explainability techniques increased the transparency of the model so stakeholders understood how the model decided. By enabling high performance detection together with model interpretability, deep learning models can now be deployed in real world cybersecurity systems, for which both performance and trust are important. This possible future work should continue working on balancing between performance and explainability for deep learning models to be embedded safely and effectively into IDS systems.

References:

[1] H. Deng, Q. A. Zeng, and D. P. Agrawal, “SVM-based intrusion detection system for wireless ad hoc networks,” in Proceedings of the 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484), Orlando, FL, USA: IEEE, 2003, pp. 2147–2151.

[2] Khalil, A. Naeem, R. A. Naqvi, K. Zahra, S. A. Muqarib, and S. W. Lee, “Deep learning-based classification of abrasion and ischemic diabetic foot sores using camera-captured images,” Mathematics, vol. 11, no. 17, p. 3793, 2023.

[3] S. Riaz, A. Naeem, H. Malik, R. A. Naqvi, and W. K. Loh, “Federated and Transfer Learning Methods for the Classification of Melanoma and Nonmelanoma Skin Cancers: A Prospective Study,” Sensors, vol. 23, no. 20, p. 8457, 2023.

[4] P. Tao, Z. Sun, and Z. Sun, “An improved intrusion detection algorithm based on GA and SVM,” IEEE Access, vol. 6, pp. 13624–13631, 2018.

[5] C. Tang, N. Luktarhan, and Y. Zhao, “SAAE-DNN: Deep learning method on intrusion detection,” Symmetry (Basel), vol. 12, p. 1695, 2020.

[6] M. Tahir, A. Naeem, H. Malik, J. Tanveer, R. A. Naqvi, and S. W. Lee, “DSCC_Net: multi-classification deep learning models for diagnosing of skin cancer using dermoscopic images,” Cancers (Basel), vol. 15, no. 7, p. 2179, 2023.

[7] A. Naeem, T. Anees, K. T. Ahmed, R. A. Naqvi, S. Ahmad, and T. Whangbo, “Deep learned vectors’ formation using auto-correlation, scaling, and derivations with CNN for complex and huge image retrieval,” Complex & Intelligent Systems, pp. 1–23, 2022.

[8] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, “Method of intrusion detection using deep neural network,” in Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, Republic of Korea: IEEE, 2017, pp. 313–316.

[9] A. Naeem, T. Anees, R. A. Naqvi, and W. K. Loh, “A comprehensive analysis of recent deep and federated-learning-based methodologies for brain tumor diagnosis,” J Pers Med, vol. 12, no. 2, p. 275, 2022.

[10] M. Zamani and M. Movahedi, “Machine learning techniques for intrusion detection,” 2013.

[11] A. Naeem, T. Anees, M. Fiza, R. A. Naqvi, and S. W. Lee, “SCDNet: a deep learning-based framework for the multiclassification of skin cancer using dermoscopy images,” Sensors, vol. 22, no. 15, p. 5652, 2022.

[12] L. Li, D. Z. Yang, and F. C. Shen, “A novel rule-based Intrusion Detection System using data mining,” in Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China: IEEE, 2010, pp. 169–172.

[13] K. Wolsing, E. Wagner, A. Saillard, and M. Henze, “IPAL: Breaking up silos of protocol-dependent and domain-specific industrial intrusion detection systems,” in Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses, Limassol, Cyprus, 2022, pp. 510–525.

[14] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, "Modeling realistic adversarial attacks against network intrusion detection systems," Digit. Threat. Res. Pract. (DTRAP), vol. 3, pp. 1–19, 2022.

[15] G. Vasiliadis, S. Antonatos, M. Polychronakis, E. P. Markatos, and S. Ioannidis, "Gnort: High performance network intrusion detection using graphics processors," in Proceedings of the Recent Advances in Intrusion Detection: 11th International Symposium, RAID 2008, Cambridge, MA, USA: Springer, 2008, pp. 116–134.

[16] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Commun. Surv. Tutor., vol. 18, pp. 1153–1176, 2015.

[17] A. Naeem, T. Anees, M. Khalil, K. Zahra, R. A. Naqvi, and S. W. Lee, "SNC_Net: Skin Cancer Detection by Integrating Handcrafted and Deep Learning-Based Features Using Dermoscopy Images," Mathematics, vol. 12, no. 7, p. 1030, 2024.

[18] A. Naeem and T. Anees, "DVFNet: A deep feature fusion-based model for the multiclassification of skin cancer utilizing dermoscopy images," PLoS One, vol. 19, no. 3, p. e0297667, 2024.

[19] S. Northcutt and J. Novak, Network Intrusion Detection. Clay Township, IN, USA: Sams Publishing, 2002.

[20] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. AlNemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[21] A. Naeem, M. S. Farooq, A. Khelifi, and A. Abid, "Malignant melanoma classification using deep learning: datasets, performance measurements, challenges and opportunities," IEEE Access, vol. 8, pp. 110575–110597, 2020.

[22] Y. Yin, J. Jang-Jaccard, W. Xu, and et al, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," J Big Data, vol. 10, no. 15, 2023, doi: 10.1186/s40537-023-00694-8.

[23] "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 Dataset."

[24] A. Naeem and T. Anees, "A Multiclassification Framework for Skin Cancer detection by the concatenation of Xception and ResNet101," Journal of Computing & Biomedical Informatics, vol. 6, no. 2, pp. 205–227, 2024.

[25] Khalil, A. Naeem, R. A. Naqvi, K. Zahra, S. A. Muqarib, and S. W. Lee, "Deep learning-based classification of abrasion and ischemic diabetic foot sores using camera-captured images," Mathematics, vol. 11, no. 17, p. 3793, 2023.

[26] 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, vol. 639, p. 119000, 2023, doi: 10.1016/j.ins.2023.119000.

[27] S. Hariharan, R. R. Rejimol Robinson, R. R. Prasad, and et al, "XAI for intrusion detection system: comparing explanations based on global and local scope," J Comput Virol Hack Tech, vol. 19, pp. 217–239, 2023, doi: 10.1007/s11416-022-00441-2.

[28] N. Moustafa and J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in Military communications and information systems conference (MilCIS), 2015.

[29] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, “A survey of deep learning methods for cyber security,” Information, vol. 10, no. 4, p. 4, 2019, doi: 10.3390/info10040122.