A Review on LAN and Router Anomaly Detection Using Machine Learning Techniques on the UNSW-NB15 Dataset

Deepanjali Kale[1] Mairaj Inamdar[2]
[1]PG Student [2]Assistant Professor
[1,2]Department of Electronics and Communication Engineering
[1,2]Siddhant College of Engineering, Sudumbare, Pune, Maharashtra, India

Abstract — Routers and local area networks serve as essential access points within contemporary communication systems, and because of this role, they are becoming more susceptible to sophisticated cyber-attacks, including reconnaissance, Denial of Service (DoS), exploits, malware spread, and botnet-driven intrusions. Traditional signature-based Intrusion Detection Systems (IDS) have shown limited effectiveness in detecting zero-day vulnerabilities and evolving attack patterns, highlighting the need for Machine Learning-based anomaly detection. The UNSW-NB15 dataset has recently gained recognition as a benchmark for evaluating machine learning-driven IDS models due to its realistic traffic composition and varied taxonomy of attacks, combined with a comprehensive feature set. This paper conducts a systematic review of machine learning-based, feature-engineering-focused, and hybrid IDS methodologies applied to the UNSW-NB15 dataset, particularly examining their appropriateness for deployment in LAN and router environments. The reviewed studies are assessed across critical methodological dimensions, including pre-processing workflows, feature selection methods, classifier design, evaluation metrics, and management of class imbalance. A thematic comparative analysis is provided across Decision Tree, Random Forest, SVM, Ensemble models, CNN variants, and feature-optimized pipelines to assess performance trends and computational trade-offs. Furthermore, the paper outlines significant research challenges, such as the misclassification of minority attacks, delays in inference, resource limitations in router platforms, incapacity for streaming and online learning, and restricted evaluation centred on deployment. In light of these findings, this review points to emerging avenues toward lightweight, feature-efficient, and deployment-oriented IDS frameworks that would be appropriate for real-time anomaly detection in LAN and router-based settings.

Keywords: LAN, Machine Learning, Deep Learning, CNN, Intrusion Detection

I. INTRODUCTION

From the router to the LAN, this ecosystem is, in essence, the foundation of the device interconnectivity we have integrated into offices, universities, factories, and living spaces. They provide the interconnectivity tissue between end devices, the core network, the cloud, and the increasing phenomenon known as the Internet of Things. Yet, with the increasing traffic, the nature of the tools, and the ever-expanding need to connect, the same elements of the ecosystem ironically represent highly prized attack points to the most sophisticated forms of cyberattacks. The attack categories remain reconnaissance and scanning, probing and enumeration, DoS and flooding, back door installation, malware diffusion, privilege escalation, and botnet-based command and control attacks, and the attacker winning control of a router or LAN component disables the end-user availability component while providing access to lateral and full-scale dataset [1].

Traditional security solutions firewall and signature based intrusion detection systems depend on predefined rules and static attack signatures stored in databases. They are quite effective against known threats but have difficulties in dealing with zero-day attacks, polymorphic malware, encrypted malicious traffic, and dynamic attack patterns. In addition to that, manual creation of rules and maintenance of databases lead to delays before taking action against new threats.

These challenges and issues have contributed to the emergence of the anomaly-based IDS, which employs ML and DL to understand the typical patterns of the network traffic and raise an alert for the possible abuse based on the anomalies. In this context, the UNSW-NB15 dataset has emerged as a prominent benchmark for developing and implementing the ML/DL-based IDS systems. Unlike KDD'99 and NSL-KDD, the UNSW-NB15 dataset includes modern network traffic and encompasses the activities of the 49 significant attack families[2].

Initial studies tended to prove that it is possible to obtain promising results using classifiers including Random Forests, SVM, Decision Trees, and Ensemble Classifiers on UNSW-NB15. Thereby solidifying the position of UNSW-NB15 as a benchmarking benchmark for IDS-related studies. Nevertheless, most studies related to IDS on UNSW-NB15 remain centred upon UNSW-NB15 itself and ignore the real-world environment[3]. Specifically speaking, there has been less emphasis on what scale UNSW-NB15 or related ML/DL models really could be applied at router and edge environments with regard to LANs because,

1) Limited CPU and memory resources
2) latency-sensitive traffic
3) energy and bandwidth constraints
4) the need for lightweight feature processing
5) a requirement concerning the detection of online or streaming content.

Additionally, challenges such as imbalanced datasets, misclassification of minority attacks, redundancy of features, issues with model interpretability, and insufficiently optimized learning architectures for edge devices persist in obstructing practical implementation. Thus, there is a pressing demand for a comprehensive review that not only examines ML and DL based IDS methods applied to the UNSW-NB15 dataset but also evaluates their applicability to anomaly detection in local area networks and router environments.

II. LITERATURE REVIEW

Intrusion Detection Systems have changed a lot over time. They used to follow rules and look for specific patterns. Now they use machine learning. Look for things that are not normal.

People who studied this early on statistics and grouping to find things that did not fit in. They also used computers to make decisions based on patterns. This was a way than just using rules that never changed. These early studies showed that using machine learning and other techniques could be a way to find intrusions, in computer networks. Intrusion Detection Systems are getting better because of this. The UNSW-NB15 dataset was a step forward in IDS benchmarking research [1]. It was created to fix the problems with datasets like KDD-99 and NSL-KDD.

Moustafa and Slay made the UNSW-NB15 dataset using the IXIA PerfectStorm cyber range. They added modern network traffic and nine types of attacks to the dataset [2]. The UNSW-NB15 dataset also includes 49 features that come from Zeek. These features help us understand the flow of traffic on a network. The UNSW-NB15 dataset is now one of the popular benchmarks, for testing systems that use machine learning and deep learning to find unusual activity. The UNSW-NB15 dataset is widely used to evaluate these kinds of systems.

Meftah and other people did one of the studies that used Machine Learning to look at UNSW-NB15. They checked how well some classifiers worked, like SVM, KNN, Naïve Bayes and Decision Trees. What they found out was that the tree-based and ensemble classifiers were better at generalizing than the models [3] . This just showed again that UNSW-NB15 is a choice for doing research, on Machine Learning based Intrusion Detection Systems.

A. Machine Learning-Based IDS Approaches

Some people have looked at how supervised machine learning models work for classifying traffic using UNSW-NB15.

Vallejo-Huanga and others compared machine learning techniques for network intrusion detection systems and found that Random Forest and Gradient Boosting are more robust and stable in different network environments than basic classifiers [6]. Moualla and others showed that the performance of machine learning models gets better when you add steps, like pre-processing, normalization and reducing the number of features to the process of building the model [7].

Kabir and his team came up with a way to stack multiple machine learning classifiers on top of each other to create an Intrusion Detection System framework. This stacking approach helped the Intrusion Detection System framework to detect things accurately and it was also better, at handling different situations compared to using the machine learning classifiers on their own. The people who worked on this project also used a kind of Random Forest implementation that was based on Spark and this helped the Intrusion Detection System framework to handle a lot of data and work faster which is really important when you are dealing with big networks and you need to analyse a lot of data at the same time [9].

Putra looked at supervised classifiers for the UNSW-NB15 attack categories. He found out that ensemble learning techniques work better than classifiers when there are many classes and not enough data, for some classes. This is what he said in his study [13].

Gunupusala and Kaila also did a study. They showed that using learning strategies can improve the detection of anomalies when there are many classes [14] .

When it comes to machine learning and intrusion detection systems people really care about making things work better and faster. Some researchers like Akuthota and Bhargava tried using XGBoost to pick the important features, which made their system work more efficiently. They found that by using the most important features they could make their system better [10].

Other people like Pansari and their team did something with XGBoost they looked at which features were not really necessary and got rid of them and it did not hurt how well their system could detect things. Feature efficiency and latency reduction are really important in this kind of research, on machine learning and intrusion detection systems. Mohamed and Agarwal used Recursive Feature Elimination to make the features smaller. The computer work less. This makes it easier to use Machine Learning based Intrusion Detection System frameworks on devices like those, at the edge level [12].

B. Deep Learning and Hybrid IDS Approaches

People who study learning for IDS have used special ways to learn from data and look at important features.

Naresh and his team looked at how supervised machine learning models and deep learning models work. They found that deep networks are better at seeing patterns but they need a lot of computer power to work [15].

Sharma and Sobti made a Deep Neural Network framework for IDS to use with UNSW-NB15 data. They said it works well and can be used in many situations if they choose the right learning settings. Deep learning, for IDS is what they used to make this framework [16].

Vibhute and the other people who worked with them looked at kinds of CNN architectures. They found out that models that use CNN are really good at figuring out how different parts of the flow are related to each other in space. They also learned that it is necessary to adjust the architecture to get the right balance, between how well the model works and how long it takes to get the results [17].

Pear and Kibria wanted to make Deep Learning models simpler. So they came up with an idea for a Parallel Artificial Neural Network architecture. This new architecture worked as well as Convolutional Neural Network models. It used a lot fewer resources. This makes it better for routers that do not have a lot of power [18].

Alsharaiah and their team tried an approach. They made a deep learning system for detecting bad network traffic. This system had layers that helped it tell apart different kinds of attacks that are similar [19].

Then there were people, like Sohail and their team. They used a kind of Generative Adversarial Network to help their Convolutional Neural Network work better. This special network helped make examples of the kinds of network traffic that are not very common. This helped fix the problem of the

dataset being imbalanced. This way of doing things made it better at remembering attack instances and it also improved the F1-score. However it made training more complicated [20].

C. Unsupervised and Behavior-Driven IDS Approaches

Sharma and the other people who worked with him think these methods are good for finding computer attacks that nobody has seen before. They found some problems with these methods. For example, it is hard to understand how the models work. The models are also very sensitive, to the scale of the features. It is hard to set the right threshold for what is considered abnormal behaviour [21].

Schummer and the other people who worked with him proposed a framework for a system that detects intrusions. This system is supposed to be used in life. They thought it was more important to make sure the system actually works and can be used every day than just making sure it is very good, at detecting intrusions. They wanted to see if the system could really be used in the world [22].

D. Streaming, Edge, and Deployment-Aware IDS Frameworks

People are now doing research on models that can detect things in time. These models can handle a lot of data and work with routers. Ness and some other people made a system that uses machine learning to find things in the data. This system is good, at finding the balance between being precise and remembering things. It also works quickly [23].

Eswarakrishnan and Singla came up with a way to make the classification process better. They did this by making sure the features they were looking at were relevant and by adjusting the settings to get the results. This makes the system more reliable when it comes to figuring out what kind of attack is happening [24].

Pushkar looked at systems that use Machine Learning to stop cyber threats in real world internet traffic. He found out that using classifiers together makes these systems work really well [25].

Shoyab and Ahmed came up with a system that uses Machine Learning to find anomalies in internet traffic as it happens. This system can look at traffic and make predictions without stopping, which is great, for always watching what is going on [26].

Anderson et al. explored IDS design in IoT-centric networks and emphasized the importance of lightweight and resource-efficient modeling architectures for constrained gateway and router-level platforms [27].

E. Literature Synthesis and Key Observations

When you look at the studies that were reviewed some things really stand out. Several trends are pretty consistent, in the studies that were looked at. The studies that were reviewed show some things that happen over and again.

1) Ensemble-based Machine Learning models give us robustness, stability and interpretability when we use them with the UNSW-NB15 dataset. This is something that we can see in the work of researchers [6] [7] [13] [14] [23]. Ensemble-based Machine Learning models are really good, at this because they can handle a lot of things and still make sense of the data. The UNSW-NB15 dataset is an example of how Ensemble-based Machine Learning models can be used to get good results.
2) Deep learning methods are really good at finding patterns that're not straightforward but they also require a lot of computer power and can be slow which makes them less suitable for use, in routers [15]–[20].
3) Feature selection based pipelines are really good because they make things work better and faster. This means they use power and that is great for things like edge and router environments [10]–[12]. Feature selection-based pipelines are a choice, for these kinds of situations.
4) Streaming-aware IDS designs are more aligned with real-world deployment requirements compared to static offline systems [4], [23], [24],[26].

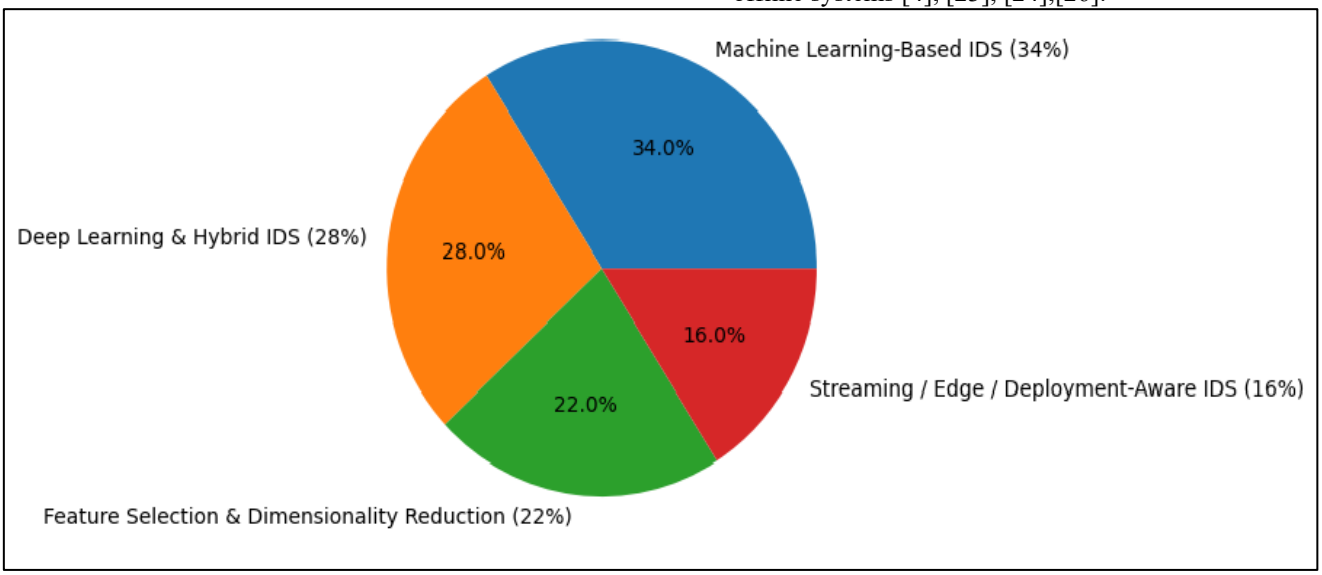

Fig. 1: Overview of Research on UNSW-NB 15 dataset

This graph categorizes the areas of research articles relating to work in anomaly detection and intrusion detection on the UNSW-NB15 dataset. A majority of the work (approximately 34%) revolves around Machine Learning-based intrusion detection systems that utilize traditional classifiers such as Decision Trees, Random Forests, SVMs,

and ensemble classifiers. These papers are known to emphasize their classification accuracy, their features' behavior, and their performance comparisons with existing systems.Next come Deep Learning techniques with a share of 28 percent. The primary approaches employed by the studies included within this category are CNNs, DNNs, LSTMs, layered architecture, as well as GAN-based learning.

Approximately 22% pertain to feature selection, reduction, as well as optimised feature pipelines. The aim here is to reduce computational requirements, improve scalability, as well as facilitate deployment on edge devices and router level systems, employing concepts such as XGBoost feature ranking, recursive feature elimination, PCA, and statistical feature filtering.

The remaining 16% focus on streaming, edge, and deployment-conscious IDS framework designs. These discuss the importance of real-time traffic analysis, efficient inference with considerations for latency, light-weight architectures, and router-level executable models.

The distribution, on the whole, illustrates that the key area of research in the field of UNSW-NB15 concerns accuracy-oriented ML and DL techniques, whereas only a small amount of research focuses on the feasibility of implementation, capacity of router-level execution, and requirement for real-time anomaly detection. It thereby identifies the author

III. Research Gap

The main things that have research gap at are listed below

1) The Intrusion Detection System models work when they are actually being used. Most of the time people judge these models by how accurate they're instead of looking at how long it takes for them to work how much memory they use or if they can even run properly on the hardware that routers use. Intrusion Detection System models should be tested in a way that takes into account the limitations of the hardware they will be running on like routers. This way we can get a sense of how well the Intrusion Detection System models will really work in real life.
2) Dimensional feature dependency are looking at a problem where many machine learning and deep learning frameworks need a lot of features to work. This means they need to do a lot of computations, which can be a problem. It also means they might not work well on devices that're not very powerful. The dimensional feature dependency is a big issue because it affects how well these frameworks can be used on smaller devices.
3) The UNSW-NB15 there is a problem with minority attack misclassification. This happens because there is not data for some types of attacks which means the system is not very good at finding these rare attacks. The class imbalance in the UNSW-NB15 is what causes this issue leading to recall and weak generalization for these rare attack categories specifically the minority attack misclassification, in the UNSW-NB15.
4) The problem with streaming and online learning support is that many IDS solutions do not work well with real time traffic. They usually work in batch mode. That is not good for local area networks that are always changing. Many IDS solutions cannot keep up with the traffic patterns in these networks because they are always changing. The IDS solutions need to be able to learn and adapt to these changes in time but many of them cannot do that. This is a problem, for IDS solutions because they need to be able to work well in real time LAN environments, like the ones that are used today. IDS solutions need to be able to support streaming and online learning.
5) Deep learning architectures that're too complicated and expensive to use: These deep networks are really good, at detecting things but they have some big problems. They are slow, hard to train and difficult to set up on routers.
6) There is an integration of explainable IDS techniques. This means that most of the approaches do not give us results that we can understand easily. They also do not provide administrator-assisted intrusion reasoning outputs. We need IDS techniques to make sense of the results. The current IDS techniques do not help the administrators to understand the reasoning behind the intrusion detection. This is a problem because explainable IDS techniques are very important, for security.
7) Insufficient focus on edge- and router-aware IDS design Only a small subset of works explicitly considers router-level execution constraints, firmware integration, or gateway-based anomaly detection.

IV. Problem Statement

There is a lack of comprehensive understanding of how existing Machine Learning, Deep learning, hybrid, and feature engineering-based IDS approaches developed using the UNSW NB15 dataset align with the resource, latency, deployment, and operational constraints of LAN and router-level environments. Although numerous IDS models report strong detection accuracy in offline experimental settings, their suitability, efficiency, and practicality for edge- and router-based anomaly detection systems remain insufficiently explored.

V. Dataset Overview

The UNSW-NB15 dataset was developed using the IXIA PerfectStorm range from the Australian Centre for Cyber Security by simulating practical hybrid traffic that comprises both benign as well as malicious behaviors. It is to be noted that the previously existing datasets such as KDD'99 or NSL-KDD were not very representative due to the lack of attacks, protocol features, and patterns that are present in today's networks, contrary to the newly developed dataset UNSW-NB15.

It consists of approximately 2.5 million network flow records, which were captured and analyzed using the Zeek (Bro) and Argus network monitors. It has 49 attributes altogether, including those that are flow-related, content-related, time-related, state-related, and statistics-related. The sheer number of attributes allows the IDS models using machine learning to include network behavior patterns, temporal relationships, as well as network protocol specifics related to LAN and router-level anomalies.

UNSW-NB15 includes nine main types of attack: Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. These attack types comprise a broad range of intrusion activities from reconnaissance to exploit and malware spreading. Several works previously demonstrated the realistic traffic distribution, up-to-date categorization of attacks, and suitability and effectiveness for ML and/or DL approaches for IDS design and implementation that has made UNSW-NB15 one of the prominent IDS benchmarks among recent works.

Despite the advantages, UNSW-NB15 still faces significant modeling difficulties, thereby influencing the classifiers' design and design evaluation processes. The presence of class imbalances, where fewer representations of the minority classes, specifically the attack classes, negatively impacts the recall and consequently the misclassification of the rare attacking processes, such as Backdoor, Shellcode, and Worms, still exists in the dataset. Furthermore, the presence of overlapping distributions in the features of attack classes, which are closely related, makes the multi-class classification a challenging task.

On the whole, the benchmark UNSW-NB15 provides a realistic and comprehensive framework for the exploration of ML-based anomaly detection techniques for LAN and router-level security applications.

VI. PROPOSED METHODOLOGY

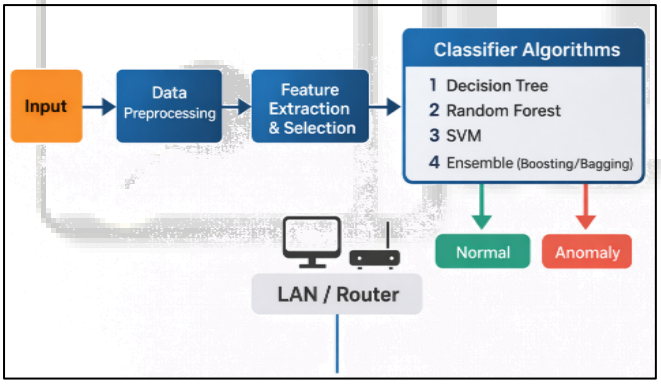

Fig. 2: Proposed Methodology

This figure presents a machine-learning-driven anomaly detection workflow that is tailored to the fine granularities of a network traffic feed, such as a LAN or router-level traffic. Starting from the network traffic input, we take the data through a pre-processing stage, cleaning up noise and normalizing the data to ensure consistency of the dataset; then we categorize numeric values. We then feed the processed data into feature extraction and selection to remove the uninformative or redundant fields, retaining the discriminative attributes of most importance. This is further optimized by the application of statistical feature ranking and dimensionality reduction to enhance model efficiency and reduce computational load.

After refinement in the feature set, the process proceeds to the classifier where various machine-learning models are employed for anomaly detection. The suite of algorithms includes Decision Trees, Random Forests, Support Vector Machines (SVM), and Ensemble Methods (Boosting and Bagging). These classifiers are trained and tested to handle both binary and multi-class intrusion detection.

Traffic is labeled as Normal or Anomalous based on the classifier outputs. This decision is then mapped to the LAN/router execution context, thus signaling how deployable it would be on edge devices or constrained hardware, which generally includes home or enterprise routers and gateway devices. The diagram stresses a feature-optimized classifier-centric IDS workflow with interpretability, modular processing, and suitability for router-based anomaly detection.

VII. SYSTEM FLOW OF PROPOSED FRAMEWORK

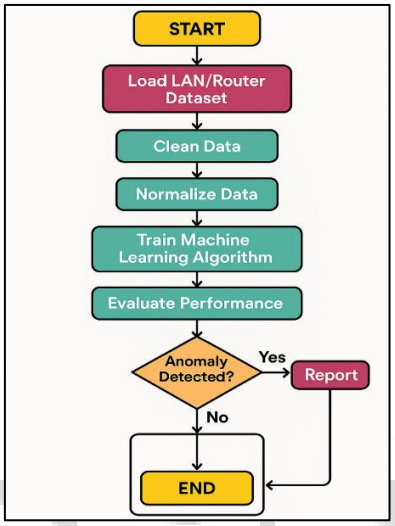

Fig. 3: System Flow Chart

The figure below describes how this machine learning LAN/router-level anomaly detection workflow is performed. It starts with initialization, which involves loading the required LAN/Router dataset and preparing it for analysis according to typical IDS pre-processing pipelines described in related works on UNSW-NB15. The pre-processing will handle inconsistencies in data, such as missing or inconsistent records, and will normalize the data to standardize feature scales across numerical attributes. This helps in normalizing classifiers and improves general detection performance in ML-based IDSs.

After cleaning and normalization, data will flow into model building. In this process, a supervised classifier will be trained on the labeled examples of network traffic. This will enable a variety of algorithmic comparisons to be made, such as Decision Tree, Random Forest, SVM, and Ensemble methods. The methodological choices here reflect those made in prior work on UNSW-NB15 intrusion detection. Once training is complete, the model is evaluated by performance metrics with a view to understand the detection capability, reliability, and generalization across different attack categories.

Subsequently, the decision stage checks for abnormality. In case of any abnormal traffic, the system generates a report or an alert for security monitoring or administrative action; this again aligns with the deployment practices of IDS in both LAN and router contexts. On other occasions, where no anomaly is detected, the process gets over. This modular, pipeline style of work is designed for

LAN and router security environments and provides effective support for ML-driven intrusion monitoring.

VIII. COMPARATIVE ANALYSIS

Ref.	Algorithm	Accuracy in %	Latency	Advantages	Limitations
[3]	Random Forest,SVM	88	Low	Stable Detection	Lower recall for rare class attacks
[7]	Decesion tree, SVM,Random forest,KNN	90	Moderate	Balanced Performance with Tuning	Sensitive to feature Scaling
[8]	Stacking Ensemble classifier	92	High	Strong generalization and Robustness	Increasing The model Complexity
[13	Decesion Tree,Random Forest,NB and SVM	89	Moderate	Multi model comparision	Higher latency on large dataset

Table 1: Comparative Analysis of different algorithm used

Table 1 presents a comparison of various machine learning-based intrusion detection techniques utilizing the UNSW-NB15 dataset. The model based on Random Forest and SVM from [3] achieved an accuracy of 88% with low latency, indicating effective near real-time detection; however, it showed a limited recall for less common attack categories. In [7], a multiclassifier approach that integrates Decision Tree, SVM, Random Forest, and KNN records an accuracy of 90% at moderate latency, demonstrating solid overall performance after optimization, though it may be sensitive to feature scaling, potentially impacting consistency. The stacking ensemble discussed in [8] raises accuracy to 92%, leveraging enhanced generalization and robustness via meta-learning, yet this comes with significantly higher computational latency and increased model complexity, making it less suitable for strict router-level requirements. The comparative analysis in [13], which evaluated Decision Tree, Random Forest, Naïve Bayes, and SVM, yielded an accuracy of 89% with moderate latency, providing useful baseline benchmarks, yet it faced longer inference times with larger datasets. In summary, the findings indicate that ensemble methods generally produce higher accuracy, while lighter ML classifiers provide lower latency, making them potentially more practical for LAN and router-level anomaly detection.

IX. CONCLUSION

The paper reviews machine learning-, deep learning-, ensemble-, and feature-engineering-based IDS proposals developed with the UNSW-NB15 dataset, focusing on their relevance to the problem space of LAN and router-level anomaly detection. The results indicate that classical ML and ensemble classifiers provide good balance in terms of accuracy, robustness, and latency, making them relatively more viable in resource-constrained router environments. The deep learning and hybrid models have achieved higher detection performance but require much higher computational resources and therefore have limited feasibility for actual deployment on edge devices.

It also underlines the main challenges faced, including class imbalance, misclassification of minority attacks, overlap of feature space, inability for streaming and online learning, and limited deployment-centric evaluation. In conclusion, future research should emphasize lightweight ML architectures, feature-efficient pipelines, edge-aware IDS design, and real-time streaming-based detection to enable practical anomaly detection in LAN and router security environments.

REFERENCES

[1] D. K. Bhattacharyya and J. K. Kalita, *Network Anomaly Detection: A Machine Learning Perspective*. CRC Press, 2013. doi: 10.1201/b15088.

[2] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. MILCIS*, 2015, pp. 1–6. doi: 10.1109/MILCIS.2015.7348942.

[3] S. Meftah, T. Rachidi, and N. Assem, "Network based intrusion detection using the UNSW-NB15 dataset," *IJCDS*, vol. 8, no. 5, pp. 478–485, 2019. doi: 10.12785/IJCDS/080505.

[4] G. Gonzalez-Granadillo et al., "LADS: A live anomaly detection system," in *Proc. ICSOFT*, 2019. doi: 10.5220/0007948904640469.

[5] Y. Sun, H. Ochiai, and H. Esaki, "Deep learning-based anomaly detection in LAN from raw network traffic measurement," *IEEE CISS*, 2021. doi: 10.1109/CISS50987.2021.9400241.

[6] D. Vallejo-Huanga, M. Ambuludi, and P. Morillo, "Empirical exploration of ML techniques for NIDS," *IEEE Latin America Trans.*, 2021. doi: 10.1109/TLA.2021.9448311.

[7] S. Moualla, K. Khorzom, and A. Jafar, "Improving ML-based IDS performance on UNSW-NB15," *Comput. Intell. Neurosci.*, 2021. doi: 10.1155/2021/5557577.

[8] M. H. Kabir et al., "Stacking ML approach for UNSW-NB15," *IEEE ICAEEE*, 2022. doi: 10.1109/ICAEEE54957.2022.9836404.

[9] "Classifying UNSW-NB15 traffic using Random Forest in Spark," *IJBDIA*, 2022. doi: 10.4018/ijbdia.287617.

[10] U. C. Akuthota and L. Bhargava, "XGBoost feature selection for intrusion classification," 2023.

[11] N. Pansari et al., "XGBoost feature selection & ablation analysis," *IEEE I²CT*, 2024. doi: 10.1109/I2CT61223.2024.10543523.

[12] F. Mohamed and M. Agarwal, "RFE-based classifier for UNSW-NB15 attack detection," *IEEE I²CT*, 2024. doi: 10.1109/I2CT61223.2024.10544076.

[13] Z. P. Putra, "Evaluating ML classifiers on UNSW-NB15," *Jurnal Ilmiah FIFO*, 2024. doi: 10.22441/fifo.2024.v16i1.009.

[14] S. Gunupusala and S. C. Kaila, "Multi-class anomaly detection," 2024. doi: 10.37256/cm.5220243723.
[15] P. Naresh et al., "Supervised and DL-based anomaly decoding," *IEEE ICACRS*, 2023. doi: 10.1109/ICACRS58579.2023.10404866.
[16] R. Sharma and S. Sobti, "DNN-based IDS for UNSW-NB15," *IEEE AsianCON*, 2024. doi: 10.1109/ASIANCON62057.2024.10837906.
[17] A. D. Vibhute et al., "CNN performance analysis on UNSW-NB15," *Procedia Comput. Sci.*, 2024. doi: 10.1016/j.procs.2024.04.211.
[18] Z. T. Pear and H. B. Kibria, "Parallel ANN alternative to CNNs," *IEEE NCIM*, 2025. doi: 10.1109/NCIM65934.2025.11159982.
[19] M. A. Alsharaiah et al., "Hierarchical deep learning NIDS," *Int. J. Data Netw. Sci.*, 2024. doi: 10.5267/j.ijdns.2024.1.007.
[20] M. Sohail et al., "Conditional GAN-augmented CNN IDS," *CSIT*, 2025. doi: 10.5121/csit.2025.151707.
[21] N. Sharma et al., "Unsupervised anomaly detection review," *IJSSIS*, 2024. doi: 10.2478/ijssis-2024-0016.
[22] P. Schummer et al., "ML-based IDS design and evaluation," *AI*, 2024. doi: 10.3390/ai5040143.
[23] S. Ness et al., "Advanced ML anomaly detection," *IEEE Access*, 2025. doi: 10.1109/ACCESS.2025.3526988.
[24] V. Eswarakrishnan and P. Singla, "Optimized ML attack classification," *IEEE PICOM*, 2024. doi: 10.1109/PICOM64201.2024.00022.
[25] V. Pushkar, "ML-based anomaly detection for threat prevention," 2025. doi: 10.55041/isjem03144.
[26] M. A. Shoyab and M. Ahmed, "Streaming-based ML anomaly detection," 2025. doi: 10.59256/indjcst.20250403006.
[27] J. Anderson, E. Johnson, and M. Brown, "IoT anomaly detection using ML," 2024.

