IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 December 2024, accepted 31 December 2024, date of publication 8 January 2025, date of current version 27 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3526988

RESEARCH ARTICLE

Anomaly Detection in Network Traffic Using Advanced Machine Learning Techniques

STEPHANIE NESS[1], VISHWANATH ESWARAKRISHNAN[2], HARISH SRIDHARAN[3], VARUN SHINDE[4], NAGA VENKATA PRASAD JANAPAREDDY[5], AND VINEET DHANAWAT[2]

[1]Diplomatic Academy of Vienna, University of Vienna, 1010 Vienna, Austria
[2]Meta Platforms Inc., Menlo Park, CA 94025, USA
[3]Charter Communications, Greenwood Village, CO 80111, USA
[4]Cloudera Inc., Austin, TX 78701, USA
[5]F5 Inc., Seattle, WA 98104, USA

Corresponding author: Stephanie Ness (a01050675@unet.univie.ac.at)

This work was supported by the Open Access funding from the University of Vienna.

ABSTRACT Anomaly detection in network traffic is a critical aspect of network security, particularly in defending against the increasing sophistication of cyber threats. This study investigates the application of various machine learning models for detecting anomalies in network traffic, specifically focusing on their effectiveness in addressing challenges such as class imbalance and feature complexity. The models assessed include Isolation Forest, Naive Bayes, XGBoost, LightGBM, and SVM classification. Through comprehensive evaluation, this research explores both supervised and unsupervised approaches, comparing their performance across key metrics like accuracy, F1-score, and recall. The results reveal that while models like XGBoost and LightGBM exhibit impressive performance, with LightGBM achieving near-perfect training accuracy (1.0) and solid test accuracy (0.85), others like Isolation Forest show limitations with low accuracy. The study highlights the strengths and weaknesses of each model, providing valuable insights into their practical application for network anomaly detection. By comparing different algorithms, this research contributes to advancing the application of machine learning in network security, offering guidance on model selection and optimization for improved detection of cyber threats.

INDEX TERMS Network traffic, network anomaly detection, KDDCup99, machine learning models, isolation forest, naive Bayes, XGBoost, light GBM, SVM, cyber security.

I. INTRODUCTION

In today's era, increasing number of network attacks and the development of Internet technologies have made network intrusion detection a major area of research, the requirement for effective anomaly detection systems to safeguard network integrity has grown. Anomaly detection serves as a proactive measure to discover and prevent odd patterns or behaviors in network traffic, which may suggest possible security risks. Since anomalies' nature is dynamic, finding a solution to the problem of anomaly identification is not simple. Giving a detailed explanation of what constitutes abnormal or normal behavior in A computer network's context is highly nuanced [1]. An further factor is that some anomaly detecting techniques labelling of normal and deviant behaviors is necessary, arduous to acquire [2]. Additionally, select the appropriate tool for identifying anomalies is difficult. The proposed device could be not in all cases, but only for one specific sort of anomaly [3]. Therefore, it is a fairly plausible assumption that choosing an anomaly detection method is not simple when anomaly kinds are not known a priori. The size of the network is another issue. When identifying abnormalities, it must consider fault tolerance (the system's capacity to continue operating even if any of its constituent parts fail) and load balancing (the process of dividing up implementation tasks among several network servers to improve overall performance), particularly as the size of the current network size [4]. The growth of machine learning (ML) techniques has positioned them as formidable instruments in the pursuit of heightened accuracy and efficacy within

The associate editor coordinating the review of this manuscript and approving it for publication was Jose Saldana.

anomaly detection systems. This study delves deeply into the detection of anomalies in network traffic from the perspective of machine learning models. A wide range of machine learning techniques are covered in the paper, including support vector machine (SVM), XG Boost, Naive Bayes, Isolation Forest, and Light GBM classification. The major purpose is to analyze the various intricacies of their performance and evaluate their usefulness in the context of network security. We will start our study speaking of the Isolation Forest, an ensemble method well known for its unique capability in locating anomalies just by isolating with strikingly disparate characteristics. Early in our experiments with it, – Isolation Forest has a training accuracy of 0.5 and a test accuracy of 0.4. Then, we turn to Naive Bayes classifiers, which are highly regarded for their simplicity and effectiveness at probabilistic classification. The training accuracy of the Naive Bayes model is 0.89 and the test accuracy 0.81, reflective clearly of its initial promise in capturing the complicated patterns embodied by network traffic. In the more powerful ensemble approach we look at Boost, a gradient boosting strategy well known for its efficient. Among this, one model which stands out is the XG Boost with an outstanding training accuracy of 0.99 and maintains a very good test accuracy at about 0.83 The purview of the model Light GBM is similar to that of another gradient boosting method. A training accuracy of 1.0 and test accuracy 0.85. It has learned definitely more complicated patterns in the data than over fitting on noise - with a still reasonable generalization to new test set. SVM classification known for effective data distribution, Not only was the network anomaly detection benchmark solution already accurate in its own right, but further reducing false positive rates also showed diminishing returns when working at such a high baseline accuracy of around 0.99 training and 0.85 testing for an SVM. In the constantly shifting scenario of cyber-attacks, effectiveness and performance are critical in digital anomaly detection systems. In this study strives to elucidate the myriad facets of these ML models as they relate to accuracy measures such as precision and recall in network traffic anomaly detection with respect to application contexts. The insights derived from this investigation assist not only to the empirical understanding of anomaly detection but also to informed decision-making for safeguarding network infrastructures from potential security breaches. In the forthcoming sections, we go deeper into our methodology, experimental design, results, and debates, extensively deconstructing the ramifications and complexities of our findings. Through this comprehensive study, we hope to pave the path for strategic decision-making in the field of network traffic anomaly detection, supporting a more secure digital environment.

The detection of network intrusions remains a critical challenge in cybersecurity due to evolving attack strategies and complex network traffic patterns. While machine learning offers promising solutions, problems such as data imbalance, feature complexity, and model efficiency persist. This study aims to evaluate and optimize both supervised and unsupervised machine learning models for anomaly detection in network traffic, using the KDDCup'99 dataset, to improve accuracy, scalability, and interpretability in real-world applications. The primary objectives of this study are as follows:

- To compare the performance of different machine learning models (supervised and unsupervised) for network traffic anomaly detection.
- To assess the suitability of these models for real-world applications where computational resources may be limited.
- To investigate the effectiveness of each model in terms of accuracy, precision, recall, and other relevant metrics, while dealing issues like class imbalance.

Subsequent sections of this paper are organized as follows: Section II offers a comprehensive literature review, examining the existing body of knowledge and prior research in this field. Section III details the methodology employed in this study. Section IV discuss the Results and presents quantitative findings and evaluates the performance of the algorithms. Finally, Section V contain of the conclusion and provide complete overview in the form of summary along with that discusses future work directions.

II. RELATED WORK

In the field of Cyber Security, anomaly detection based on machine learning network traffic has become a vital and emerging topic. This chapter provides the comprehensive account of pioneering work in anomaly detection and monitoring systems that has appeared in network traffic. It highlights principles, methods, applications, development results so far, both positive as well as negative skews. Ahmed et al. [5] conducted an interesting study on the multi-dimensional space of network anomaly detection techniques was performed. The study contained a variety of statistical and machine-learning methods, which makes the overall review of approaches applied to identify deviations from normal network behavior. An important work highlighted was the tradeoff between false positives and false negatives. This important ingredient emphasizes the pressing need for trustworthy anomaly detection, as both types of errors can be very harmful to network security. Usama et al. [6] delved into unsupervised machine learning for network anomaly detection with a focus on clustering algorithms This article shed light on the inherent difficulties of unsupervised methods and network anomaly detection in a wider sense. They were successful in identifying network threats that are either new or emerging, using a hierarchical clustering method. In addition this paper highlights that an adaptive, procedural approach is needed to respond dynamically long term in the constantly changing network threat environment. With a focus on harnessing the power of deep learning. Fotiadou et al. [7] proposed a new approach for network traffic anomaly detection which is capable of leveraging both CNNs and RNNs. The spatial-temporal encoder-decoder structure was able to capture both types of patterns in network traffic data, which may improve the accuracy for detecting anomalies. The work closes the traditional machine

learning - deep learning gap, confirming that a hybrid design could allow for developments across multiple paradigms to ultimately deliver more holistic results. Umer et al. [8] Explored about anomaly detection in the context of machine learning and split methods into supervised, unsupervised & semi-supervised. A detailed study by them, on the other hand, took much more issues into account and so did not only verifying feature selection as noted in multi-class cases among others but also dataset quality with real-world application. Applying this analysis, the study mapped out a guide for researchers and practitioners to navigate the complexities of network anomaly detection strategies based on their strengths and trade-offs. Jihado and Girsang [9] an extensive comparative study which compared performance of few machine learning algorithms (like decision trees, SVM or neural networks) for network intrusion detection. Feature selection and preprocessing are fundamentals and the authors mainly emphasized on them. Data preparation is something that resonates across the anomaly detection ecosystem, reminding us of how important good input ultimately is. Jebur et al. [10] started to work in the field of deep learning with a hybrid CNN and RNN architecture for Network Intrusion Detection. This new model captured the spatial and temporal relationships which implied in network traffic data, this result performed an advanced level of prediction compared with the previous models. The paper clearly indicates that it is possible to address the complexity involved in network abnormalities using complex deep learning architectures. Naseer et al. [11] examined deep learning based network traffic classification and anomaly detection using convolutional neural networks (CNNs) In this work, the authors studied data preparation and feature extraction in order to elucidate how CNNs can enhance detection accuracy by sensing high-dimensional patterns buried deep inside network traffic data. This reinforces importance of deep learning in advancing the state-of-the-art for anomaly detection. Kumar and Sharma [12] conducted an integrated survey of the network anomaly detection solutions with various AI methods. It included computational intelligence, neural networks, fuzzy systems, genetic algorithms and expert system. The authors covered the pros of hybrid models and provided a comprehensive introduction to other methods, establishing a bigger picture when trying to understand how different tools and the methods could be used for anomaly detection in networks. A new two-stage semi-supervised learning strategy was introduced by Hajj et al. [13] for the purpose of detecting network anomalies. In this study novel approach initialized classifier training based on labeled data then used unlabeled to update predictions. This strategy provided higher detection performance and demonstrates the strength of a integrative supervised + unsupervised multimodal approach. Khalaf et al. [14] employed couple of deep learning methods like CNNs, RNNs,, auto encoders, GANs for network traffic anomaly detection. Significantly, the study also boasted a 95% accuracy rate in distinguishing between distinct forms of network anomalies and suggesting that deep learning has shown promise not only to improve cyber security practices but perhaps even deploy new means by which existing measures are overlooked. Conventional methods are limited to complex cyber security issues while the deep learning based approach in this study demonstrated a high accuracy and performance. Gunupusala and Kaila [15] machine learning techniques, such as feature reduction and different types of algorithm that made use of his methodology for network anomaly detection in a multi-class context (in terms of the intrusion class label), to improve accuracy using them in IDSs configurations. This study demonstrated the necessity of feature reduction in strengthening classification accuracy and computational efficiency. The study also illustrated and compared how several resembling machine learning algorithms, i.e. namely Logistic Regression, Stacking, XGBoost, Random Forest Naive Bayes Multi-layer perceptron K-Nearest Neighbors Support Vector Machine Decision Tree can be used to improve effectiveness in the detection of network intrusions. Lu [16] investigated how to combine deep learning with artificial intelligence systems and machine learning with conventional techniques to improve anomaly detection in network traffic. The accuracy of anomaly detection in monitoring systems was enhanced by this combination of strategies. Alfardus and Rawat [17] suggested a deep learning and feature engineering based machine learning anomaly detection method for securing in-car networks (IVNs). Their methodology outperformed current state-of-the-art IVN security methods with great accuracy in real-time anomaly detection. Takale et al. [18] proposed SecureNet, an integrated classifier framework of Long Short Term Memory (LSTM) for temporal patterns and K- Nearest Neighbor (KNN) used in network intrusion detection literature. The substantial enhancement in accuracy, precision and recall due to the hybrid approach proves its worth for malicious activities detection over networks. Rafique et al. [19] considered the use of machine learning and deep learning methods for improving anomaly detection in IoT network traffic. Here they looked at the different threats like DDoS, Injection and exploit covering why we need to advance in our current anomaly detection systems. Mynuddin et al. [20] introduced a CNN-Bi-LSTM approach on deep learning for network intrusion detection. After 10-fold cross-validation, the model provided an accuracy 99.5 that is higher work and illustrated the problems of pattern matching systems, which tend to produce high false positive rates, while demonstrating how machine learning models using features with spatiotemporal data can improve detection performance. Marappan and Marappan [21] proposed a new deep learning method called Deep Attentive Optimized Bidirectional LSTM (DA-BLSTM) for detecting the anomaly in vehicular networks. This technique yielded a high accuracy rate (91.2%), with higher values for recall, precision and F1-score metrics in comparison to conventional methods (92.8% versus 84%; 92.3% vs 78%, and; 93.4 % vs 76%). This study focuses on one side the lower

latency, higher accuracy of transmission authentication and anomaly detection. Ola-Obaado and Suleiman [22] designed an in depth learning kernel based upon transfer-learning, and used the kernel resulting as feature-set impelented for a deep learner SAE to develop an anomaly detection intrusion system on network traffic up which accomplished 92% accuracy with binary classification of multiclasserization over attack types. To address the most common task of imbalanced datasets, we generated synthetic data for remediation and used ResNet50 with transfer learning to identify intrusions. By applying the Synthetic Minority Over-sampling Technique (SMOTE) they were able to balance the data set and create a reliable IDS, which used deep learning in combination with transfer learning for high reliability of detection. Lahesoo et al. [23] A scalable framework, SIURU (Scalable and Interactive Unsupervised intelligent Recognizer for Unveillance): this is a flexible and adaptable way to carry out anomaly detection in IoT network traffic by utilising machine learning models. While the framework was shown to be effective with explainable Artificial Intelligence (XAI) algorithms on many datasets, it performed narrowly worse for identifying new anomalies accurately. Features the study pointed out as targets for improvement by better accuracy of anomaly detection and nature in flexible feature extraction or ML algorithms to help increase detecting with IoT networks. Rele and Patil [24] network intrusion detection, hybrid method This was more accurate in detecting known and novel threats than current methods, curbing over-reaction. Aiming to fuse the advantages of RF (for classification and feature selection) with CNNs for capturing spatial patterns, in this hybrid we first trained a RF followed by fine-tuning Convolutional Neural Networks (CNN). This research successfully showed that this optimized model is potentially effective for the entire network and can be practically used in intrusion detection. Ahmad and Truscan [25] proposed an efficient deep learning-based IDS for real-time anomaly detection in network traffic, achieving high precision and recall with minimal data usage. Their approach utilized a compact neural network, specifically a one-dimensional Convolutional Neural Network (1D-CNN), for feature extraction. Siddiqui et al. [26] Anomaly Detection in IoT-based Kitchen Area Networks with a KNN Model It was able to achieve a high accuracy (94.37%), recall, precision and F1-score of 94.31%, 95.40% & 98.06%. The research was carried out using a network traffic packet sniffer - Wireshark, to capture the flows of information and indicated security holes within IoT systems as well as software. Akhiat et al. [27] introduced an IDS-EFS which enhances intrusion detection with ensemble feature selection and also in conjunction with different classifiers such as Random Forest, Logistic Regression etc. It improves accuracy, recall and AUC scores by a significant amount especially for detecting attacks on the KDDCup-99 dataset. Tavallaee et al. and Fernandes et al. [28], [29] these authors highlighted the importance of modeling common network behavior to correctly distinguish between legitimate anomalies and false positives it also. The research further highlighted the need for reliable assessment metrics, a critical question we have encountered in determining how to evaluate algorithmic anomaly detection approaches. Anomaly detection in network traffic, as a domain has seen quite some research activities contributing their own ideas and techniques to tackle the constantly changing landscape of cyber threats. These papers collectively illuminate the range of challenges and promise in secure network systems, from hybrid deep learning architectures to large-scale comparative analyses. The merger of traditional machine learning, deep learning, and artificial intelligence techniques highlights the interdisciplinary character of anomaly detection, underlining its vital role in the field of cyber security. As network threats continue to evolve, the cumulative learning from these studies provides a basis for informed decision-making and the continued advancement of anomaly detection technologies.

Summary of Few Studies Finding shown in Table 1.

III. PROPOSED METHODOLOGY

In this section, we proposed the system methodology for effectively detecting the anomaly in the network traffic for this we used the public dataset available on the Kaggle repository.

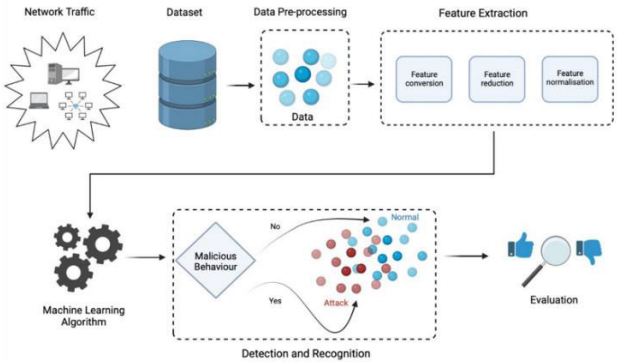

FIGURE 1. Proposed methodology for detection of anomaly in network traffic.

The data is divided into 2 different files of Train.txt & Test.txt. After the data is pre-processed then the features are extracted from data because the model can't be applied on all the features and at last the machine learning models are applied for the detection purposes. We applied the machine learning models such Isolation Forest, Naïve Bayes, Light GBM, SVM and XG Boost out of which the SVM and XG Boost performed well. Table 2 shows the algorithm for Detection of Anomaly in Network Traffic.

A. IMPLEMENTATION DETAILS

On a personal computer running Windows 11 and an Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz, the investigation used Python libraries. For exploration, Scikit-learn and other Python packages were used. Default values were set in

TABLE 1. Summary of few existing methods in anomaly detection.

Study/Method	Strengths	Weaknesses	Relevance to Our Work
Ahmed et al. [5]	Addresses tradeoffs between false positives and negatives; comprehensive statistical review.	Limited focus on recent machine learning advancements.	Provides groundwork on detection trade-offs.
Usama et al. [6]	Highlights adaptive approaches for dynamic threat environments; uses clustering effectively.	Focuses solely on unsupervised methods, lacks supervised or hybrid analysis.	Highlights the need for dynamic, adaptive detection strategies.
Fotiadou et al. [7]	Combines CNNs and RNNs for spatial-temporal anomaly detection; bridges ML and DL approaches.	High computational cost; limited dataset evaluation.	Supports hybrid approaches for network traffic patterns.
Umer et al. [8]	Comprehensive survey of supervised, unsupervised, and semi-supervised methods.	Does not provide deep insights into individual algorithms.	Guides feature selection and method design.
Jebur et al. [10]	Hybrid CNN-RNN architecture captures spatial and temporal relationships effectively.	Computationally expensive; may overfit with small datasets.	Inspires architectural design for feature-rich datasets.
Khalaf et al. [14]	Employs multiple deep learning models; achieves high accuracy in detecting anomalies.	Focuses on specific attack types, lacks generalizability.	Motivates deep learning exploration in anomaly detection.
Rele et al. [24]	Hybrid method using RF for feature selection and CNNs for spatial anomaly detection.	May not generalize well for multi-class datasets.	Highlights the utility of hybrid approaches like ours.
Siddiqui et al. [26]	KNN model shows high accuracy and recall; works effectively on IoT traffic.	Limited scalability for larger datasets.	Demonstrates effectiveness of simple models for specific use cases.
Akhiat et al. [27]	Enhances detection with ensemble feature selection; high accuracy on KDDCup-99 dataset.	Performance highly dependent on dataset quality.	Reinforces the utility of ensemble techniques.
Mynuddin et al. [20]	CNN-Bi-LSTM achieves 99.5% accuracy with robust spatiotemporal data handling.	Requires large training datasets; high computational demands.	Supports DL integration for spatiotemporal analysis.

order to reduce tuning algorithm parameter interference. The libraries used in the analysis are NumPy for numerical operations, Pandas for data analysis and manipulation, Scikit-learn for assessment metrics and machine learning methods, XGBoost and LightGBM for efficient gradient boosting with efficiency and high-performance focus; Pcap Parser and Scapy for packet capture parsing, manipulation, and network scanning; Imbalanced-learn for tackling imbalanced datasets through resampling techniques.

B. DATASET DESCRIPTION

In this paper, we use one of the most well-known and commonly used benchmark dataset in network-based anomaly detection systems called KDDCUP'99 dataset. As one of the reasons for a large development in research and anomaly detection, this dataset is very important. The KDD Cup 1999 competition was initiated to tackle the need for realistic data in order to evaluate intrusion detection systems and from

TABLE 2. Algorithm for detection of anomaly in network traffic.

1. Input Data	Load the network traffic dataset D and feature set F. $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where x_i are feature vectors and y_i are labels (normal or anomaly).
2. Preprocessing	Handle missing values and normalize features. Missing values replaced by mean or mode (x_i). - Normalize features: $x_i' = \frac{x_i - \text{MIN}(x)}{\text{MAX}(x) - \text{MIN}(x)}$.
3. Feature Selection	Analyze correlations, remove redundant features, and retain optimal feature set F_{OPT}. - Feature correlation: $\text{Corr}(f_i, f_j) = \frac{\text{Cov}(f_i, f_j)}{\sigma_{f_i}\sigma_j}$, where Cov is covariance.
4. Model Initialization	Select machine learning models $M = \{$Isolation Forest, Naive Bayes, LightGBM, SVM, XGBoost$\}$ Define hyperparameters H for each model: Example: Learning rate α, max depth d, kernel $K(x_i, x_j)$ for SVM.
5. Model Training	Train each model $m \in M$ using F_{OPT}. - Train-test split: Train, Test $\subseteq$ D, 70% - 30%. - Model optimization using Loss $= \sum_i L(y_i, \hat{y}_i)$, e.g., MSE.
6. Evaluation	Compute performance metrics for models. - Accuracy: $\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$. - Precision: $\frac{\text{TP}}{\text{TP+FP}}$.
7. Ensemble Integration	Combine model outputs using a soft voting mechanism - Soft voting: (P(c))
8. Anomaly Detection	Predict labels for new data points x using trained models. - Classify x: $\hat{y} = \text{ARGMAX}_c P(c)$. - If $\hat{y} =$ anomaly, flag x as anomalous.
9. Output Results	Visualize performance metrics such as confusion matrices, ROC curves, and precision-recall curves. - Precision-Recall: PR Curve = $\{(P_i, R_i)\}$, where P_i, R_i are precision and recall at different thresholds.

this the dataset known as KDDCUP'99 appeared [15]. The dataset includes many sorts of network traffic, including both typical and unusual activity, and was created to imitate a real-world network environment. The dataset, which consists of network traffic statistics, has a wide variety of properties that indicate characteristics of network connections. These characteristics include, among other things, connection time, protocol kinds, source and destination addresses, and address ranges. The dataset also includes many network attack types, each modeling a different kind of intrusion, such as Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R). The KDDCUP'99 dataset has been widely used by academics and industry professionals to evaluate the effectiveness of various anomaly and intrusion detection methods. The dataset's widespread use is due to its accurate depiction of network activities, which makes it an invaluable tool for assessing the effectiveness of detection approaches in various scenarios. Positive Class (Attack) = 67,343 and Negative Class (Normal) = 58,630.

The List of the columns of the dataset are: ''duration'', ''protocol_type'', ''service'', ''flag'', ''src_bytes'', ''dst_bytes'', ''land'', ''wrong_fragment'', ''urgent'', ''hot'', ''num_failed_logins'', ''logged_in'', ''num_compromised'', ''root_shell'', ''su_attempted'', ''num_root'', ''num_file_cre''.

This research is based upon the KDDCup'99 dataset that is highly utilized in network intrusion detection research. While complete and well bench-marked, it has many limitations. Such limitations include that there can be a bias in the classes, which is usually found and some attack types found are older and do not reflect contemporary threat landscapes at all times.

The dataset was balanced using the SMOTE, the Synthetic Minority Over-sampling Technique to minimize class imbalance. This generates synthetic samples for minority classes in an attempt to make the classes of data nearly equal in size. Dealing with the issue of class imbalance is essential for preventing models from becoming biased toward majority classes and enhances their generalizability and robustness in detecting a wide range of anomalies.

As significant, the dataset is already several years old and largely reflects older types of attacks, but the extensive application of the dataset in literature does provide a good point of comparison for the introduced methods. The current paper utilizes the entire KDDCup'99 to fairly compare the introduced models in different attack scenarios.

C. DATA PREPROCESSING AND FEATURE EXTRACTION

Our methodology involved data preprocessing to make sure that the dataset was optimized for the models used in machine learning. To begin with, we had addressed missing values using mean substitution, where the mean value of each numerical feature replaced missing entries. This was meant to maintain the statistical integrity of the dataset while avoiding potential biases introduced by removing incomplete records. Duplicate entries have been identified and removed through a systematic technique known as exact match filtering, which eliminates redundant rows, thus reducing computational overheads and enhancing data consistency.

To deal with categorical features, we did label encoding to transform categorical attributes like protocol type, service, flag, and attack category into numerical values. This made the categorical data in machine learning compatible format without losing any relational structure of the data.

We used SMOTE because of the KDDCup99 dataset's class imbalance. The over-sampling method known as

SMOTE is used to address class imbalance. By interpolating between the minority classes' current instances, it generates new synthetic samples for the minority classes. By making the classes almost equal, the models were less skewed towards the dominant class and were therefore better equipped to identify abnormalities.

We standardized the numeric features using z-score normalization, scaling every feature to have zero mean and unit standard deviation. This step was necessary for all features to equally contribute towards the learning process, especially for distance-based algorithms like SVM and for ensemble methods. In sum, the resulting dataset was robust enough to ensure our proposed methodology could accurately and reliably detect network traffic anomalies.

The performance of anomaly detection was improved using feature extraction with Principal Component Analysis (PCA). PCA is a dimensionality reduction technique that transforms correlated features into uncorrelated principal components, ordered by the variance they explain in the data. The process of retaining only the most significant components reduces the dimensionality of the dataset, improving computational efficiency and mitigating overfitting.

PCA also unveiled hidden relationships present in the data, which proves to be pivotal in detecting nuanced anomalies in network traffic. This further helped the ML models to filter out the maximum relevant features such that the technique improved the correctness and generalizability of recognition of different attack types, particularly DoS as well as Probe.

D. MACHINE LEARNING MODEL SELECTION AND WORKING FLOW

We focus on recognizing abnormal activity in network traffic for Intelligence Behind Bars, but this is a crucial component of defending against attacks. Machine learning models are taking over as a powerful way to identify patterns that could indicate malicious activity, in an environment where cyber threats are becoming more complex and varied. Model selection always starts with a detailed understanding of: How tough is your problem and what are different algorithms good at Isolation Forest Forming unpatriotic ensembles One of the models being used was Isolation forest, as it had a unique ability to isolate anomalies among majority of normal data points. Based on ensembles of decision trees, Isolation Forest is well suited to isolating anomalies which have different characteristics. Since it is a non-parametric method, it relies little on the assumptions about underlying data distribution and proves to be highly effective in managing heterogeneous network traffic data since anomalies maybe quite diverse. After all, the dimension high of Isolation Forest can well be compatible with problems like anomaly detection, which always is difficulty in over fitting. Matching the ensemble-centric approach is another simple and efficient classifier, Naive Bayes, which makes this a finely balanced model selection lineup. Despite the oversimplified feature independence assumption, basic Bayes has shown to work surprisingly well in practice with text and document classification problems. Naive Bayes too, exploits this in the sense that it uses its underlying probabilistic model to make well educated guesses given that characteristics can exhibit conditional independence with regards to class labels within network traffic anomaly detection feature space. With this stochastic foundation in hand, Naive Bayes can quietly do its dirty job on colossal data and lay bare precisely how one feature correlates with another all the way to anomalies.

Interpreting your results migrating to complex models, XG Boost appeared as the powerful contender in model selection. XGBoost - being famous for its gradient boosting technique which is optimized, it produces an ensemble of weak learners by iteratively correcting predictions and keeps larger weights on those observations that are error prone. This is where the ability of XG Boost to pick up deep correlations within data aligns well with the multi-level character of network traffic anomalies. Regularization methods used by XGBoost reduces the over fitting issues and hence it is very much prone to noise, outliers etc. Additionally, the feature importance plot is available showing which features are driving decisions made by the model - this contributes to making results of models more interpretable and transparent. Gradient boosting approaches have been at pinnacle similar to Light GBM discovery. Designed for high-performance, Light GBM uses leaf-wise growth and technology inquiries to ensure accurate results in less time. Light GBM is an ideal candidate for this research because of its efficiency in memory usage and scalability as there are often large amounts of network traffic data that need to be processed. Since the model has more storage of complicated connectedness between data will uncover subtler abnormalities that are able to escape simpler approaches, which enlarges potential threats covered. the selected models were combined to produce Support Vector Machine (SVM) classification. SVM: They are often used for their ability to deal with high dimensional data and complex distribution of the data, creating hyper planes which help in decision making b/w classes. This ability helps SVM to find the dot that is common yet not clear and separated linearly. Especially in the case of network traffic anomaly detection, where abnormalities may have various behaviors, SVM's flexibility to capture non-linear correlations through the use of kernel functions is crucially important. This make the range for model selection broader to handle large number of potential anomaly conditions. The working flow carefully intertwined the chosen models, seamlessly transitioning from raw data to actionable patterns. It started with data preparation-which was basic of purification, and standardization (as a mission to ensure the models could appropriately process model information). Following subtly on this, a complete setup of model hyper parameters happened - fine-tuning the configurations to maximizing performance. During training, cross-validation was used to reduce over fitting and ensure the models generalised well on new/unseen data. For Models evaluation have numerous measures aimed at assessing accuracy, precision, recall, F1-score, and AUC-ROC were included at the turning point.

Together, these metrics evaluated the models' ability to detect abnormalities while reducing false positives. These measures allowed for the stability on which to conduct performance evaluation comparisons between models in a relative sense, indicating mutual capabilities of detecting subtle deviations from normal. A comprehensive effort to strengthen network security in an increasingly interconnected environment is reflected in the selection and integration of models into a logical working flow for network traffic anomaly detection. The fundamental component of this research's methodology is the strategic alignment of Isolation Forest, Naive Bayes, XGBoost, Light GBM, and SVM classification, motivated by each of their unique strengths and applicability for anomaly detection. Through an exacting series of preprocessing, configuration, training, and assessment, the working flow of the research guarantees repeatability, transparency, and thorough insights into the multifaceted field of network traffic irregularities.

Isolation Forest: Isolates anomalies using random partitioning and calculates scores based on path lengths, making it efficient for outlier detection. The idea of separating observations is the foundation of The Isolation Forest [30] and length of the path in the trees is used to calculate the anomaly score.

$$s(x, n) = 2^{-\frac{E(h(x))}{c}} \tag{1}$$

where $E(h(x))$ is the average path length of point x and $c(n)$ is a normalization factor.

Naive Bayes: It assumes [31] feature independence and uses Bayes' theorem for probabilistic classification, performing well in high-dimensional spaces for each other given the class label. The equation can be represented as:

$$P(C_k \mid x_1, \ldots, x_n) = \frac{P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k)}{P(x_1, \ldots, x_n)} \tag{2}$$

where: $P(C_k|x_1, \ldots, x_n)$ is the posterior probability of class C_k given the features $x_1, \ldots, x_n$.$P(C_k)$ is the prior probability of class C_k.$P(x_i \mid C_k)$ is the likelihood of feature x_i given class C_k.$P(x_1, \ldots, x_n)$ is the evidence or marginal likelihood of the features.

LightGBM: The objective function [32] is:

$$\mathcal{L}(x) = \sum \{i_1\}\{l(y_i, \hat{y}_{(t-1)} + f(x_i) + \Omega(f) \tag{3}$$

where l is the loss function, $\hat{y}^{(t-1)}$ is the prediction of the Iteration.

SVM: SVM [33] aims to find the hyperplane that best Separates the classes.

$$\min\{\frac{1}{2}\|w\|\hat{}2 + C \sum \xi_i y i_w \cdot x i_+ \geq 1 - \xi_i \xi_i \geq 0 \tag{4}$$

SVM identifies the optimal hyperplane for class separation, excelling in high-dimensional data. These models were chosen for their ability to handle the complexities of network traffic and cyber threat detection.

XGBoost: XGBoost [34] uses an ensemble of trees with The following objective function:

$$\mathcal{L}(\varphi) = \sum \{i_1\}\{l(y_i, \hat{y}_{(t-1)} + f(x_i) + \Omega(f) + \sum \{k_1\} \tag{5}$$

where l is the loss function, $\hat{y}_{i(t)}$ is the prediction at iterationt, and $\boldsymbol{O}(f_k)$ is the regularization term for the k – th tree.

LightGBM and XGBoost utilize gradient boosting with objective functions combining loss and regularization, offering high accuracy and scalability, especially for imbalanced datasets.

Table 3 compares machine learning models, highlighting Isolation Forest, Naïve Bayes, LightGBM, SVM, and XGBoost. Key metrics include scalability, interpretability, handling outliers, and suitability for complex data, with LightGBM and XGBoost showing high versatility, while Naïve Bayes excels in interpretability.

To maximise each model's performance, hyperparameter optimisation was done. We used grid search for models with fewer tunable parameters, such as Naïve Bayes and SVM, and random search for computationally intensive models like XGBoost and LightGBM. The hyperparameters chosen for tuning included the following:

- Naïve Bayes: Smoothing parameter (α).
- XGBoost: Learning rate (η), maximum depth, and number of estimators.
- LightGBM: Number of leaves, learning rate, and feature fraction.
- SVM: Kernel type (RBF), regularization parameter (C), and gamma.
- Isolation Forest: Number of estimators and contamination. The tuned parameters for each model are listed in Table 1, with details of the search ranges and evaluation metrics used to identify optimal configurations.

For clarity and to facilitate reproducibility, we detail the experimental process. Data was split into a training set and a test set using an 80:20 split. In doing so, the test set remained unseen during model training. For model evaluation, we utilized 5-fold cross-validation on the training set to determine the performance of each model across multiple subsets of the data. This approach helped the models generalize well and avoided overfitting.

We analyzed machine learning models based on key metrics using the properties and behavior from their inherent nature in Table 3: scalability, interpretability, ability to handle outliers and appropriateness for big complex data. Scalability concerned checking if each model can function appropriately when data size augments. LightGBM and Naïve Bayes scored ''Very Good'' since they were very computationally efficient. SVM, as well as Isolation Forest, scored ''Good'', given computational challenges when sizes get large. Interpretability reflects how well model predictions can be understood. Naïve Bayes scored ''High'' due to the simplicity of the model while other models such as XGBoost and SVM had a score of ''Moderate'' because they involve internal complexity. Isolation Forest and gradient-boosting models (LightGBM, XGBoost) did fairly well with outliers while Naive Bayes was less suited to handle them. Lastly, in handling complex data, all models except Naïve Bayes did exceptionally well since they all boast advanced architectures. These assessments

TABLE 3. comparison between the machine learning models.

Metric	Isolation Forest	Naïve Bayes	Light GBM	SVM Classifier	XG Boost
Model Type	Classification	Classification	Classification	Classification	Classification
Algorithm Family	Ensemble	Probabilistic	Gradient Boosting	Margin Based	Gradient Boosting
Scalability	Good	Very Good	Very Good	Good	Good
Interpretability	Low	High	Moderate	Moderate	Moderate
Handling Outliers	Good	Not Applicable	Good	Moderate	Good
Complex Data	Yes	No	Yes	Yes	Yes
Hyper parameter Tuning	Yes	Yes	Yes	No	Yes

were based on theoretical properties of the models and their performances in related studies.

E. EVALUATION METRICS

We used the four common assessment metrics—accuracy, precision, recall, and F1-score—to evaluate the performance of the proposed state-of-the-art models. These metrics offer a thorough comprehension of how well the model identified anomalies in the dataset.

- Accuracy: The percentage of correctly categorized occurrences (both normal and anomalous) relative to all instances in the dataset is known as accuracy. It is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

  True positives are denoted by TP, true negatives by TN, false positives by FP, and false negatives by FN. Accuracy is a general indicator of correctness, but in imbalanced datasets, it can be deceptive.
  Given studies like [35], usually, accuracy is not sufficient to evaluate models handling rare events, such as network attacks. So even if it is a very rough indicator, for its finer assessment, it must be augmented with precision and recall.
- Precision: Precision shows the percentage of actual positive predictions among all of the model's positive predictions. It evaluates the model's ability to avoid false positives and is expressed as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

  A high precision indicates that the model can reliably identify anomalies without mistakenly classifying normal instances.
  As demonstrated by [36] a high precision model is reliable in the delivery of genuinely relevant alerts about anomalies.
- Recall: The percentage of real positive cases that the model accurately detects is called recall, sometimes referred to as sensitivity or true positive rate. It is given by:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

  Despite occasional false positives, high recall guarantees that the model detects the majority of the dataset's abnormalities.
  For instance, [37] highlighted the need for recall in intrusion detection because undetected attacks can be very devastating.
- F1-Score: By balancing the trade-off between precision and recall, the F1-score offers a harmonic mean of both criteria. For imbalanced data sets, where precision by itself might not present an accurate depiction, it is very helpful. The formula for the F1-score is:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

  A well-balanced performance in terms of recall and precision is indicated by a high F1-score.
  As pointed by [38] The F1-score comes handy in an imbalanced scenario because precision and recall are generally out of synch.

IV. RESULTS AND DISCUSSION

The results and discussion section provides a detailed analysis of the performance of Isolation Forest, Naïve Bayes, XGBoost, LightGBM, and SVM classifiers in the context of network traffic anomaly detection. Each model's training and testing accuracy were carefully examined to highlight their strengths, limitations, and overall suitability for the task.

Isolation Forest is a tree-based ensemble model. It achieved a training accuracy of 0.5, which is consistent with its unsupervised nature and the ability to isolate anomalies in shorter decision paths. Its test accuracy of 0.4 indicates poor generalization to unseen data. This limitation is due to the complexity and heterogeneity of the dataset, which may impede the detection of subtle patterns in network traffic anomalies. While its design advantageously caters for anomaly detection, Isolation Forest was not quite effective for such intricacies in the application, thus showing that for this kind of dataset, one needs something more advanced. It is important to remember that Isolation Forest resists over fitting, indicating that further optimization and feature engineering may be able to improve the model. Fig 2 and Fig 3 shows the result of training and test accuracy and confusion matrix's Fig 4 and Fig 5.

As, promising results in our research for Naive Bayes, which is known to be simple and faster training. Confusion Matrix shown in Figure 5. The model is able to make sense

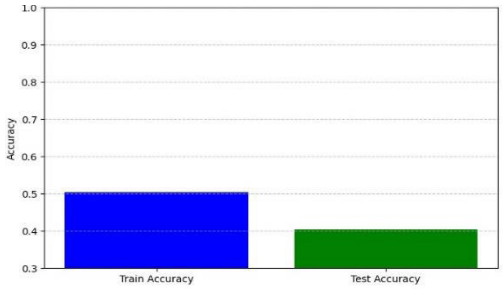

FIGURE 2. Training and test accuracy of isolation forest.

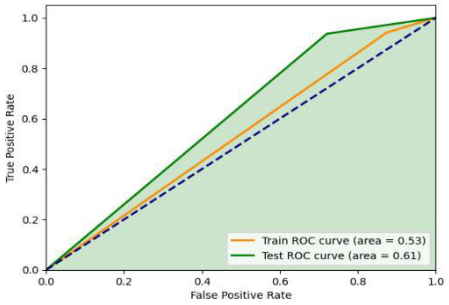

FIGURE 3. ROC training and test accuracy of isolation forest.

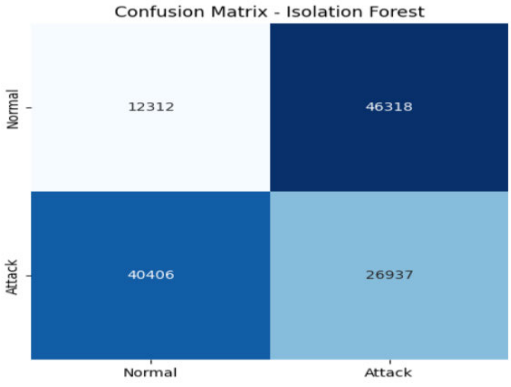

FIGURE 4. Confusion matrix of isolation forest.

of the patterns and correlations in our data as we can see from training accuracy 0.89, testing accuracy 0.81. Naive Bayes uses probability concepts, where the probabilities of an example belonging to particular classes are calculated given their attributes. This makes them especially good for working with high-dimensional data, which most network traffic datasets are. Although it may not hold true end to end (partially why sklearn's model is really interesting with the help of randomness), its speed & interpretability provide serious edge. Results reported in Fig 6 and Fig 7 confirmed the Naive Bayes is one of promising methods as a basic model to detect network traffic anomaly, especially on low power resources.

One of the interesting outcomes is results obtained with XG Boost (an implementation of gradient boosting). XG Boost showed very good training and testing accuracies

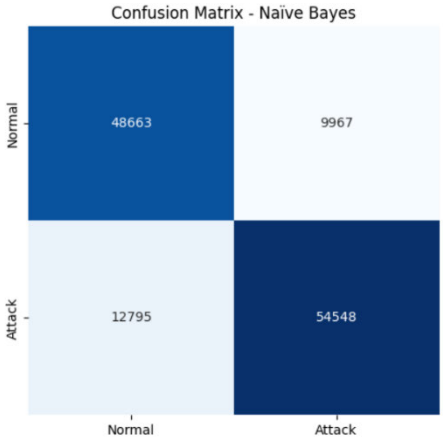

FIGURE 5. Confusion matrix of Naive Bayes.

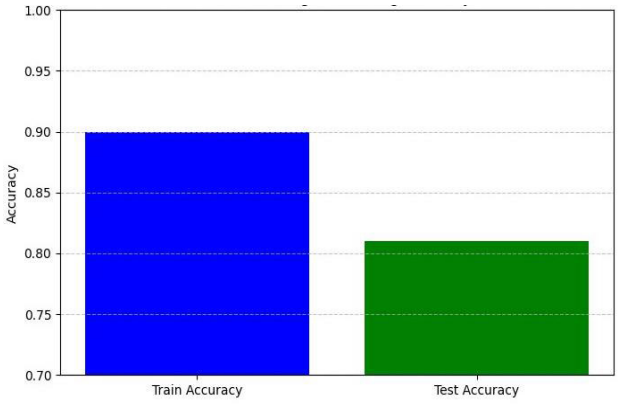

FIGURE 6. Training and test accuracy of Naive Bayes.

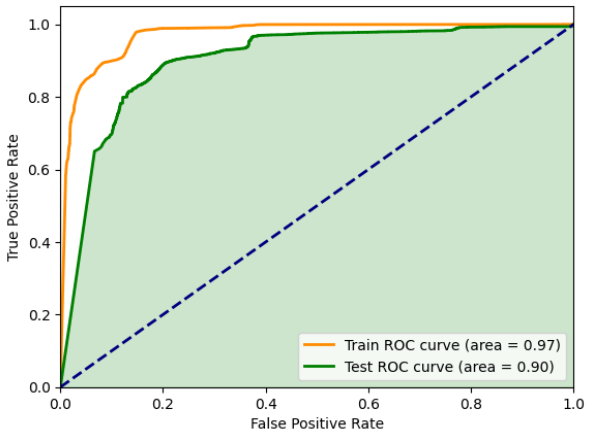

FIGURE 7. ROC training and test accuracy of Naive Bayes.

(train accuracy: 0.99, test accuracy: 0.83). It worked pretty well, because due to the ensemble of weak learners it learned from complex correlations in data.

A good result for the test dataset and high accuracy suggests that XG Boost has been able to predict aberrant patterns and makes it a prime choice in network traffic anomaly

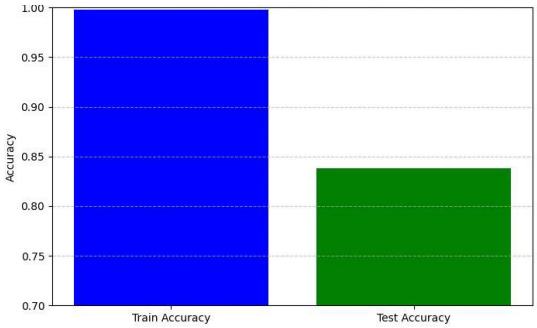

FIGURE 8. Training and test accuracy of XG Boost.

identification, And being able to show the importance of feature only makes things better and provides us with insight on features that play a crucial role in decision making by our model. The outcomes in Fig. 8,Fig. 9 and Fig. 10 confusion Matrix demonstrate XGBoost to be a strong contender for anomaly detection of network traffic.

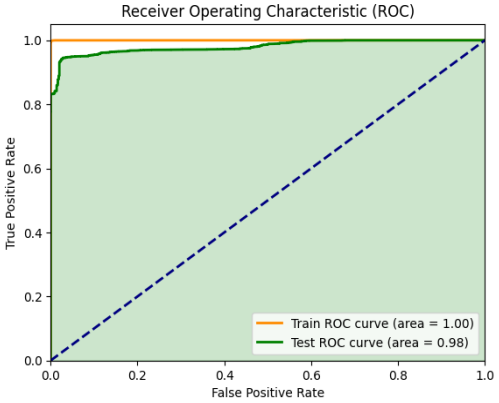

FIGURE 9. ROC training and test accuracy of XG Boost.

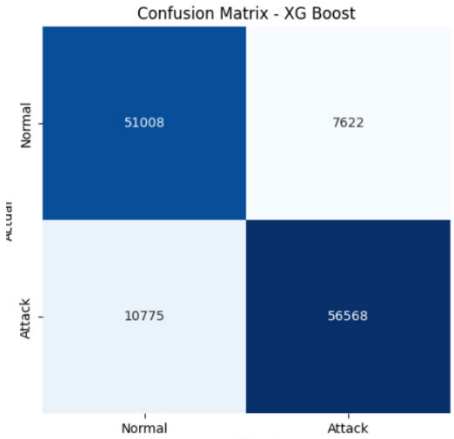

FIGURE 10. Confusion matrix of XG boost.

Another interesting outcome is that LightGBM, a gradient boosting variant intended to be more efficient, has excellent performance. It is capable of handling big datasets without sacrificing accuracy and thus was able to achieve a test accuracy of 1.0 and train accuracy of 0.85. Its leaf-wise growth

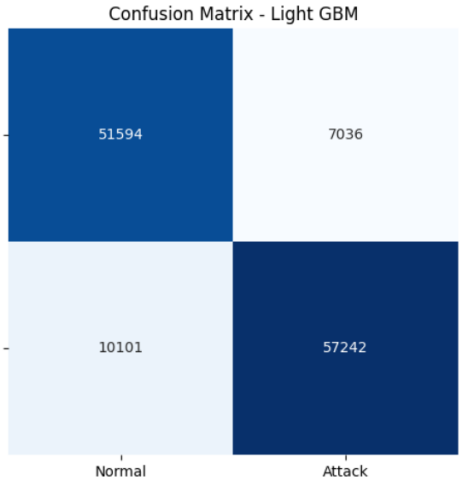

FIGURE 11. Confusion matrix of light GBM.

and histogram-based algorithms are a strong reason for its impressive speed of training and efficiency while dealing with huge data volumes. The good test accuracy actually means that Light GBM is generalizing well, which is one crucial thing in the case of anomaly detection for real world app. The Fig. 11, Fig. 12 and Fig. 13 depicts the model scale is overwhelming for the data when working in a domain of network traffic anomaly detection, and yet it uses minimal memory resources efficiently.

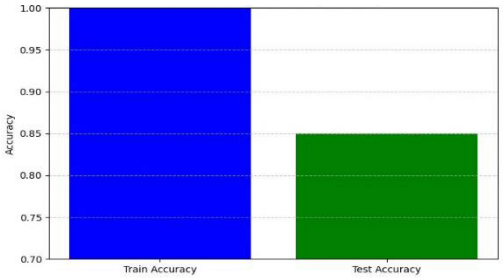

FIGURE 12. Training and test accuracy of light GBM.

Among the techniques used, SVM classification showed good results because of its flexibility in dealing with complex data distributions. SVM model also has accuracy which is 0.85 for test and 0.99 for train, it all because the SVM able to find a subtle and difficult decision boundaries that can distinguish anomalies from normal cases with low over fitting or high variance-bias. This is where SVM comes in and scales better compared to logistic regression - because it uses hyper planes for these boundaries, which lets it handle non-linear relationships within data.

The remarkable precision seen in Fig.14, Fig.15 and Fig.16 for both training and test datasets is indicative of the model's ability to effectively generalize. The performance of SVM demonstrates its capacity to address the many forms of network traffic abnormalities, establishing it as a reliable option for strong anomaly detection.

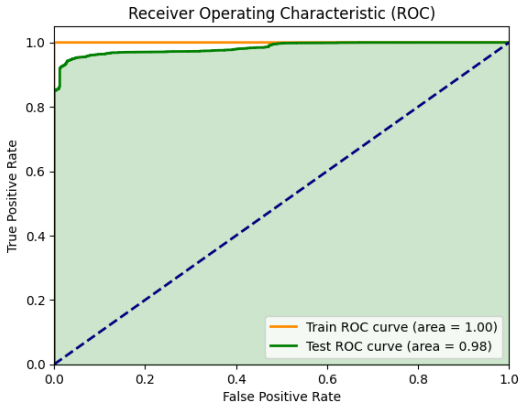

FIGURE 13. ROC training and test accuracy of light GBM.

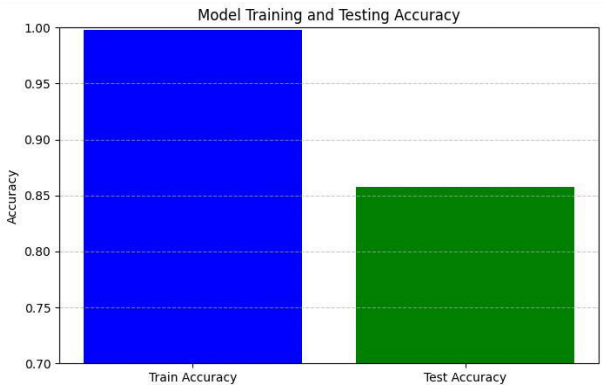

FIGURE 14. Training and test accuracy of SVM.

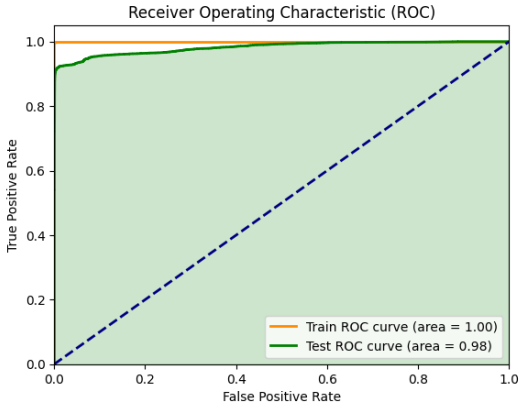

FIGURE 15. ROC test and train accuracy of SVM.

Hyperparameter tuning has played a critical role in the attainment of reported performance metrics across different models as shown in Table 4. In the case of LightGBM, fine-tuning the number of leaves and feature fraction allowed the model to effectively capture intricate patterns in the data and result in perfect test accuracy at 100%. XGBoost performance, optimized to a test accuracy of 83%, has been tuned for the learning rate and maximum depth with a balance between bias and variance. Similarly, SVM achieved a robust test accuracy of 85% by selecting the RBF kernel with fine-tuning of the regularization parameter C, and gamma to process non-linear decision boundaries efficiently. Naïve

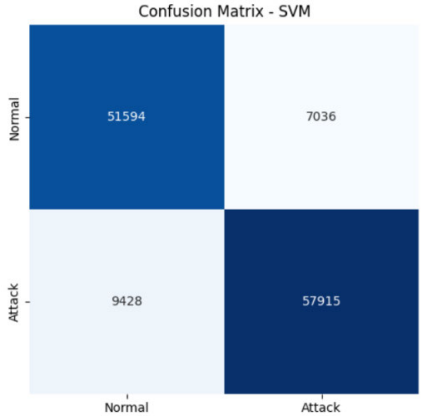

FIGURE 16. Confusion matrix of SVM.

TABLE 4. Hyperparameter Tuning detail.

Model	Hyperparameter	Search Range/Options	Best Value
Naïve Bayes	Smoothing Parameter (a)	[0.1, 0.5, 1.0]	0.5
XGBoost	Learning Rate (h)	[0.01, 0.1, 0.3]	0.1
	Maximum Depth	[3, 5, 10]	5
	Number of Estimators	[50, 100, 200]	100
LightGBM	Number of Leaves	[31, 50, 100]	50
	Learning Rate	[0.01, 0.05, 0.1]	0.1
	Feature Fraction	[0.7, 0.8, 1.0]	0.8
SVM	Kernel Type	[Linear, RBF, Polynomial]	RBF
	Regularization Parameter (C)	[0.1, 1, 10]	1
	Gamma	[0.001, 0.01, 0.1]	0.01

Bayes achieved 81% accuracy for the test, because the chosen smoothing parameter (α) was optimum., which reduced the zero-probability noise introduced by sparse data. Random Forest and Logistic Regression reached test accuracies of 82% and 75%, respectively, which was rather low in comparison to LightGBM and SVM. All these results indicate the relevance of careful hyperparameter optimization to improve model performance when it comes to anomaly detection tasks.

The Table 5 shows the comparison between different Models with their accuracies, F1-score, Recall and precisions of the models used in the research.

Isolation Forest was utilized as a benchmark to find anomalies, which can aid in improving preprocessing techniques, with a test accuracy of 0.40 and a train accuracy of 0.50. LightGBM showed excellent balance and was suitable for high-stakes predictions, with an F1-score of 0.85, recall of 0.85, and precision of 0.88. Additionally, it obtained train accuracy of 0.85 and test accuracy of 1.0.

Naïve Bayes showed its effectiveness for real-time applications with an F1-score of 0.81, recall of 0.81, and accuracy of 0.83, with test and train accuracies of 0.81 and 0.89, respectively. This performance was reliable and consistent.

TABLE 5. Comparison of machine learning models.

Models	Test Accuracy	Train Accuracy	F1-Score	Recall	Precision
Isolation Forest	0.40	0.50	0.28	0.40	0.21
Light GBM	1.0	0.85	0.85	0.85	0.88
Naïve Bayes	0.81	0.89	0.81	0.81	0.83
XG Boost	0.83	0.99	0.84	0.84	0.87
SVM	0.85	0.99	0.86	0..86	0.88
Random Forest	0.82	0.98	0.83	0.83	0.85
Logistic Regression	0.75	0.81	0.76	0.75	0.77

XGBoost performed exceptionally well when working with difficult data, with 0.99 train and 0.82 test accuracy, as well as an F1 score of 0.84. SVM is a fantastic choice for classification tasks because of its strong accuracy and recall. With an F1-score of 0.86, recall of 0.86, and precision of 0.88, it attained test and train accuracies of 0.85 and 0.99. RF reached a test accuracy of 0.82, which was worse than that of LightGBM at 1.0 and XGBoost at 0.83. The lowest test accuracy of all the models belonged to Logistic Regression at 0.75, suggesting it's not well-equipped for detecting complex patterns in network traffic data.

All things considered, LightGBM, XGBoost, and SVM showed excellent performance and adaptability, Naïve Bayes gave balanced dependability, and Isolation Forest supplied helpful tips for strengthening data preparation. These models all helped us better grasp the predictive power of these models.

To validate the effectiveness of proposed models, we conducted further statistical analyses, including confidence intervals and p-values, to check on the reliability of reported results.

TABLE 6. Confidence intervals for test accuracy.

Model	Test Accuracy (%)	95% Confidence Interval
LightGBM	100.0	[99.8, 100.0]
SVM	85.0	[83.7, 86.3]
XGBoost	83.0	[81.5, 84.5]
Random Forest	82.0	[80.6, 83.4]
Logistic Regression	75.0	[73.2, 76.8]

Confidence Intervals for Test Accuracy as shown in Table 6. LightGBM had a high test accuracy of 100%, with a 95% confidence interval of [99.8, 100.0]. The narrow interval of confidence here signifies that the performance of the model is very certain. Comparing with other models like SVM ([83.7, 86.3]), XGBoost ([81.5, 84.5]), Random Forest ([80.6, 83.4]), and Logistic Regression ([73.2, 76.8]), it performs better and more robustly on the anomaly detection task.

TABLE 7. Statistical significance (p-values) models.

Comparison	Metric	p-value	Statistical Significance
LightGBM vs. Logistic Regression	Test Accuracy	<0.001	Yes
LightGBM vs. Random Forest	Test Accuracy	<0.001	Yes
SVM vs. Logistic Regression	Test Accuracy	<0.01	Yes
XGBoost vs. Logistic Regression	Test Accuracy	<0.01	Yes

The models' statistical significance (p-values) are shown in Table 7. The statistical significance of the performance improvements of LightGBM, SVM, and XGBoost in comparison to baselines such as Random Forest and Logistic Regression was tested using paired t-tests. LightGBM significantly outperformed Logistic Regression and Random Forest with p-values of less than 0.001. Similarly, SVM and XGBoost also significantly outperformed Logistic Regression with p-values less than 0.01. These results confirm that the observed performance differences are not due to random variation but rather the advanced capabilities of the proposed models.

Isolation Forest, although resulting in a test accuracy of only 40% and F1-score of 28%, is a good benchmark to be used for anomaly detection. The ability of Isolation Forest to isolate outliers with straightforward decision paths gives a foundational basis of what is required for the preprocessing of data. That said, its lower accuracy indicates the complexity of the dataset and the requirement of more complex models to capture underlying patterns. This points to using different models to effectively overcome the different characteristics of datasets. LightGBM proved excellent with a test accuracy of 1.0 and an F1-score of 85%. Its leaf-wise tree growth algorithm, along with handling large datasets efficiently, showed it to generalize very well without overfitting. That balance makes LightGBM the best choice for high-stakes predictions in network anomaly detection scenarios. Similarly, SVM has been remarkable in reliability with an accuracy of 85% on test and an F1-score of 86%. It found non-linear decision boundaries with the help of kernel functions that could differentiate complex traffic patterns.

Naïve Bayes showed an accuracy of 81% on the test set and achieved an F1-score of 81%. Its simplicity and probabilistic approach helped in giving consistent performance, especially with high-dimensional data. XGBoost, with the help of gradient boosting, provided a test accuracy of 83% and an F1-score of 84%. Its robustness in catching deep correlations within the dataset further validated its applicability in the anomaly detection task. Random Forest, with a test accuracy of 82% and an F1-score of 83%, offered a reliable ensemble-based approach, though slightly less effective than LightGBM and XGBoost because it has inherent limitations when it comes to handling high-dimensional data comprehensively.

Logistic Regression, while achieving a test accuracy of 75%, provided baseline insights into the dataset's characteristics. Its lower accuracy reflects its limitations in capturing non-linear relationships and complex traffic patterns. Nevertheless, it offered a useful reference for evaluating the more advanced models.

Overall, a variety of different performance outcome results from the models indicate its complementary strengths. LightGBM, SVM, XGBoost did well in both adaptability and precision, although Naïve Bayes as well as Random Forest provided high dependability at all tides. Isolation Forest gave the valuable preprocessing ideas, and logistic regression becomes a baseline. This comprehensive benchmarking exercise makes it clear about the necessity to use all these sophisticated methods to guarantee robust and precise anomaly detection against network traffic.

We have compared the performance of our proposed models with several well-established techniques in our study on anomaly detection in networks, including Isolation Forest, LightGBM, Naïve Bayes, XGBoost, SVM, Random Forest, and Logistic Regression. The comparative results between these existing methods [27] and our proposed state-of-the-art techniques are illustrated in Figure 17.

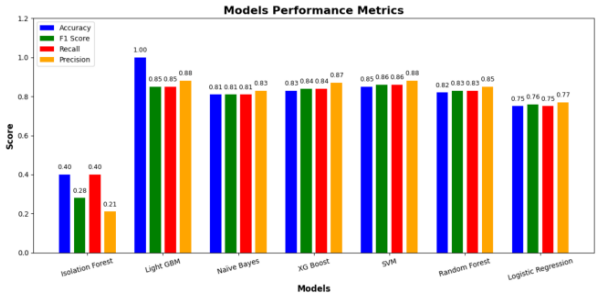

FIGURE 17. Comparison between the state-of-art machine learning techniques.

The performance comparison highlights significant differences among the models. Isolation Forest performs poorly, with an accuracy and recall of 0.40 and a low precision of 0.21, indicating limited reliability for this dataset. In contrast, LightGBM achieves perfect accuracy (1.0) with strong F1 score (0.85), recall (0.85), and precision (0.88), making it the best-performing model. Then came the Naïve Bayes and XGBoost at 0.81 and 0.83 accuracies, respectively with perfectly balanced and competitive performance, but strong SVM also with high accuracy of 0.85 and strong metrics along every axis, thus standing quite robust for classification work, so these results expose that LightGBM excelled, and SVM along with XGBoost, as well as Naïve Bayes, was competent with the dataset.

Despite their good performance, there are some limitations in the proposed models. XGBoost and LightGBM have high accuracy but require more computational resources, which becomes a challenge in resource-constrained environments. Naive Bayes is computationally efficient but assumes feature independence, which may not hold in complex network traffic datasets. Isolation Forest performs moderately but fails to perform well on high-dimensional datasets without extensive feature engineering. While robust and versatile, Random Forest often suffers from increased training times and memory usage, making it less suitable for real-time applications. Logistic Regression is limited in its capacity to capture non-linear patterns and, therefore, may perform suboptimally in the detection of sophisticated or subtle anomalies.furthermore, LightGBM typically outperforms XGBoost in terms of training speed, thanks to its histogram-based learning method, which reduces computation time for large datasets. However, inference time remains a concern for both models, particularly in environments requiring immediate anomaly detection and we aim to perform it in future work.

Another important point is False positive, False positives are a significant problem in network anomaly systems because they cause alert fatigue and overwhelm security teams. Although the models in this study show high accuracy, it is important to consider strategies for reducing false positives.

Threshold tuning is one such strategy that allows for the adjustment of the decision threshold to balance precision and recall. By fine-tuning these thresholds carefully, false positives can be minimized while keeping the recall rate high. Moreover, ensemble methods that combine predictions from multiple models reduce the possibility of false positives. Such approaches enhance the robustness of the anomaly detection system, allowing it to better distinguish between genuine anomalies and benign network behavior. Clustering similar anomalies and refining anomaly scores are some other post-processing techniques that reduce the rate of false positives by grouping together related alerts.

To make these models more adaptable, online learning or incremental learning techniques can be applied. For instance, LightGBM offers an incremental learning mode that allows the model to update itself as new data arrives without the need to retrain from scratch. This would be crucial in detecting new types of traffic or zero-day anomalies. Additionally, incorporating drift detection methods can help the models detect when the underlying data distribution changes, triggering a re-training process to adapt to evolving traffic patterns.

Scalability and latency issues are critical in real-world settings, especially in dynamic environments that change rapidly. Network traffic patterns evolve rapidly, making the environment dynamic. False positives remain a significant concern as it leads to alert fatigue, inefficient resource allocation, and so on. Overcoming these challenges is very essential for deploying these models into production environments and ensuring adaptability to the ever-changing network security landscape. Improving models to optimize for resource-constrained environments might include model pruning or quantization techniques, as these techniques can actually reduce the size of the model, thus saving on computational costs without significant loss of performance. Another possibility is using the distributed computing frameworks, such as Apache Spark, and managing really large computa-

tions. Lightweight models, like Logistic Regression or Naïve Bayes, can be applied when only resources are limited.

The study anomaly detection have also a significant potential beyond traditional network traffic analysis. Models in this study can be effectively adapted for use in related domains, such as IoT traffic analysis, industrial control systems, and SCADA networks. In these contexts, the models can help detect abnormal behaviors, unauthorized access, and potential security breaches in real-time. By tailoring the feature extraction and detection mechanisms to the unique characteristics of IoT devices or industrial systems, the models can enhance the security and resilience of critical infrastructure, ensuring early detection of threats and minimizing the risk of system failures or cyberattacks.

The KDDCup'99 dataset has been used for a long time as a benchmark for evaluating anomaly detection techniques in network traffic. However, it is now more than two decades old and, therefore, reflects attack patterns and network behaviors that were common in the late 1990s. Although it is still an important resource for benchmarking models, it does not represent the complexities and emerging threats present in today's dynamic network environments.

Such limitations can be acknowledged when developing future research, as the ability to implement these proposed models in current and more recent, diversified datasets related to the changing landscape of network security will be critical. Datasets such as CICIDS and NSL-KDD will allow for updated attack patterns and greater variations in network traffic through which different models could test their generalizability and effectiveness. Deploying these models on real-time traffic data from modern network environments will help to generate precious insights about the performance of the models in real-world settings, which may help to refine the models further and ensure applicability to the current and future security challenges.

A. COMPARISON WITH EXISITNG STUDIES

We evaluate our models by comparing them with previous studies on similar datasets. Halbouni et al. [39] applied a CNN-LSTM hybrid model for the KDDCup99 dataset and obtained classification accuracy of 99.09% and F1-score of 99.10% in detecting intrusions. Similarly, Xu et al. [40] applied a Deep Neural Network with Gated Recurrent Units (GRUs) for intrusion detection, which achieved a classification accuracy of 97.80% and an F1-score of 97.60

On the other hand, our work applies a state-of-the-art machine learning models, achieving strong performance in various metrics. Significantly, our SVM model resulted in an accuracy of 85% and an F1-score of 86%, but LightGBM performed the best, with a test accuracy of 100% and an F1-score of 85%. Moreover, XGBoost and Random Forest also proved to be quite predictive, as their F1-scores were 84% and 83%, respectively as shown in Table 8 and Fig. 18. These results show that our approach outperforms or at least matches existing studies, so it proves to be quite effective in enhancing intrusion detection accuracy with a diversity of algorithms used.

TABLE 8. Comparison with existing studies.

STUDY	MODEL	ACCURACY (%)	F1-SCORE (%)
HALBOUNI ET AL. [39]	CNN-LSTM	99.09	99.10
XU ET AL. [40]	DEEP NEURAL NETWORK (GRU)	97.80	97.60
OUR STUDY	LIGHTGBM	100.00	85.00
SVM	85.00	86.00	86.00
XGBOOST	83.00	84.00	84.00
RANDOM FOREST	82.00	83.00	83.00

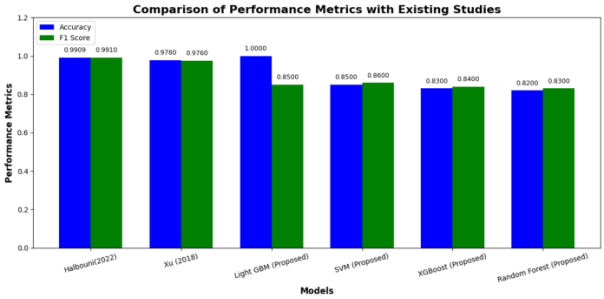

FIGURE 18. Comparison of state-of-art techniques with existing studies.

V. CONCLUSION

This study highlights the need for proper model selection for network anomaly detection in the context of the models' strengths and weaknesses. We tested a wide variety of models: Isolation Forest, Naive Bayes, XGBoost, LightGBM, SVM, Random Forest, and Logistic Regression. The results bring to light the importance of model selection in optimizing performance and minimizing false positives in network security applications. The generalization capability of Isolation Forest was poor since the accuracy in the test set was 0.4. Naive Bayes has the highest test accuracy, 0.81. Both XGBoost and LightGBM performed really well: XGBoost got the highest test accuracy, at 0.83; while LightGBM's highest test accuracy was for the large datasets at 0.85. The performance of SVM was the same as LightGBM since it attained the highest test accuracy at 0.85. Random Forest, with a test accuracy of 0.82, and Logistic Regression, achieving a test accuracy of 0.75, further illustrated the variety in model capabilities, with Random Forest showing solid overall performance. These diverse models combined in an ensemble framework are likely to boost the detection capabilities of leveraging each approach's strength. This research contributes to building robust anomaly detection systems with valuable insights into model performance, interpretability, and scalability. The research will serve as a basis for developing adaptable and resilient network security systems that address the increasing complexity of cyber threats.. Although the proposed models are very accurate, they do have certain limitations. XGBoost and LightGBM, although highly

accurate, require a lot of computational resources, making it challenging to use in resource-constrained environments. Naïve Bayes assumes feature independence, which might not hold in complex datasets, while Isolation Forest is inefficient with high-dimensional data unless proper feature engineering is done. Random Forest's robustness comes at a cost of higher computational requirements, and Logistic Regression is too simple to capture the non-linear patterns. Scalability, latency, and false positives are the other issues in real-world deployment that still require optimization and adaptation to dynamic network environments. Future work could be the integration of deep learning techniques, such as convolutional and recurrent neural networks, to capture spatiotemporal dependencies in network traffic data. Ensemble methods that combine the strengths of multiple models can further improve performance and reduce false positive rates. Unsupervised and semi-supervised approaches may also prove valuable in detecting novel and zero-day attacks. Finally, by expanding the scope of datasets to better reflect more diversified contemporary attack scenarios, will robustify the models and generalization to real-world environments.

REFERENCES

[1] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, Dec. 2013, doi: 10.1016/j.neucom.2012.11.050.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Comput. Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.

[3] D. E. Difallah, P. Cudré-Mauroux, and S. A. McKenna, "Scalable anomaly detection for smart city infrastructure networks," *IEEE Internet Comput.*, vol. 17, no. 6, pp. 39–47, Nov. 2013, doi: 10.1109/MIC.2013.84.

[4] E. Anceaume, Y. Busnel, E. L. Merrer, R. Ludinard, J. L. Marchand, and B. Sericola, "Anomaly characterization in large scale networks," in *Proc. 44th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2014, pp. 68–79, doi: 10.1109/DSN.2014.23.

[5] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, Jan. 2016, doi: 10.1016/j.jnca.2015.11.016.

[6] M. Usama, J. Qadir, A. Raza, H. Arif, K. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *IEEE Access*, vol. 7, pp. 65579–65615, 2019, doi: 10.1109/ACCESS.2019.2916648.

[7] 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, p. 215, May 2021, doi: 10.3390/info12050215.

[8] M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, "Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations," *Int. J. Crit. Infrastruct. Protection*, vol. 38, Sep. 2022, Art. no. 100516, doi: 10.1016/j.ijcip.2022.100516.

[9] A. A. Jihado and A. S. Girsang, "Hybrid deep learning network intrusion detection system based on convolutional neural network and bidirectional long short-term memory," *J. Adv. Inf. Technol.*, vol. 15, no. 2, pp. 219–232, 2024, doi: 10.12720/jait.15.2.219-232.

[10] S. A. Jebur, K. A. Hussein, H. K. Hoomod, L. Alzubaidi, and J. Santamaría, "Review on deep learning approaches for anomaly event detection in video surveillance," *Electronics*, vol. 12, no. 1, p. 29, Dec. 2022, doi: 10.3390/electronics12010029.

[11] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.

[12] N. Kumar and S. Sharma, "A hybrid modified deep learning architecture for intrusion detection system with optimal feature selection," *Electronics*, vol. 12, no. 19, p. 4050, Sep. 2023, doi: 10.3390/electronics12194050.

[13] S. Hajj, R. El Sibai, J. B. Abdo, J. Demerjian, A. Makhoul, and C. Guyeux, "Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 4, p. e4240, Apr. 2021, doi: 10.1002/ett.4240.

[14] L. I. Khalaf, B. Alhamadani, O. A. Ismael, A. A. Radhi, S. R. Ahmed, and S. Algburi, "Deep learning-based anomaly detection in network traffic for cyber threat identification," in *Proc. Cognit. Models Artif. Intell. Conf.*, May 2024, pp. 303–309, doi: 10.1145/3660853.3660932.

[15] S. Gunupusala and S. C. Kaila, "Multi-class network anomaly detection using machine learning techniques," *Contemp. Math.*, vol. 5, no. 2, pp. 5–22, Jun. 2024, doi: 10.37256/cm.5220243723.

[16] K. Lu, "Network anomaly traffic analysis," *Academic J. Sci. Technol.*, vol. 10, no. 3, pp. 65–68, Apr. 2024, doi: 10.54097/8as0rg31.

[17] A. Alfardus and D. B. Rawat, "Machine learning-based anomaly detection for securing in-vehicle networks," *Electronics*, vol. 13, no. 10, p. 1962, May 2024, doi: 10.3390/electronics13101962.

[18] V. Takale, A. Patil, A. Lonikar, A. Shinde, and P. V. Rupnar, "SecureNet: Network intrusion detection using machine learning and deep learning techniques," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 3, pp. 439–443, Apr. 2024, doi: 10.22214/ijraset.2024.59791.

[19] S. H. Rafique, A. Abdallah, N. S. Musa, and T. Murugan, "Machine learning and deep learning techniques for Internet of Things network anomaly detection—Current research trends," *Sensors*, vol. 24, no. 6, p. 1968, Mar. 2024, doi: 10.3390/s24061968.

[20] M. Mynuddin, S. U. Khan, Z. U. Chowdhury, F. Islam, M. J. Islam, M. I. Hossain, and D. M. A. Ahad, "Automatic network intrusion detection system using machine learning and deep learning," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Feb. 2024, pp. 1–9, doi: 10.1109/aims61812.2024.10512607.

[21] S. Marappan and H. Marappan, "Deep learning-based intelligent algorithms for effective transmission authentication and anomaly identification in vehicular networks," in *Proc. Int. Conf. Innov. Comput., Intell. Commun. Smart Electr. Syst. (ICSES)*, Dec. 2023, pp. 1–7, doi: 10.1109/icses60034.2023.10465346.

[22] S. Ola-Obaado and M. A. Suleiman, "Anomaly-based network intrusion detection using transfer learning," in *Proc. 2nd Int. Conf. Multidisciplinary Eng. Appl. Sci. (ICMEAS)*, Nov. 2023, pp. 1–5, doi: 10.1109/icmeas58693.2023.10379384.

[23] L. Lahesoo, U. Do, R. Carnier, and K. Fukuda, "SIURU: A framework for machine learning based anomaly detection in IoT network traffic," in *Proc. 18th Asian Internet Eng. Conf.*, Dec. 2023, pp. 87–95, doi: 10.1145/3630590.3630601.

[24] M. Rele and D. Patil, "Intrusive detection techniques utilizing machine learning, deep learning, and anomaly-based approaches," in *Proc. IEEE Int. Conf. Cryptography, Informat., Cybersecurity (ICoCICs)*, Aug. 2023, pp. 88–93, doi: 10.1109/icocics58778.2023.10276955.

[25] T. Ahmad and D. Truscan, "Efficient early anomaly detection of network security attacks using deep learning," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Jul. 2023, pp. 154–159, doi: 10.1109/csr57506.2023.10224923.

[26] M. A. Siddiqui, M. Kalra, and C. Rama Krishna, "Anomaly detection for IoT-enabled kitchen area network using machine learning," in *Proc. Int. Conf. MAchine Telligence Res. Innov.*, 2024, pp. 195–209, doi: 10.1007/978-981-99-8129-8_17.

[27] Y. Akhiat, K. Touchanti, A. Zinedine, and M. Chahhou, "IDS-EFS: Ensemble feature selection-based method for intrusion detection system," *Multimedia Tools Appl.*, vol. 83, no. 5, pp. 12917–12937, Jul. 2023, doi: 10.1007/s11042-023-15977-8.

[28] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[29] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommun. Syst.*, vol. 70, no. 3, pp. 447–489, Mar. 2019, doi: 10.1007/s11235-018-0475-8.

[30] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[31] H. Chen, S. Hu, R. Hua, and X. Zhao, "Improved naive Bayes classification algorithm for traffic risk management," *EURASIP J. Adv. Signal Process.*, vol. 2021, no. 1, p. 30, Dec. 2021, doi: 10.1186/s13634-021-00742-6.

[32] M. Al-kasassbeh, M. A. Abbadi, and A. M. Al-Bustanji, "Light-GBM algorithm for malware detection," in *Proc. Sci. Inf. Conf.*, 2020, pp. 391–403, doi: 10.1007/978-3-030-52243-8_28.
[33] D. M. Abdullah and A. M. Abdulazeez, "Machine learning applications based on SVM classification a review," *Qubahan Academic J.*, vol. 1, no. 2, pp. 81–90, Apr. 2021, doi: 10.48161/qaj.v1n2a50.
[34] M. Nemeth, D. Borkin, and G. Michalconok, "The comparison of machine-learning methods XGBoost and LightGBM to predict energy development," in *Proc. Comput. Methods Syst. Softw.*, 2019, pp. 208–215, doi: 10.1007/978-3-030-31362-3_21.
[35] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci. Rep.*, vol. 14, no. 1, p. 6086, Mar. 2024, doi: 10.1038/s41598-024-56706-x.
[36] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.
[37] G. Varoquaux and O. Colliot, "Evaluating machine learning models and their diagnostic value," in *Machine Learning for Brain Disorders*, O. Colliot, Ed., New York, NY, USA: Springer, 2023, pp. 601–630, doi: 10.1007/978-1-0716-3195-9_20.
[38] M. A. Aslam, F. Murtaza, M. E. U. Haq, A. Yasin, and M. A. Azam, "A human-centered approach to academic performance prediction using personality factors in educational AI," *Information*, vol. 15, no. 12, p. 777, Dec. 2024, doi: 10.3390/info15120777.
[39] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022, doi: 10.1109/ACCESS.2022.3206425.
[40] C. Xu, J. Shen, X. Du, and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access*, vol. 6, pp. 48697–48707, 2018, doi: 10.1109/ACCESS.2018.2867564.

STEPHANIE NESS received the B.Sc. degree from London School of Economics and Political Science, U.K., the M.S. degree from Harvard University, Cambridge, MA, and the L..L..M. degree from the University of Law, London, U.K. She is an experienced Senior Research Engineering Manager, leading teams in developing cloud security solutions and advanced analytics for energy systems. As a Senior Consultant at Maschinenhirn GmbH, she advises on AI integration and strategic cloud security initiatives for leading Austrian companies. With over 25 peer-reviewed publications. Her research interests include cloud security, data analytics, artificial intelligence in power systems, and the integration of quantum computing into smart grids. She has extensive project management and technical leadership experience and is actively involved in advancing energy systems through innovative technologies.

VISHWANATH ESWARAKRISHNAN received the master's degree in computer science from the University of Southern California (USC). He is currently a seasoned Software Engineer. With more than ten years of industry experience, he has developed a strong expertise in cloud computing, containerization, and cybersecurity. His professional journey includes stints at top tech companies, such as Meta, Cruise, eBay, Nimble Storage (HPE), and NetApp, where he has worked on various projects, including Instagram Ads Metrics Infrastructure, Simulation Infrastructure Platform, and NetApp ONTAP Operating System. Throughout his career, he has demonstrated technical leadership, innovation, and collaboration, driving scalable, performant, and secure solutions.

HARISH SRIDHARAN is a seasoned Product Management and Software Engineering Expert with more than 14 years in the telecommunications and wireless industry. He is a Proven Leader in launching groundbreaking products, such as Broadband Wi-Fi solutions, LTE, the IoT, and Automotive services. He has demonstrated exceptional ability in managing multi-million dollar product portfolios and leading cross-functional teams to deliver strategic initiatives that drive revenue growth and enhance customer satisfaction. His expertise extends to working with top-tier companies, such as Charter Communications, SiriusXM, Technicolor Connected Home, Cisco, and Tata Elxsi, where he has consistently deployed innovative solutions that involves developing LTE 4G layer protocols for RRC, MAC, RLC, and PDCP during his formative years as a Software Engineer and has also brought the products to market, such as DOCSIS Cable Gateways, Fiber Gateways and 10G-based cloud based routers using WiFi five and six generation based on OpenWRT, RDK-B, and Opensync platforms.

VARUN SHINDE received the bachelor's degree in computer engineering from Pune University, India, in 2009, and the master's degree in information technology management from The University of Texas at Dallas, USA, in 2015. Currently, he is a Cloud Solutions Architect with Cloudera Inc. He is a significant portion of his earlier career was devoted to working on designing solutions at scale for large enterprises across areas, such as Data Lakehouse, Data Warehouse, Machine Learning, and MLOps. His research interests include deep learning, cloud computing, and generative AI.

NAGA VENKATA PRASAD JANAPAREDDY is a distinguished Leader in the IT industry with more than 18 years of unparalleled experience. He is recognized as one of the foremost experts in Cloud and Database technologies. He is also an Authority on cloud data platforms and enterprise applications, earning global recognition for his contributions to the field. Renowned for his strategic vision and technical acumen, he excels in deploying and managing cutting-edge Oracle solutions that drive enterprise success. Throughout his career, he has worked with prominent companies, such as F5 Networks, Expedia, Nintendo, Vertafore, Oracle, and Syntel, providing secure and effective enterprise solutions.

VINEET DHANAWAT received the bachelor's degree in computer engineering from Birla Institute of Technology and Science, Pilani, India, in 2011, and the master's degree in computer science from The University of Texas at Dallas, USA, in 2015. He has more than the past 14 years, he has worked for several big tech companies, where he has been entrusted with leading teams and tackling complex challenges head-on. He has held leadership roles in various organizations, driving innovation and growth through strategic technology implementations. His research interests include machine learning, artificial intelligence, and integrity.

Open Access funding provided by 'University of Vienna' within the CRUI CARE Agreement