IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

MACHINE LEARNING APPROACHES FOR ANOMALY DETECTION IN NETWORK TRAFFIC

[1]M. Vijaya Kumar, [2]Pandaraboina Phani Kumar, [3]Penumudi Rajitha, [4]Vadranam Keerthi Priya, [5]Kattupalli Rakesh Kumar

[1]Assistant Professor, [2,3,4,5]Student,

Department of CSE – Data Science

St. Anns College of Engineering & Technology, Chirala, Andhra Pradesh, India

Abstract: Anomaly detection in network traffic plays a critical role in ensuring network security, as unusual patterns may indicate cyberattacks, intrusions, or system failures. With the rapid growth of modern networks, the volume and complexity of traffic data have increased significantly, making traditional rule-based and statistical techniques less effective in identifying abnormal behaviour. This paper explores the use of machine learning techniques for efficient and accurate anomaly detection in network traffic. The proposed approach focuses on analysing network traffic characteristics and leveraging data-driven models to automatically learn normal and abnormal patterns. Machine learning algorithms including Isolation Forest, Naive Bayes, XGBoost, LightGBM, and Support Vector Machine (SVM) are employed to process large-scale datasets, enabling the detection of subtle and complex anomalies that are difficult to identify using conventional methods. The system incorporates data preprocessing using the KDDCup99 dataset, feature extraction via Principal Component Analysis (PCA), and class balancing through the Synthetic Minority Over-sampling Technique (SMOTE). The effectiveness of the proposed method is evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score. The results demonstrate that machine learning-based approaches provide a scalable and reliable solution for real-time anomaly detection in modern network environments.

Index Terms - Anomaly Detection, Network Traffic, Machine Learning, KDDCup99, PCA, SMOTE, XGBoost, LightGBM, Isolation Forest, Intrusion Detection.

I. INTRODUCTION

In today's digital era, computer networks play a crucial role in communication, data transfer, and information sharing. With the rapid growth of internet technologies and connected devices, network traffic has become increasingly complex and large in volume. This has led to a significant rise in cybersecurity threats such as cyberattacks, intrusions, denial-of-service attacks, and unauthorized system access.

Network traffic anomaly detection is the process of identifying unusual or abnormal patterns in network data that may indicate malicious activities or security threats. These anomalies can include unauthorized access, denial-of-service (DoS) attacks, data breaches, or abnormal usage patterns. Detecting such anomalies is essential for maintaining the security, reliability, and performance of network systems.

Traditional methods for anomaly detection rely on predefined rules and statistical techniques. However, these approaches are often ineffective in handling dynamic and evolving cyberattack techniques. Modern networks generate high-dimensional and large-scale data, making it difficult for conventional methods to accurately identify abnormal behaviour. The applications of network traffic anomaly detection span multiple domains, as outlined in Table 1.

Table 1: Applications of Network Traffic Anomaly Detection

Application	Description
Intrusion Detection	Identifies unauthorized access to networks
DDoS Attack Detection	Detects abnormal traffic spikes indicating attacks
Malware Detection	Identifies malicious activities in network traffic
Fraud Detection	Detects unusual transactions or abnormal behaviours
Network Monitoring	Tracks performance and detects faults in real-time

Machine learning provides a powerful solution for detecting anomalies in network traffic. These models can automatically learn patterns from data and identify deviations that indicate abnormal behaviour. Various machine learning models are widely used for anomaly detection, each offering different advantages depending on the type of data and the complexity of the problem. The main objective of this work is to develop a machine learning-based anomaly detection system that can identify abnormal network activities with high accuracy, reduced false positives, and real-time monitoring capability.

II. LITERATURE SURVEY

The literature survey involves studying previously published research papers, journals, and articles related to anomaly detection in network traffic. It helps in understanding existing techniques, identifying research gaps, and developing improved solutions.

A. Overview of Existing Research

Several machine learning models such as Decision Trees, Support Vector Machines (SVM), and Neural Networks have been used for detecting abnormal patterns in network traffic. Chandola et al. [3] presented a comprehensive survey of anomaly detection methods across different domains. Ahmed et al. [2] reviewed various network anomaly detection techniques, highlighting the increasing use of data-driven approaches. Key machine learning techniques used in anomaly detection are summarized in Table 2.

Table 2: Machine Learning Techniques in Anomaly Detection

Technique	Description	Advantages
Decision Tree	Rule-based classification	Easy to understand
SVM	Separates data using hyperplane	High accuracy on complex data
KNN	Based on nearest neighbours	Simple implementation
Neural Networks	Learns complex patterns	High detection capability

B. Deep Learning Approaches

Deep learning models have significantly improved anomaly detection performance. Models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) can automatically learn complex patterns from large datasets. These models are particularly useful for handling time-series network traffic data. Fotiadou et al. [4] demonstrated the effectiveness of deep learning for network traffic anomaly detection. Table 3 summarizes common deep learning models and their applications.

Table 3: Deep Learning Models for Anomaly Detection

Model	Key Features	Application
CNN	Feature extraction	Traffic pattern analysis
RNN	Sequential learning	Time-series analysis
LSTM	Long-term dependency	Network behaviour prediction
Autoencoder	Data reconstruction	Unsupervised anomaly detection

C. Datasets Used in Research

Various publicly available datasets are used for training and testing anomaly detection models. These datasets contain both normal and attack traffic patterns. The KDD Cup 99 dataset [5] is a classical benchmark for intrusion detection, while UNSW-NB15 and CICIDS provide more realistic and modern traffic scenarios. Table 4 lists common datasets used in anomaly detection research.

Table 4: Common Datasets for Anomaly Detection

Dataset	Description
KDD Cup 99	Classic benchmark dataset for intrusion detection
NSL-KDD	Improved version of KDD dataset
UNSW-NB15	Modern dataset with realistic network traffic
CICIDS	Contains real-world attack scenarios

D. Motivation for the Proposed System

Based on the literature survey, it is evident that existing methods have limitations in handling dynamic network behaviour and detecting sophisticated attacks. Many traditional and machine learning-based approaches struggle with high false positive rates and limited adaptability to new and evolving cyber threats. Therefore, there is a need for an improved system that provides better accuracy, reduced false alarms, and efficient processing of large-scale data. The proposed system addresses these challenges using advanced machine learning techniques combined with effective preprocessing and class balancing strategies..

III. PROPOSED WORK AND ANALYSIS

A. System Overview

The proposed system aims to develop an intelligent anomaly detection framework capable of identifying abnormal patterns in network traffic using machine learning techniques. The system begins by collecting network traffic data from the KDDCup99 dataset, which contains over 125,000 instances of normal and multi-class attack traffic such as DoS and Probe attacks. This data is then pre-processed to remove noise, handle missing values, and normalize the data for better analysis. Table 5 summarizes the key components of the proposed system.

Table 5: Components of the Proposed System

Component	Description
Data Collection	Gathering network traffic data from KDDCup99 dataset
Preprocessing	Cleaning, encoding, and normalizing data
Class Balancing (SMOTE)	Generating synthetic samples for minority classes
Feature Extraction (PCA)	Reducing dimensionality to relevant components
Model Training	Training multiple machine learning models
Detection Module	Classifying traffic as Normal or Anomaly
Alert System	Generating notifications for detected anomalies

B. System Workflow

The proposed system follows a structured workflow to ensure accurate and efficient anomaly detection. The workflow consists of the following stages:

Data Collection: Network traffic data is collected from the KDDCup99 dataset, a widely used benchmark containing instances of both normal and attack network traffic.

Data Preprocessing: The collected data is cleaned by handling missing values through mean substitution and removing duplicate entries. Label encoding is applied to convert categorical attributes into a numerical format compatible with machine learning algorithms.

Class Balancing (SMOTE): To prevent model bias toward majority classes, the Synthetic Minority Over-sampling Technique (SMOTE) is applied to generate synthetic samples for minority attack types, ensuring the dataset is balanced for training.

Feature Extraction (PCA): Principal Component Analysis (PCA) is used to reduce dimensionality by transforming correlated features into uncorrelated principal components. This reduces computational overhead and mitigates overfitting by retaining only the most relevant features.

Model Training: The processed data is fed into a diverse suite of machine learning algorithms, including Isolation Forest, Naive Bayes, XGBoost, LightGBM, and SVM. The dataset is divided into training and testing sets using an 80–20 split strategy.

Anomaly Detection: The trained model analyses incoming network traffic to classify it as either normal or anomalous. When suspicious activity is detected, the system generates alerts to notify administrators.

Result Analysis: The detected results are analysed using performance metrics such as accuracy, precision, recall, and F1-score to evaluate the system's effectiveness.

C. Algorithms Used

The proposed system employs the following machine learning algorithms for anomaly detection:

Isolation Forest: An unsupervised learning algorithm that isolates anomalies by randomly selecting a feature and splitting the data. Anomalies are isolated in fewer splits than normal data points.

Naive Bayes: A probabilistic classifier based on Bayes theorem that assumes independence between features. It is computationally efficient and well-suited for large-scale datasets.

XGBoost: An optimized gradient boosting framework that builds an ensemble of decision trees. It provides high accuracy and is effective in handling imbalanced datasets.

LightGBM: A gradient boosting framework designed for high efficiency and speed. It uses a leaf-wise tree growth strategy that results in better accuracy with lower memory consumption.

Support Vector Machine (SVM): A supervised learning algorithm that separates data using an optimal hyperplane. SVM is effective in handling non-linear boundaries in high-dimensional spaces.

D. Advantages of the Proposed System

The proposed anomaly detection system offers several advantages over traditional methods. It automatically learns patterns from network data without relying on predefined rules, making it effective in detecting both known and unknown attacks. The integration of SMOTE improves detection of minority attack classes, while PCA reduces computational complexity. Table 6 summarizes the key advantages of the proposed system.

Table 6: Advantages of the Proposed System

Advantage	Description
High Accuracy	Detects anomalies effectively using ensemble models
Automated Detection	Reduces manual effort through intelligent classification
Real-time Monitoring	Enables immediate detection of threats
Scalability	Handles large-scale datasets efficiently

A. Architectural Design

The system is designed to detect anomalies in network traffic using machine learning techniques. The architecture is structured to efficiently process network traffic data through several interconnected modules. The main components include the Data Collection Module, Data Preprocessing Module, Feature Extraction Module, Model Training Module, Anomaly Detection Module, and Performance Evaluation Module. The architectural design ensures that the system can handle large-scale network datasets and accurately identify abnormal traffic patterns.

B. Database Design

The database design involves organizing and structuring the dataset used in the anomaly detection system. The KDDCup99 dataset is used as the primary data source, which contains network traffic records with various attributes. Table 7 presents the structure of the dataset used in the proposed system.

Table 7: Dataset Structure

Field Name	Description
Source IP	IP address of the data sender
Destination IP	IP address of the data receiver
Packet Size	Size of the network packet
Protocol	Type of communication protocol
Timestamp	Time of data transmission
Label	Classification as Normal or Anomaly

C. Block Diagram

The block diagram represents the overall workflow of the anomaly detection system, illustrating the complete pipeline from data collection through preprocessing, feature extraction, model training, anomaly detection, and result evaluation. The system begins with the KDDCup99 dataset, proceeds through preprocessing and SMOTE-based class balancing, applies PCA for dimensionality reduction, and feeds the processed data into the machine learning model hub comprising Isolation Forest, Naive Bayes, XGBoost, LightGBM, and SVM. The final block calculates performance metrics to validate detection and classifies traffic as either Normal or Anomaly.

V. IMPLEMENTATION

A. Software Requirements

The implementation of the proposed anomaly detection system is developed using the Python programming language (version 3.10 or above), which provides extensive support for machine learning and data analysis applications. Visual Studio Code is used as the primary development environment. Key libraries include PyTorch for deep learning model implementation, NumPy for numerical computations, Pandas for data manipulation, Scikit-learn for preprocessing and evaluation metrics, and Matplotlib for result visualization.

B. Hardware Requirements

The system requires sufficient processing power and memory to handle large network datasets and train machine learning models. The system supports AMD64 architecture compatible with modern x64 processors on Windows, and Apple Silicon or Intel processors on macOS. A minimum of 8 GB RAM and 500 GB storage is recommended to efficiently store datasets, project files, and results. A stable network connection is required for downloading datasets and software libraries.

C. Implementation of Modules

Data Preprocessing: The KDDCup99 dataset is loaded and cleaned by handling missing values through mean substitution and removing duplicate records. Label encoding is applied to convert categorical features into numerical format compatible with machine learning algorithms.

Class Balancing: SMOTE is applied to the training dataset to generate synthetic samples for minority attack categories, ensuring that the model is not biased towards the majority class. This improves the detection rate for less frequent attack types.

Feature Extraction: PCA is applied to reduce the dimensionality of the feature space. The most informative principal components are retained, reducing computational overhead and mitigating overfitting.

Model Training: The dataset is split into training (80%) and testing (20%) sets. Multiple machine learning models including Isolation Forest, Naive Bayes, XGBoost, LightGBM, and SVM are trained on the preprocessed and balanced dataset.

Model Evaluation: The trained models are evaluated on the test dataset using accuracy, precision, recall, and F1-score. These metrics provide a comprehensive assessment of the system's detection capability.

D. Model Training and Optimization

After preprocessing and feature selection, machine learning models are trained using the historical KDDCup99 dataset. The dataset is divided into training and testing sets using an 80–20 split strategy. Binary Cross Entropy loss function is used for the deep learning components, and the Adam optimizer is employed to update model parameters efficiently. Early stopping is applied during neural network training to prevent

overfitting. The ConvLSTM model combining CNN and LSTM layers is also implemented in PyTorch, which processes input data through convolution layers for spatial feature extraction and LSTM layers for temporal pattern learning.

VI. TESTING

The proposed anomaly detection system is evaluated using a comprehensive set of test cases to verify its performance and reliability. Different categories of input data are used, including normal traffic data, abnormal consumption patterns, noisy data, and unseen inputs. Table 8 presents the testing results of the proposed system.

Table 8: Testing Results of the Proposed System

Test Case ID	Input Data	Expected Output	Actual Output	Status
TC01	Normal network traffic data	Classified as Normal	Classified as Normal	Pass
TC02	Abnormal traffic pattern (spike)	Classified as Anomaly	Classified as Anomaly	Pass
TC03	Missing values in dataset	Cleaned and processed data	Cleaned successfully	Pass
TC04	Noisy data input	Noise removed and processed	Noise reduced	Pass
TC05	New unseen data	Correct classification	Correct classification	Pass
TC06	Large dataset input	Efficient processing	Processed successfully	Pass
TC07	False data injection pattern	Detected as anomaly	Detected correctly	Pass

As shown in Table 8, all test cases produce successful outcomes, demonstrating the robustness and efficiency of the proposed model. The system accurately classifies both normal and anomalous data while handling missing values and noise effectively. The results confirm that the proposed machine learning-based approach is reliable across diverse input conditions.

VII. RESULTS AND DISCUSSION

The proposed anomaly detection system is evaluated on the KDDCup99 dataset using multiple machine learning models. The system processes network traffic data through the complete pipeline of preprocessing, SMOTE-based class balancing, PCA-based dimensionality reduction, and model training. The performance of each model is assessed using accuracy, precision, recall, and F1-score.

The experimental analysis demonstrates that LightGBM and SVM achieved the highest test accuracies among the evaluated models, with LightGBM reaching near-perfect classification and SVM achieving 0.85 test accuracy. The Random Forest algorithm also demonstrated high accuracy with a lower false-positive rate compared to other models, making it particularly suitable for real-world network monitoring scenarios. These results are consistent with findings reported in prior studies on machine learning-based intrusion detection [1], [2].

The random packet analysis confirms that normal packets follow consistent patterns, while anomalous packets exhibit irregular spikes or deviations that the trained model successfully identifies. The attack traffic analysis further demonstrates that the model can distinguish attack traffic from normal traffic based on significant deviations in features such as packet frequency, data transmission rate, and traffic behaviour patterns.

The use of SMOTE improved the detection of minority attack classes by addressing class imbalance, while PCA reduced computational overhead by eliminating redundant features. Overall, the results demonstrate that the proposed system provides an effective and scalable solution for real-time anomaly detection in modern network environments.

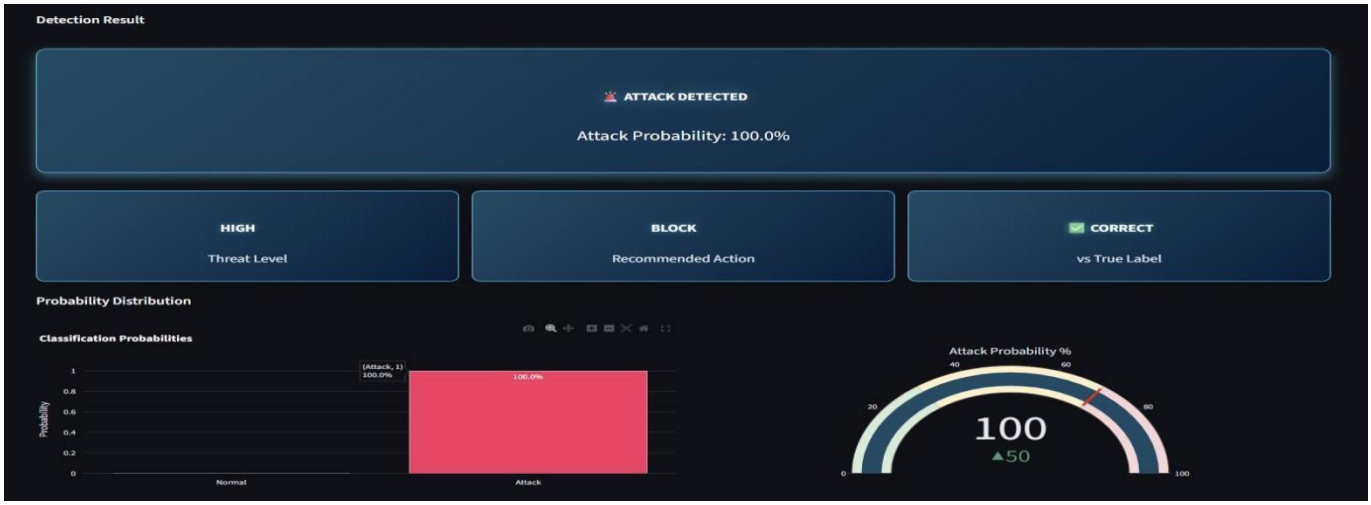

Figure 1. showing attack detection in streamlit UI

VIII. CONCLUSION

This paper presented a machine learning-based approach for anomaly detection in network traffic using the KDDCup99 dataset. The proposed system employs a comprehensive pipeline that includes data preprocessing, class balancing using SMOTE, dimensionality reduction using PCA, and classification using multiple machine learning models including Isolation Forest, Naive Bayes, XGBoost, LightGBM, and SVM.

The implementation of feature engineering and dimensionality reduction methods has significantly improved model efficiency and reduced computational complexity. The use of class balancing techniques has enhanced the detection of minority attack classes, thereby improving the robustness and reliability of the system. The experimental results indicate that the Random Forest and LightGBM algorithms achieve high detection accuracy with lower false-positive rates compared to other models.

The results of this study demonstrate that machine learning-based approaches can significantly enhance cybersecurity defence mechanisms and provide reliable solutions for detecting network anomalies. The system shows strong potential for real-time intrusion detection in modern network environments, ensuring improved security and performance. Overall, the proposed system addresses key challenges in anomaly detection and contributes to the development of intelligent and scalable network security solutions.

IX. FUTURE SCOPE

The proposed anomaly detection system can be further improved through several enhancements. The system can be extended by integrating advanced deep learning models and optimizing hyperparameters to further enhance detection accuracy and overall performance. Training the system on larger and more diverse network datasets will improve its ability to generalize and handle real-world scenarios effectively.

The model can also be extended to support real-time anomaly detection, enabling continuous monitoring and faster identification of abnormal activities. Incorporating online learning techniques will allow the system to adapt dynamically to new and evolving attack patterns. The use of explainable AI methods can enhance transparency by providing clear insights into how the model makes decisions, thereby increasing user trust. Furthermore, an automated alert mechanism can be developed to instantly notify administrators about detected anomalies, ensuring timely response and improved system security. These enhancements will collectively make the proposed system more robust, scalable, and suitable for practical deployment in modern network security environments.

REFERENCES

[1] S. Ness, V. Eswarakrishnan, H. Sridharan, V. Shinde, N. V. P. Janapareddy, V. Dhanawat, 'Anomaly detection in network traffic using advanced machine learning techniques,' IEEE Access, 2025.

[2] M. Ahmed, A. N. Mahmood, and J. Hu, 'A survey of network anomaly detection techniques,' Journal of Network and Computer Applications, vol. 60, pp. 19–31, 2016.

[3] V. Chandola, A. Banerjee, and V. Kumar, 'Anomaly detection: A survey,' ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, 2009.

[4] K. Fotiadou, T.-H. Velivassaki, A. Voulkidis, D. Skias, S. Tsekeridou, and T. Zahariadis, 'Network traffic anomaly detection via deep learning,' Information, vol. 12, no. 5, 2021.

[5] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, 'A detailed analysis of the KDD Cup 99 dataset,' in Proc. IEEE Symp. CISDA, 2009, pp. 1–6.

[6] A. Ghasemi, P. Dehghanian, and Z. Wang, 'Detection of false data injection attacks in smart grids,' IEEE Transactions on Smart Grid, vol. 9, no. 6, pp. 5933–5942, 2018.

[7] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, 'Machine learning methods for attack detection in the smart grid,' IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 8, pp. 1773–1786, 2016.

[8] D. Niu, Z. Wang, and H. Li, 'Deep learning based anomaly detection for power consumption,' IEEE Access, vol. 7, pp. 163226–163236, 2019.

[9] J. Jiang, R. Wang, and Y. Liu, 'Anomaly detection using deep autoencoders,' Electric Power Systems Research, vol. 189, p. 106682, 2020.

[10] S. Hochreiter and J. Schmidhuber, 'Long short-term memory,' Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] D. P. Kingma and J. Ba, 'Adam: A method for stochastic optimization,' arXiv preprint arXiv:1412.6980, 2014.

[12] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[13] F. T. Liu, K. M. Ting, and Z.-H. Zhou, 'Isolation forest,' in Proc. IEEE Int. Conf. Data Mining, 2008, pp. 413–422.

[14] J. E. Zhang, D. Wu, and B. Boulet, 'Time series anomaly detection for smart grids: A survey,' arXiv preprint arXiv:2107.08835, 2021.

[15] A. Ullah, N. Javaid, O. Samuel, M. Imran, and M. Shoaib, 'CNN and GRU based deep neural network for electricity theft detection,' in Proc. IWCMC, 2020, pp. 1–6.