International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

Hybrid Machine Learning Approaches for Efficient Network Outlier Detection: A Review of LightGBM-Based Frameworks

Asif Ali Syed
Research Scholar, Department of Computer Science, Malwanchal University, Indore
Dr. Manav Thakur
Supervisor, Department of Computer Science, Malwanchal University, Indore

Abstract
The rapid growth of network traffic volume, connected devices and attack diversity has made outlier detection one of the most demanding problems in modern cybersecurity. Signature-based systems are unable to recognise novel or mutated attacks, while many deep learning detectors deliver high accuracy at a computational cost that limits their use in high-speed or resource-constrained environments. Light Gradient Boosting Machine (LightGBM), a histogram-based gradient boosting decision tree framework, has emerged as a strong candidate for efficient network outlier detection because it combines fast training, low memory consumption and competitive accuracy on tabular flow data. It first traces the evolution of network anomaly detection and the development of gradient boosting algorithms, and then analyses benchmark datasets, classical and deep learning detectors, LightGBM-based hybrid frameworks, and the cross-cutting issues of class imbalance, interpretability and deployment. Three comparative tables summarise datasets, algorithm families and representative LightGBM-centred frameworks. The review finds that hybridisation, through feature selection, resampling, parallel processing, ensemble stacking and explainability layers, consistently improves the efficiency and reliability of LightGBM detectors. It also identifies persistent gaps in dataset quality, cross-dataset generalisation, concept drift handling and adversarial robustness, and proposes directions for future research.

Keywords: network outlier detection, intrusion detection, LightGBM, gradient boosting, hybrid machine learning, class imbalance, explainable AI

1. Introduction

1.1 Evolution of Network Outlier Detection

Network outlier detection refers to the identification of traffic records, flows or sessions that deviate significantly from the expected behaviour of a network. Such deviations may indicate intrusions, denial-of-service campaigns, botnet communication, data exfiltration, misconfiguration or hardware faults. Early intrusion detection systems relied on signature matching, in which incoming traffic was compared against a database of known attack patterns.

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

Although signature-based systems produce few false alarms for well-documented threats, they cannot recognise zero-day attacks and require constant manual updating (Khraisat et al., 2019). Anomaly-based detection emerged as an alternative that models normal behaviour and flags departures from it, offering the possibility of detecting previously unseen attacks at the cost of higher false positive rates (Ahmed et al., 2016).

Over the past decade, machine learning has become the dominant paradigm for anomaly-based detection. Buczak and Guven (2016) surveyed data mining and machine learning methods for cyber intrusion detection and showed that decision trees, support vector machines, clustering, Bayesian networks and ensemble methods had all been applied with varying success. Their review also highlighted that the field suffered from a dependence on outdated datasets and from limited attention to computational complexity, two concerns that remain relevant today. Khraisat et al. (2019) later classified intrusion detection techniques and datasets in detail, noting that more complex models often achieved higher accuracy on large datasets but demanded considerably more training time and hardware resources.

The shift towards high-speed networks, cloud infrastructure and the Internet of Things has intensified the need for detectors that are not only accurate but also fast and lightweight. Traffic must often be analysed in near real time, and detection models may need to run on edge devices with limited memory and processing capacity. These operational constraints have renewed interest in algorithms that offer a favourable balance between accuracy and efficiency, and gradient boosting decision trees have emerged as one of the most promising families in this respect.

1.2 Gradient Boosting and the Emergence of LightGBM

Gradient boosting builds an additive ensemble of weak learners, typically shallow decision trees, by fitting each new tree to the residual errors of the existing ensemble. The approach is particularly well suited to tabular data, which is the natural representation of network flow records. Chen and Guestrin (2016) introduced XGBoost, a scalable tree boosting system with regularised objectives, sparsity-aware split finding and cache-aware computation, which quickly became a standard tool in applied machine learning. Later libraries such as CatBoost introduced ordered boosting and native handling of categorical features, broadening the family of practical gradient boosting tools (Bentéjac et al., 2021).

LightGBM, proposed by Ke et al. (2017), introduced two techniques that substantially reduce the cost of training gradient boosting models on large datasets. Gradient-based One-Side Sampling (GOSS) retains instances with large gradients and randomly samples those with small gradients, so that the information gain can be estimated accurately from a reduced dataset. Exclusive Feature Bundling (EFB) merges mutually exclusive sparse features into single bundles, reducing the effective dimensionality of the data. Combined with histogram-based split finding and leaf-

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

wise tree growth, these techniques allow LightGBM to train many times faster than earlier implementations while using less memory and maintaining comparable accuracy (Ke et al., 2017).

Comparative studies have confirmed that the three major gradient boosting libraries offer different trade-offs. Bentéjac et al. (2021) compared gradient boosting algorithms across a range of datasets and found that LightGBM was generally the fastest to train, while differences in generalisation accuracy were modest once hyperparameters were tuned. Grinsztajn et al. (2022) further demonstrated that tree-based models continue to outperform deep learning architectures on typical medium-sized tabular data, largely because trees are robust to uninformative features and can represent irregular decision functions. These findings provide a theoretical and empirical basis for the use of LightGBM in network outlier detection.

1.3 Rationale for Hybrid Learning Frameworks

Despite its strengths, a standalone LightGBM classifier does not resolve all of the difficulties inherent in network outlier detection. Network datasets are typically highly imbalanced, with rare attack categories represented by only a small number of samples. They often contain redundant, noisy or highly correlated features. Traffic characteristics change over time as user behaviour and attack strategies evolve, and security analysts require explanations of why an alert was raised before acting upon it (Apruzzese et al., 2023). Each of these problems can degrade the performance of any single learning algorithm.

Hybrid frameworks address these problems by combining LightGBM with complementary techniques at different stages of the detection pipeline. Data-level hybrids integrate resampling methods such as SMOTE or ADASYN to rebalance the training data (Fernández et al., 2018). Feature-level hybrids apply filter, wrapper or embedded selection methods to reduce dimensionality and training time. Model-level hybrids combine LightGBM with other learners through stacking, voting or cascading arrangements, or pair it with deep feature extractors such as autoencoders. System-level hybrids embed LightGBM within parallel or streaming architectures to meet throughput requirements (Jin et al., 2020). Post-hoc hybrids attach explanation methods such as SHAP to make the model's decisions interpretable (Lundberg & Lee, 2017).

The common premise of these approaches is that LightGBM provides an efficient and accurate core classifier, while the surrounding components compensate for its weaknesses or extend its capabilities. Understanding how these components interact, and which combinations yield the greatest benefit, is therefore central to the design of practical detection systems.

1.4 Scope and Organisation of the Review

This review concentrates on peer-reviewed and widely cited work published between 2015 and 2024 that is relevant to LightGBM-based network outlier detection. Three groups of literature are

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

covered: foundational work on gradient boosting and explainability; surveys and empirical studies of network intrusion and anomaly detection, including datasets and deep learning methods; and studies that propose or evaluate LightGBM-centred hybrid frameworks. The terms outlier detection, anomaly detection and intrusion detection are used in a closely related sense throughout, reflecting their overlapping use in the literature.

The remainder of the paper is organised as follows. Section 2 presents the literature review in four parts: benchmark datasets and evaluation practices, classical and deep learning approaches, LightGBM-based and hybrid frameworks, and cross-cutting considerations of class imbalance, interpretability and deployment. Three tables summarise the main datasets, algorithm families and representative frameworks. Section 3 draws conclusions from the reviewed literature, and Section 4 outlines directions for future work.

2. Literature Review

2.1 Benchmark Datasets and Evaluation Practices

The quality of a network outlier detector cannot be separated from the quality of the data used to train and evaluate it. For many years, research relied on the KDD Cup 1999 dataset and its refined successor NSL-KDD. Although these datasets enabled early comparisons, they reflect network conditions from the late 1990s and contain well-documented redundancies and artefacts. Ring et al. (2019) conducted a comprehensive survey of network-based intrusion detection datasets and argued that researchers should select datasets according to explicit properties such as recency, labelling quality, traffic diversity and availability of raw packet data, rather than defaulting to the most frequently cited benchmarks.

Two more recent datasets have become central to LightGBM-based research. Moustafa and Slay (2015) introduced UNSW-NB15, generated with the IXIA PerfectStorm tool to combine realistic modern normal activity with synthetic contemporary attacks across nine categories, including fuzzers, backdoors, exploits and reconnaissance. Sharafaldin et al. (2018) presented CICIDS2017, which was built from five days of profiled user behaviour and includes brute force, denial-of-service, web, infiltration, botnet and distributed denial-of-service attacks, with more than 80 flow features extracted by CICFlowMeter. Both datasets offer far greater diversity than KDD-derived benchmarks and are large enough to expose differences in training efficiency between algorithms.

However, newer datasets are not free from problems. Ring et al. (2019) warned that labelling errors, duplicated records and unrealistic class distributions can inflate reported performance. Similar methodological concerns have been raised by Apruzzese et al. (2023), who noted that many machine learning security studies use random train-test splits that leak temporal information and that results obtained in the laboratory frequently fail to transfer to operational networks. In addition, flow-based datasets typically contain many redundant or weakly

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

informative features, which strengthens the case for including a feature selection stage in hybrid pipelines (Seth et al., 2021).

Evaluation practices vary considerably across studies. Accuracy remains the most commonly reported metric, but on imbalanced datasets it can be misleading because a classifier that labels all traffic as benign may still achieve high accuracy. More informative metrics include precision, recall (detection rate), F1-score, false alarm rate, area under the ROC curve and per-class metrics for minority attack categories. Because efficiency is a core motivation for using LightGBM, training time, inference latency and memory consumption should also be reported, yet many studies omit these measures or report them on incomparable hardware. Table 1 summarises the datasets most frequently used in the reviewed literature.

Table 1. Benchmark datasets commonly used in LightGBM-based network outlier detection

Dataset	Source	Main characteristics	Known limitations
NSL-KDD	Refinement of KDD Cup 1999	41 features; four attack groups (DoS, Probe, R2L, U2R); duplicates removed from original KDD99	Outdated traffic; synthetic origin; limited relevance to modern networks (Ring et al., 2019)
UNSW-NB15	Moustafa and Slay (2015)	49 features; nine modern attack categories; hybrid real and synthetic traffic	Several low-information features; imbalance in rare categories
CICIDS2017	Sharafaldin et al. (2018)	Over 80 flow features; five days of profiled traffic; DoS, DDoS, web, botnet, infiltration and brute force attacks	Severe class imbalance; labelling and duplicate issues reported (Ring et al., 2019)
KDD Cup 1999	DARPA 1998 evaluation	Large volume; historical benchmark for early IDS research	Heavy redundancy; traffic artefacts; not recommended for modern evaluation (Khraisat et al., 2019)
Mirai/IoT traffic (Kitsune)	Mirsky et al. (2018)	Real IoT and surveillance network captures with botnet and scanning attacks	Narrow network setting; mainly suited to unsupervised evaluation

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

2.2 Classical and Deep Learning Approaches to Network Anomaly Detection

Before gradient boosting became widespread, network anomaly detection drew on a wide range of statistical and machine learning methods. Ahmed et al. (2016) organised network anomaly detection techniques into classification-based, statistical, information-theoretic and clustering-based families. Classification-based techniques such as support vector machines and decision trees performed well when labelled data were available, while clustering-based techniques such as k-means could operate without labels but struggled with high false alarm rates. Statistical techniques were computationally light but depended on assumptions about traffic distributions that rarely held in practice.

Ensemble methods, particularly random forests, became a strong baseline because they reduce variance, handle mixed feature types and offer feature importance estimates. Buczak and Guven (2016) observed that ensemble and tree-based methods were among the most consistently effective approaches in the surveyed literature, although few studies examined their scalability. Khraisat et al. (2019) similarly concluded that hybrid and ensemble detectors generally outperformed single classifiers, and recommended that future systems combine anomaly and signature components to balance detection of novel attacks with low false alarm rates.

Deep learning has attracted considerable attention since around 2017. Shone et al. (2018) proposed a non-symmetric deep autoencoder for unsupervised feature learning, combined with a random forest classifier, and showed that the hybrid reduced training time compared with conventional deep belief networks on KDD-derived data. Vinayakumar et al. (2019) evaluated deep neural networks across multiple datasets, including NSL-KDD, UNSW-NB15 and CICIDS2017, and proposed a scalable hybrid framework for monitoring network and host-level activity. Their results indicated that deep models were effective but that performance varied substantially between datasets. Mirsky et al. (2018) introduced Kitsune, an ensemble of lightweight autoencoders designed for online unsupervised intrusion detection, demonstrating that carefully engineered neural ensembles could run on modest hardware.

Broader surveys of deep anomaly detection have clarified both the promise and the limits of these methods. Pang et al. (2021) reviewed deep learning for anomaly detection across domains and identified challenges such as low anomaly recall, high-dimensional and non-independent data, scarcity of labelled anomalies and lack of interpretability. Vinayakumar et al. (2019) likewise reported that deep architectures achieved strong detection rates but required careful tuning of network depth and hyperparameters, with substantial computation. At the same time, Grinsztajn et al. (2022) showed that tree-based models remain highly competitive on tabular data, which suggests that the additional cost of deep architectures is not always justified for flow-based detection. Table 2 compares the main algorithm families discussed in this section.

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

Table 2. Comparison of algorithm families for network outlier detection

Algorithm family	Strengths	Weaknesses	Representative studies
Statistical and clustering methods	Low computational cost; can operate without labels	Distributional assumptions; high false alarm rates	Ahmed et al. (2016)
Classical classifiers (SVM, decision tree, k-NN)	Simple to implement; effective on labelled data	Poor scalability (SVM, k-NN); sensitive to imbalance	Buczak and Guven (2016); Khraisat et al. (2019)
Bagging ensembles (random forest)	Robust; low variance; feature importance available	Large model size; slower inference with many trees	Buczak and Guven (2016); Shone et al. (2018)
Gradient boosting (XGBoost, CatBoost, LightGBM)	High accuracy on tabular data; handles missing values; strong efficiency (especially LightGBM)	Hyperparameter sensitivity; risk of overfitting minority classes	Chen and Guestrin (2016); Ke et al. (2017); Bentéjac et al. (2021)
Deep learning (DNN, CNN, RNN, autoencoder)	Automatic feature learning; strong on large and sequential data	High training cost; limited interpretability; dataset-dependent results	Shone et al. (2018); Vinayakumar et al. (2019); Pang et al. (2021)

2.3 LightGBM-Based and Hybrid Detection Frameworks

A growing body of work places LightGBM at the core of network detection pipelines and augments it with complementary techniques. One of the most explicit efficiency-oriented designs is SwiftIDS, proposed by Jin et al. (2020). SwiftIDS adopts LightGBM as the detection algorithm to cope with massive traffic volumes and introduces a parallel intrusion detection mechanism that analyses traffic arriving in different time windows concurrently. The authors evaluated the framework on KDD99, NSL-KDD and CICIDS2017 and supplemented offline experiments with a near real-time test, showing that the combination of a fast learner and a parallel architecture could keep pace with high-speed traffic while maintaining satisfactory

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

detection performance. The study is notable because it treats timeliness as a primary design objective rather than an afterthought.

Class imbalance is the focus of the framework developed by Liu et al. (2021), who combined adaptive synthetic oversampling (ADASYN) with LightGBM. ADASYN generates more synthetic samples for minority instances that are harder to learn, shifting the decision boundary towards difficult regions of the feature space. In their experiments, the ADASYN-LightGBM hybrid improved the detection of minority attack classes and outperformed several alternative classifiers while retaining LightGBM's speed advantage. This study is frequently cited in subsequent work as evidence that data-level hybridisation is an effective complement to gradient boosting.

Feature selection constitutes a second major line of hybridisation. Seth et al. (2021) proposed a time-efficient intrusion detection approach that couples LightGBM with a hybrid feature selection strategy, reducing the number of input features before classification. The authors explicitly contrasted their design with SwiftIDS, arguing that reliance on feature reduction rather than parallel detection makes the model's performance independent of network speed. Their results on CICIDS2018 data indicated that eliminating redundant features shortened prediction latency without sacrificing detection quality. Related work on UNSW-NB15 has similarly found that many of the dataset's features contribute little to classification, reinforcing the value of a selection stage before boosting.

Model-level hybrids combine LightGBM with other learners. The autoencoder-plus-random-forest design of Shone et al. (2018) and the deep hybrid framework of Vinayakumar et al. (2019) illustrate a general pattern in which a representation learner extracts compact features that are then passed to a tree-based classifier. Substituting LightGBM for the downstream classifier is a natural extension, since it typically trains faster than random forests on large datasets (Ke et al., 2017; Bentéjac et al., 2021). Stacking and voting ensembles that include LightGBM alongside XGBoost and CatBoost exploit differences in how each library constructs trees and handles categorical variables (Chen & Guestrin, 2016; Bentéjac et al., 2021), although such ensembles partially offset the efficiency gains of using LightGBM alone.

A further category concerns explainable hybrids. Because LightGBM is a tree ensemble, it is compatible with exact and efficient tree-based SHAP explanations derived from the unified framework of Lundberg and Lee (2017). Attaching SHAP to a LightGBM detector allows analysts to see which flow features drove each alert, to validate that the model relies on meaningful indicators rather than dataset artefacts, and to prune uninformative features. This combination addresses one of the main barriers to operational adoption identified by Apruzzese et al. (2023). Table 3 summarises representative LightGBM-centred and closely related hybrid frameworks.

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

Table 3. Representative LightGBM-centred and related hybrid frameworks

Study	Hybrid strategy	Datasets	Main contribution
Jin et al. (2020)	LightGBM with parallel detection over time windows	KDD99, NSL-KDD, CICIDS2017	Real-time oriented IDS for high-speed networks; near real-time validation
Liu et al. (2021)	ADASYN oversampling with LightGBM	Public intrusion detection benchmarks	Improved minority-class detection while preserving training speed
Seth et al. (2021)	Hybrid feature selection with LightGBM	CICIDS2018	Reduced prediction latency; performance independent of network speed
Shone et al. (2018)	Deep autoencoder features with random forest classifier	KDD Cup 1999, NSL-KDD	Template for deep-feature plus tree-classifier hybrids
Mirsky et al. (2018)	Ensemble of lightweight autoencoders (Kitsune)	Real IoT and surveillance network traffic	Online unsupervised detection on modest hardware
Lundberg and Lee (2017), applied to boosting detectors	SHAP explanations attached to tree ensembles	Domain-independent	Feature-level attribution for alerts; supports feature pruning and trust

2.4 Class Imbalance, Interpretability and Deployment Considerations

Across the reviewed literature, three cross-cutting issues determine whether a LightGBM-based detector that performs well in experiments will remain effective in practice. The first is class imbalance. Benign traffic usually dominates network datasets, and certain attack types, such as infiltration or user-to-root exploits, may appear in only a handful of records. Fernández et al. (2018) reviewed fifteen years of SMOTE-based research and concluded that synthetic oversampling remains one of the most effective general tools for imbalanced learning, but cautioned that it can generate noisy samples in overlapping regions and must be combined with

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

cleaning or adaptive variants. LightGBM also provides algorithm-level options, including class weighting and focal-style custom losses, which can be combined with data-level resampling as demonstrated by Liu et al. (2021).

The second issue is interpretability. Security operations centres must triage alerts quickly, and an unexplained alert from a black-box model is difficult to act upon. Lundberg and Lee (2017) unified several attribution methods under the SHAP framework, which assigns each feature a contribution to an individual prediction with desirable theoretical properties. For gradient boosted trees, these values can be computed efficiently, making explanation affordable even at high traffic volumes. Interpretability also serves a scientific purpose: Apruzzese et al. (2023) warned that machine learning models in cybersecurity may learn spurious correlations present in benchmark datasets, and explanation methods offer one way of detecting such shortcuts before deployment.

The third issue concerns deployment conditions. Network traffic is non-stationary, and a model trained on historical data may degrade as protocols, applications and attack techniques change. Most studies reviewed here evaluate models on a single static dataset with random splits, which does not reveal how quickly performance decays over time (Ring et al., 2019; Apruzzese et al., 2023). Generalisation across datasets is similarly underexplored; Vinayakumar et al. (2019) observed considerable variation in performance across benchmarks, suggesting that models may be tuned to dataset-specific characteristics. Efficiency, the principal motivation for LightGBM, must also be assessed under realistic conditions. Jin et al. (2020) and Seth et al. (2021) are among the few studies that evaluate latency or real-time behaviour explicitly, and their contrasting designs show that efficiency can be gained either through system architecture or through dimensionality reduction.

Taken together, these considerations suggest that the value of a LightGBM-based detector depends as much on the surrounding pipeline and evaluation protocol as on the classifier itself. A model that attains near-perfect accuracy on a randomly split benchmark may still miss rare attacks, raise unexplained alerts or degrade rapidly once traffic patterns shift. Hybrid frameworks are therefore most convincing when they demonstrate improvements in minority-class recall, report efficiency measures alongside accuracy, and provide some form of explanation for their decisions (Khraisat et al., 2019; Fernández et al., 2018; Lundberg & Lee, 2017).

Finally, adversarial robustness is an emerging concern. Attackers may deliberately manipulate flow features to evade detection, and tree ensembles, like neural networks, can be vulnerable to such manipulation. Apruzzese et al. (2023) emphasised that practical machine learning security systems should be assessed against adaptive adversaries rather than only static test sets. Very few LightGBM-based detection studies in the reviewed period address this issue directly, which represents a significant gap given the adversarial nature of the security domain.

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 8.3 www.ijesh.com ISSN: 2250-3552

3. Conclusion

This review has examined hybrid machine learning approaches for efficient network outlier detection, with particular attention to frameworks built around LightGBM. The literature from 2015 to 2024 shows a clear progression from signature-based and shallow anomaly detection towards ensemble and deep learning methods, and more recently towards efficiency-aware designs that can cope with high-speed and resource-constrained environments. Within this progression, LightGBM occupies a distinctive position. Its gradient-based one-side sampling, exclusive feature bundling and histogram-based learning give it a substantial training-speed and memory advantage over earlier boosting implementations (Ke et al., 2017), while comparative studies confirm that it achieves accuracy comparable to XGBoost and CatBoost and remains competitive with deep learning on tabular data (Bentéjac et al., 2021; Grinsztajn et al., 2022).

The reviewed studies indicate that LightGBM is most effective when embedded within a hybrid pipeline. Parallel detection architectures improve throughput for real-time operation (Jin et al., 2020); adaptive oversampling improves recognition of rare attack classes (Liu et al., 2021); feature selection reduces latency and model complexity (Seth et al., 2021); deep feature extractors and ensembles offer richer representations (Shone et al., 2018; Vinayakumar et al., 2019); and SHAP explanations support analyst trust and model validation (Lundberg & Lee, 2017). Each hybrid component addresses a specific weakness, and the most robust frameworks combine several of them.

At the same time, the review has identified recurring limitations. Many studies depend on a small number of benchmark datasets whose labelling quality and realism have been questioned (Ring et al., 2019). Evaluation often emphasises accuracy over minority-class recall and rarely reports latency or memory under comparable conditions. Temporal drift, cross-dataset generalisation and adversarial robustness receive limited attention (Apruzzese et al., 2023). Consequently, although LightGBM-based hybrid frameworks represent one of the most practical routes to efficient network outlier detection, their reported performance should be interpreted with caution until these methodological gaps are addressed.

4. Future Work

Several directions emerge from the reviewed literature. First, evaluation methodology needs to be strengthened. Future studies should adopt time-aware train-test splits, report per-class precision, recall and F1-score, and measure training time, inference latency and memory consumption on clearly specified hardware. Cross-dataset experiments, in which a model trained on one benchmark is tested on another, would provide a more honest estimate of generalisation than single-dataset evaluation (Ring et al., 2019; Apruzzese et al., 2023).

Second, LightGBM frameworks should be extended to handle concept drift. Incremental or windowed retraining strategies, drift detectors that trigger model updates, and ensembles that

weight recent models more heavily could allow detectors to adapt to evolving traffic without full retraining. LightGBM's fast training makes it well suited to frequent updating, but systematic studies of this capability in network settings are still scarce.

Third, adversarial robustness requires dedicated investigation. Research should evaluate LightGBM detectors against feature-space and problem-space evasion attacks, and explore defences such as adversarial training, monotonic constraints on selected features and ensemble diversity. Explainability tools could be used both to anticipate which features an attacker is likely to manipulate and to monitor for suspicious shifts in attribution patterns (Lundberg & Lee, 2017).

Fourth, deployment on edge and Internet of Things devices deserves greater attention. Model compression, feature budget constraints and hardware-aware hyperparameter optimisation could make LightGBM detectors practical on gateways and embedded systems, building on the lightweight design philosophy demonstrated by Kitsune (Mirsky et al., 2018). Federated learning offers a complementary route by allowing multiple organisations to train shared detectors without exchanging raw traffic data.

Finally, semi-supervised and unsupervised hybrids merit further exploration. Combining LightGBM with autoencoders or other representation learners that can model normal behaviour without labels would reduce dependence on scarce attack annotations and improve sensitivity to zero-day threats (Pang et al., 2021). Integrating such components while preserving LightGBM's efficiency remains an open and promising research challenge.

References

1. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31. https://doi.org/10.1016/j.jnca.2015.11.016
2. Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A. V., & Di Franco, F. (2023). The role of machine learning in cybersecurity. Digital Threats: Research and Practice, 4(1), Article 8. https://doi.org/10.1145/3545574
3. Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54(3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5
4. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

5. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785
6. Fernández, A., García, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. Journal of Artificial Intelligence Research, 61, 863–905. https://doi.org/10.1613/jair.1.11192
7. 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 (Datasets and Benchmarks Track) (pp. 507–520).
8. 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, Article 101984. https://doi.org/10.1016/j.cose.2020.101984
9. 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).
10. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2(1), Article 20. https://doi.org/10.1186/s42400-019-0038-7
11. Liu, J., Gao, Y., & Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. Computers & Security, 106, Article 102289. https://doi.org/10.1016/j.cose.2021.102289
12. 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).
13. Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. In Proceedings of the Network and Distributed System Security Symposium (NDSS 2018). Internet Society. https://doi.org/10.14722/ndss.2018.23204
14. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942
15. Pang, G., Shen, C., Cao, L., & van den Hengel, A. (2021). Deep learning for anomaly detection: A review. ACM Computing Surveys, 54(2), Article 38. https://doi.org/10.1145/3439950

16. 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. https://doi.org/10.1016/j.cose.2019.06.005
17. 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
18. Sharafaldin, I., Lashkari, A. H., & 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. https://doi.org/10.5220/0006639801080116
19. Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41–50. https://doi.org/10.1109/TETCI.2017.2772792
20. 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. https://doi.org/10.1109/ACCESS.2019.2895334