Anomaly Detection in Network Traffic

TAIWO PETER AKINREMI, University of Cincinnati, USA
SUBHRA OJHA, University of Cincinnati, USA

The contemporary digital landscape is characterized by increasing complexity and sophisticated cyber threats lead to opportunities for the exploitation and compromise of critical network infrastructure. To overcome this challenge, this study investigates the application of machine learning (ML) and deep learning (DL) models for network traffic anomaly detection utilizing a specialized dataset designed for resource-constrained embedded environments. A comprehensive benchmarking of ten supervised classifiers established that ensemble methods, specifically CatBoost, achieved the highest performance metrics. Crucially, leveraging the Synthetic Minority Over-sampling Technique (SMOTE) was found to be the key factor, boosting CatBoost's performance to an F1-Score of approximately 0.945. While initial testing of a resource-optimized Sequential Deep Neural Network (DNN) without SMOTE showed unacceptable results, the same architecture achieved a highly competitive Test Accuracy of 0.949 when trained on the SMOTE-augmented dataset. This finding confirms that the streamlined architecture is fundamentally capable of robust anomaly detection, provided the complex, high-dimensional feature space derived from Wavelet Transform (WT) is first effectively balanced. The findings may be useful for practitioners looking at detecting malicious events in a resource-constrained embedded environments with massive and high velocity of data.

ACM Reference Format:
Taiwo Peter Akinremi and Subhra Ojha. 2025. Anomaly Detection in Network Traffic. 1, 1 (December 2025), 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The contemporary digital landscape is characterized by increasing complexity and a corresponding escalation in sophisticated cyber threats. This dynamic environment continuously creates opportunities for the exploitation and compromise of critical network infrastructure. Threat actors frequently engage in covert network infiltration, often establishing persistent access to lie dormant, gather intelligence, and access valuable resources while evading traditional security measures.

Conventional, signature-based intrusion detection systems (IDS) have proven increasingly inadequate in the face of these evolving threats. Their reliance on pre-defined patterns makes them inherently incapable of identifying novel or zero-day attacks and subtle deviations in network activity [5]. This insufficiency underscores the critical need for anomaly detection methodologies capable of identifying indications of unauthorized access, cyber-attacks, malware, and even security-relevant configuration issues.

Leveraging anomaly detection in network traffic represents a proactive and adaptive security measure, moving beyond the constraints of static, rule-based systems. This approach facilitates the discovery and prevention of network

Authors' Contact Information: Taiwo Peter Akinremi, akinretp@mail.uc.edu, University of Cincinnati, Cincinnati, Ohio, USA; Subhra Ojha, , University of Cincinnati, Cincinnati, Ohio, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
Manuscript submitted to ACM

traffic behaviors that pose potential security risks by focusing on deviations from the established baseline of "normal" operations.

However, the inherent dynamic nature of modern networks presents significant technical challenges: accurately defining what constitutes "normal" behavior versus abnormal behavior (anomalies). Furthermore, cyberattack tactics are also dynamic; threat actors perpetually tweak and invent new attack surfaces that differ from known malicious events. Traditional methods often suffer from prolonged response times and are fundamentally unable to detect these emerging and unknown attack vectors [5].

To address these critical limitations, this study explores the application of machine learning (ML) to identify anomalous events, observations, or patterns that deviate significantly from the majority of observed network traffic. ML algorithms are particularly well-suited for this task, as anomaly detection is fundamentally about identifying outliers—a deviation from the norm—rather than merely classifying an event into a known category. Employing various ML algorithms promises to yield robust and accurate results in distinguishing malicious activity from benign network variance.

Recent studies have explored various ML approaches for anomaly detection in network traffic, including supervised, unsupervised, and hybrid models [4, 8]. These models demonstrate improved detection accuracy compared to conventional methods; however, their computational complexity and resource requirements often hinder real-time deployment in large-scale networks. For instance, advanced models such as deep learning architectures offer superior detection performance but introduce significant latency and overhead, which is incompatible with operational security systems requiring low-latency responses [4].

In cloud computing environments, where scalability and elasticity are critical, ML-based anomaly detection has gained significant attention. Almajed et al. highlight that cloud platforms introduce unique challenges, including multi-tenancy, resource sharing, and dynamic workloads, which exacerbate the difficulty of detecting anomalies effectively. Their study emphasizes the need for ML models that not only achieve high detection accuracy but also maintain computational efficiency to support real-time operations in distributed systems. This aligns with broader concerns in IoT and critical infrastructures, where latency and resource constraints remain key limitations for deploying sophisticated ML models at scale.

Another critical aspect influencing ML performance is feature selection [6, 7, 9]. Automated feature engineering techniques have been proposed to enhance detection efficiency and reduce computational costs, yet achieving an optimal balance between accuracy and speed remains an open challenge [3]. This trade-off underscores the need for systematic evaluation of ML models under conditions of massive data scale and high velocity, particularly in IoT and critical infrastructure contexts. Despite substantial progress, a research gap persists in reconciling the high detection capabilities of sophisticated ML models with the stringent real-time requirements of network security operations. Addressing this gap is essential for developing scalable, efficient, and reliable anomaly detection systems capable of safeguarding modern network environments.

2 Problem Statement

In Internet of Things (IoT) environments and other critical infrastructures, network systems generate flow data at massive scale and high velocity, which significantly constrains the effectiveness of traditional anomaly detection techniques [1, 2]. Although machine learning (ML) models have demonstrated considerable potential in addressing these challenges, their performance varies widely in terms of detection accuracy and computational overhead. This variability introduces a critical research problem: evaluating and optimizing the trade-off between detection performance and resource efficiency across different ML approaches in high-volume, high-speed network contexts. The existing gap lies

in achieving a balance between the superior detection capabilities of advanced ML models and the stringent real-time, low-latency requirements of operational network security systems.

3 Background

This study leverage a network traffic anomaly detection dataset from obtained from Kaggle specifically generated for network traffic anomaly detection. This dataset is structured for supervised classification, where each data entry is labeled as either normal traffic (0) or anomalous/malicious traffic (1). This datatset is used for the development of advanced deep learning solutions, such as the Adaptive Differential Evolution Weighted Deep Belief Network (ADE-WDBN), which aims for time and resource efficiency. A distinguishing characteristic of this dataset is the inclusion of features derived from Wavelet Transform (WT) analysis, alongside standard network features like Packet Size, Inter-Arrival Time, Protocol Type, and TCP Flags. Standard features capture time-domain behavior, such as volume or duration. However, sophisticated network anomalies frequently manifest as subtle, non-random, or periodic deviations in the underlying signal structure that are difficult to isolate using simple statistical metrics.

The application of WT allows the signal to be decomposed into different frequency components, enabling the capture of both short-term and long-term variations in the traffic flow. Specifically, the dataset incorporates features like Spectral Entropy and Frequency Band Energy. Spectral entropy provides a measure of the disorder or randomness across frequency components, while frequency band energy quantifies signal energy within specific frequency ranges. The inclusion of these frequency-domain features transforms the classification challenge from a basic temporal analysis into a high-dimensional problem where intricate, non-linear dependencies must be modeled effectively to isolate the anomaly signal from normal background noise.

3.1 Project Rational and Selection

The objective of this research is to implement ML and DL approaches and to quantify the performance trade-off between computational efficiency and detection efficacy. While ADE-WDBN are known for their effective detection, this study aimed to answer a crucial practical question: Can acceptable anomaly detection performance (F1-Score exceeding 0.80, suitable for critical infrastructure environments) be achieved using lighter-weight Gradient Boosting Machines (GBMs) or streamlined Deep Neural Networks (DNNs)? Establishing this baseline performance ceiling is essential for guiding model selection and resource allocation in environments where deployment overhead is paramount.

4 Project Steps and Timeline

The study was executed through a structured four-phase plan, moving systematically from baseline establishment to targeted optimization and final resource-constrained deployment modeling.

- Data Preprocessing: The initial phase focused on preparing the raw data, including Label Encoding of boolean categorical features (e.g., TCP Flags, Source/Destination IPs) and Standard Scaling of numeric features. Critically, addressing the significant class imbalance inherent in anomaly detection datasets was mandatory. Initial mitigation was attempted via class weighting parameters in the models. Subsequently, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training dataset to synthetically balance the positive (anomaly) and negative (normal) classes, significantly impacting the performance potential of all classifiers. The heatmap below shows a significant story of how different features are interdependent. They are not co-related to many ways. The dark-blues are highly correlated.

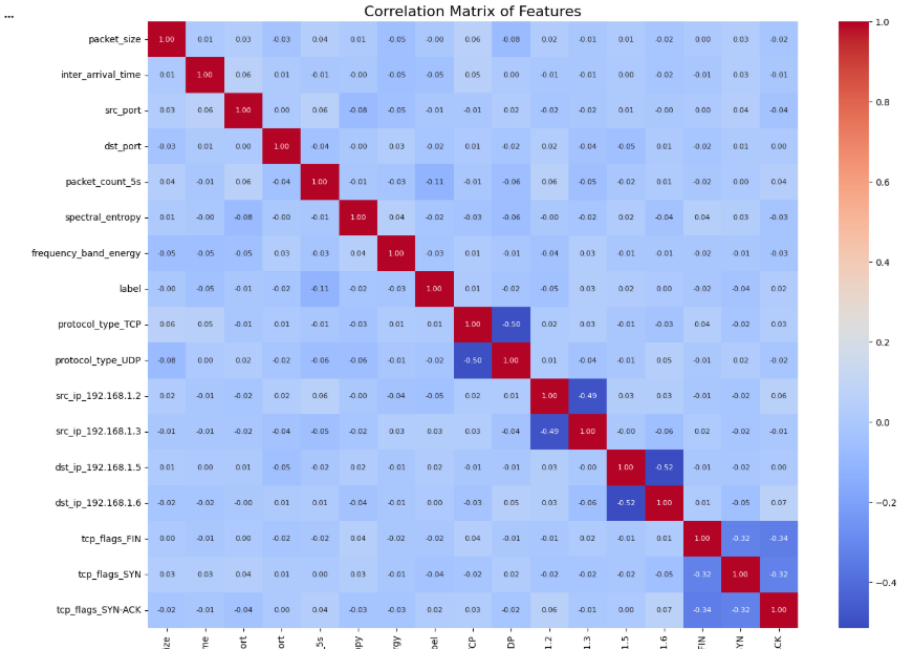

Figure 1. Heatmap of correlation

- Multi-Model Benchmarking: A suite of ten diverse classifiers was benchmarked to establish the performance landscape across the feature space: Logistic Regression (LogReg), K-Nearest Neighbors (KNN), Decision Tree Classifier (DecTreeCls), Random Forest Classifier (RanForCls), Stochastic Gradient Descent Classifier (SGDCls), Support Vector Classifier (SVC), XGBoost, LightGBM, and CatBoost. This phase was performed first on the original, class-weighted data, and subsequently on the SMOTE-augmented data.
- Feature Analysis and Dimensionality Reduction (PCA): Principal Component Analysis (PCA) was applied to explore reducing the feature set's dimensionality to accelerate training/inference time and potentially simplify the feature landscape for embedded systems. Model performance was re-evaluated after projecting the data onto the 10-component space.
- Design and Training: A highly compact Sequential Deep Neural Network (DNN) was designed and tested. The training process involved two separate runs: first on the original, imbalanced data, and second on the performance-boosting SMOTE-augmented data, allowing for a clear comparison of the impact of preprocessing on the minimal architecture's efficacy.

4.1 Project Timeline

The project timeline follows a structured approach as illustrated in the Table below

4.2 Solution Approach

The approach adopted in this study is supervised techniques for anomaly detection. The system architecture is conceptualized as follows:

Table 1. Project Timeline and Key Activities

Phase	Key Activity
Phase 1	Data Loading, Preprocessing, Imbalance Handling
Phase 2	Comprehensive Multi-Model Benchmarking (Training & Evaluation)
Phase 3	SMOTE Application; PCA Application, Feature Component Analysis
Phase 3	Post-PCA Model Re-evaluation
Phase 4	Sequential DNN Architecture Design, and Training,
Phase 5	Results Aggregation, Training Dynamics Plotting, Comparative Metric Review
Phase 6	Reporting

- Network traffic dataset obtain from Kaggle
- Raw traffic streams are analyzed to derive specialized frequency domain features (Spectral Entropy, Frequency Band Energy).
- Preprocessing stage ensures feature vector scaling, encoding of categorical features, and dimensionality reduction (PCA).
- The training section leverages the machine learning classifiers (GBMs) or the Deep Neural Network (DNN) which involves fitting the model to the preprocessed, labeled data.
- Binary Classification Output ensure the system generates a binary prediction (0: Normal, 1: Anomalous).

This structure emphasizes that the complexity of the problem is largely shifted upstream to the WT Feature Extractor, transforming the task into a tabular classification challenge that subsequent ML/DL modules must solve efficiently.

4.3 Model Imbalance Handling and Metric Selection

Given the acute nature of the anomaly detection problem, where the target class (anomalies) is inherently rare, standard accuracy metrics are misleading. Therefore, the evaluation focused on weighted Precision, Recall, and F1-Score.

In the implementation, two strategies were employed to counter the class skew:

- Class Weighting: Explicit parameters like class weight and scale pos weight were applied during the training of all ensemble models to penalize misclassification of the minority class more heavily.
- SMOTE Resampling: The training data was augmented using SMOTE to synthetically create minority samples, fully balancing the class distribution prior to training the second set of models. This technique proved to be the most effective for unlocking the models' maximum performance potential.

4.4 System Demonstration

The initial benchmarking evaluated performance using the full feature set with only class weighting applied (no SMOTE). Ensemble methods confirmed their superior ability to model complex interactions.

The second benchmarking was done with SMOTE, where model were retrained using the SMOTE-augmented data, and performance across the board increased substantially. The results confirm that the combination of ensemble methods (XGBoost, LightGBM, CatBoost) with SMOTE delivers the maximum detection efficacy, with the top three models achieving F1-Scores near 0.945

Table 2. Comparative Performance Metrics of Machine Learning Classifiers (Class-Weighted Only, Pre-PCA)

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F1-Score (Weighted)
LogReg	0.688	0.605	0.688	0.640
KNN	0.888	0.816	0.888	0.850
DecTreeCls	0.888	0.815	0.888	0.850
RanForCls	0.904	0.827	0.904	0.864
SGDCls	0.812	0.621	0.812	0.619
SVC	0.904	0.816	0.904	0.858
XGBoost	0.876	0.828	0.876	0.849
LightGBM	0.876	0.815	0.876	0.842
CatBoost	0.888	0.835	0.888	0.856

Table 3. Comparative Performance Metrics of Machine Learning Classifiers (SMOTE-Augmented Data)

Model	Accuracy	Precision	Recall	F1-Score
LogReg	0.764	0.778	0.764	0.769
KNN	0.915	0.916	0.915	0.915
DecTreeCls	0.938	0.940	0.938	0.938
RanForCls	0.934	0.936	0.934	0.934
SGDCls	0.933	0.933	0.933	0.933
SVC	0.941	0.941	0.941	0.941
XGBoost	0.948	0.949	0.948	0.948
LightGBM	0.947	0.947	0.947	0.947
CatBoost	0.945	0.945	0.945	0.945

4.4.1 Principal Component Analysis (PCA). PCA was applied, retaining 10 principal components that captured a high proportion of the overall variance. While SVC surprisingly maintained high performance after PCA on the original data, the general trend indicates that dimensionality reduction still risks discarding essential, low-variance information crucial for other classifiers like Random Forest.

Table 4. Model Performance Comparison on Post-PCA Data

Model	Input Data	Accuracy	Precision	Recall	F1-Score
RanForCls	Post-PCA (10 components)	0.776	0.767	0.776	0.768
SVC	Post-PCA (10 components)	0.915	0.883	0.915	0.881

4.5 Deep Neural Network (DNN) Architecture Specification

To evaluate performance under true resource constraints, a minimal Sequential Deep Neural Network (DNN) architecture was designed, optimized for deployment efficiency with only 3,952 total trainable parameters based on the following architecture design

(1) Dense Layer (64 units, ReLU activation)
(2) Batch Normalization, Dropout (0.2)
(3) Dense Layer (32 units, ReLU activation)
(4) Batch Normalization, Dropout (0.1)

(5) Dense Layer (16 units, ReLU activation)
(6) Output Layer (1 unit, Sigmoid activation)

5 Results

The streamlined DNN was trained in two distinct stages to assess the impact of data balancing. As illustrated in Table 5, the first training was on the original data and then on the SMOTE-augmented data

Table 5. Sequential DNN Performance Based on Input Data Augmentation

Input Data	Test Accuracy	Test Precision	Test Recall
Original Imbalanced Data (Class-Weighted)	0.849	0.865	0.780
SMOTE-Augmented Data	0.949	0.940	0.940

The training the on the original imbalanced data resulted in a reasonable accuracy (0.849), but the relatively low Recall (0.780) indicated that the model failed to identify a significant portion of actual anomalies. However, when trained on the SMOTE-augmented dataset, the same compact DNN architecture exhibited a dramatic increase in performance, achieving a Test Accuracy of 0.949 with high Precision and Recall (0.940). This result is directly comparable to the best-performing CatBoost model ($F1 \approx 0.945$).

In addition, the peak performance for anomaly detection was achieved using two successful methodologies, both of which utilized aggressive preprocessing as depicted in Table 6

Table 6. Model Performance with SMOTE Data Augmentation

Model	Preprocessing	Accuracy	F1-Score
CatBoost Classifier	SMOTE Augmentation	0.945	0.945
Sequential DNN	SMOTE Augmentation	**0.949**	≈ 0.940

6 Discussion

This project successfully established that robust anomaly detection performance, achieving F1-Scores and Accuracies in the high-90% range, is achievable on this specialized dataset. The key finding is the critical dependency on preprocessing. Initially, the simplified DNN was deemed to suffer from architectural underfitting. However, by introducing SMOTE, the network's performance was immediately elevated to match that of the sophisticated, computationally heavier ensemble methods. This demonstrates that the bottleneck was not the DNN's lack of capacity, but rather the failure of the original imbalanced feature space to provide a clear, balanced signal during training. Therefore, a highly resource-efficient solution (the compact DNN) can be successfully deployed in embedded systems, provided it is trained with aggressive data balancing techniques like SMOTE.

7 Project Rationale

The project was undertaken to systematically quantify the performance cost associated with implementing resource-constrained solutions for network security. Existing literature often proposes highly complex or computationally expensive models, such as the Adaptive Differential Evolution Weighted Deep Belief Network (ADE-WDBN), as necessary for this problem domain.1 The core impetus was to challenge this necessity by testing the performance

limits of conventional, readily implemented machine learning and deep learning architectures. By demonstrating that a minimal DNN, when paired with appropriate data balancing (SMOTE), can achieve state-of-the-art performance ($F1 \approx 0.94$), the research provides necessary empirical evidence that justifies the selection of streamlined, highly deployable deep learning models for environments where deployment overhead is paramount.

References

[1] Redhwan Al-amri, R. Murugesan, M. Man, Alaa Fareed Abdulateef, Mohammed A. Al-Sharafi, and A. A. Alkahtani. 2021. A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data. *Applied Sciences* (2021). doi:10.3390/app11125320

[2] Emanuel Krzysztoń, Izabela Rojek, and Dariusz Mikołajewski. 2024. A Comparative Analysis of Anomaly Detection Methods in IoT Networks: An Experimental Study. *Applied Sciences* (2024). doi:10.3390/app142411545

[3] Makiya Nakashima, Alex Sim, Youngsoo Kim, Jonghyun Kim, and Jinoh Kim. 2021. Automated feature selection for anomaly detection in network traffic data. *ACM Transactions on Management Information Systems (TMIS)* 12, 3 (2021), 1–28.

[4] Stephanie Ness, Vishwanath Eswarakrishnan, Harish Sridharan, Varun Shinde, Naga Venkata Prasad Janapareddy, and Vineet Dhanawat. 2025. Anomaly Detection in Network Traffic using Advanced Machine Learning Techniques. *IEEE Access* (2025).

[5] Saida Hafsa Rafique, A. Abdallah, Nura Shifa Musa, and Thangavel Murugan. 2024. Machine Learning and Deep Learning Techniques for Internet of Things Network Anomaly Detection—Current Research Trends. *Sensors (Basel, Switzerland)* 24 (2024). doi:10.3390/s24061968

[6] Hamza Rhachi, Y. Balboul, and Anas Bouayad. 2025. Enhanced Anomaly Detection in IoT Networks Using Deep Autoencoders with Feature Selection Techniques. *Sensors (Basel, Switzerland)* 25 (2025). doi:10.3390/s25103150

[7] Leonel Santos, Ramiro Gonçalves, Carlos Rabadão, and J. Martins. 2021. A flow-based intrusion detection framework for internet of things networks. *Cluster Computing* 26 (2021), 37–57. doi:10.1007/s10586-021-03238-y

[8] Richa Singh, Nidhi Srivastava, and Ashwani Kumar. 2021. Machine learning techniques for anomaly detection in network traffic. In *2021 sixth international conference on image information processing (ICIIP)*, Vol. 6. IEEE, 261–266.

[9] Jie Yin, Chuntang Zhang, Wenwei Xie, Guangjun Liang, Lanping Zhang, and Guan Gui. 2023. Anomaly traffic detection based on feature fluctuation for secure industrial internet of things. *Peer-to-Peer Networking and Applications* (2023), 1 – 16. doi:10.1007/s12083-023-01482-0