Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

Cybersecurity Threat Detection Using Machine Learning: A Comparative Analysis of Gradient Boosting Approaches on Network Intrusion Data

Md Saad Bin Rizvi[1], Md Saqlain Mustaque[2], Md Arfakshad[3], Dr. Anum Kamal[4]

Computer Science and Engineering, Integral University, Lucknow, India

[1]saadrizvi106@gmail.com, [2]saqlain3177@gmail.com, [3]mdarfakshadbth@gmail.com, [5]anumkamal@iul.ac.in

---

Abstract – Due to emerging attacks, signature-based threat detection systems are no longer efficient for current attack strategies. In this research paper, we will propose an end-to-end machine learning solution based on two state-of-the-art gradient boosting methods (XGBoost & LightGBM), which can classify network connection into five categories of threat attacks. This study uses the dataset provided by the KDD Cup 1999 challenge, which consists of 494,021 network connection samples labeled with threats. To train an accurate model, we implement a strict data preprocessing procedure, which involves eliminating duplicates, performing one-hot encoding, class label aggregation, and applying min-max normalization. Experimentation reveals that our model achieves the maximum accuracy level of 99.2% using XGBoost and 99.0% using LightGBM, compared to the baseline models of Decision Tree (97.8%), Naive Bayes (88.4%), K-Nearest Neighbors (96.1%), and Random Forest (98.5%). It turns out that our model works great when identifying high-frequency attacks (Denial-of-Service, Probe) but does not perform well enough for detecting minority attack classes (R2L, U2R).

Index Terms: Cybersecurity, Threat Detection, Machine Learning, XGBoost, LightGBM, Intrusion Detection System, KDD Cup 1999, Multi-Class Classification, Gradient Boosting, Network Security, Anomaly Detection

I. INTRODUCTION

Network infrastructure is now essential to a wide range of industries, from healthcare and finance to education and national defense, due to the digital transformation of the world's economy. But this increased reliance on networked systems has also resulted in large attack surfaces, which adversaries are still using with ever-increasing sophistication. Cyberattacks that seriously jeopardize data integrity and business continuity, such as Denial-of-Service (DoS) floods, remote exploitation, privilege escalation, and network reconnaissance, result in annual economic losses of billions of dollars [1].

By keeping an eye on traffic patterns and warning security staff of questionable activities, Network Intrusion Detection Systems (NIDS) constitute an essential defensive layer. The fingerprints of known attacks are encoded in manually maintained signature repositories, which are essential to legacy NIDS infrastructures. These systems are dependable for threats that have already been cataloged, but they are essentially unable to identify zero-day attacks, which are incursions that take advantage of vulnerabilities that have not yet been discovered or use attack patterns that are sufficiently different from stored signatures to avoid rule-based matching. [2].

Machine learning (ML) is changing the way we approach cybersecurity. It allows us to create systems that can learn from examples of both normal and harmful online activity. Instead of relying on set rules, these ML-based detectors look for patterns in behavior and statistics, helping them recognize new types of attacks. Early attempts to use ML in intrusion detection, such as with Decision Trees, Support Vector Machines (SVM), and K-Nearest Neighbors, showed promise. However, they faced challenges with managing high-dimensional data, uneven class distribution, and would often struggle as the amount of data grew.

Nowadays, methods like XGBoost and LightGBM have become leaders in classifying structured data. They work by building a series of simple decision trees, with each new tree aimed at fixing the mistakes made by previous ones. This approach leads to a highly accurate system that performs well with the diverse and unbalanced nature of network traffic.

In this paper, we introduce a complete framework for detecting cybersecurity threats, which includes: (1) a thorough process for preparing the KDD Cup 1999 dataset by handling duplicates and ensuring proper formatting; (2) a careful comparison of XGBoost and LightGBM against four standard classifiers; (3) an analysis of performance across different traffic categories; and (4) a look at ongoing challenges and suggestions for future research.

II. RELATED WORK

A. Foundations of Intrusion Detection

The earliest theoretical framework for anomaly detection was

introduced by Denning [6], who argued that unusual statistical deviations from normal system behavior could serve as reliable indicators of intrusions. This principle became the basis for later data-driven approaches. Building on this idea, Lee and Stolfo [7] demonstrated that data mining applied to network audit records could automatically generate effective detection models. Their work directly influenced the creation of the KDD Cup 1999 dataset, which became a widely used benchmark in the field.

B. Machine Learning Approaches

As machine learning gained traction, researchers explored its potential for intrusion detection. Buczak and Guven [8] reviewed more than 40 algorithms, including Naive Bayes, neural networks, fuzzy logic, and genetic algorithms. They concluded that no single method consistently outperforms others across all attack categories, which encouraged the use of ensemble strategies. Revathi and Malathi [10] tested multiple classifiers on the NSL-KDD dataset and found that tree-based models generally performed better than kernel methods, especially with high-dimensional data. However, all models struggled with rare attack types such as R2L and U2R.

Tavallaee et al. [9] highlighted major shortcomings in the original KDD Cup dataset, including excessive duplicate records and severe class imbalance. To address these issues, they introduced NSL-KDD and recommended preprocessing practices that have since become standard. Later, Moustafa and Slay [11] developed UNSW-NB15 to reflect modern attack scenarios absent from KDD. Despite these newer datasets, KDD remains the most common benchmark for reproducible comparisons across multiple classifiers.

C. Gradient Boosting in Security Applications

The introduction of gradient boosting brought new momentum to intrusion detection research. Chen and Guestrin [4] presented XGBoost, which uses second-order optimization, regularized tree construction, and column subsampling to achieve state-of-the-art performance on structured data. Ke et al. [5] later proposed LightGBM, which employs leaf-wise tree growth and Gradient-based One-Side Sampling (GOSS) to deliver similar accuracy with significantly lower computational cost — an important advantage for large-scale traffic analysis.

Recent studies [12] have applied these gradient boosting methods to intrusion detection, but most have done so independently. A systematic comparison under consistent preprocessing conditions is still missing, and this paper aims to address that gap.

III. DATASET DESCRIPTION

For this study, we rely on the KDD Cup 1999 dataset [13], which was originally built from the DARPA 1998 network simulation environment. The complete dataset contains about 4.9 million labeled connection records. To make experiments computationally manageable while still preserving the distribution of attack categories, researchers typically use a stratified 10% sample — amounting to 494,021 records — and we follow that convention here.

Feature Composition

Each connection record is described by 41 features, grouped into three main categories:

1. **Basic connection attributes** — such as duration, protocol type, service used, and connection flags.
2. **Content-based features** — which capture information like the number of failed login attempts, root access requests, or file creation events.
3. **Traffic-level statistical features** — calculated over short two-second windows, reflecting broader patterns like connection rates, SYN error frequencies, and destination host activity.

Together, these features provide a detailed view of both individual connection behavior and aggregate traffic trends, making the dataset a cornerstone for evaluating intrusion detection methods.

TABLE I — KDD Cup 1999 Dataset Class Distribution

Normal	Legitimate network traffic	97,278	19.69%
DoS	Denial of Service attacks	391,458	79.24%
Probe	Network surveillance/scanning	4,107	0.83%
R2L	Remote to Local unauthorized access	1,126	0.23%
U2R	User to Root privilege escalation	52	0.01%
Total	—	494,021	100%

As shown in Table I, the distribution of attack types in the dataset is highly uneven. Denial-of-Service (DoS) records make up close to 80% of all entries, while User-to-Root (U2R) attacks appear in less than 0.01% of cases. This extreme imbalance creates a significant challenge for model training.

Without corrective measures, classifiers tend to favor the majority classes — DoS and Normal traffic — which can inflate overall accuracy scores but mask poor performance on the rarer, and often more critical, attack categories.

IV. METHODOLOGY

A. System Architecture

The proposed threat detection framework is organized into five sequential modules: **Data Ingestion → Preprocessing → Feature Engineering → Model Training → Evaluation**. Each stage is designed to be independently configurable, which not only supports ablation studies but also makes it easier to integrate the system with live network monitoring environments.

B. Preprocessing Pipeline

- **Duplicate Removal**: In line with Tavallaee et al. [9], duplicate records are eliminated. Roughly 78% of the original KDD dataset consists of repeated entries, which, if left in place, can artificially boost accuracy by allowing identical samples to appear in both training and testing sets.
- **Label Consolidation**: The dataset's 23 attack subcategories are grouped into five broader classes — Normal, DoS, Probe, R2L, and U2R — following the standard KDD taxonomy. This mapping ensures reproducibility and consistency in multi-class evaluations.
- **Categorical Encoding**: Nominal features such as protocol_type (tcp, udp, icmp), service (http, ftp, smtp, etc.), and flag (SF, S0, REJ, etc.) are converted into integer values using label encoding, preparing them for tree-based models.
- **Feature Normalization**: The remaining 38 numeric features are scaled to a [0, 1] range using min-max normalization. This step balances feature contributions and improves convergence for algorithms sensitive to distance metrics, such as KNN and SVM.

C. Feature Engineering

To reduce dimensionality, a preliminary Random Forest model is used to compute feature importance scores. The top 20 features, ranked by mean decrease in impurity, are retained while discarding attributes with near-zero variance or high correlation. This streamlining speeds up training without sacrificing accuracy. Key retained features include: src_bytes, dst_bytes, count, srv_count, dst_host_count, logged_in, protocol_type, service, flag, and serror_rate.

D. Model Configurations

Six classifiers are trained and tested using identical data splits. Table II outlines the hyperparameter settings for the two primary models, providing a clear basis for comparison across approaches.

TABLE II — Hyperparameter Configuration

Objective	multi:softmax	multiclass
Num. Classes	5	5
Max Depth	6	N/A (leaf-wise)
Num. Leaves	N/A	31
Learning Rate	0.1	0.05
N Estimators	300	500
Subsample	0.8	0.8 (bagging)
Col. Subsample	0.8	0.8
Reg. Lambda	1.0	0.1
Early Stopping	20 rounds	20 rounds
Eval Metric	mlogloss	multi_logloss

E. Evaluation Protocol

To assess model performance, the cleaned dataset is split into two parts: **80% for training (395,217 samples)** and **20% for testing (98,804 samples)**. Stratified random sampling is used so that the proportions of each attack category remain consistent across both sets. In addition, a **five-fold stratified cross-validation** is carried out on the training data to measure variance and ensure robustness.

Performance is reported using a comprehensive set of metrics:

- **Overall Accuracy** – the proportion of correctly classified records.
- **Per-Class Precision, Recall, and F1-Score** – to capture how well each attack type is identified.
- **Weighted Average F1-Score** – balancing performance across imbalanced classes.
- **Training Time (seconds)** – to evaluate computational efficiency.

This protocol ensures that results reflect not only aggregate accuracy but also the system's ability to detect minority attack types, which are often the most critical in real-world scenarios.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Overall Classification Performance

Table III presents the overall performance of all six classifiers evaluated on the held-out test set. XGBoost achieves the highest accuracy (99.2%) with an overall weighted F1-score of 0.991, closely followed by LightGBM (99.0%, F1 = 0.989). Both substantially outperform all baseline methods.

TABLE III — Overall Classifier Performance Comparison

Naive Bayes	88.4	0.861	0.884	0.872	0.8
KNN (k=5)	96.1	0.958	0.961	0.959	42.3
Decision Tree	97.8	0.977	0.978	0.977	3.1
Random Forest	98.5	0.984	0.985	0.984	18.6
LightGBM	99.0	0.989	0.990	0.989	9.4
XGBoost	99.2	0.991	0.992	0.991	22.7

B. Per-Class F1-Score Analysis

Table IV provides per-class F1-scores, revealing important nuances that aggregate accuracy conceals. Both XGBoost and LightGBM achieve near-perfect detection on DoS and Normal traffic. Probe detection is also strong (F1 > 0.97). However, R2L and U2R — the two minority classes — show significantly lower F1-scores across all classifiers, reflecting the challenge posed by extreme class imbalance.

TABLE IV — Per-Class F1-Score by Classifier

Naive Bayes	0.921	0.901	0.743	0.412	0.081
KNN	0.981	0.972	0.891	0.623	0.312
Decision Tree	0.987	0.991	0.943	0.714	0.423
Random Forest	0.993	0.996	0.961	0.741	0.467
LightGBM	0.997	0.998	0.974	0.782	0.531
XGBoost	0.998	0.999	0.981	0.801	0.563

C. Cross-Validation Results

Five-fold stratified cross-validation on the training partition confirms the generalizability of results. XGBoost achieves a mean CV accuracy of 99.1% ± 0.08%, and LightGBM records 98.9% ± 0.11%. The low standard deviation across folds indicates stable model behavior rather than overfitting to any particular data partition.

D. Computational Efficiency

LightGBM stands out for its speed, finishing training in just **9.4 seconds**, compared to **22.7 seconds** for XGBoost. That's roughly a **2.4× improvement**, achieved with only a negligible drop in accuracy (about 0.2 percentage points). This efficiency advantage becomes even more significant as dataset size grows, making LightGBM particularly well-suited for real-time detection systems or environments with limited resources. By contrast, KNN proves to be the most computationally demanding model. Because its prediction complexity scales linearly with the number of samples, even the training-time index construction takes **42.3 seconds**, highlighting its inefficiency for large-scale intrusion detection tasks.

E. Discussion

The experiments confirm that **gradient boosting methods dominate** when applied to tabular intrusion detection data. Their strength lies in the iterative error-correction process: each new tree focuses on the mistakes of the previous ones, steadily improving performance. Compared to Random Forest, this sequential refinement explains the superior results. XGBoost further benefits from its built-in regularization (L1 and L2 penalties on leaf weights), which helps control overfitting and is reflected in stronger scores for minority attack classes.

Still, the results reveal a persistent weakness. Rare attack types such as R2L and U2R achieve low F1-scores (0.801 and 0.563, respectively). With only **52 U2R training samples** against nearly **300,000 DoS records**, even advanced models struggle to learn meaningful decision boundaries. This imbalance remains the most pressing challenge in intrusion detection research.

Finally, it is important to recognize the limitations of the **KDD Cup 1999 dataset** itself. While it remains a standard benchmark, it does not capture the realities of modern network traffic. Emerging threats like ransomware, advanced persistent threats (APTs), and data exfiltration over encrypted channels are absent. As a result, these findings should be interpreted as benchmark performance rather than direct indicators of real-world effectiveness.

VI. LIMITATIONS

- **Dataset Age**: The KDD Cup 1999 dataset was created

more than 25 years ago. As a result, it does not capture modern attack types such as ransomware, advanced persistent threats (APTs), or IoT-based botnets. This limits how well findings can be generalized to today's network environments.

- **Class Imbalance**: Rare attack categories like R2L and U2R together make up less than 0.25% of the dataset. Their scarcity makes it extremely difficult for models to learn meaningful patterns without techniques such as oversampling or cost-sensitive learning.
- **Static Evaluation**: All experiments are conducted in a batch setting. Real-world systems operate on streaming traffic, where concept drift — changes in attack behavior over time — can erode model accuracy. This dynamic aspect has not been tested here.
- **Feature Engineering Scope**: Feature selection relies on importance scores from a single Random Forest model. More advanced approaches, such as recursive feature elimination or Shapley value analysis, could produce different and potentially more effective subsets.
- **Hyperparameter Search**: Model configurations are based on defaults and manual tuning. Automated optimization methods, such as Bayesian search, may further enhance both accuracy and efficiency.

VII. CONCLUSION

This study introduced a machine learning–based framework for cybersecurity threat detection, evaluated using the KDD Cup 1999 benchmark dataset. The results clearly show that **gradient boosting algorithms, particularly XGBoost and LightGBM, outperform traditional classifiers**, achieving overall accuracies of **99.2% and 99.0%** respectively. By combining a systematic preprocessing pipeline, a standardized evaluation protocol, and detailed per-class analysis, the framework establishes a reproducible baseline for future comparative research.

High-volume attack categories such as **DoS** and **Probe** were detected with near-perfect F1-scores. However, minority classes like **R2L** and **U2R** remain difficult to identify due to extreme class imbalance. Tackling this issue is the most pressing challenge ahead. Promising strategies include **SMOTE-based oversampling**, **class-weighted loss functions**, and **one-class anomaly detection models** trained specifically on minority attack samples.

Beyond imbalance, future work should extend evaluation to **modern datasets** such as UNSW-NB15, CIC-IDS-2018, and CICIOT2023, which better reflect today's threat landscape. Incorporating **deep learning components** — for example, LSTMs for sequential traffic modeling or autoencoders for unsupervised anomaly detection — could further enhance detection capabilities. Additionally, exploring **online learning approaches** that adapt to concept drift in live traffic streams would make the system more practical for real-world deployment.

In summary, this research demonstrates that **gradient boosting–based intrusion detection is both highly accurate and computationally efficient**, offering a strong foundation for operational use in network security environments.

REFERENCES

[1] Cybersecurity Ventures, "Cybercrime Report 2023," Cybersecurity Ventures, 2023. [Online]. Available: https://cybersecurityventures.com

[2] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42–57, 2013.

[3] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. Pearson, 2017. [4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[5] 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, vol. 30, 2017.

[6] D. E. Denning, "An intrusion-detection model," IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[7] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 227–261, 2000.

[8] A. L. Buczak and E. Guven, "A survey of data mining and

machine learning methods for cyber security intrusion detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.

[9] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6. [10] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," International Journal of Engineering Research and Technology, vol. 2, no. 12, 2013.

[11] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. Military Communications and Information Systems Conference, 2015, pp. 1–6.

[12] H. Imamverdiyev and F. Abdullayeva, "Deep learning method for denial of service attack detection based on restricted Boltzmann machine," Big Data, vol. 6, no. 2, pp. 159–169, 2018.

[13] UCI KDD Archive, "KDD Cup 1999 Data," University of California, Irvine. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html