International Journal for Rapid Research in Engineering Technology & Applied Science (IJRRETAS)

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume 12 issue 07 July 2026 OPEN ACCESS E-ISSN: 2455-4723 Impact Factor 7.95

A Robust LightGBM-Based Machine Learning Framework for Network Outlier Detection

Rohit Chandra

Research Scholar, M. Tech, Department of Computer Science and Engineering, YBN University Ranchi

Dr. Krishna Murari

Supervisor, H.O.D., Department of computer Science and Engineering, School of Engineering and Technology, YBN University Ranchi-10.

ABSTRACT

The growing volume and complexity of network traffic have made manual and signature-based monitoring insufficient for detecting malicious activity. This paper presents a robust machine learning framework, based on the Light Gradient Boosting Machine (LightGBM), for detecting outliers in network traffic data. The framework integrates data cleaning, one-hot encoding, logarithmic transformation, mutual-information-based feature selection, standardisation and a regularised LightGBM classifier trained with early stopping. It was evaluated on the NSL-KDD benchmark and compared with Logistic Regression, Random Forest and a Support Vector Machine under identical conditions. On a held-out test set of 25,193 connections, LightGBM achieved an accuracy of 99.933%, a precision of 99.957%, a recall of 99.898%, an F1-score of 99.927% and a ROC-AUC of 0.99999, making only 17 errors. It outperformed all baseline models while training about four times faster than Random Forest. SHAP-based interpretation showed that byte volume, connection rate, service concentration and SYN-error features drive detection. Robustness experiments confirmed stable performance across data splits and random seeds, and resilience to severe class imbalance, with an F1-score of 97.74% when outliers formed only 1% of traffic. On an independent test set containing unseen attack types, accuracy fell to 78.54%, identifying generalisation to novel attacks as the main limitation. The results show that LightGBM offers an accurate, efficient and interpretable foundation for network outlier detection.

Keywords: network outlier detection; anomaly detection; intrusion detection; LightGBM; gradient boosting; SHAP; NSL-KDD

1. INTRODUCTION

1.1 Background of Network Security

Computer networks underpin banking, healthcare, government, commerce and personal communication, and the continuous growth of cloud services, mobile access and connected devices has made them larger and more complex than ever. This openness creates value but also exposes every connected system to potential attack. Denial of Service attacks exhaust the resources of targets, probing attacks map networks for weaknesses, and intrusion attacks attempt to gain unauthorised access or escalate privileges. Because traditional perimeter defences such as firewalls only enforce rules about permitted traffic, organisations increasingly

rely on continuous monitoring to identify malicious activity hidden within legitimate flows (Khraisat et al., 2019).

The scale of this challenge continues to grow. Encrypted protocols now protect most web traffic, which prevents inspection of payloads and forces detection to rely on observable metadata such as byte counts, connection timing and service usage. At the same time, automated attack tools and botnets allow even unskilled actors to generate large volumes of hostile traffic. Monitoring systems must therefore examine every connection quickly, identify suspicious behaviour from connection-level characteristics and present analysts with a manageable number of reliable alerts.

1.2 Network Outliers and Anomalous Traffic

A network outlier is a connection or traffic pattern that departs from the behaviour produced by normal network use. Such departures may be caused by attacks, misconfigurations or unusual but benign events. Outliers are rarely extreme in a single attribute; more often they are defined by unusual combinations of attributes, such as a connection that transfers no data while forming part of a burst of connections that end in errors. In this paper, every connection labelled as an attack is treated as an outlier and every legitimate connection as normal, which allows outlier detection to be framed as a supervised binary classification problem.

Network outliers take several forms. Point anomalies are individual connections that are unusual in themselves, contextual anomalies are unusual only in a particular setting, and collective anomalies are groups of connections that are abnormal together even when each appears ordinary. Flooding and scanning attacks typically appear as collective anomalies, which is why the time-based and host-based traffic statistics that summarise recent connections to the same host or service are especially informative. Intrusions into user sessions, by contrast, often appear as subtle point anomalies that closely resemble normal activity.

1.3 Intrusion Detection Approaches

Intrusion detection systems monitor traffic and raise alerts when suspicious behaviour is observed. Signature-based systems compare traffic with known attack patterns and are precise, but they cannot recognise attacks for which no signature exists. Anomaly-based systems model the difference between normal and abnormal behaviour and can generalise beyond known patterns, but they have historically suffered from high false alarm rates (Khraisat et al., 2019; Liu & Lang, 2019). Modern traffic volumes also require detection to be fast and automated, since analysts cannot inspect millions of connections manually, and alerts must be accurate enough to avoid overwhelming them.

The effectiveness of any detection approach is ultimately judged by two competing error rates. A missed attack may allow an intruder to establish persistent access, while a false alarm consumes analyst time and, if frequent, leads to alert fatigue in which genuine alerts are ignored. An effective detector must keep both error rates low simultaneously, and it must continue to do so when attacks are rare, when traffic changes over time and when new attack types appear.

1.4 Machine Learning for Anomaly Detection

Machine learning enables detectors to learn complex relationships among many traffic features directly from labelled data and to classify new connections in microseconds. A wide range of algorithms has been applied to the task, from linear models and support vector machines to random forests, gradient boosting and deep neural networks (Ahmad et al., 2021). However, recent critical work has shown that many evaluations rely on single random splits of benchmark data, omit computational cost and seldom test generalisation, which can overstate real-world performance (Arp et al., 2022). A credible detection framework must therefore be assessed for efficiency, interpretability and robustness as well as accuracy.

Machine learning approaches differ in the kinds of relationships they can represent and in their computational cost. Linear models are fast and transparent but cannot capture the non-linear and interacting patterns typical of attacks. Kernel methods can learn non-linear boundaries but scale poorly to large datasets. Deep neural networks are highly expressive but demand substantial data, tuning and computing resources, and their decisions are difficult to explain. Tree-based ensembles combine flexibility with efficiency and offer natural measures of feature importance, which makes them attractive for tabular connection records.

1.5 LightGBM and Gradient Boosting

LightGBM is a gradient boosting decision tree framework designed for speed and memory efficiency on large datasets (Ke et al., 2017). It builds an ensemble of trees sequentially, with each tree correcting the errors of the previous ones, and it introduces histogram-based split finding, gradient-based one-side sampling, exclusive feature bundling and leaf-wise tree growth. These properties suit network traffic, which is voluminous, heterogeneous, skewed and sparse after categorical encoding. LightGBM also supports exact SHAP explanations of individual predictions (Lundberg et al., 2020). This paper develops and systematically evaluates a LightGBM-based framework for network outlier detection, comparing it with conventional classifiers and assessing its robustness under varied conditions.

Leaf-wise growth is particularly relevant to network data. Instead of expanding every node at a given depth, LightGBM splits the leaf that yields the greatest reduction in loss, which allows it to concentrate model capacity on the difficult boundary between normal sessions and subtle attacks while separating distinctive flooding traffic with only a few splits. Because this strategy can overfit, the framework in this paper controls tree complexity through leaf limits, minimum leaf sizes, subsampling, regularisation and validation-based early stopping.

2. LITERATURE REVIEW

2.1 Network Anomaly Detection Techniques and Datasets

Surveys of intrusion detection classify approaches into signature-based and anomaly-based methods and identify the scarcity of realistic datasets as a central obstacle (Khraisat et al., 2019). Ring et al. (2019) reviewed network intrusion datasets and found that none satisfied all criteria of realism, labelling, attack diversity and currency, while noting that the KDD family, including NSL-KDD, remained widely used because it enables comparison with prior work. Newer datasets such as CIC-IDS2017 were created to reflect modern traffic and attacks

(Sharafaldin et al., 2018), yet Engelen et al. (2021) showed that CIC-IDS2017 contains labelling and feature-extraction errors that alter measured classifier performance. Arp et al. (2022) further identified common pitfalls in security-related machine learning, including sampling bias, data snooping and inappropriate performance measures. These studies indicate that careful data preparation and realistic evaluation are as important as the choice of algorithm.

A further implication of this literature concerns the difference between within-dataset and cross-distribution performance. Random train-test splits of a single dataset place very similar records in both partitions, so high accuracy on such splits mainly demonstrates that a model has learned known attack patterns. Testing on data containing different attacks provides a more demanding measure of generalisation. NSL-KDD is useful in this respect because its separate test file includes attack types absent from training, allowing both kinds of performance to be measured with the same model.

2.2 Machine Learning and Deep Learning Approaches

Classical and ensemble classifiers have been studied extensively. Resende and Drummond (2018) surveyed random-forest-based intrusion detection and reported high accuracy, robustness to noisy features and built-in feature importance, alongside notable computational cost for large forests. Kilincer et al. (2021) compared classical methods across several datasets under common pre-processing and found that tree-based methods generally performed strongly. Deep learning has also been widely applied. Yin et al. (2017) used recurrent neural networks for binary and multi-class detection on NSL-KDD and reported gains over several classical classifiers, and Vinayakumar et al. (2019) evaluated deep neural networks across multiple intrusion datasets and found that no single architecture performed best everywhere. Liu and Lang (2019) observed that deep learning studies rarely report training and detection times. Outside the security domain, Grinsztajn et al. (2022) showed that tree-based models still outperform deep learning on typical tabular data, which is the form taken by connection-level traffic records. Class imbalance remains a recurring challenge, and Talukder et al. (2024) combined oversampling with feature embedding and dimensionality reduction, reporting very high accuracies for tree-based models on large imbalanced datasets.

Two conclusions emerge from this body of work. First, reported accuracies are often very high but depend strongly on the dataset and evaluation procedure, so comparisons between studies that use different pre-processing and partitions are difficult to interpret. Second, tree-based ensembles appear consistently among the best-performing methods on tabular traffic data, whether used alone or after resampling, and they generally achieve this at a fraction of the computational cost of deep networks. These observations motivate both the choice of LightGBM and the decision to compare it with baseline models under identical conditions.

2.3 LightGBM-Based Detection and Explainability

Several studies have applied LightGBM directly to intrusion detection. Jin et al. (2020) proposed SwiftIDS, a real-time system combining LightGBM with parallel detection across time windows, and evaluated it on KDD Cup 1999, NSL-KDD and CIC-IDS2017. Liu et al. (2021) combined adaptive synthetic oversampling with LightGBM to improve the detection of

minority attacks while keeping training and detection times short. Seth et al. (2021) paired hybrid feature selection with LightGBM on CIC-IDS2018, achieving 97.73% accuracy with low prediction latency. Explainability has become increasingly important: Lundberg and Lee (2017) introduced SHAP, and Lundberg et al. (2020) developed an exact algorithm for tree ensembles. Chicco and Jurman (2020) argued that the Matthews correlation coefficient gives a more truthful summary than accuracy under class imbalance. Despite these advances, few studies combine a fair comparison with conventional classifiers, efficiency measurement, SHAP-based interpretation and systematic robustness testing within a single framework. This paper addresses that gap.

The existing LightGBM studies share a focus on efficiency and report strong detection performance, but they differ in datasets, pre-processing and evaluation, and each emphasises a particular enhancement, such as parallel detection, oversampling or feature selection. Their evaluations are largely confined to within-dataset performance, and few examine how the model behaves as the proportion of attacks decreases or when it encounters unseen attack types. The present framework builds on this work by embedding LightGBM in a leakage-free pipeline, triangulating SHAP explanations with statistical evidence, and reporting both strengths and limitations through a structured robustness evaluation.

3. RESEARCH METHODOLOGY

The study adopted a quantitative experimental design in which network outlier detection was formulated as supervised binary classification. The NSL-KDD dataset was used, comprising a training file (KDDTrain+) of 125,973 connection records and an independent test file (KDDTest+) of 22,544 records. Each record is described by 41 features grouped into basic connection, content, time-based traffic and host-based traffic features, of which three (protocol_type, service and flag) are categorical. Records labelled normal were assigned to class 0 and all attack records to class 1 (outlier). Attack types were also mapped to the DoS, Probe, R2L and U2R families for analysis. KDDTest+ contains 17 attack types that do not appear in the training file, so it was reserved as an independent test of generalisation.

NSL-KDD was chosen for three reasons. It is publicly available, which supports reproducibility; it is widely used, which allows the results to be related to prior work; and its separate test file allows generalisation to unseen attacks to be measured directly. Two design principles guided the experiments: all models were compared on exactly the same records and features, and no information from the test data was allowed to influence any stage of model development.

Pre-processing began with a check for missing values, of which none were found, and the removal of 9 exact duplicate training records, leaving 125,964 records. The constant feature num_outbound_cmds was removed. The categorical features were one-hot encoded into 84 indicator columns, and a log(1 + x) transformation was applied to ten highly skewed count and byte features, reducing the skewness of src_bytes from 190.66 to 0.31. The data were then divided by stratified sampling into training (70%), validation (10%) and test (20%) partitions.

International Journal for Rapid Research in Engineering Technology & Applied Science (IJRRETAS)

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume 12 issue 07 July 2026 E-ISSN: 2455-4723 Impact Factor 7.95

To prevent information leakage, all subsequent data-dependent steps were fitted on the training partition only.

Feature selection proceeded in two stages. A variance threshold of 1×10^{-4} removed ten near-constant indicators, and mutual information between each feature and the class label was then estimated. Features with a mutual information of at least 0.01 were retained, giving 35 selected features. These were standardised with z-score scaling fitted on the training data.

The LightGBM classifier was configured with a binary objective, a learning rate of 0.05, a maximum of 63 leaves per tree, a minimum of 20 records per leaf, row and column subsampling of 0.8, and L1 and L2 regularisation of 0.1 and 1.0. Up to 1,000 boosting iterations were allowed, with early stopping when the validation log-loss had not improved for 50 iterations. Early stopping selected 259 iterations. Three comparative models were trained on identical features and partitions: Logistic Regression (L2 penalty, C = 1.0), Random Forest (300 trees) and a Support Vector Machine with an RBF kernel (C = 10), trained on a stratified subsample of 25,000 records because of its quadratic training cost.

Performance was measured on the test partition using accuracy, precision, recall, F1-score, specificity, the Matthews correlation coefficient (Chicco & Jurman, 2020), ROC-AUC, precision-recall AUC and confusion matrix analysis. Feature importance was analysed using gain, split frequency and SHAP values (Lundberg et al., 2020). Robustness was assessed through alternative train-test splits (60:40 to 90:10), ten random seeds, controlled class imbalance (outlier shares from 40% to 1%, with and without class weighting), five-fold cross-validation and evaluation on KDDTest+. Training time and per-record prediction latency were recorded to assess efficiency. All experiments were implemented in Python using scikit-learn and LightGBM with a fixed random seed of 42.

Precision and recall were given particular weight because they correspond directly to the two operational costs of detection, namely false alarms and missed attacks. Precision-recall analysis was included alongside ROC analysis because it focuses on the outlier class and is not inflated by the large number of correctly classified normal connections. The decision threshold was also varied between 0.1 and 0.9 to examine how the balance between precision and recall changes with the operating point.

4. RESULTS AND DISCUSSION

Early stopping halted training at 259 iterations. Table 1 shows that the LightGBM model generalised well: the gap between training accuracy (99.994%) and test accuracy (99.933%) was only 0.061 percentage points, and the test log-loss remained very low at 0.00276. The test performance was marginally higher than the validation performance, which indicates that the stopping point was not overfitted to the validation data.

Table 1. Training, validation and testing performance of LightGBM

Metric	Training	Validation	Testing
Records	88,174	12,597	25,193
Accuracy (%)	99.994	99.897	99.933

International Journal for Rapid Research in Engineering Technology & Applied Science (IJRRETAS)

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume 12 issue 07 July 2026 OPEN ACCESS E-ISSN: 2455-4723 Impact Factor 7.95

Metric	Training	Validation	Testing
Precision (%)	99.995	99.898	99.957
Recall (%)	99.993	99.881	99.898
F1-score (%)	99.994	99.889	99.927
MCC	0.99989	0.99793	0.99864
ROC-AUC	1.00000	0.99999	0.99999
Log-loss	0.00069	0.00268	0.00276

Figure 1 and Table 2 present the confusion matrix. Of 13,469 normal connections, only 5 were flagged incorrectly (a false positive rate of 0.037%), and of 11,724 attacks only 12 were missed. Detection was near-perfect for DoS (99.96%) and Probe (99.87%) attacks and remained high for R2L attacks (98.02%). U2R attacks were hardest to detect (88.89%), reflecting their scarcity and their resemblance to legitimate sessions, although this figure is based on only 9 test records.

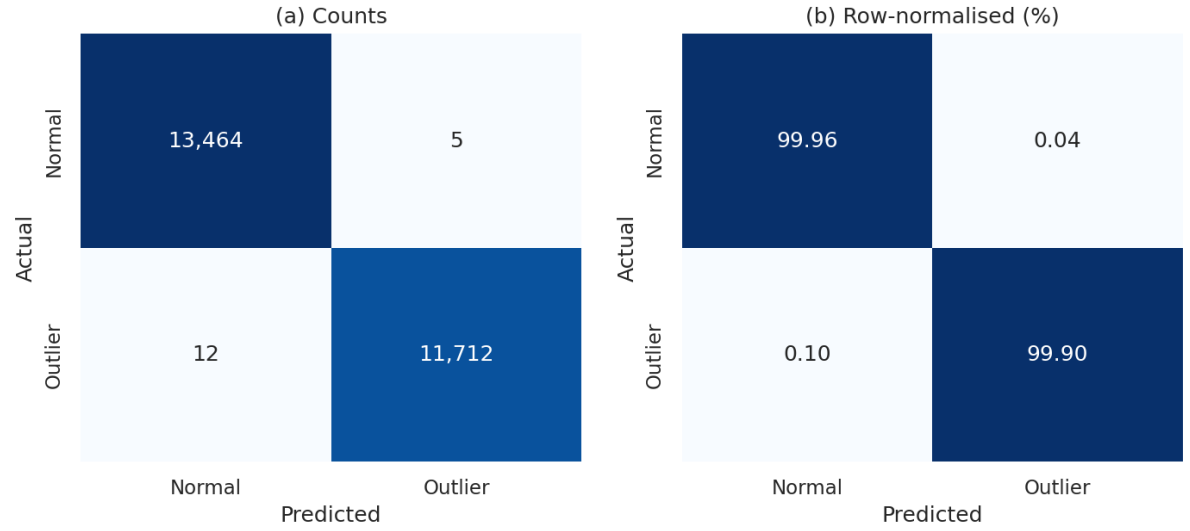

Figure 1. Confusion matrix of LightGBM on the test set: (a) counts and (b) row-normalised percentages

Table 2. Detection rate of LightGBM by traffic category (test set)

Category	Test records	Correct	Misclassified	Detection rate (%)
Normal	13,469	13,464	5	99.96
DoS	9,148	9,144	4	99.96
Probe	2,365	2,362	3	99.87
R2L	202	198	4	98.02
U2R	9	8	1	88.89
Overall	**25,193**	**25,176**	**17**	**99.93**

International Journal for Rapid Research in Engineering Technology & Applied Science (IJRRETAS)

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume 12 issue 07 July 2026 OPEN ACCESS E-ISSN: 2455-4723 Impact Factor 7.95

Figure 2 shows the ROC and precision-recall curves. The ROC-AUC and the average precision both reached 0.99999, far above the no-skill precision baseline of 0.465. This shows that the model ranked almost every outlier above almost every normal connection, independently of the decision threshold. The F1-score remained above 99.8% for all thresholds between 0.1 and 0.9, so operators can adjust the threshold to their tolerance for false alarms without a material loss of performance.

These results can be explained by the structure of the data. Exploratory analysis showed that the median outlier connection transferred zero bytes, whereas a minority of attacks transferred very large volumes, with normal traffic lying between these extremes. A decision tree represents this non-monotonic pattern naturally with two thresholds, and boosting then concentrates later trees on the records that remain misclassified, which are predominantly the subtler R2L and U2R connections.

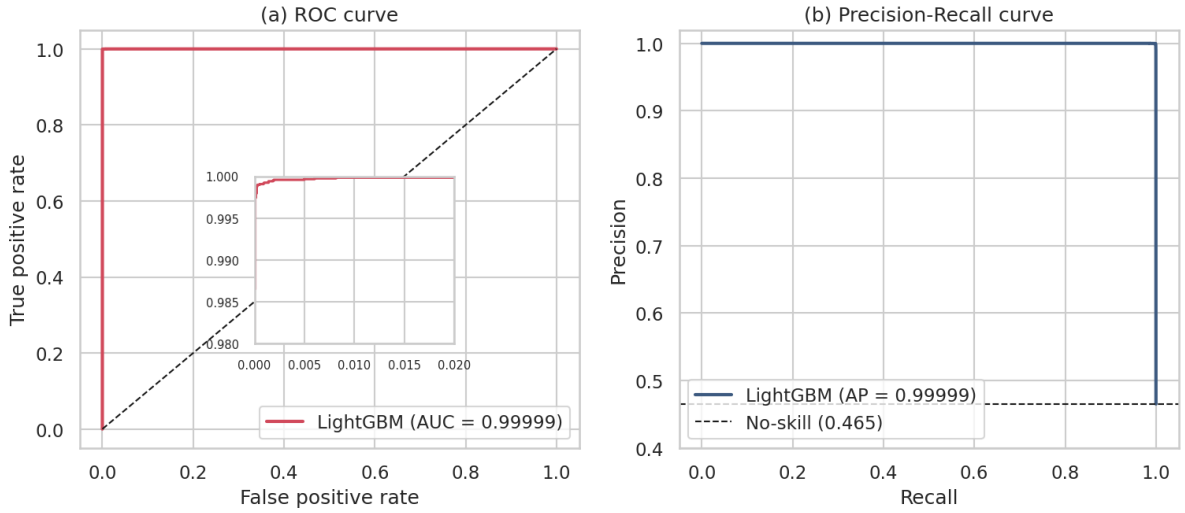

Figure 2. ROC curve (a) and precision-recall curve (b) of LightGBM on the test set

Feature importance results are shown in Figure 3 and Table 3. Byte-volume features dominated the gain: src_bytes contributed 53.85% and dst_bytes 15.54%, and the top ten features accounted for 92.15% of the total gain. SHAP analysis showed that zero-byte connections, high connection counts and SYN errors pushed predictions towards the outlier class, whereas repeated use of a single service such as http pushed them towards normal. These patterns correspond to the known behaviour of flooding and scanning attacks. A model using only the ten most important features still achieved 99.80% accuracy.

International Journal for Rapid Research in Engineering Technology & Applied Science (IJRRETAS)

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume 12 issue 07 July 2026 E-ISSN: 2455-4723 Impact Factor 7.95

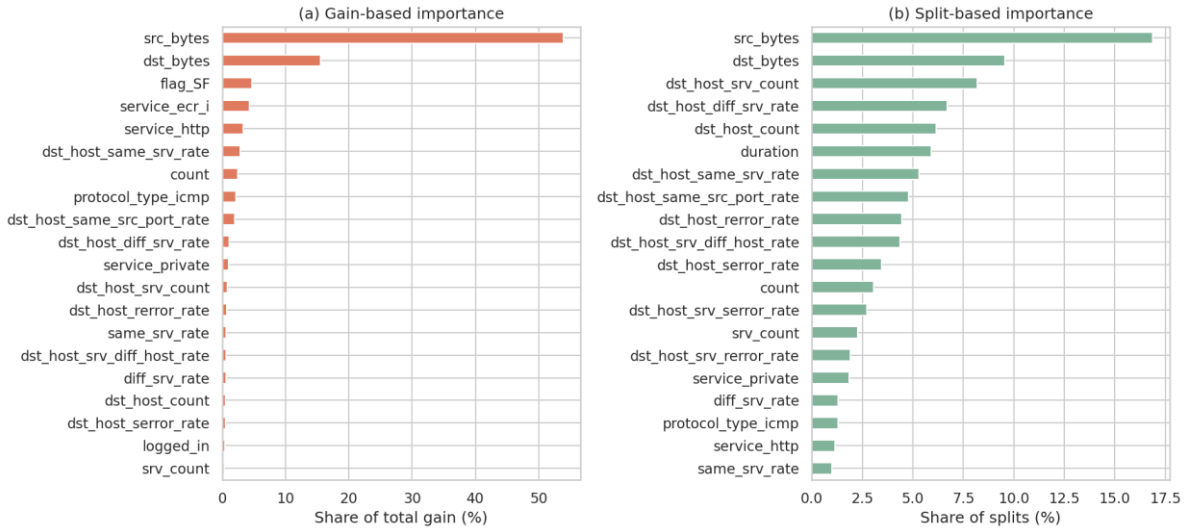

Figure 3. LightGBM feature importance by (a) information gain and (b) split frequency

Table 3. Top ten features of the LightGBM model

Feature	Gain (%)	Split (%)	Mean |SHAP|	SHAP rank
src_bytes	53.85	16.87	3.777	1
dst_bytes	15.54	9.54	1.163	2
flag_SF	4.72	0.32	0.236	15
service_ecr_i	4.27	0.79	0.186	19
service_http	3.35	1.15	0.372	7
dst_host_same_srv_rate	2.80	5.31	0.447	5
count	2.43	3.06	0.869	4
protocol_type_icmp	2.12	1.29	0.292	11
dst_host_same_src_port_rate	1.97	4.78	0.278	12
dst_host_diff_srv_rate	1.09	6.69	0.229	16

The agreement between these importance measures and independent statistical evidence strengthens confidence in the model. Mutual information, computed before training, ranked src_bytes and dst_bytes first, and Mann-Whitney U tests confirmed that all key rate and volume features differed significantly between normal and outlier traffic ($p < 0.001$). The convergence of model-based and statistical evidence indicates that the classifier has learned genuine behavioural signatures of attacks rather than artefacts of the dataset, which is an important consideration given the dataset quality concerns raised by Engelen et al. (2021). It also means that monitoring efforts can be focused on a compact set of connection-level statistics that do not require payload inspection.

International Journal for Rapid Research in Engineering Technology & Applied Science (IJRRETAS)

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume 12 issue 07 July 2026 OPEN ACCESS E-ISSN: 2455-4723 Impact Factor 7.95

Table 4 and Figure 4 compare LightGBM with the three baseline models. LightGBM achieved the best value on every metric. Random Forest was the closest competitor, matching LightGBM's precision exactly but missing 22 attacks rather than 12, so LightGBM made 37% fewer errors overall. The SVM made 103 errors and Logistic Regression 784, since a linear model cannot represent the non-monotonic relationship between byte volume and attack status. These results agree with evidence that tree ensembles are particularly effective on tabular data (Grinsztajn et al., 2022; Kilincer et al., 2021).

Table 4. Comparative performance of the four models (test set)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	MCC	ROC-AUC	Errors
Logistic Regression	96.888	97.524	95.744	96.626	0.93752	0.99579	784
Random Forest	99.893	99.957	99.812	99.885	0.99785	0.99999	27
SVM (RBF)	99.591	99.599	99.522	99.561	0.99178	0.99943	103
LightGBM	**99.933**	**99.957**	**99.898**	**99.927**	**0.99864**	**0.99999**	**17**

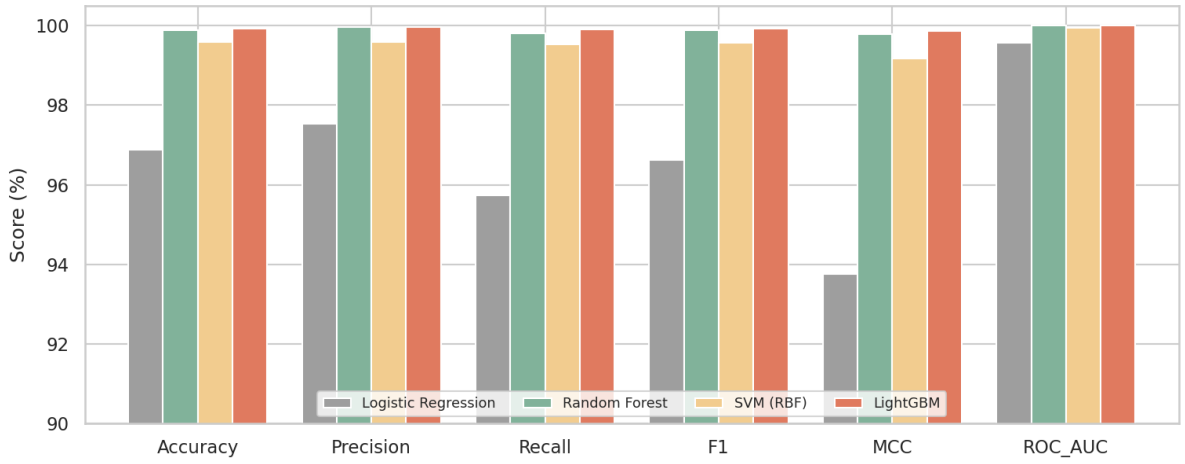

Figure 4. Classification metrics of the four models on the test set

Efficiency results are given in Table 5. LightGBM trained in 2.91 seconds, about four times faster than Random Forest, and faster than the SVM even though the SVM used less than a third as much training data. Its per-record latency of 11.45 microseconds was the lowest among the three high-accuracy models, which corresponds to more than 87,000 connections per second on a single machine. This confirms the efficiency advantage reported in earlier LightGBM studies (Jin et al., 2020; Seth et al., 2021).

Table 5. Computational efficiency of the four models

Model	Training records	Training time (s)	Latency per record (µs)
Logistic Regression	88,174	0.28	0.09
Random Forest	88,174	11.52	14.60

International Journal for Rapid Research in Engineering Technology & Applied Science (IJRRETAS)

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume 12 issue 07 July 2026 OPEN ACCESS E-ISSN: 2455-4723 Impact Factor 7.95

Model	Training records	Training time (s)	Latency per record (µs)
SVM (RBF)	25,000	4.89	29.11
LightGBM	**88,174**	**2.91**	**11.45**

Robustness results are summarised in Table 6 and Figure 5. Accuracy ranged only from 99.859% to 99.944% across split ratios from 60:40 to 90:10, and the standard deviation across ten random seeds was 0.018 percentage points. As the outlier share fell from 40% to 1%, precision stayed high and the F1-score declined gradually from 99.89% to 97.74%. Class weighting improved recall at the 5% and 2% levels. Five-fold cross-validation gave a mean accuracy of 99.886% (standard deviation 0.019). On the independent KDDTest+ set, which contains 17 unseen attack types, accuracy fell to 78.54%, while precision remained 96.73%. LightGBM achieved the highest ROC-AUC of all models on this set (0.9673), but the SVM achieved higher accuracy (81.68%) at the default threshold. The drop reflects the inherent difficulty supervised detectors face with previously unseen attacks, which is consistent with concerns in the evaluation literature (Arp et al., 2022).

Taken together, these findings indicate that the framework is highly robust within the distribution of its training data but, like all supervised detectors, is limited when confronted with attack types it has never seen. LightGBM's strong ranking performance on KDDTest+ suggests that part of the lost recall could be recovered by recalibrating the decision threshold for the new distribution or by periodically retraining with newly labelled attacks, which its short training time makes inexpensive. A layered architecture that pairs the supervised model with an unsupervised detector of unusual behaviour would further reduce this risk.

Table 6. Robustness of LightGBM under varied conditions

Condition	Setting	Accuracy (%)	F1-score (%)
Train-test split	60:40	99.859	99.849
Train-test split	90:10	99.944	99.940
Random seeds (10 runs)	80:20, mean ± SD	99.894 ± 0.018	99.886 ± 0.020
Class imbalance	5% outliers (weighted)	-	99.65
Class imbalance	1% outliers (standard)	-	97.74
Cross-validation	5-fold, mean ± SD	99.886 ± 0.019	99.877 ± 0.020
Independent test	KDDTest+ (unseen attacks)	78.54	77.39

International Journal for Rapid Research in Engineering Technology & Applied Science (IJRRETAS)

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume 12 issue 07 July 2026 E-ISSN: 2455-4723 Impact Factor 7.95

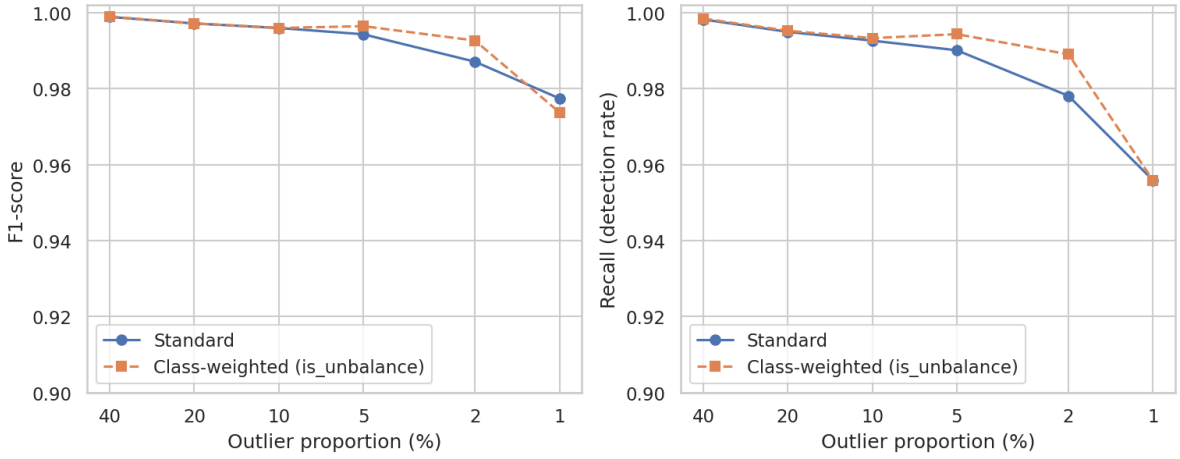

Figure 5. F1-score (a) and recall (b) of LightGBM as the outlier proportion decreases

5. CONCLUSION

This paper developed and evaluated a robust LightGBM-based framework for network outlier detection. By combining leakage-free pre-processing, mutual-information-based feature selection and a regularised LightGBM model with early stopping, the framework achieved 99.933% accuracy with only 5 false alarms and 12 missed attacks among 25,193 test connections. It outperformed Logistic Regression, an SVM and Random Forest on every metric while training about four times faster than Random Forest. SHAP analysis showed that its decisions rest on interpretable traffic behaviours, namely little or no data transfer, bursts of connections, SYN errors and scattering across services. The framework was stable across splits, seeds and folds and resilient to severe class imbalance. Its main limitation is reduced recall on attack types absent from training, as shown on KDDTest+. Future work should validate the framework on modern datasets, recalibrate thresholds under distribution shift, combine it with unsupervised detectors in a layered architecture, and extend it to multi-class detection. Overall, LightGBM provides an accurate, efficient and interpretable foundation for practical network anomaly detection.

The findings have practical implications for network defenders. The very low false alarm rate and microsecond-level latency make real-time deployment feasible, while SHAP explanations allow analysts to see why each connection was flagged, which supports faster triage and greater trust. The small number of features needed for high accuracy also allows lightweight models to be deployed on resource-constrained devices such as edge gateways. Researchers, for their part, should evaluate detectors on independent data containing unseen attacks and report computational cost alongside accuracy, so that published results better reflect operational conditions.

REFERENCES

1. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150.

2. Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., & Rieck, K. (2022). Dos and don'ts of machine learning in computer security. In Proceedings of the 31st USENIX Security Symposium (pp. 3971–3988). USENIX Association.
3. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21, Article 6.
4. Engelen, G., Rimmer, V., & Joosen, W. (2021). Troubleshooting an intrusion detection dataset: The CICIDS2017 case study. In 2021 IEEE Security and Privacy Workshops (SPW) (pp. 7–12). IEEE.
5. Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In Advances in Neural Information Processing Systems 35 (pp. 507–520).
6. Jin, D., Lu, Y., Qin, J., Cheng, Z., & Mao, Z. (2020). SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism. Computers & Security, 97, 101984. https://doi.org/10.1016/j.cose.2020.101984
7. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems 30 (pp. 3146–3154).
8. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2, Article 20.
9. Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks, 188, 107840.
10. Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences, 9(20), 4396.
11. Liu, J., Gao, Y., & Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. Computers & Security, 106, 102289. https://doi.org/10.1016/j.cose.2021.102289
12. Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2(1), 56–67.
13. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems 30 (pp. 4765–4774).
14. Resende, P. A. A., & Drummond, A. C. (2018). A survey of random forest based methods for intrusion detection systems. ACM Computing Surveys, 51(3), Article 48.
15. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147–167.
16. Seth, S., Singh, G., & Kaur Chahal, K. (2021). A novel time efficient learning-based approach for smart intrusion detection system. Journal of Big Data, 8, Article 111. https://doi.org/10.1186/s40537-021-00498-8

17. Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP) (pp. 108–116). SciTePress.
18. Talukder, M. A., Islam, M. M., Uddin, M. A., Hasan, K. F., Sharmin, S., Alyami, S. A., & Moni, M. A. (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. Journal of Big Data, 11, Article 33. https://doi.org/10.1186/s40537-024-00886-w
19. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525–41550.
20. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, 21954–21961.