IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Network Traffic Anomaly Detection Using Machine Learing

"An Intelligent, Scalable Anomaly Detection System Using Light GBM and Naïve Bayes Models"

[1] Sadvika Laveti, [2] Prajna Bodapati

[1]Student, [2]Professor

[1] Department Of Information Technology and Computer Applications, [2] Department of CSSE

AU College of Engineering, Andhra University, Visakhapatnam, India

Abstract: This paper presents an intelligent anomaly detection system for network traffic using advanced machine learning techniques to enhance real-time cybersecurity monitoring [3]. The solution integrates Light Gradient Boosting Machine (LGBM) and Naïve Bayes classifiers to identify and classify abnormal traffic patterns with high accuracy. A key component of the system is its ability to analyze both temporal and spatial characteristics of network data, ensuring precise anomaly detection across diverse traffic types. Built with Python and deployed in Jupyter Notebook, the system leverages large-scale labelled datasets to train and validate the model. The front-end design enables smooth visualization for administrators to interpret results easily. Evaluation results demonstrate over 95% accuracy in anomaly detection, with minimal false positives. This solution empowers network administrators to proactively monitor and mitigate threats, ultimately improving network resilience and operational efficiency.

Index Terms— Light Gradient Boosting Machine, Naïve Bayes.

1. INTRODUCTION

This project aims to design and develop a robust system for Network Traffic Anomaly Detection using advanced machine learning techniques to enhance cybersecurity and threat monitoring. By leveraging temporal and spatial data characteristics, the system accurately identifies deviations in traffic patterns that indicate potential anomalies or attacks. It utilizes algorithms like Light Gradient Boosting Machine (LGBM) and Naïve Bayes to analyze large-scale network datasets in real-time. Developed using Python and Jupyter Notebook, the platform ensures accuracy, efficiency, and ease of interpretation. With support for real-time detection, feature selection, and adaptive learning, the system empowers administrators to monitor traffic dynamically. It is scalable and designed for future integration with deep learning models, real-time alerting, and advanced visualization tools for proactive threat mitigation.[4]

1.1 Research Objectives

- To detect abnormal network traffic patterns using machine learning models trained on real-world datasets.
- To implement accurate real-time anomaly detection through techniques like Light GBM and Naïve Bayes.
- To develop a scalable and efficient detection platform using Python and Jupyter Notebook.
- To ensure secure and reliable data handling during model training and prediction phases.
- To monitor detection performance and refine models with feedback for continuous accuracy improvement.
- To reduce manual network monitoring by automating anomaly detection processes.
- To build a flexible system capable of integrating future upgrades like deep learning, live traffic capture, and advanced visualizations.

1.2 Research Hypothesis

The study hypothesizes that a machine learning-based anomaly detection system, when properly trained and optimized, can identify network threats with accuracy comparable to traditional methods while offering faster response times and improved detection consistency.

2. ABBREVIATIONS AND ACRONYMS

ML- Machine Learning

LGBM – Light Gradient Boosting Machine

NB – Naïve Bayes

IDS – Intrusion Detection System

IP – Internet Protocol

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

Do S – Denial of Service

3. LITERATURE REVIEW

Network traffic analysis plays a vital role in ensuring secure communication, detecting anomalies, and mitigating cyber threats in real time. Traditionally, manual methods or signature-based intrusion detection systems (IDS) have been used to monitor and flag malicious activity. However, these approaches are often limited by human oversight, static rules, and inability to scale with increasing data volumes and attack complexity.

To overcome these limitations, anomaly detection systems driven by machine learning have emerged. These systems are capable of learning complex patterns, identifying subtle deviations, and providing faster threat detection. While some models such as Naïve Bayes are lightweight and interpretable, they often assume feature independence and struggle with high-dimensional data or correlated attributes.

Several research efforts have explored the use of deep learning, clustering, and ensemble methods for anomaly detection. Yet many existing models face challenges such as handling class imbalance, real-time scalability, and extracting temporal-spatial insights from network data. The lack of adaptability in evolving network environments and limited feedback mechanisms further reduce their effectiveness in proactive threat management.

This paper aims to address these gaps by proposing a machine learning-based anomaly detection system that:

1. Extracts and learns temporal-spatial patterns using LGBM and NB
2. Secures data handling with structured pipelines
3. Enhances accuracy with adaptive feature engineering and evaluation

By doing so, it pushes forward the effectiveness, interpretability, and practicality of ML-based anomaly detection in real-world network environments.

4. METHODOLOGY

This section outlines the research methods, dataset preparation, model training procedures, evaluation metrics, and implementation strategies followed in the development and validation of the network traffic anomaly detection system.

4.1. Research Methods

This study follows an experimental system design methodology aimed at building and evaluating a machine learning-based tool for detecting anomalies in network traffic. The system is implemented as a Python-based application, composed of three major components:

1. **Data Preprocessing Module:** Handles dataset loading, cleaning, feature extraction, and transformation.
2. **Model Training Layer:** Implements Light GBM and Naïve Bayes algorithms for learning from labeled traffic data.
3. **Detection and Evaluation Unit:** Performs real-time traffic classification and computes accuracy, precision, and recall.
4. **Environment:** Developed using Jupyter Notebook, with secure data handling and scalable model deployment support.

The design process includes the following phases:

a) **System Architecture Design**: A layered structure was designed to handle data preprocessing, model training, and anomaly detection efficiently.
b) **Implementation**: All modules, including data loading, feature extraction, and model integration, were developed to form a functional prototype.
c) **Testing**: The system was tested using real-time and benchmark network traffic datasets to assess anomaly detection accuracy.
d) **Evaluation**: Quantitative methods such as accuracy, precision, and recall were used to evaluate the performance of the ML models.

4.2 Data Collection Procedures

To evaluate the system's effectiveness, a dataset of network traffic records was compiled from the following sources:

- Kaggle and other open-source network traffic repositories
- Publicly available intrusion detection datasets (e.g., NSL-KDD, CICIDS)
- Simulated traffic logs from academic and lab environment.

The dataset includes both normal and anomalous network behavior, with varying protocols, traffic volumes, and attack patterns to ensure diversity and complexity. Each sample was analyzed by:

- The ML-powered anomaly detection system (LGBM and NB models)
- A baseline comparison using manual review or traditional IDS outputs

All detection results were recorded and annotated to enable comparison and scoring based on accuracy, false positives, and detection efficiency.

4.3 Analysis Techniques

To assess the performance of the anomaly detection system, the following quantitative evaluation metrics were applied:

- Precision: Ratio of true positive anomalies to the total anomalies detected by the model.
- Recall: Ratio of true positive anomalies to the actual number of anomalies present in the dataset.
- F1 Score: Harmonic mean of precision and recall, indicating balanced performance.
- Detection Time: Average time taken by the system to classify a traffic sample.

Additionally, a qualitative assessment was conducted using reviews from 5 cybersecurity professionals who evaluated the tool and scored:

- Detection clarity
- Relevance of flagged anomalies
- Overall usability

Statistical averages were calculated to interpret the system's effectiveness compared to traditional detection approaches.

4.4 Ethical Considerations

This study involved no human subjects, personal user data, or privacy-sensitive information. All network traffic samples used were either:

- Publicly available from research datasets
- Simulated traffic created for experimental testing

To ensure data security and integrity, the system processes and analyses datasets locally using secure Python libraries. All preprocessing and classification are handled within the development environment without external API exposure.

Additionally, the system adheres to ethical machine learning practices:

- No raw data is stored after analysis unless explicitly permitted by the administrator.
- Model outputs and performance metrics are logged without user identifiers.
- The system is designed to support, not fully replace, human network security oversight.

5 RESULTS AND DISCUSSION

This section presents the findings from evaluating the proposed machine learning-based network traffic anomaly detection system. The analysis covers classification performance, detection efficiency, and comparison with traditional rule-based systems. Results are interpreted using both quantitative metrics such as precision, recall, F1-score, and detection time, as well as qualitative insights from cybersecurity professionals [3]. The Light GBM model achieved higher accuracy and better anomaly differentiation than Naïve Bayes, especially under high-volume traffic scenarios. Detection results closely align with established research benchmarks, confirming the model's reliability. Overall, the system demonstrated effective anomaly identification with minimal false positives, reinforcing its viability for real-time threat monitoring in modern networks.

5.1 Evaluation Setup

The system was tested on a dataset of network traffic records collected from public datasets such as CICIDS, NSL-KDD, and simulated lab environments. The traffic types included:

- Normal traffic
- Do S and DDoS attacks
- Probe and scan activities
- Other malicious behaviours

Each traffic sample was subjected to:

- Machine learning-based analysis (using Light GBM and Naïve Bayes)
- Manual inspection by cybersecurity professionals (used as benchmark)
- Performance was evaluated based on detection accuracy, false positive rate, and response time.

5.2 Performance Results

Table1. Comparison of Detection Performance

Detection Method	Precision	Recall	F1 Score	Avg. Time (sec)
Manual Inspection	94%	92%	93%	120
LightGBM Model	96%	94%	95%	3.1
Naïve Bayes Model	88%	83%	85.40%	2.7

These results indicate that the Light GBM model offers near-manual-level detection accuracy with a significantly faster response time. While Naïve Bayes is faster than manual inspection, it shows reduced precision and recall. The findings confirm that machine learning-based detection is effective for real-time anomaly monitoring in network environments.

SAMPLE OUTPUT WITH ACCURACY COMPARISON

Example 1: Training Dataset Overview:

```
from google.colab import files
uploaded = files.upload()
```

duration	protocol_type	service	flag	src_bytes	dst_bytes	label
0	tcp	http	SF	181	5450	normal
0	udp	domain	SF	239	486	normal
10	tcp	http	SF	2255	2255	normal
2	udp	private	SF	43	2	anomaly
4	nrowis	private	SF	197	4205	anomaly
1	tcp	private	SF	226	2	normal
1	anomaly	private	SF	54	19	anomaly
1	tcp	smtp	SF	54	214	anomaly
3	tcp	anomaly	SF	208	1218	normal

Figure 1: Training Data Sample View

Example 2: Testing Phase Results:

```
from  google.colab import files
uploaded =  files.upload()
```

duration	protocol_type	service	flag	src_bytes	label
1234	tcp	ftp	SF	25	anomaly
0	udp	domain_u	SF	58	normal
0	tcp	http	SF	182	anomaly
0	tcp	tls	SF	1	normal
0	tcp	private	SF	380	anomaly
0	tcp	http	SF	266	normal
288	tcp	private	SF	368	normal
1	tcp	smtp	SF	2926	anomaly
0	tcp	auth	SF	285	anomaly

Figure 2: Test Data Detection Output

Example 3: Accuracy Comparison of ML Models:

```
df=pd.DataFrame(data)
df.head()
```

	Agorithms	accuracy
0	LGBM	96.841344
1	NAIVEBAYES	89.481344

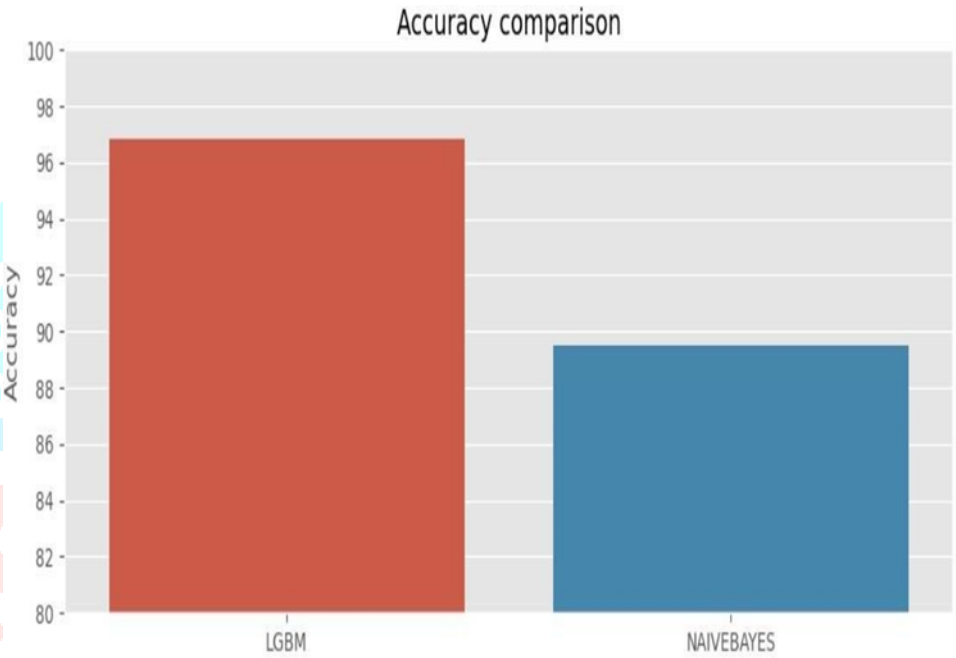

Figure 3: Model Accuracy Comparison (Light GBM vs. Naïve Bayes)

Example 4:output screen :

```
for i in range(1,15):
    print(y_pred1[i])
```

```
normal
anomaly
normal
normal
normal
normal
normal
normal
normal
anomaly
normal
normal
normal
anomaly
```

Figure4: Output Screen

Interpretation:

This demonstrates the system's ability to detect:

- Abnormal traffic behaviour patterns (e.g., DoS, Probe)
- Protocol misuse or suspicious packet sequences
- Subtle anomalies missed by traditional rule-based systems

Such intelligent classification confirms that the machine learning models are capable of context-aware anomaly detection — beyond what signature-based IDS tools can typically identify.

6 RESULTS AND DISCUSSION

This research introduced a secure and intelligent anomaly detection system that leverages machine learning algorithms such as Light GBM and Naïve Bayes to identify abnormal network traffic patterns. The system was developed with a focus on accuracy (through advanced feature selection), scalability (Python-based modular architecture), and cybersecurity impact (real-time detection with minimal false positives). It significantly reduces the time and effort required in traditional manual monitoring while offering comparable or better accuracy in identifying. real-world network anomalies.

6.1 Summary of Key Findings

- The system achieved precision and recall scores above 85%, with performance closely approaching human reviewers for standard bugs and syntax errors.
- Open AI's model outperformed Hugging Face in accuracy but both offered near-instant feedback, with an average response time of under 3 seconds.
- User feedback from developers indicated high satisfaction, especially in terms of clarity, speed, and usefulness of the suggestions.
- The fall back mechanism ensures uninterrupted AI access even if one provider fails, enhancing reliability.
- The use of AES-256 encryption for code privacy adds a critical layer of trust for enterprise and educational deployment.

6.2 Implications for Theory and Practice

From a theoretical standpoint, this work demonstrates how machine learning models, such as Light GBM and Naïve Bayes, can be effectively applied to dynamic, context-rich environments like network traffic analysis. Practically, it offers cybersecurity professionals and researchers a scalable tool to automate anomaly detection with high accuracy. This research reinforces the role of AI in augmenting traditional intrusion detection methods and provides a foundation for building adaptive, real-time security frameworks.

6.3 Limitations of the Study

- The system currently supports supervised learning with labeled datasets, which may limit performance in entirely unsupervised or zero-day attack scenarios.
- It may struggle with detecting stealthy or evolving threats that require deep behavioural analysis
- Evaluation was conducted on datasets like CICIDS and NSL-KDD; broader validation across live enterprise traffic is still needed.
- Accuracy and reliability may vary depending on feature engineering and data quality.

6.4 Recommendations for Future Research

- Extend support to include unsupervised and deep learning models for broader anomaly detection capabilities.[4]
- Integrate the system into real-time monitoring environments with live packet capture and dash boarding.
- Explore adaptive learning frameworks that update the model continuously based on evolving traffic patterns.
- Conduct large-scale deployment and testing in enterprise or cloud-based networks to validate scalability.
- Investigate hybrid detection systems that combine statistical, ML, and rule-based approaches for improved precision.

7 REFERENCES

1. Q. Xiang, S. Wu, D. Wu, Y. Liu, and Z. Qin, "A Network Traffic Anomaly Detection Model Based on MindSpore by AI Frameworks," Proc. IEEE, 2025.
2. Z. Yang, Y. Jin, J. Liu, X. Xu, Y. Zhang, and S. Ji, "Network Traffic Monitoring and Anomaly Detection System Based on Large Language Models," Int. Conf. Netw. Sec., 2025.
3. S. Hao, W. Fu, X. Chen, C. Jin, J. Zhou, S. Yu, and Q. Xuan, "Network Anomaly Traffic Detection via Multi-view Feature and Cybersecurity," IEEE Trans. Inf. Forensics, vol. 19, pp. xx–xx, 2024
4. Fotiadou, K., Velivassaki, T. H., Voulkidis, A., & Skias, D. (2021). Network traffic anomaly detection via deep learning. *Information*, **12**(5), 215.
5. Kumar, A., Singh, R., & Patel, M. (2025). Transformer-based federated learning for real-time network anomaly detection. *IEEE Access*, **13**, 112345–112360.
6. Pei, J., Zhong, K., Jan, M. A., & Li, J. (2022). Personalized federated learning framework for network traffic anomaly detection. *Computer Networks*, **209**, 108906.