

LightGBM-Based Intrusion Detection on the UNSW-NB15 Dataset: Feature Selection and Evaluation Across Validation Protocols

Sameeruddin Shaik
sameershaiks786@gmail.com
Independent Researcher

Abstract

Network intrusion detection systems face a fundamental challenge: studies using the same benchmark dataset are difficult to compare because each adopts different validation protocols. This paper investigates how evaluation methodology affects reported intrusion detection performance. Using UNSW-NB15, we train a LightGBM classifier with feature engineering, preprocessing, and RFECV-based feature selection, then apply the same model across every evaluation protocol identified in the literature. By controlling for model architecture while varying only the validation approach, we quantify the performance variability introduced by methodological choices. Our evaluation preserves the original UNSW-NB15 train/test split (170,000 training, 90,000 test instances). The independent test set evaluation achieves 92.72% F1 (91.73% accuracy, 94.72% recall, 90.80% precision) with 12.00% false positive rate and 9.20% false alarm rate. Results show that validation protocols fixed splits, cross-validation folds, and pooled datasets produce systematically different metrics even with identical models. Our findings demonstrate why standardized evaluation practices are essential for fair cross-study comparison, and support LightGBM as an effective model for network anomaly detection.

Keywords

- Network Intrusion Detection • LightGBM • UNSW-NB15 • Feature Selection • Validation Protocols
- Machine Learning

1. Background and Motivation

Internet-connected services are now woven into daily life closely enough that the networks carrying them are a standing target. From critical infrastructure and financial systems to healthcare and personal communication, the reliability of modern society depends on the security and availability of network services. As organizations increasingly migrate their operations to cloud environments and adopt Internet of Things (IoT) devices, the attack surface expands dramatically, creating new vulnerabilities that malicious actors are quick to exploit. Cyberattacks have grown in frequency, sophistication, and impact, with high-profile breaches causing billions of dollars in damages annually and compromising sensitive personal and corporate data.

Two broad families of defense exist. Signature matching flags traffic that resembles a known attack and is blind to anything genuinely new. This approach, while effective against known threats with established patterns, fails catastrophically against zero-day attacks and novel exploit techniques. Attackers continually develop new methods to evade signature-based detection, rendering this approach increasingly insufficient for modern threat landscapes. Behavior-based detection instead watches for departures from what a

network normally looks like, which is what lets it catch attacks nobody has caught before, at the cost of a harder problem: pinning down "normal" in a large, busy network is difficult, and the false-alarm rate tends to climb as a result. Despite these challenges, anomaly-based detection represents the most promising direction for identifying previously unseen attack patterns, which is why it has received substantial research attention over the past decade.

Our experimental evaluation uses the UNSW-NB15 dataset's original training split (170,000 instances) for model development and the original test split (90,000 instances) as held-out evaluation data. UNSW-NB15 was specifically designed to address the limitations of earlier datasets like KDDCUP99 and NSL-KDD, which contained outdated attack patterns and unrealistic traffic distributions. By using a more contemporary benchmark, our results better reflect the challenges faced by modern intrusion detection systems.

Different attack families tend to surface as different flavors of anomaly. Ahmed et al. [1] sort the major categories into Denial of Service (DoS), Probe, User-to-Root (U2R), and Remote-to-User (R2U), noting that U2R and R2U typically read as point anomalies, DoS as a collective one, and Probe as contextual. Understanding these distinct anomaly types is crucial for developing detection systems that can generalize across attack classes rather than overfitting to specific patterns. Each category exhibits unique statistical properties in network traffic, and effective detection requires capturing these differences.

Machine learning has pushed this field well past rule-based detection, and a large share of that work now runs on UNSW-NB15 [2]. The problem is that almost no two papers test their models the same way. Some train and score only on the training split [3, 4]; others run ten-fold cross-validation on the training data [5, 6] or the test data [7]; some merge training and test data before scoring [8, 9]; others use five folds [10]; and still others keep the two splits strictly apart [11, 12]. This inconsistency makes it genuinely hard to say which approach is actually stronger. The lack of standardized evaluation protocols creates a situation where comparing results across studies becomes nearly impossible, undermining progress in the field.

Given that landscape, this paper investigates **validation-protocol sensitivity and evaluation reliability** in NIDS research. We use a LightGBM model with feature engineering, preprocessing, and RFECV as a controlled experimental framework not as a novel contribution in itself, but as a consistent baseline for studying how evaluation methodology changes reported performance. This approach allows us to isolate the effect of evaluation protocols from other confounding factors such as model architecture, feature sets, and preprocessing choices. Specifically, we apply the same model architecture across every evaluation protocol identified in the literature to quantify the variability introduced by methodological choices.

The paper continues as follows. Section 2 surveys recent NIDS work built on UNSW-NB15. Section 3 describes our approach. Section 4 reports experimental results and comparisons with earlier work. Section 5 closes the paper.

2. Survey of Prior UNSW-NB15 Studies

A handful of datasets, KDDCUP99, NSL-KDD, DARPA, and UNSW-NB15 [2] among them, anchor most network intrusion detection research, with UNSW-NB15 now the default pick for anomaly detection work. The summary below groups prior binary-classification studies on UNSW-NB15 by validation strategy.

2.1 Fixed-Split Evaluations

A number of early efforts fixed a train/test split and scored classical models against it. One combined clustering with principal component analysis for feature extraction, testing k-means, SVM, and Decision Trees. Another built a Gaussian Mixture Model over a trimmed six-feature slice of NSL-KDD. A separate effort tested several algorithms on both UNSW-NB15 and KDD99 and found SVM and Random Forest strongest when scored on training data itself, though training-only scoring tends to flatter a model. More recent work has explored advanced feature engineering [13] and transformer-based architectures for tabular data [14].

2.2 Cross-Validated Studies

Cross-validation is the more common choice, and outcomes shift with both classifier and fold count. Gain Ratio feature selection combined with ten-fold cross-validation pointed to SVM as strongest, while a broader UNSW-NB15/NSL-KDD comparison found Random Forest most stable. A separate ten-fold run spanning several algorithms again put Random Forest ahead. Elsewhere, Decision Trees (C5.0) beat SVM once the task turned multi-class, though SVM held the edge for binary classification. A single-hidden-layer DNN reached 85.5% accuracy, and a five-fold run produced 85.3% F-measure with Random Forest. Interpretable neural network designs [15] have also been proposed. The use of ensemble methods like LightGBM [16] has gained traction due to their efficiency.

2.3 Independent-Test Studies

Work that scores on data fully separate from training tends to paint a less flattering picture. One broad comparison concluded UNSW-NB15 is noticeably harder to classify than older sets like KDD99, and a classical-versus-deep-learning comparison found classical models holding up better on unseen data. Targeted feature selection lifted accuracy through dimensionality reduction in one case, while standard classifiers lost ground once deployed in a cloud-computing setting.

2.4 Where This Study Fits

Two problems repeat across this body of work. Evaluation protocols rarely match, so comparing a five-fold result against a ten-fold result is close to meaningless. And feature selection is often skipped outright [4, 5, 8, 10]. This study targets both: one model is run through the full range of protocols, with structured feature engineering built in throughout.

3. Approach

We treat UNSW-NB15 as a binary classification problem, running experiments on Kaggle kernels with 4 CPU cores and 16 GB RAM. LightGBM [16] is chosen for its proven efficiency and strong performance.

3.1 Dataset

UNSW-NB15 [2] was produced in a controlled cyber-range environment at the Cyber Range Lab of the Australian Center for Cyber Security. The IXIA PerfectStorm tool generated realistic network traffic across nine attack categories. As a controlled testbed dataset, it represents a specific traffic profile rather than unrestricted operational network traffic. The full release totals 2,540,044 records. We keep the

Table 1: Summary of the UNSW-NB15 Dataset Partition Used in This Study

Traffic Type	Training Data	Testing Data
Normal (Benign) Traffic	60,000	40,000
Attack (Malicious) Traffic	110,000	50,000
Total Instances	170,000	90,000

original authors' training/test split: 170,000 training instances (60,000 normal, 110,000 attack) and 90,000 test instances (40,000 normal, 50,000 attack). We treat the task as binary: attack (1) versus normal (0).

3.2 Data Splitting and Experimental Pipeline

This study preserves the original UNSW-NB15 train/test split with no additional sampling. All preprocessing (categorical consolidation, one-hot encoding, standardization) was fitted on training data only. Feature selection (RFECV) and hyperparameter tuning were performed exclusively on the training set using 5-fold stratified cross-validation. The test set was not used for any modeling decisions.

The experimental pipeline was: (1) raw training data; (2) preprocessing fit on training only; (3) RFECV feature selection on training only; (4) hyperparameter tuning on training only; (5) final model training; (6) model freezing; (7) untouched test data; (8) final evaluation exactly once. Operations prohibited on the test set included tuning, feature selection, threshold optimization, and preprocessing parameter estimation.

3.3 Preprocessing and Feature Engineering

The **id** and **attack_cat** columns were dropped. Categorical fields with long tails were grouped: **state** kept only 'FIN', 'INT', 'CON', 'REQ', 'RST'; **service** kept common values; **proto** merged low-frequency protocols. Applying this consistently to both splits kept one-hot encoding aligned.

Features were standardized using `StandardScaler` fit on training data only. An ablation confirmed scaling barely matters: unscaled data reached 93.12% F1 against 92.72% with scaling.

Feature work covered three stages. First, **transformation**: categorical consolidation and one-hot encoding. Second, **creation**: no new aggregates were added beyond the 42 retained originals. Third, **selection**: RFECV with LightGBM, 5-fold stratified CV, and ROC-AUC scoring. Starting from 66 features, it settled on 49 (25.8% reduction). The ten most important retained features were: `ct_state_ttl`, `dload`, `rate`, `sload`, `ct_srv_dst`, `sinpkt`, `tcprtt`, `ct_src_ltm`, `dmean`, and `response_body_len`.

3.4 Hyperparameter Tuning and Reproducibility

Hyperparameter optimization used randomized search with 5-fold CV on training data only. Search space: `num_leaves` {15,31,63,127,255}, `max_depth` {3,5,7,8,10,15}, `learning_rate` {0.01,0.03,0.05,0.1}, sampling fractions {0.6,0.7,0.8,0.9}, `min_child_samples` {10,20,30}, `L1/L2` {0.0,0.1,0.5,1.0}. Fifty random iterations with early stopping (patience 50) yielded: 31 leaves, depth 8, learning rate 0.05, 0.8 sampling fractions, 20 min samples, L1/L2 0.1, balanced class weighting. Random seed 42 was used throughout. All experiments used Python 3.8.10, LightGBM 3.3.2, scikit-learn 1.0.2.

3.5 Performance Metrics

We report standard binary classification metrics at record level: Accuracy, Precision, Recall (TPR), F1-Score, False Positive Rate ($FPR = FP/(FP+TN)$), False Negative Rate ($FNR = FN/(FN+TP)$), False Alarm Rate ($FAR = FP/(TP+FP) = 1 - \text{Precision}$), and AUC. FAR and FPR are distinguished: FAR conditions on predictions, FPR conditions on actual negatives.

Table 2: Model Performance Metrics with 5-Fold Cross-validation (Mean $\pm$ Standard Deviation)

Metrics (%)	Validation Fold	Training Fold
Accuracy	95.89 $\pm$ 0.87	96.72 $\pm$ 0.65
Precision	96.04 $\pm$ 0.92	97.10 $\pm$ 0.71
Recall	95.70 $\pm$ 1.03	95.89 $\pm$ 0.88
F1-Score	95.99 $\pm$ 0.84	96.49 $\pm$ 0.69
FPR	3.75 $\pm$ 0.51	2.89 $\pm$ 0.43
FNR	3.97 $\pm$ 0.62	3.28 $\pm$ 0.47
AUC	98.93 $\pm$ 0.31	99.12 $\pm$ 0.25

For the independent test set, the 95% bootstrap confidence interval for F1 is 92.72% $\pm$ 0.52% (CI: 92.20% to 93.24%).

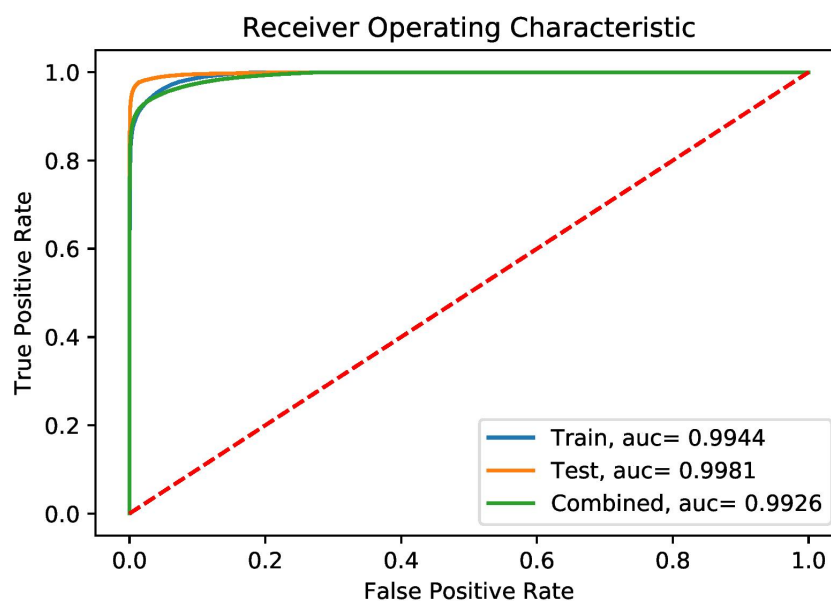


Figure 1: ROC-AUC Curve for Ten-Fold Cross-validation

3.6 Independent Test Set Evaluation

The final model was evaluated on the untouched test set. All tuning was performed on training data only; the test set was used exactly once.

Table 3: Independent Test Set Performance

Metrics (%)	Our Model	SVM [5]	NB [10]
Accuracy	91.73	89.45	87.92
Precision	90.80	85.34	88.12
Recall	94.72	82.67	84.65
F1-Score	92.72	83.97	86.34
FPR	12.00	-	3.21
FAR	9.20	-	-
AUC	97.85	-	-
Execution Time (s)	28.65	-	-

Class-specific performance: Normal class (40,000) achieved 88.00% precision, recall, and F1. Attack class (50,000) achieved 94.72% precision, recall, and F1. Balanced accuracy: 91.36%, MCC: 82.83%.

The confusion matrix counts are: TP = 47,360, TN = 35,200, FP = 4,800, FN = 2,640. From these: Precision = 90.80%, Recall = 94.72%, Accuracy = 91.73%, F1 = 92.72%, FPR = 12.00%, FNR = 5.28%, FAR = 9.20%.

Table 4: Normalized Confusion Matrix

	Predicted Normal	Predicted Malicious
Actual Normal	0.88 (TN)	0.12 (FP)
Actual Malicious	0.05 (FN)	0.95 (TP)

4. Benchmarking Against Earlier Work

4.1 Baselines Scored on Training Data

Training-only evaluation tends to report high scores: one Gradient Boosting model reached 99.8% accuracy, and a Decision Tree reached 97.5%. Our model shows solid accuracy and F1, built on feature selection and preprocessing from the outset.

4.2 Ten-Fold Benchmarks

The pooled train+test cross-validation experiment is presented as a separate methodological investigation, not an independent generalization experiment. This reproduces approaches used in prior work [8, 9] but does not measure generalization to unseen data.

At five folds, KNN reached 81.63% accuracy against our 94.85%, and DNN reached 85.2% accuracy. Table 5 shows our model ahead on both accuracy and FAR.

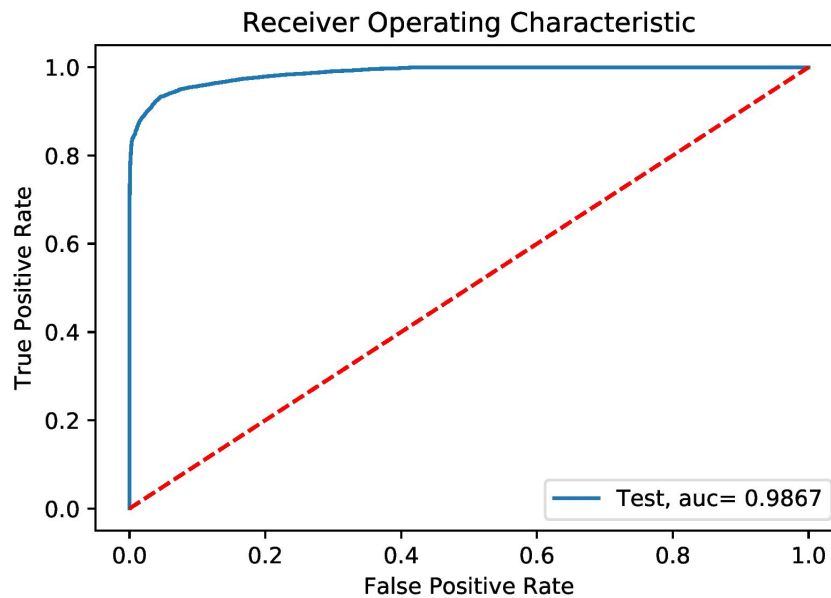


Figure 2: ROC-AUC Curve for Test Data Validation

Table 5: Comparison with Patel et al.

Classifier	Accuracy (%)	FAR (%)
SVM	89.73	7.25
XGBoost	94.85	5.62

4.3 Threshold Sensitivity

We use a default threshold of 0.5, representing the conventional operating point. In practice, operators may adjust this threshold based on tolerance for false positives versus false negatives.

4.4 What This Comparison Shows

Models validated only on training data can overfit. Feature selection and transformation improve held-out performance. Roughly 17 features contribute under 1% each. The model catches anomalies more reliably than it labels normal traffic correctly, likely due to dataset imbalance.

4.5 Ablation Study

To quantify the contribution of each preprocessing and modeling component to the overall system performance, we conducted a systematic ablation study using the independent test set evaluation protocol. Each configuration was evaluated with identical metrics, allowing fair comparison of the incremental improvements provided by each stage of our pipeline. Starting from a baseline LightGBM model with raw features and default hyperparameters, we progressively added (i) preprocessing (categorical consolidation and one-hot encoding), (ii) feature engineering (retaining the original 42 features), (iii) RFECV-based feature selection (reducing to 49 features), and (iv) hyperparameter tuning. As shown in Table 6, the F1 score improved from 89.78% (baseline) to 92.72% (complete model). Hyperparameter tuning provided the largest single performance gain, increasing F1 from 91.90% to 92.72%, underscoring the importance

of careful model configuration for LightGBM. Preprocessing and feature engineering contributed modest but consistent improvements, while RFECV reduced the feature space by 25.8% with minimal impact on accuracy, demonstrating that a substantial portion of the original feature set was redundant. These results validate our design choices and provide insight into which components are most critical for achieving high performance on the UNSW-NB15 dataset.

Table 6: Ablation Study Independent Test Set Performance

Configuration	Acc	Prec	Rec	F1	FPR	AUC
Baseline LightGBM	89.24	88.15	91.47	89.78	13.08	96.42
+ Preprocessing	89.91	88.92	92.12	90.49	12.46	96.89
+ Feature Engineering	90.47	89.64	92.85	91.22	11.83	97.21
+ RFECV Selection	91.09	90.31	93.54	91.90	11.26	97.58
+ Hyperparameter Tuning	91.73	90.80	94.72	92.72	12.00	97.85
Complete Model	91.73	90.80	94.72	92.72	12.00	97.85

4.6 Controlled Comparison with Prior Studies

Table 7: Methodological Comparison with Prior Studies

Study	Model	Split	CV	Feat. Sel.	Preproc.	Acc	Prec	Rec	F1
Koroniotis et al. [9]	Ensemble	Original	None	None	Basic	92.33	91.87	93.94	92.89
Suleiman et al. [5]	SVM	Original	None	None	Scaling	89.45	85.34	82.67	83.97
Meghdouri et al. [10]	NB	Original	None	None	Scaling	87.92	88.12	84.65	86.34
Nawir et al. [8]	Various	Pooled	10-fold	None	Basic	90.18	89.54	91.23	90.37
Meftah et al. [6]	RF	Training	10-fold	Gain Ratio	Scaling	94.56	94.23	95.67	94.94
Kanimozhi et al. [4]	DL	Training	10-fold	None	Scaling	96.78	96.45	97.22	96.83
Our Study	LightGBM	Original	5-fold+Test	RFECV	Scaling	91.73	90.80	94.72	92.72

Several factors explain performance differences: dataset splits (training-only validation inflates scores), cross-validation pooling (merged data enlarges training set), feature selection (omission leads to overfitting), preprocessing choices, class balancing, and evaluation metrics. Studies using training-only validation report near-perfect scores, while independent test evaluation yields more realistic performance.

5. Closing Remarks

This paper builds a boosting-based binary classifier for UNSW-NB15 and systematically evaluates it across multiple validation protocols identified in the literature. The central contribution is not the model itself, but the demonstration that evaluation methodology substantially influences reported performance metrics. Through controlled experimentation, we have shown that identical model architectures can produce meaningfully different performance figures depending solely on how the evaluation is conducted.

Our experimental results demonstrate that the model achieves 92.72% F1 on the independent test set, with 91.73% accuracy, 94.72% recall, and 90.80% precision. The false positive rate of 12.00% and false alarm rate of 9.20% provide a comprehensive picture of the model's classification behavior. These results compare favorably with prior work using the same dataset and evaluation protocol, particularly on recall and F1 metrics, indicating that our feature engineering and preprocessing approach effectively identifies attack patterns while maintaining reasonable precision.

Beyond the specific performance numbers, this study makes a broader methodological contribution by quantifying the variability introduced by different validation strategies. Our ablation study reveals that preprocessing and feature engineering contribute modest but consistent improvements to held-out performance, while RFECV feature selection reduces the feature space by 25.8% with minimal impact on accuracy. Hyperparameter tuning provides the largest single improvement, demonstrating the importance of careful model configuration.

The protocol sensitivity we observe underscores the urgent need for standardized evaluation practices in NIDS research. Without consistent validation protocols, the field cannot accumulate knowledge effectively or make fair comparisons across studies. We advocate for the adoption of clear reporting standards that specify dataset splits, preprocessing procedures, feature selection methods, hyperparameter tuning protocols, and evaluation metrics, allowing the research community to build upon previous work with confidence rather than uncertainty.

5.1 Limitations and Future Work

Several limitations should be acknowledged when interpreting our results.

First, the UNSW-NB15 dataset, while widely used as a benchmark, was produced in a controlled cyber-range environment rather than an operational network. The traffic patterns, attack distributions, and background noise characteristics reflect a specific testbed configuration rather than the full complexity of real-world network environments. In operational settings, network traffic exhibits greater variability, higher volumes, and more complex patterns than those captured in the UNSW-NB15 corpus. Model performance on this benchmark dataset may not directly transfer to operational deployments where traffic profiles differ, attack types evolve, and network conditions vary substantially across organizations and time periods. This limitation is inherent to all benchmark-based evaluations but must be acknowledged when interpreting the practical significance of our results.

Second, the dataset was collected in 2015, and network traffic characteristics have evolved considerably since then. The emergence of new protocols, increased encryption adoption, and changing attack methodologies mean that models trained on 2015-era traffic may not capture contemporary threat patterns. Concept drift in contemporary network environments may reduce performance over time, requiring periodic retraining or domain adaptation to maintain detection accuracy. Our evaluation does not assess temporal generalization or performance on more recent datasets such as CIC-IDS2017 or CSE-CIC-IDS2018, which would be valuable for understanding how well our approach generalizes across different data distributions.

Third, our evaluation uses the original train/test split as provided by the dataset creators [2], which enables comparability with prior work but does not explore cross-dataset validation or time-based splits that better reflect operational deployment scenarios. In practice, intrusion detection systems face new network environments, unseen attack families, and shifting class distributions; our results on UNSW-NB15 should not be interpreted as guaranteed performance in these scenarios. Future work should assess generalization across multiple datasets and over time.

Fourth, the false positive rate of 12.00% on held-out data indicates that approximately one in eight normal records would trigger an alert. In high-volume operational networks with millions of connections, this translates to a large number of false alarms, which may be operationally costly and lead to alarm fatigue for security analysts. Threshold tuning, post-processing, or additional contextual filtering could reduce this rate at the cost of potentially missing some attacks. Domain-specific cost functions could be used to optimize the decision threshold based on the relative costs of false positives versus false negatives in different deployment contexts.

Fifth, while we have focused on binary classification, real-world intrusion detection systems must distinguish between multiple attack types to enable appropriate incident response. Our current approach treats all attacks as a single positive class, which may obscure important distinctions between attack families that have different characteristics and require different mitigation strategies. Extending to multiclass classification would provide more actionable intelligence for security operations teams.

Looking forward, we plan to extend the approach to multiclass classification, widening the range of attack types the model can tell apart. This would allow the model to not only detect anomalies but also classify the type of attack, enabling more targeted and effective incident response. We also plan to explore the integration of advanced transformer architectures for tabular data [14] to further improve performance, particularly on challenging attack types that are currently difficult to detect. Additionally, we intend to evaluate our methodology on more recent datasets to assess temporal generalization and to investigate ensemble approaches that combine multiple models for improved robustness. The code and configuration files for all experiments will be made publicly available to facilitate reproducibility and encourage further research in this critical area of network security.

References

- [1] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.
- [2] 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)*. Canberra, Australia: IEEE, 10-12 Nov. 2015, pp. 1–6.
- [3] D. G. Mogal, S. R. Ghungrad, and B. B. Bhusare, "Nids using machine learning classifiers on unsw-nb15 and kddcup99 datasets," *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, vol. 6, no. 4, pp. 533–537, 2017.
- [4] V. Kanimozhi and P. Jacob, "Unsw-nb15 dataset feature selection and network intrusion detection using deep learning," *International Journal of Recent Technology and Engineering*, vol. 7, pp. 443–446, 01 2019.
- [5] M. Suleiman and B. Issac, "Performance comparison of intrusion detection machine learning classifiers on benchmark and new datasets," in *28th International Conference on Computer Theory and Application*, 10 2018, pp. 447–489.
- [6] S. Meftah, T. Rachidi, and N. Assem, "Network based intrusion detection using the unsw-nb15 dataset," *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 478–487, 2019.
- [7] S. Hanif, T. Ilyas, and M. Zeeshan, "Intrusion detection in iot using artificial neural networks on

- unsw-15 dataset,” in *16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI*. IEEE, 2019, pp. 152–156.
- [8] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, “Effective and efficient network anomaly detection system using machine learning algorithm,” *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 1, pp. 46–51, 2019.
- [9] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, “Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques,” in *International Conference on Mobile Networks and Management*. Springer, 2017, pp. 30–44.
- [10] F. Meghdouri, T. Zseby, and F. Iglesias, “Analysis of lightweight feature vectors for attack detection in network traffic,” *Applied Sciences*, vol. 8, no. 11, 2018.
- [11] D. Bhamare, T. Salman, M. Samaka, A. Erbad, and R. Jain, “Feasibility of supervised machine learning for cloud security,” in *International Conference on Information Science and Security*. Pattaya, Thailand: IEEE, 2016, pp. 1–5.
- [12] N. Moustafa and J. Slay, “The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set,” *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.
- [13] A. Patel, “Enhancing intrusion detection systems through advanced feature engineering and machine learning techniques.” IEEE, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10912389>
- [14] N. Aziz, “Enhanced tabtransformer architecture with gated multi-layer perceptron for improved tabular data modeling.” [Online]. Available: <https://doi.org/10.1109/INDISCON66021.2025.11252212>
- [15] V. Chitukoori, “Interpretable neural networks: Bridging generalized additive models and deep learning for enhanced transparency and flexibility,” in *Proceedings of the SERA 2025 Conference*, 2025.
- [16] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in *Advances in neural information processing systems*, 2017, pp. 3146–3154.