

A Confidence-Driven Multiclass Network Intrusion Detection Framework for Intelligent Attack Identification

¹Rachita R, ²Rohini R, ³Shreya Gopalakrishna, ⁴Surabhi Kappgal Srinath, ⁵Dr.Savitha

¹Rachita R Department of Computer Science and Engineering RV Institute of Technology and Management Bengaluru, India rvit23bcs035.rvitm@rvei.edu.in

²Rohini R Department of Computer Science and Engineering RV Institute of Technology and Management Bengaluru, India rvit23bcs048.rvitm@rvei.edu.in

³Shreya Gopalakrishna Department of Computer Science and Engineering RV Institute of Technology and Management Bengaluru, India rvit23bcs080.rvitm@rvei.edu.in

⁴Surabhi Kappgal Srinath Department of Computer Science and Engineering RV Institute of Technology and Management Bengaluru, India rvit23bcs010.rvitm@rvei.edu.in

⁵Dr.Savitha G Department of Computer Science and Engineering, Associate Professor RV Institute of Technology and Management Bengaluru, India savithag.rvitm@rvei.edu.in

Abstract -An Intrusion Detection Systems (IDS) play a critical role in securing modern network infrastructures, especially in Internet of Things (IoT) environments where resource constraints and heterogeneous devices increase vulnerability to cyberattacks. Traditional signature-based approaches are insufficient against evolving and zero-day attacks, leading to the adoption of machine learning and deep learning techniques for intelligent threat detection. However, challenges such as imbalanced datasets, high false positive rates, lack of interpretability, and scalability issues remain significant. This paper presents a comprehensive study of machine learning and deep learning-based intrusion detection systems with a focus on IoT networks. It analyzes recent advancements, including deep neural networks, autoencoders, generative adversarial networks for data augmentation, and federated learning for privacy-preserving detection. Additionally, the paper highlights the importance of explainable artificial intelligence techniques such as SHAP and LIME to improve model transparency and trustworthiness. The study also reviews benchmark datasets and evaluation metrics, identifying key limitations and research gaps. The findings emphasize the need for lightweight, interpretable, and robust IDS models capable of operating efficiently in real-time IoT environments.

Key Words: Intrusion Detection System, Internet of Things, Machine Learning, Deep Learning, Explainable AI, Cybersecurity, Federated Learning

1. INTRODUCTION

The rapid growth of the Internet of Things (IoT) has changed how devices interact across domains such as health-care, transportation, and industrial systems. While this connectivity improves efficiency, it also introduces new security risks. Many IoT devices operate with limited computational resources and often lack strong built-in

security, which makes them vulnerable to a wide range of cyberattacks. Because of this, intrusion detection systems (IDS) have become an important layer of defense for monitoring and identifying suspicious network activity. Identify applicable funding agency here. If none, delete this. Recent work has shown that deep learning can significantly improve the performance of intrusion detection systems, especially when dealing with large and complex network traffic. In particular, Farhan et al. [1] demonstrated that deep learning-based IDS models are capable of achieving higher detection accuracy compared to traditional approaches, making them suitable for modern network environments. Earlier IDS techniques mainly relied on signature-based detection, where known attack patterns are used to identify threats. While effective for previously observed attacks, these methods cannot detect new or evolving threats. This limitation led to the development of anomaly-based detection methods, which focus on identifying deviations from normal behavior [2]. The availability of benchmark datasets such as UNSWNB15 has further supported the evaluation and comparison of IDS models [21]. Machine learning (ML) techniques were later introduced to improve detection performance by learning patterns from network data. Methods such as random forests and support vector machines have shown promising results in classifying network traffic [22], [29]. However, these models often depend on manual feature selection and may struggle when dealing with high-dimensional or dynamic data. Deep learning (DL) approaches address some of these limitations by automatically learning feature representations from raw data. Models based on recurrent neural networks and autoencoders have been widely used for intrusion detection due to their ability to capture temporal and nonlinear relationships in network traffic [14], [15]. In addition, advances in deep learning architectures have contributed to improved performance in cybersecurity applications [32]. Generative techniques, such as generative adversarial networks, have also been

explored to handle imbalanced datasets by generating synthetic attack samples [3]. Despite these improvements, several challenges remain. One major issue is class imbalance, where normal traffic dominates the data set and reduces the model's ability to detect rare attacks [5]. Another concern is the lack of interpretability in deep learning models, as their decisions are often difficult to explain. This is particularly important in security applications, where understanding why a decision was made is as important as the decision itself. Explainable AI techniques have been proposed to address this issue by providing insights into model behavior [11], [19]. More recently, federated learning has been explored as a way to build intrusion detection systems without sharing sensitive data. By allowing models to be trained across distributed devices, federated learning helps preserve privacy while maintaining detection performance [8], [10]. This is especially useful in IOT environments, where data is generated across multiple locations. Considering these developments, there is still a need for IDS solutions that are accurate, efficient, and interpretable, while also being suitable for real-time deployment in IOT networks. This work builds upon the foundation provided by [1] and examines recent advances in machine learning and deep learning-based intrusion detection, with a focus on addressing current limitations and identifying future research directions.

2. LITERATURE REVIEW

Intrusion detection has been studied for decades, but the nature of the problem has changed significantly with the growth of modern networks and IOT systems. Earlier approaches were largely rule-driven, where systems depended on predefined signatures to identify malicious activity. While such methods worked reasonably well for known attacks, they struggled in situations where attack patterns evolved or remained unknown. This limitation pushed researchers toward anomaly-based detection, where the focus shifted to identifying deviations from normal behavior rather than matching fixed signatures [2], [30]. As network data became more available, machine learning started to play a central role in intrusion detection research. Classical models such as decision trees, support vector machines, and random forests were widely explored. In particular, the work by Zhang et al. [22] showed that ensemble methods like random forests could handle noisy network data more effectively than simpler classifiers. Surveys such as the one by Buczak and Guven [29] further reinforced the idea that data driven methods offer clear advantages over static detection systems. That said, these models were not without issues. In many cases, performance depended heavily on how well features were engineered, which is not always practical in dynamic environments. The transition to deep learning introduced a different way of approaching the problem. Instead of relying on manually selected features, deep learning

models learn representations directly from raw data. This shift is not just technical but also conceptual. For example, Yin et al. [14] demonstrated that recurrent neural networks can capture temporal dependencies in network traffic, which are often overlooked in traditional models. Similarly, Shone et al. [15] explored the use of deep auto-encoders and showed that layered feature extraction can improve detection accuracy. These approaches align with the broader understanding of deep learning discussed by Lecun et al. [32], where hierarchical feature learning is considered a key strength. However, improving accuracy alone does not solve all problems. One recurring issue in intrusion detection datasets is imbalance. In most cases, normal traffic dominates, while attack samples are relatively rare. This imbalance can bias models toward predicting normal behavior. Karatas et al. [5] addressed this concern by emphasizing the need for techniques that can better represent minority classes. In a similar direction, Ring et al. [3] explored the use of generative adversarial networks to create synthetic attack data. This idea is interesting because it does not just improve training data but also changes how datasets themselves are constructed. Another layer of complexity appears when intrusion detection is applied to IOT environments. Compared to traditional networks, IOT systems are more distributed and often operate under strict resource constraints. Hindy et al. [18] highlighted that many existing IDS solutions are too heavy for such environments. Likewise, Al-Hawawreh et al. [20] showed that even though deep learning models can achieve high accuracy, their deployment in industrial IOT settings requires careful consideration of computational cost. This suggests that performance metrics alone are not sufficient; practical deployment constraints matter just as much. Interpretability is another area that has gained more attention recently. As models become more complex, understanding their decisions becomes increasingly difficult. This is particularly concerning for security applications, where decisions may need to be audited or justified. Work by

Arrieta et al. [11] and Guidotti et al. [12] provide a broader view of explainable AI, but their relevance becomes clearer in applied settings such as intrusion detection. Wang et al. [13] attempted to bridge this gap by introducing an explainable framework for IDS, while Latif et al. [19] emphasized that transparency is essential for trust in automated security systems. In practice, this remains an open challenge, as there is often a trade-off between model complexity and interpretability. More recently, privacy has also entered the discussion, particularly with the rise of distributed systems. Federated learning offers one possible solution by allowing models to be trained across multiple devices without sharing raw data.

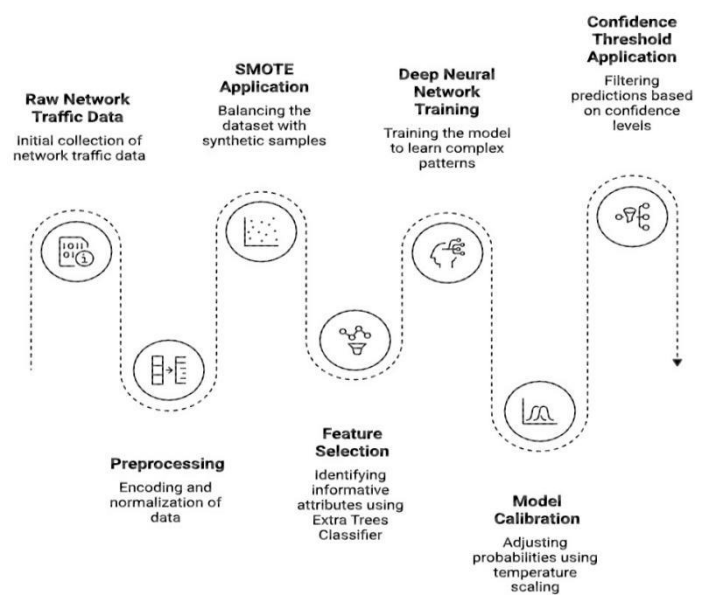
The concept, introduced by Mc-Mahan et al. [9] and further discussed by Yang et al. [8], has been adapted for

intrusion detection in IOT environments. Nguyen et al. [10] proposed DIOT, which applies federated learning to anomaly detection. While promising, such approaches also raise questions about communication overhead and model consistency across nodes. The role of datasets in IDS research should not be overlooked either. Older datasets often fail to reflect current network behavior, which limits their usefulness. The UNSWNB15 dataset [21] and CICIDS2017 [4] were introduced to address this gap by providing more realistic traffic patterns. At the same time, Ring et al. [16] pointed out that no single dataset can fully represent real-world conditions, which makes cross-dataset evaluation an important but often neglected aspect. Some researchers have also raised concerns about the broader applicability of machine learning intrusion detection.

Sommer and Paxson [28], for instance, argued that many models perform well in controlled settings but fail to generalize in real deployments. Axelsson’s discussion of the base-rate fallacy [30] further complicates the issue, suggesting that even accurate models can produce misleading results in environments where attacks are rare. Against this background, the work by Farhan et al. [1] stands out as an effort to apply deep learning techniques in a more practical intrusion detection setting. Their results show that deep learning models can achieve strong performance when properly trained and evaluated. However, like many existing approaches, there is still room to improve aspects such as interpretability, efficiency, and adaptability to real-time conditions. Overall, the literature reflects a gradual shift toward more intelligent and adaptive intrusion detection systems. At the same time, it also makes it clear that no single approach fully addresses all challenges. Balancing accuracy, efficiency, and transparency remains an ongoing concern, particularly in the context of IOT networks where constraints are more pronounced.

3. PROPOSED METHODOLOGY

The overall design of the proposed system is not based on a single model choice, but rather on a sequence of steps that try to address some of the recurring issues seen in intrusion detection literature. In particular, earlier work—including the deep learning-based approach in [1]—shows that high accuracy is achievable, but often at the cost of reliability or interpretability. With that in mind, the method here combines data balancing, feature reduction, and post-processing rather than relying only on the classifier itself.



3.1 Data Preparation

The experiments are conducted on the NSLKDD dataset, which is still commonly used despite its known limitations.

One reason for using it is that it provides a consistent benchmark across many previous studies [21]. At the same time, it is worth noting that no data set fully represents real network traffic; something already pointed out in earlier analyses of IDS datasets [16]. Before training, the data is cleaned and transformed. Categorical attributes such as protocol type and service are encoded, while numerical features are scaled. This step is fairly standard, but skipping it tends to create unstable training behavior, especially for neural networks. In practice, even small inconsistencies in pre-processing can lead to noticeable differences in performance

3.2 Addressing Class Imbalance

One issue that becomes obvious when working with NSLKDD is the imbalance between classes. Some attack categories appear far less frequently, which makes them harder for the model to learn. Instead of leaving the data as it is, SMOTE is applied to generate additional samples for these minority classes. This is not a new idea—imbalanced learning has already been discussed in intrusion detection contexts [5]—but in practice, it still makes a difference. Without this step, the model tends to bias toward normal traffic, which may look good in terms of accuracy but is not very useful from a security perspective.

3.3 Feature Selection

Rather than feeding all available features into the model, a feature selection step is introduced. An Extra Trees

Classifier is used to rank features based on importance, and only the top 20 are retained. This decision is partly practical. High-dimensional input can slow down training and sometimes introduce noise.

Earlier work using tree-based models has shown that not all features contribute equally to intrusion detection performance [22]. By reducing the feature set, the model becomes easier to train and slightly more stable, although the exact number of selected features is somewhat empirical.

3.4 Deep Neural Network Model

The core of the system is a deep neural network. Compared to traditional machine learning models, neural networks are better at capturing complex relationships in the data, especially when patterns are not linearly separable. This is one of the main reasons deep learning has been widely adopted in IDS research [14], [15].

Training is performed over multiple epochs, and the model gradually converges as expected. While the training accuracy reaches a high value, this alone is not taken as a sufficient indicator of performance. Previous studies have already shown that high accuracy does not necessarily translate to reliable predictions in real-world scenarios [28].

3.5 Model Calibration

One aspect that is often overlooked in IDS models is how confident the predictions actually are. Neural networks, in particular, tend to produce overconfident probabilities. This can be misleading, especially when decisions are made based on those probabilities. To deal with this, temperature scaling is applied after training. The idea is simple: instead of changing the predictions themselves, the confidence values are adjusted to better reflect actual likelihoods. This aligns with broader discussions in explainable and trustworthy AI, where reliability is considered as important as accuracy [11].

3.6 Confidence-Based Filtering

After calibration, a confidence threshold is introduced. Predictions above this threshold are accepted, while others are remarked as uncertain. This is a small addition, but it changes how the system behaves. Instead of forcing a decision for every input, the model is allowed to admit uncertainty. In real deployment, these uncertain cases could be flagged for further inspection. Similar ideas have been explored in IoT-based IDS systems, where reducing false positives is often more valuable than maximizing raw accuracy [18].

3.7 Evaluation Strategy

The model is evaluated using common metrics such as accuracy, precision, recall, F1-score, and ROCAUC.

However, the evaluation does not stop there. The effect of calibration and confidence filtering is also considered.

This is important because standard metrics alone do not fully capture how a system performs in practice. For example, a model might have good overall accuracy but still perform poorly on rare attack classes or produce unreliable confidence scores.

3.8 Explainability with SHAP

Finally, SHAP is used to interpret the model's predictions.

Rather than treating the model as a black box, SHAP provides a way to see which features influence a particular decision. This step is partly motivated by the growing interest in explainable AI for cybersecurity applications [11], [19]. In practice, the explanations also help verify whether the model is relying on meaningful features or simply picking up spurious patterns. Overall, the methodology does not rely on a single improvement but combines several smaller adjustments— data balancing, feature selection, calibration, and interpretability. Individually, these steps are not new, but their combination makes the system more practical for intrusion detection, especially in scenarios where reliability matters as much as accuracy.

4. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents a detailed evaluation of the proposed intrusion detection system. Instead of focusing only on overall accuracy, the analysis also considers prediction of reliability, calibration effects, and the behavior of the model under uncertainty.

4.1 Experimental Setup

The experiments are conducted using the NSLKDD dataset, which includes five categories of network traffic: Normal, DOS, Probe, R2L, and U2R. After pre-processing and applying SMOTE to balance the dataset, the training set contains 336,715 samples. A deep neural network is trained for 15 epochs. The training shows stable convergence, with accuracy increasing over time. By the final epoch, the training accuracy reaches approximately 97.84%, indicating that the model has learned the underlying patterns in the data.

However, as noted in earlier studies, high training accuracy does not always translate to strong generalization performance in intrusion detection tasks

[28]. Therefore, evaluation of unseen test data becomes critical.

4.2 Baseline Model Performance

The first evaluation is performed using the uncalibrated model. The results are summarized in Table-1.

Table -1: Baseline Model Performance

Metric	Value
Accuracy	75.10%
Precision	80.93%
Recall	75.10%
F1-Score	72.59%
ROC-AUC	0.9228

At first glance, the ROCAUC value appears strong, suggesting that the model can distinguish between classes effectively.

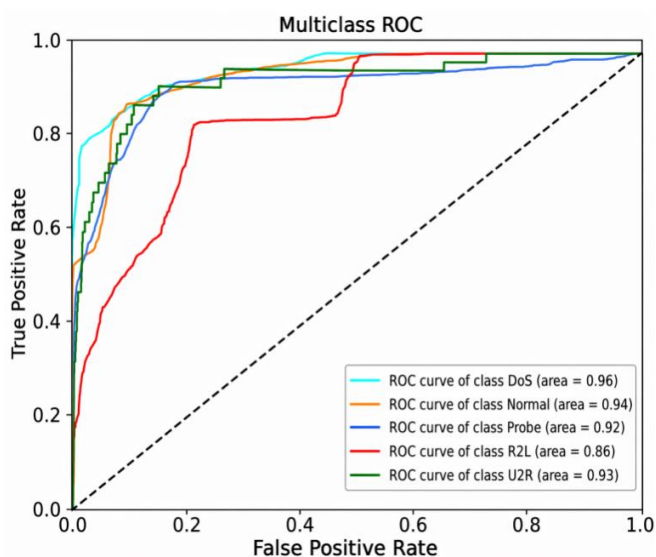


Fig-1: Multiclass ROC Curve showing model discrimination ability across all classes

However, the relatively lower F1 score indicates that performance across different classes is not balanced. One noticeable pattern is that precision is higher than recall.

This implies that the model is more cautious when predicting attacks, leading to fewer false positives but potentially missing some actual attacks. In practical

settings, this tradeoff can be problematic, especially when undetected attacks carry higher risk.

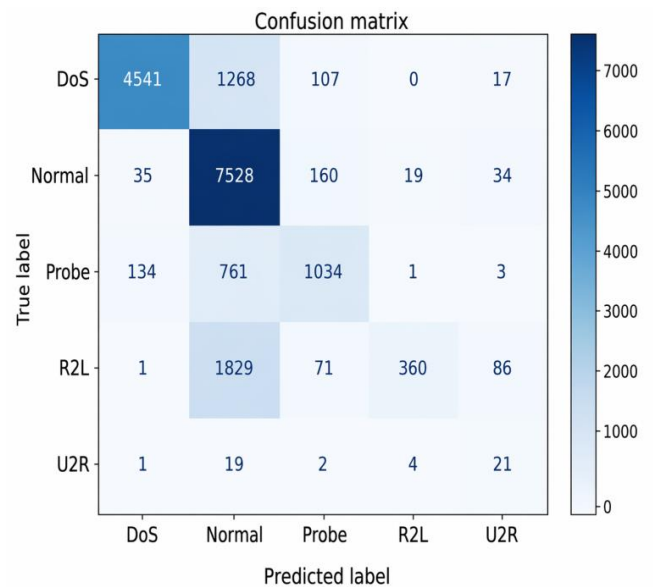


Fig-2: Confusion Matrix of the Model

4.3 Effect of model Calibration

To address the issue of unreliable confidence scores, temperature scaling is applied. The calibration results are shown in Table-2.

Table -2: Model Calibration Results

Metric	Value
NLL (Before Calibration)	2.1926
Optimal Temperature	1.8886
NLL (After Calibration)	1.2132

The reduction in negative log-likelihood (NLL) is quite significant. This suggests that the model's predicted probabilities are better aligned with actual outcomes after calibration. It is important to note that calibration does not aim to improve classification accuracy directly but rather to improve the reliability of confidence scores [11]. To verify this, the model is evaluated again after calibration.

The results are presented in Table-3.

Table -3: Calibrated Model Performance

Metric	Value
Accuracy	75.01%
Precision	80.91%
Recall	75.01%
F1-Score	72.49%
ROC-AUC	0.9238

As expected, the classification of metrics remain almost unchanged. This confirms that calibration improves confidence estimation without affecting prediction outcomes.

In intrusion detection, this distinction is important because decisions are often made based on confidence levels rather than raw predictions.

4.4 Confidence-Based Decision Analysis

To further evaluate the usefulness of calibrated confidence scores, a threshold of 0.8 is applied. Predictions above this threshold are accepted, while others are marked as uncertain. The results are summarized in Table-4.

Table -4: Confidence-Based Decision Results

Metric	Value
Total Samples	18,036
Accepted Predictions	16,006 (88.74%)
Uncertain Predictions	2,030 (11.26%)
Accuracy (Accepted)	79.98%
Accuracy (Uncertain)	35.81%

A few observations stand out here. First, the majority of predictions fall into the high-confidence category. More importantly, the accuracy for accepted predictions increases to nearly 80. At the same time, the uncertain predictions show significantly lower accuracy.

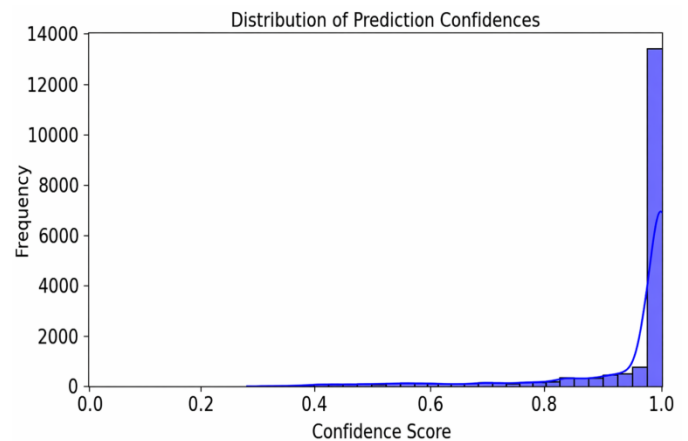


Fig-3: Distribution of prediction confidence scores after calibration

This indicates that the model is correctly identifying difficult or ambiguous cases. In practice, this behavior is desirable because it allows the system to flag uncertain instances instead of making unreliable decisions.

4.5 Feature Importance Analysis

Feature selection results show that certain attributes contribute more significantly to the model's decisions.

Features such as `dst_host_srv_count`, `error_rate`, and `same_srv_rate` appear among the top-ranked features. These features are closely related to connection behavior and error patterns, which are known as indicators of malicious activity. Previous studies have also highlighted the importance of such features in intrusion detection tasks [22]. The feature importance plot further confirms that a smaller subset of features can capture most of the relevant information.

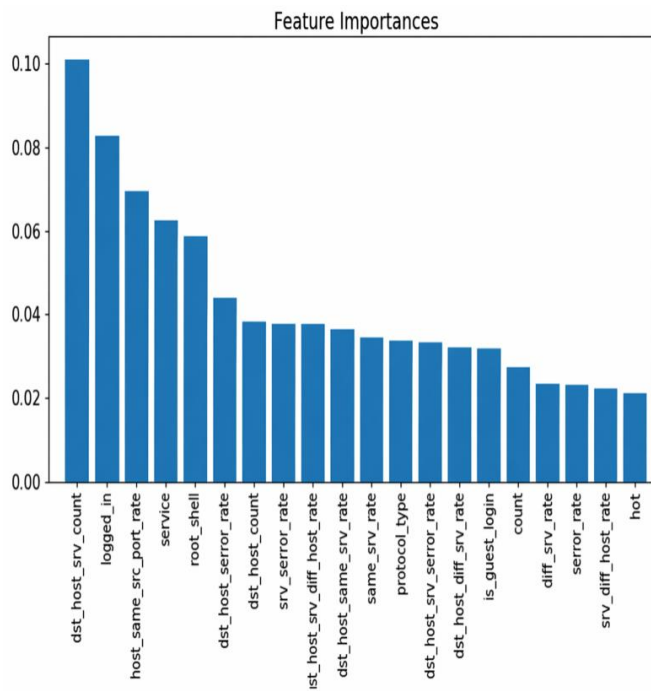


Fig-4: Feature importance graph

4.6 Model Interpretability Using SHAP

To better understand the model’s behaviour, SHAP is used to analyse feature contributions. The SHAP summary plot provides insight into how different features influence predictions across multiple samples. The analysis shows that features related to abnormal traffic patterns and connection statistics have a strong impact on attack detection.

This aligns with expectations and provides some confidence that the model is learning meaningful patterns rather than relying on noise. Explainability plays an important role in intrusion detection systems, particularly when decisions need to be validated by security analysts [19]. The use of SHAP helps bridge the gap between model performance and interpretability.

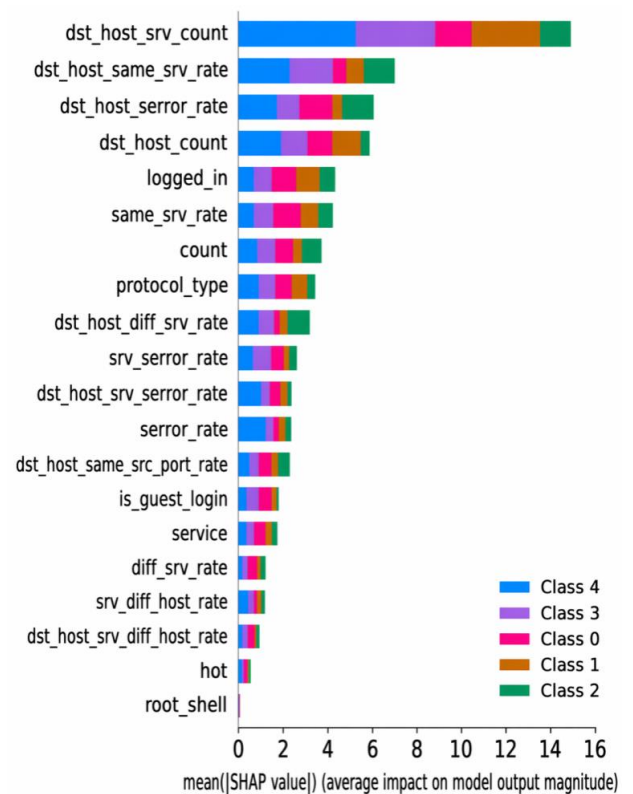


Fig-5: SHAP Feature Importance Summary Plot

4.7 Discussion

Overall, the results suggest that the proposed approach performs reasonably well, particularly when confidence-based filtering is applied. While the baseline accuracy is moderate, the system becomes more reliable when uncertain predictions are separated. At the same time, the gap between training and testing performance indicates that generalization remains a challenge. This is consistent with earlier observations that intrusion detection models often perform differently in controlled datasets compared to real-world environments [28]. Rather than relying solely on accuracy improvements, the proposed system focuses on improving reliability and interpretability. These aspects are often overlooked but are essential for practical deployment in IOT and network security applications.

5. CONCLUSIONS

This work explored the use of deep learning techniques for network-based intrusion detection, with particular attention to issues that often appear in practical deployments rather than just benchmark performance. While deep neural networks are known to achieve high accuracy, the results in this study suggest that accuracy alone does not fully reflect how reliable a system is in real-world scenarios. The experimental results show that the

proposed model is capable of learning meaningful patterns from network traffic, achieving strong training performance, and reasonable test accuracy. However, a noticeable gap between training and testing results indicates that generalization remains a concern, which is consistent with observations reported in earlier studies [28]. One of the more useful outcomes of this work is the effect of calibration and confidence-based filtering. Even though calibration does not improve traditional metrics such as accuracy or F1-score, it improves the reliability of prediction probabilities. When combined with a confidence threshold, the system becomes more selective, allowing high-confidence predictions to be trusted while separating uncertain cases. This behavior is often more desirable in intrusion detection, where incorrect decisions can have significant consequences. The inclusion of feature selection and SHAP-based interpretation further adds to the practical value of the system. Reducing the feature set simplifies the model without a major loss in performance, while explainability provides insight into how decisions are made. This is particularly important in security applications, where transparency is often required alongside accuracy. Overall, the results suggest that combining deep learning with calibration, confidence filtering, and interpretability leads to a more balanced intrusion detection system. Rather than focusing only on improving numerical performance, the approach moves toward making the system more reliable and usable in practice.

5.1 Future Work

Although the proposed approach shows promising results, there are several areas that could be explored further. One immediate direction is improving generalization. The current model is trained on the NSLKDD dataset, which, while useful, does not fully represent modern network environments. Evaluating the model on more recent datasets or real network traffic would provide a better understanding of its practical performance. Another area for improvement is handling class imbalances more effectively. While SMOTE helps to some extent, it may introduce synthetic patterns that do not always reflect real attacks. Alternative approaches, such as cost-sensitive learning or more advanced data generation techniques, could be explored. The current model also relies on a fixed confidence threshold. In future work, this threshold could be made adaptive, allowing the system to adjust based on changing network conditions. This would make the model more flexible in dynamic environments, such as IOT networks. In addition, the computational cost of deep learning models remains a concern, particularly for resource-constrained devices. Developing lightweight or optimized versions of the model would make it more suitable for deployment in edge or IOT environments. Finally, while SHAP provides useful explanations, integrating explainability more tightly into the decision-

making process could be beneficial. Instead of using explanations only for analysis, they could be used to actively guide or validate predictions. These directions suggest that future research should focus not only on improving detection performance but also on making intrusion detection systems more adaptive, efficient, and interpretable.

REFERENCES

- [1] M. Farhan, H. Waheed ud Din, S. Ullah, M. S. Hussain, M. A. Khan, T. Mazhar, U. F. Khattak, and I. H. Jaghdam, "Network-based intrusion detection using deep learning technique," *Scientific Reports*, vol. 13, 2023.
- [2] H. Hindy, D. Brosset, E. Bayne, A. Seem, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy of network threats and intrusion detection systems," *Future Generation Computer Systems*, vol. 103, pp. 247–281, 2020.
- [3] M. Ring, D. Schlor, D. Landes, and A. Hotho, "Flow-based network traffic generation using generative adversarial networks for intrusion detection," *Computers & Security*, vol. 82, pp. 156–172, 2019.
- [4] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116.
- [5] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the performance of machine learning-based IDSs on an imbalanced dataset," in *Proc. IEEE Int. Congr. Big Data*, 2020, pp. 171–176.
- [6] S. Ahmed, Y. Lee, S. Hyun, and I. Koo, "Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2765–2777, 2019.
- [7] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "Intrusion detection and ejection framework against lethal attacks in UAV-aided networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1143–1153, 2017.
- [8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.
- [9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017, pp. 1273–1282.
- [10] M. Nguyen, T. T. Nguyen, H. Kim, and H. Kim, "DloT: A federated self-learning anomaly detection system for IoT," in *Proc. IEEE ICDCS*, 2019, pp. 756–767.
- [11] A. B. Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges," *Information Fusion*, vol. 58, pp. 82–115, 2020.
- [12] R. Guidotti et al., "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–42, 2018.
- [13] M. Wang, K. Zheng, H. Yang, and X. Wang, "An explainable machine learning framework for intrusion detection systems," *IEEE Access*, vol. 8, pp. 73127–73141, 2020.

- [14] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
- [15] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, 2018.
- [16] M. Ring, S. Wunderlich, D. Grudl, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.
- [17] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. EAI Int. Conf.*, 2016, pp. 21–26.
- [18] H. Hindy, R. Atkinson, and X. Bellekens, "Machine learning based intrusion detection for IoT networks: A survey," *IEEE Access*, vol. 8, pp. 215–232, 2020.
- [19] S. Latif, Z. Zou, and J. Qadir, "Explainable artificial intelligence for intrusion detection systems: A survey," *IEEE Access*, vol. 9, pp. 112125–112147, 2021.
- [20] M. Al-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *J. Inf. Security Appl.*, vol. 41, pp. 1–11, 2018.
- [21] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. MilCIS*, 2015.
- [22] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 38, no. 5, pp. 649–659, 2008.
- [23] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in *Proc. IJCNN*, 2002, pp. 1702–1707.
- [24] K. Wang, S. J. Stolfo, and B. Li, "Anomalous payload-based network intrusion detection," in *RAID*, 2004, pp. 203–222.
- [25] D. Barbara, N. Wu, and S. Jajodia, "Detecting novel network intrusions using Bayes estimators," in *Proc. SIAM SDM*, 2001.
- [26] T. A. Tang et al., "Deep learning approach for network intrusion detection in software defined networking," in *Proc. IEEE Conf.*, 2016.
- [27] W. Wang et al., "Malware traffic classification using convolutional neural network," in *Proc. IEEE Conf. Inf. Netw.*, 2017.
- [28] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE S&P*, 2010.
- [29] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2016.
- [30] S. Axelsson, "The base-rate fallacy and its implications for intrusion detection," *ACM TISSEC*, vol. 3, no. 3, pp. 186–205, 2000.
- [31] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. ICML*, 2006.
- [32] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [33] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 2008.
- [34] A. Abeshu and N. Chilamkurti, "Deep learning: The frontier for distributed attack detection," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 169–175, 2018.
- [35] M. Conti et al., "Internet of Things security and forensics: Challenges and opportunities," *FGCS*, vol. 78, pp. 544–546, 2018.
- [36] N. Moustafa et al., "Big data analytics for intrusion detection system," in *Proc. IEEE DSAA*, 2015.
- [37] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in *Proc. IEEE ICC*, 2017.
- [38] M. Zolanvari et al., "DDoS detection using deep learning in software-defined networks," in *Proc. IEEE Big Data*, 2019.
- [39] A. Khraisat et al., "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, 2019.
- [40] N. Moustafa and J. Slay, "Evaluation of network anomaly detection systems," in *Proc. MilCIS*, 2015.
- [41] G. Apruzzese et al., "On the effectiveness of machine and deep learning for cyber security," in *Proc. IEEE Cyber Conflict*, 2018.