Journal of Future Artificial Intelligence and Technologies

E-ISSN: 3048-3719

Research Article

A Hybrid Approach for Efficient DDoS Detection in Network Traffic Using CBLOF-Based Feature Engineering and XGBoost

Zainab Sahib Dhahir *

Department of Computer Networking Technologies and Software, Technical Institute of Babylon, Al-Furat Al-Awsat Technical University, Babil 51015, Iraq; e-mail : zainab.dhahir@atu.edu.iq

* Corresponding Author : Zainab Sahib Dhahir

Abstract: This is one of the greatest challenges in computer network security and cannot be dealt with without a set of most recent detection techniques. This paper lays down a new hybrid technique that combines Clustering-Based Local Outlier Factor (CBLOF) and Extreme Gradient Boosting (XGBoost) to enhance accuracy while detecting Distributed Denial of Service (DDoS) from network traffic. The proposed hybrid model utilizes a CBLOF for outlier detection as feature engineering. Over the detected anomalies, classification is to be done using XGBoost classification to attain the objective. The proposed hybrid model was tested extensively on CICIDS 2017 and CICIDS 2018 datasets Compared with traditional ones, the proposed model outperformed the traditional ones with an accuracy rate of 99.99%, precision of 100%, and F1 score reflecting perfection. These results confirm this model's efficiency in terms of known and novel attack patterns and introduce a further reliable framework for the timely detection of DDoS attacks. Even if it is computation-heavy, optimization could be made towards real-time large-scale data.

Keywords: Clustering-Based Local Outlier (CBLOF); DDoS attacks; Extreme Gradient Boosting (XGBoost); Intrusion Detection; Network security.

1. Introduction

For the last couple of years, Distributed Denial of Service (DDOS) attacks have been one of the most common menaces to network security and have brought huge disruptions and financial loss into view for various industries. As such attacks increase in intricacy and size, and likewise the constant development of intricacy in network infrastructures, the need for enhanced and more proficient detection methods has risen. Though traditional systems for detecting DDoS attacks have certain merits, the big volume and diversity of the traffic flow impede them from effectively identifying malicious activities in legitimate network operations [1], [2]. One of the most severe risks facing the networks and the organizations using them is the DDoS attack. This occurs in the form of trying to flood a network or server with more traffic than can be handled.

Consequently, it slows down the service or makes it unavailable for legitimate purposes. The outcome, therefore, can be critical downtime and disruption of business[3]. The immediate financial consequence of a DDoS attack would involve lost revenue due to service outages, mitigation costs, and possible fines in case of SLA violations. Indirectly, a reputation impact leads to customer loss and their trust. The technical and administrative resources that any IT department would have to invest to neutralize a DDoS attack are pretty hefty. Their shift from other critical tasks would result in low productivity and operational inefficiencies. Apart from that, if the attacks keep recurring or last for a really long period, there is always damage to an organization's reputation, reflecting customer trust in the brand[4].

Received: September, 1st 2024
Revised: September, 24th 2024
Accepted: September, 27th 2024
Published: September, 30th 2024
Curr. Ver.: September, 30th 2024

DOI : 10.62411/faith.2024-33

This will lead to a perceived view by the clients and users about the organization being untrustworthy or vulnerable. The mitigation process can be quite expensive since special services or added infrastructure for handling or filtering the attack traffic are needed. Additional costs also come into play in recovery and system upgrades post-attack. In other cases, DDoS may unintentionally affect other services or networks. An attack of this type against a shared hosting provider might bring multiple sites down. Depending on the case, if a DDoS leads to data protection regulations or industry standards being breached, possible legal and compliance issues for the organization may also arise. If the attack does not bring down the network but knocks performance, it would cause problems such as latency, packet loss, and reduced throughput. Anything can range from all users depending on the network to all applications [4], [5]. While this may be an independent attack, it is also used as a smokescreen to exploit other vulnerabilities in the network or systems that could result in security breaches or data compromise. DDoS attacks have, in other words, the potential to cause widespread and far-reaching impacts on networks, impacting service availability, financial viability, operational effectiveness, and reputation.

Robust security should be implemented with a full response plan to mitigate these risks as much as possible[6]. One of the strongest and deadliest threats towards Global Networks during the modern era has been the DDoS attacks. DDoS performs an implicit and well-hidden attack over the network or service, drastically compromising the network's availability because of the many spoofed requests it sends to the attacked network [7]. This, in turn, points to the prime importance of their early detection and mitigation. Basically, DDoS detection in network traffic has now become an activity that needs to be performed in near real-time. The continuous evolvement of cyber-attacks has made machine learning one of the most favorite and effective approaches for classifying malicious activities in the complex domain of network traffic[8]. Such services and features have allowed for performance and efficiency hitherto unattained, making them indispensable ingredients in the fight against cyber threats. More advanced machine learning models provide even better protection from DDoS, guarantee network integrity, and deliver higher availability for networks[2], [9].

This research proposes a hybrid model that uses CBLOF and XGBoost to detect high-accuracy DDoS attacks in network traffic. The proposed model can detect outliers in the network traffic and segregate them further through ML-based techniques to classify them as attacks. This model is trained, tested, and evaluated on the CICIDS 2017 dataset pertaining to DDoS attacks. The proposed hybrid model is compared with two other models, CBLOF and XGBoost, which are applied separately on the same dataset. Evaluations are done based on performance metrics such as accuracy, precision, recall, specificity, and F1 score. Furthermore, and most importantly, CBLOF + XGBoost outperforms CBLOF and XGBoost models separately by a significant margin on performance metrics, as indicated in the results section. The main contributions of this paper are as follows:

- This paper proposes a new hybrid framework that combines the strengths of CBLOF and XGBoost to effectively detect DDoS attacks in network traffic by adding outlier detection by CBLOF as an additional feature.
- This aims to enhance feature engineering by using CBLOF to capture anomalous network traffic behavior, which is then used to improve XGBoost's ability to differentiate between good and bad traffic through outlier identification..
- The approach presented here uses SHapley Additive exPlanations (SHAP) for the interpretability of the proposed machine learning model, which explains which features have contributed to the classification results the most. It could also be used to determine the most contributing features of DDoS attacks.

2. Literature Review

While reviewing supervised, unsupervised, and hybrid models, most authors believe the unsupervised models did better than the supervised ones. Most of the authors then surfaced the convolutional autoencoders among the most efficient unsupervised deep learning models. Different authors have employed several techniques in network traffic analysis to detect DDoS intrusion, including Sirisha[10]. For that, a study [10] proposed an efficient detection solution for DDoS, wherein this targeted Naive Bayes, SVM, and Logistic Regression for accurate detection in semi-supervised ML techniques. As such, the search should efficiently target DDoS attacks, as usually, when the attack happens, there is congestion in the network and unresponsive servers. The proposed system doesn't need fine-tuning; it can detect multiple network attacks on one dataset. Similar detection mechanisms in the literature show that for the detection of DDoS attacks in IoT, that of A. Srivastava et al. [11] proposed an XGBoost and Random Forest-based detection mechanism for these network attacks.

L. Fray [12] has put forward effective methods for detecting DDoS attacks using Random Forest, Decision Tree, AdaBoost, XGB, MLP, and DNN. The research findings demonstrate a high rate of detection for the proposed model. In a separate study, R. Rolim, Rawsen Alves, G. Sundararajan, and Javam de Castro Oliveira examined the performance of a self-organizing map in monitoring web traffic. Their findings suggest that this model excels in identifying unusual events. Sharma et al. [13] presented a study on an Online Detection System for LDoS attack Based on XGBoost, aimed at detecting LDoS attacks in real-time. This method exhibited high accuracy in recognizing LDoS attacks and showed promising outcomes. Thus, The study provides a strong foundation for adopting XGBoost in DDoS detection.

In that connection, Liu et al. [3] proposed another related work involving a machine learning-based technique for detecting DDoS in network traffic based on unsupervised learning algorithms such as K-means clustering. The study also discussed the inefficiencies of the traditional methods used for DDoS detection. It thereby stressed the need for better and more reliable solutions toward the detection of network traffic anomalies. Several methodologies have been proposed to detect DDoS attacks in network traffic effectively. The list includes machine learning algorithms such as CBLOF and XGBoost. In this section, we explore the related literature for these methodologies and analyze how effectively they can cope with detecting and mitigating DDoS attacks. Moreover, we discuss the failures of the current approaches and propose a hybrid approach that embeds the beneficial features of both CBLOF and XGBoost. The literature review will delve into current methodologies and their effectiveness in detecting DDoS attacks. Numerous research works use the CBLOF methodology for IDS applications; some clearly show its efficiency in detecting anomalies in network traffic. Whereas explicit mentions of CBLOF in the literature reviewed are somewhat limited, the general context of anomaly detection and the use of machine learning approaches in intrusion detection systems are discussed in detail.

Nevertheless, the application of CBLOF in the domain of DDoS detection has been explored in recent works; for example, the author of [14] presents a combination of deep learning models and traditional or advanced outlier detection techniques like CBLOF. Such a model combination is proposed to increase the overall detection performance. In addition, a hybrid solution for detecting intrusions in network flow was suggested in [15] by jointly applying Autoencoder and XGBoost. It involves classifying the attributes derived from the autoencoder and detecting anomalies. The research objective in [16] is to prove how XGBoost can enhance intrusion detection system performance. It is attained by increasing both classification accuracy and efficiency. Reference [17] aims to develop an enhanced IDS by incorporating XGBoost into ensemble learning methodologies, enabling the system to improve its intrusion detection capability and further withstand different kinds of intrusions. The study [18] is focused on mitigating the hurdles posed by imbalanced multiclass classification in intrusion detection systems related to the Industrial IoT. This study considers using XGBoost, a powerful machine learning algorithm that will power up precision and efficiency throughout the detection process. Table 1 outlines several studies and their respective methodologies applied.

Although unsupervised machine learning and deep learning have performed well in DDoS attack detection, there are some limitations that they might face challenges where the traffic patterns are complex and also when data sets are imbalanced. Such an approach couple's subtle detection of outliers in network traffic with CBLOF and strong classification by means of XGBoost. That's where it gains in terms of accuracy and robustness. Moreover, feature interpretation, done with SHAP, adds considerable value to understanding drivers for such classification, which is usually overlooked in former works. The holistic approach offers a scalable structure for real-time detection by working out computational challenges common in most mainstream methods.

Table 1. Literature Review on Techniques for Detecting DDoS Attacks and Their Performance Outcomes

Ref.	Method used	Results
[2]	Sparse convolute network for IoT threat analysis and DDoS recognition.	ESCNN system detects DDoS attacks with 98.9% maximum detection rate.
	LSTM training process for user behavior identification and precision improvement.	ESCNN classifies normal and abnormal activities with 99.29% recognition accuracy.
	Bayesian network models for DDoS with incomplete data.	LSTM training process effectively reduces false attack prediction rate.
[3]	Feature engineering with binary grey wolf optimization algorithm.	Random Forest outperformed other algorithms in DDoS attack detection metrics.
	Machine learning classifiers: SVM, RF, Decision Tree, XGBoost, k-NN.	The proposed method effectively detects and identifies DDoS attacks in SDNs.
[4]	Logistic regression, CNN, XGBoost, naive Bayes, AdaBoost, KNN, random forest ML.	XGBoost model achieved 99.9999% detection accuracy using the SMOTE approach.
	Feature extraction, model training, testing, data preprocessing, and machine learning models.	
	Heatmap matrix, tree classifier, logistic approach, data preparation, data cleaning	
[6]	XGBoost, RF, and ANN are used for classification in the proposed method.	The proposed model outperforms others with 99.99% and 100% accuracy.
	Feature selection methods include SMOTE, XGBoost, and RF.	
[8]	SVM classification algorithm, SNORT IPS integration for DDoS prevention	Achieved an average accu-racy rate of 97 %.
[10]	Naive Bayes, SVM, Logistic Regression for DDoS detection accuracy analysis.	SVM, Logistic Regression, and Naive Bayes achieved 95.94% accuracy in detection.
[12]	Use of Random Forest, Decision Tree, AdaBoost, XGB, MLP, DNN.	RF classifier achieved 99.97% accuracy, and Decision Tree had 99.88% accuracy.
[13]	XGBoost as a classifier for detection.	Detection accuracy: 95.11% for UNSW-NB15, 99.92% for CICIDS2017
[18]	XGBoost model applied for imbalanced multiclass IoT datasets	XGBoost achieved F1 scores of 99.9% and 99.87% on two datasets.

3. Background on DDoS Attacks

Computer networks are the backbone for business, industry, commerce, research, and academia. Physical cables, connections, and switches or routers are used to connect computers and servers. Although very useful and highly applied, this form of infrastructure comes with its threats and vulnerabilities. On the other hand, attacks can be invasive, such as unauthorized access and misuse of information[3]. DDoS attack floods devices with access requests such that one couldn't operate the network. Distributed DoS derives from multiple sources, making it hard for the victim to identify the source. With the growth of networked infrastructure, DDoS attacks have become a significant threat to the availability of networks over the last few decades [19]. In a DDoS attack, numerous systems send a massive number of connection requests within a short time, eventually consuming all the victim's bandwidth, which causes the network to be unavailable. These attacks can consume either bandwidth or server resources. Most types of DDoS can be launched at all network layers. The most widespread are TCP SYN flood, HTTP flood, UDP flood, ICMP flood, etc. DDoS attacks on Layer 7, the Application Layer, are very advanced and difficult to be caught. The reasons for executing these attacks may include the following: to show in-competence, extortion, revenge, etc. [20].

3.1. Anomaly Detection in Network Traffic

Continuing network attacks result from increasing Internet use and connecting several devices. One major threat is DDoS, which works by overwhelming traffic to restrict access. Hackers or botnets usually hack those local area networks operating behind weak securities of any broadband subscription. Consequently, DDoS, due to this attack, victimizes nearly all organizations, from finance, banks, and e-business to learning institutions, leading to downtime and economic loss. Therefore, there is a great need to detect DDoS attacks to ensure uninterrupted network usability[21]. DDoS attack detection techniques can be anomaly-based or misuse-based methods. Misuse-based techniques depend on a database containing past attacks and thus cannot detect novel attacks. An anomaly-based approach characterizes normal network traffic behavior and enables the detection of novel attacks[5]. Normal event streams are usually determined by thresholds, which are mostly unrealistic in a large network; hence, newer, efficient machine learning methods. These methods estimate traffic patterns in order to identify abnormal event streams. The ma-chine-learning-based anomaly detection methods could be further classified into three categories: supervised, unsupervised, and semi-supervised. Most of the approaches that come under a supervised setting require labeled data, wherein classifiers are trained for the detection while only a few anomalies are known. Labeling is expensive, and most data sources don't have such labels accompanying them[22]

3.2. Machine Learning Techniques for DDoS Detection

The interest in employing machine-learning techniques for DDoS attack detection and network traffic classification has recently increased. The most commonly used machine-learning algorithms for detecting DDoS attacks in network traffic are NB, SVM, KNN, RF, DT, and ANNs. After extensive analysis, it was found that RF, followed by KNN and SVM performed relatively better among these algorithms. However, the stated algorithms have their strengths and limitations for different traffic conditions. While tree-based learning approaches like RF and DT are strong classifiers, interpreting the results is complicated. On the other hand, while NB and SVM are interpretable models, their performance can drop significantly with a small amount of random noise in the data. Shallow ANNs have interpreted models and suffered less from noise than SVM or RF, but the performance is greatly affected by training data conditions [4]. A hybrid model combining DT with MLP ANN classifier performs relatively better than DT or MLP ANN models alone. However, properly training an ANN model requires extensive datasets with sufficient representative samples of all required classes, which is often difficult and impractical for real networks. Tree-based algorithms like RF that are tolerant of the outlier data and generally require less preprocessing are generally recommended for DDoS detection under network traffic with high noise and outliers. The C4.5 decision tree algorithm is one of the most popular algorithms for constructing decision tree classifiers. C4.5 algorithms can process both continuous and discrete attributes, and in a decision tree, they can handle missing data by determining the attribute weights for splitting nodes[3]. On the other hand, it can do a post-processing optimization on the tree to reduce memory requirement and tree size. The main limitation of C4.5 is that it works well only for smaller datasets; however, the performance is degraded when the data grows in size and complexity. Random Forest (RF) is an ensemble of tree-structured classifiers where, during operation, each classifier votes for a class, and the class with the majority of votes is the output of the RF. Using this ensemble idea broadly reduces the generalization error. Each classifier in the RF is created using different subsets of the training dataset obtained using bootstrap sampling, setting the basis that it can help to reduce the overfitting of a single classifier (single Decision Tree). The out-of-bag (OOB) method evaluated the individual tree's performance. The RF algorithm also offers a measure of variable importance that can help to find the most representative traffic features for the classification task and thus help in the network congestion investigation[6].

3.2.1. Clustering-Based Local Outlier Factor (CBLOF)

The CBLOF is a methodological approach designed to detect outliers by fusing ideas from clustering and the local outlier detection process. Combining strengths from the clustering techniques and the LOF algorithm, the CBLOF was developed to enhance the performance of the outlier detection algorithms. This technique employs clustering methods to enhance the accuracy of LOF. Clustering has been done on the dataset by using a clustering algorithm, like K-means, where each data point will be assigned to a cluster. The Cluster Outlier Factor evaluates the dissimilarity of a point from other points within its cluster, while the Cluster Size Factor considers the cluster's magnitude. Points in smaller clusters are more likely outliers than those in larger clusters. The final outlier score of an individual data point is obtained by combining the cluster outlier factor with the factor that represents the cluster size. Some of the advantages of CBLOF are as follows:

- Contextual Detection: This involves a more context-sensitive anomaly detection, considering the density and magnitude of the data points within clusters.
- Higher Sensitivity: Enhances the ability to find anomalies, especially in data sets with a tendency to have varying densities and dimensions of clusters.

CBLOF shows efficacy for detecting outliers in such datasets, which would create challenges for the traditional LOF due to its consideration of the different attributes of clusters throughout the outlier detection process [23].

3.2.2. Extreme Gradient Boosting (XGBoost)

The XGBoost framework provides a robust and efficient machine learning algorithm, mainly used for classification and regression tasks. It is the particular approach of ensemble learning methodologies using gradient boosting. This framework builds up a sequence of decision trees, each one trying to correct the errors of the previous one. XGBoost has several key features, such as:

1. The technique involves embedding a diverse set of weak learners, often in the form of decision trees, into one strong learning model. Every subsequent tree in this ensemble is consciously designed to correct mistakes made by the current models.
2. Regularization The estimation here involves the application of L1-Lasso and L2-Ridge, one for reducing overfitting and improving overall generalization capability.
3. Dealing with Absent Data: Has a systematic missing value handling, using the most effective technique for imputation or adjustment during training.
4. This is designed for efficiency, so trees can be processed in parallel to accelerate training, which is particularly useful when datasets are very large.
5. It provides substantial insights into core features that are required for making predictions, hence helping with feature selection and improving the performance understanding of the model.
6. It can manage various data and tasks, from classification and prediction to prioritization and customized goal criteria.

XGboost can also be used in several cases of categorization and identification, regression in forecasting cases, and ranking and recommendation systems. This is because accurate management and handling of huge datasets have become really popular in machine learning competitions and for practical purposes[24].

4. Proposed Hybrid Approach

This integrated approach combines the merits of CBLOF and XGBoost techniques to detect DDoS attacks in network traffic. It stitches together the strengths that inherently arise from the two methods: one enhances the accuracy in the detection and resilience against a varied set of attack types. It is envisioned that this hybrid model will better characterize the anomalies in the network traffic and classify the attack patterns effectively afterward. The new concept couples the power of both CBLOF and XGBoost to develop further the results obtained in DDoS attack detection. CBLOF is also one of the outliner detection algorithms that could perform basically within the clustering techniques. CBLOF could undoubtedly find singular points amongst the data, following all the clustering principles. It efficiently finds hidden patterns in the data and classifies the outliers accurately. By contrast, the XGBoost algorithm is a powerful gradient-boosting algorithm used for building progressively a strong predictive model from a sequence of weak classifiers. The integration of CBLOF with XGBoost is an inclusive approach for DDoS detection. The clustering strengths of CBLOF ensure that even a little deviation from normal network activity gets picked by the hybrid framework as an indication of active attack execution. The boosting will increase the accuracy of the developed methodology for detection through iterative reshaping of the model's predictive power. Fig.1 describes the proposed hybrid approach.

4.1. Integration of CBLOF and XGBoost

The fundamental principle behind this hybrid approach is the combined use of CBLOF with XGBoost. In this approach, CBLOF works as a preprocessing step that pinpoints instances in network traffic as potential outliers by using data clustering, and an outlier score is given for each cluster attribute. Scores developed from CBLOF can be used to discern any abnormal traffic features that could indicate a DDoS attack. After detecting outliers, the classified outliers are further classified using XGBoost into different classes of normal traffic or DDoS attacks. The outlier scores become a feature for XGBoost in developing extreme classification, giving a high accuracy and efficiency value. Therefore, two combined methodologies will enable the detection of established and new attack patterns with a very high degree of accuracy. The proposed CBLOF with the XGBoost mechanism primarily responds to complexity and variability in handling network traffic. Therefore, a very valid tool for real-time DDoS detection was developed.

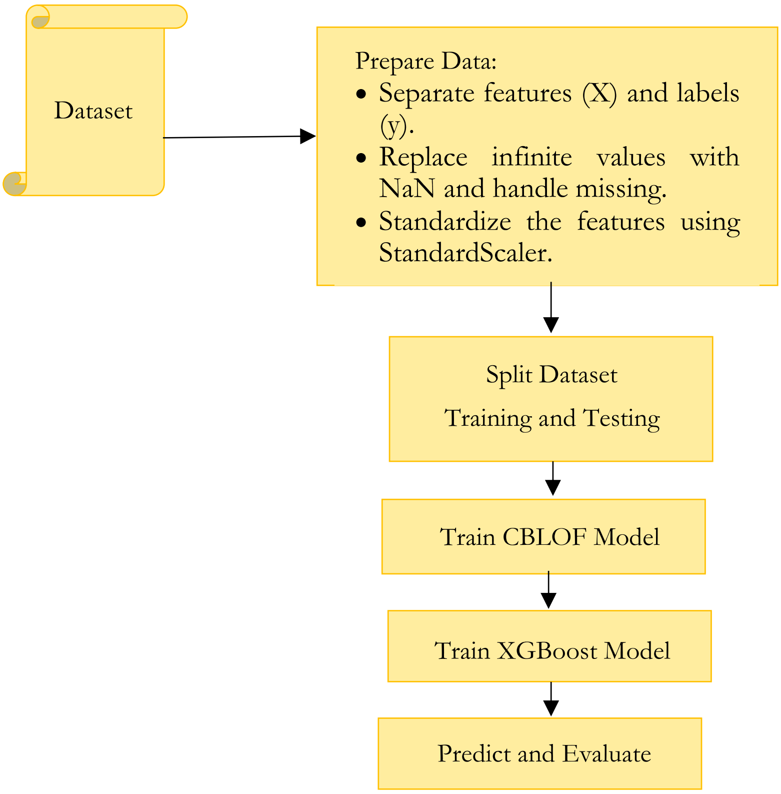

Dataset
Prepare Data:
• Separate features (X) and labels (y).
• Replace infinite values with NaN and handle missing.
• Standardize the features using StandardScaler.
Split Dataset
Training and Testing
Train CBLOF Model
Train XGBoost Model
Predict and Evaluate

Figure 1. Proposed Hybrid Approach.

4.2. Feature Engineering for DDoS Detection

Since that highly influences CBLOF and XGBoost performances, feature engineering can be considered the heart of the proposed hybrid methodology. Consequently, the present section strongly pleads for feature selection and transformation so the model may discriminate highly between normal and malicious traffic. Feature engineering involves analyzing the network traffic data to identify the features that will represent the nature of the DDoS attack. Critical features encompass packet size, flow duration, and connection quantity. These features are chosen based on their significance in DDoS detection and their capacity to enhance the discriminatory capability of the model.

Furthermore, the characteristics are uniform and adjusted to guarantee equal contribution to the model's educational procedure. Sophisticated methods, like reducing dimensionality and choosing characteristic algorithms, minimize the characteristic area and remove unnecessary or irrelevant characteristics. This procedure improves detection's accuracy and decreases computational complexity, rendering the model more effective for real-time applications. The steps of the proposed hybrid approach are shown in Algorithm 1.

4.3. Preparation of Data and Extraction of Features

The CSE-CIC-IDS2017 [25] and CSE-CIC-IDS2018 [26] datasets provided by the University of New Brunswick in Canada were utilized for empirical investigation in this study. Minimal duplicates and uncertainty were noted, and the dataset is available in a convenient

CSV format[27]. With 78 features representing different attack types, difficulties were encountered due to the significant imbalance in data volume. An average of 210,000 sample data from DDoS was chosen to tackle this issue.

Algorithm 1. CBLOF & XGBoost

```
Step1: Load DataSet
Step2: Prepare Data:
    • Separate features (X) and labels (y).
    • Replace infinite values with NaN and handle missing values by filling them with the mean.
    • Standardize the features using StandardScaler.
Step3: Split Dataset
    • Training and Testing sets with a 70-30 ratio
Step4: Train CBLOF Model
    • Initialize the CBLOF model with specific parameters such as contamination, random_state, and others.
    • Train the CBLOF model on the training dataset.
Step5: Feature Transformation Using CBLOF
    • Generate outlier scores from the CBLOF model for both training and testing datasets.
    • Combine the original features with the outlier scores to form a new set of features for both training and testing datasets.
Step6: Train XGBoost Model
    • Convert string labels in the training and testing datasets to numeric values (e.g., 'Benign' to 0 and 'DDoS attacks' to 1).
    • Initialize the XGBoost model and perform 5-fold cross-validation on the combined training set to evaluate its accuracy.
    • Train the XGBoost model on the combined training set.
Step7: Predict and Evaluate the Model
    • Use the trained XGBoost model to predict labels on the combined testing set.
    • Convert numeric predictions back to string labels for evaluation.
    • Generate a classification report to assess the model's performance.
    • Calculate the accuracy score on the testing set.
    • Construct and display a confusion matrix to visualize the model's performance.
```

4.4. Dataset Cleaning and Balancing

While preparing a dataset, it is common to encounter various data quality issues such as Nan values, outliers, and duplicates. The process of data cleaning is crucial during data preprocessing to mitigate any potential negative impacts of these issues on the dataset's quality. In the current study, any infinite values in the dataset are substituted with NaN (Not a Number), as infinite values can lead to complications in machine learning models. Either the assumption is used that the values are normal one time and attack another time, but this affects the accuracy of the results, or one can expect a value closer to reality by relying on the adjacent points within the same cluster, and this is what was adopted. After substituting infinite values, all NaN values are replaced with the mean of their respective columns, a standard imputation technique to prevent missing values from influencing the model's performance. Subsequently, standardization involves scaling the features to have a mean of 0 and a standard deviation of 1, which is essential for many machine learning models, especially those that rely on distance calculations, as it ensures that all features have an equal contribution to the model.

4.5. Utilizing the CBLOF Model and SMOTE for Dataset Preparation and Balancing

The data was used to train the CBLOF model to find the outliers. Later on, the scores about outliers received from the CBLOF model were additional features in the dataset, turning this into an additional knowledge dataset. In that way, it is supposed to be by emphasizing data points that might be anomalous. Fig. 2 shows the data point before and after applying

CBLOF. The enriched feature set, a combination of the original features and the outlier score obtained from CBLOF, would enhance the model's power to discriminate between normal and abnormal data points. Since there is an imbalance in the number of normal versus abnormal data points, thus unbalanced classes, it becomes necessary to use SMOTE to even out the disparity by creating synthetic samples for the minority class. This becomes very important when the class imbalance is so great that one class markedly outweighs the other. Given these preprocessing steps, the code does tend to prepare data that could be used in perfect conjunction with machine-learning algorithms. This will not only enable the model to work effectively, but also make respective data sensitive to alteration for proper critical analysis.

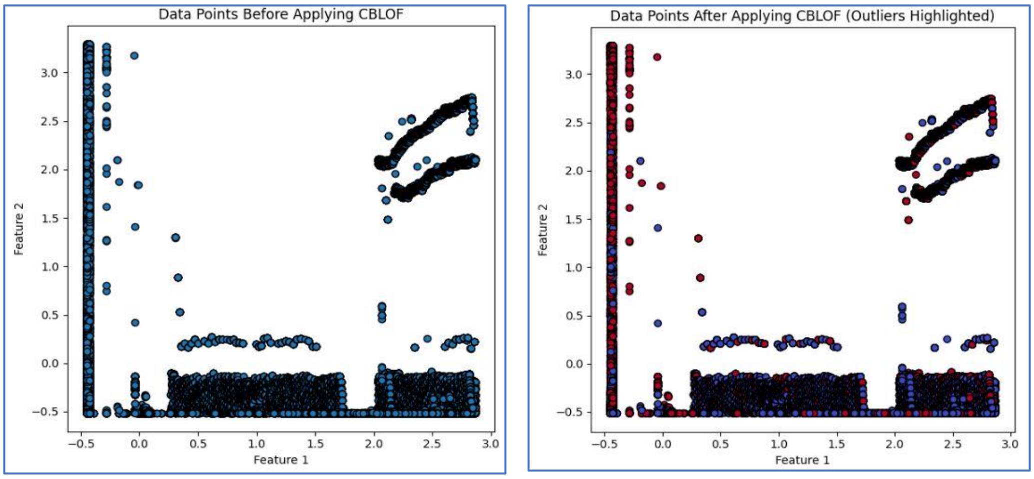


Figure 2. The features of data points before and after applying the CBLOF

5. Implementation and Results

This portion provides a detailed account of the methodology employed in conducting the experiments, the dissemination of the results, and the examination and discourse of other related studies.

5.1. Experimental Environment

The analysis carried out in this investigation utilized Google Colab, an internet-based platform that offers a cloud-based setting for executing Python code. Colab was chosen for its convenience in offering no-cost access to high-speed computing resources, such as GPUs and TPUs, which are advantageous for machine learning assignments. The essential attributes of the experimental setting are outlined in Table 2.

Table 2. The key aspects of the experimental environment used in the proposed approach.

Component	Details
Platform	Google Colab
CPU	2.3 GHz quad-core Intel Core i7
RAM	16 GB
GPU	NVIDIA Tesla K80 (12 GB)
Operating System	Linux-based environment (provided by Colab)
Python Version	Python 3.8
Libraries Used	XGBoost 1.4.2, Scikit-learn 0.24.2, Pandas 1.3.3, Numpy 1.19.5, Matplotlib 3.4.3
Datasets	CICIDS2017, CICIDS2018
Preprocessing	Replaced infinite values with NaNs, handled missing values through mean imputation, feature scaling with StandardScaler
Metrics Evaluated	Accuracy, Precision, Recall, F1 Score

The CICIDS2017 and CICIDS2018 datasets were employed to train and test the models. Before utilization, the dataset underwent preprocessing wherein infinite values were replaced with NaNs, and missing values were addressed through mean imputation. Feature scaling was then carried out using StandardScaler to normalize the contribution of all features to the model's learning process. The experiments were divided into multiple stages, including data preprocessing, model training, and evaluation. The model performance was assessed using accuracy, precision, recall, and F1 score metrics.

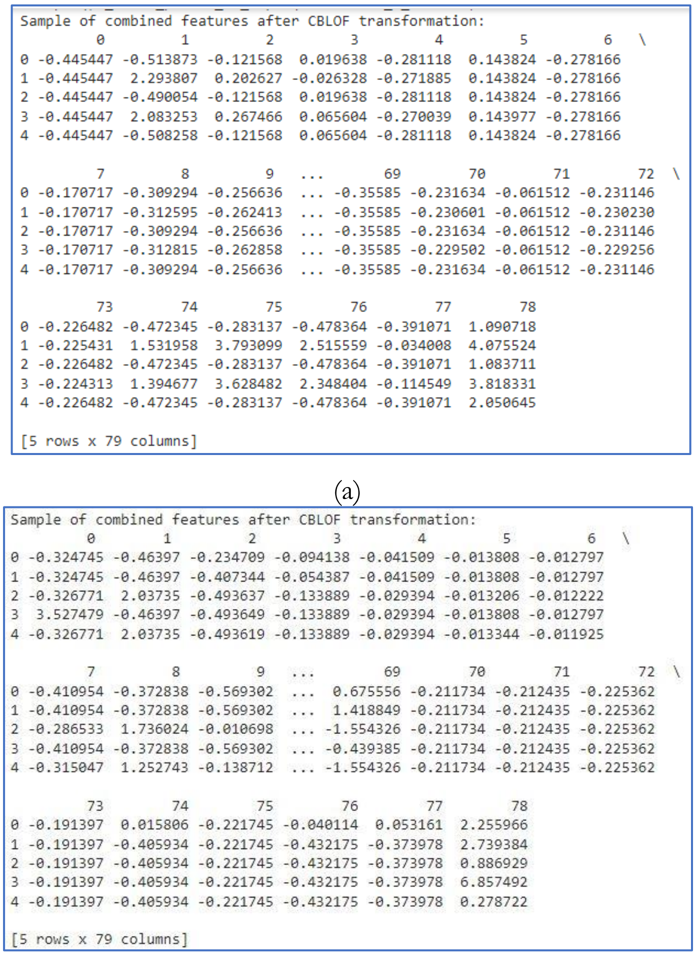
5.2. Performance of Hybrid Approach

The hybrid model underwent training and testing using the CICIDS2017 with (225745 track, 79 feature) and CICIDS2018 with (113269, 79) datasets, a commonly utilized resource for assessing the performance of intrusion detection systems. The hybrid model combines the benefits of CBLOF and XGBoost to tackle the difficulties related to identifying DDoS attacks in intricate network setups. CBLOF is utilized first to pinpoint potential anomalies in the network traffic. CBLOF is proficient in detecting peculiar traffic patterns that could signify a DDoS attack through data clustering and assigning outlier scores derived from cluster properties. To improve the results of CBLOF, the (GridSearchCV) technique was used to systematically search for the optimal hyperparameters by trying out different combinations, evaluating each one, and selecting the combination that results in the best performance according to a specified metric. The following CBLOF optimal Parameters were identified:

{'alpha': 0.5, 'beta': 3, 'contamination': 0.01}

Where alpha: Controls the relative importance of cluster size versus distance; beta: Determines the number of clusters; contamination: Specifies the expected proportion of outliers in the data.

The number of features in the dataset is 78, combined with the outlier scores generated by the CBLOF model to become 79 features. Fig. 3 shows a sample of combined features after CBLOF transformation.



```
Sample of combined features after CBLOF transformation:
         0         1         2         3         4         5         6  \
0 -0.445447 -0.513873 -0.121568  0.019638 -0.281118  0.143824 -0.278166
1 -0.445447  2.293807  0.202627 -0.026328 -0.271885  0.143824 -0.278166
2 -0.445447 -0.490054 -0.121568  0.019638 -0.281118  0.143824 -0.278166
3 -0.445447  2.083253  0.267466  0.065604 -0.270039  0.143977 -0.278166
4 -0.445447 -0.508258 -0.121568  0.065604 -0.281118  0.143824 -0.278166

         7         8         9  ...       69        70        71        72  \
0 -0.170717 -0.309294 -0.256636  ... -0.35585 -0.231634 -0.061512 -0.231146
1 -0.170717 -0.312595 -0.262413  ... -0.35585 -0.230601 -0.061512 -0.230230
2 -0.170717 -0.309294 -0.256636  ... -0.35585 -0.231634 -0.061512 -0.231146
3 -0.170717 -0.312815 -0.262858  ... -0.35585 -0.229502 -0.061512 -0.229256
4 -0.170717 -0.309294 -0.256636  ... -0.35585 -0.231634 -0.061512 -0.231146

        73        74        75        76        77        78
0 -0.226482 -0.472345 -0.283137 -0.478364 -0.391071  1.090718
1 -0.225431  1.531958  3.793099  2.515559 -0.034008  4.075524
2 -0.226482 -0.472345 -0.283137 -0.478364 -0.391071  1.083711
3 -0.224313  1.394677  3.628482  2.348404 -0.114549  3.818331
4 -0.226482 -0.472345 -0.283137 -0.478364 -0.391071  2.050645

[5 rows x 79 columns]
```

(a)

```
Sample of combined features after CBLOF transformation:
         0        1         2         3         4         5         6  \
0 -0.324745 -0.46397 -0.234709 -0.094138 -0.041509 -0.013808 -0.012797
1 -0.324745 -0.46397 -0.407344 -0.054387 -0.041509 -0.013808 -0.012797
2 -0.326771  2.03735 -0.493637 -0.133889 -0.029394 -0.013206 -0.012222
3  3.527479 -0.46397 -0.493649 -0.133889 -0.029394 -0.013808 -0.012797
4 -0.326771  2.03735 -0.493619 -0.133889 -0.029394 -0.013344 -0.011925

         7         8         9  ...        69        70        71        72  \
0 -0.410954 -0.372838 -0.569302  ...  0.675556 -0.211734 -0.212435 -0.225362
1 -0.410954 -0.372838 -0.569302  ...  1.418849 -0.211734 -0.212435 -0.225362
2 -0.286533  1.736024 -0.010698  ... -1.554326 -0.211734 -0.212435 -0.225362
3 -0.410954 -0.372838 -0.569302  ... -0.439385 -0.211734 -0.212435 -0.225362
4 -0.315047  1.252743 -0.138712  ... -1.554326 -0.211734 -0.212435 -0.225362

        73        74        75        76        77        78
0 -0.191397  0.015806 -0.221745 -0.040114  0.053161  2.255966
1 -0.191397 -0.405934 -0.221745 -0.432175 -0.373978  2.739384
2 -0.191397 -0.405934 -0.221745 -0.432175 -0.373978  0.886929
3 -0.191397 -0.405934 -0.221745 -0.432175 -0.373978  6.857492
4 -0.191397 -0.405934 -0.221745 -0.432175 -0.373978  0.278722

[5 rows x 79 columns]
```

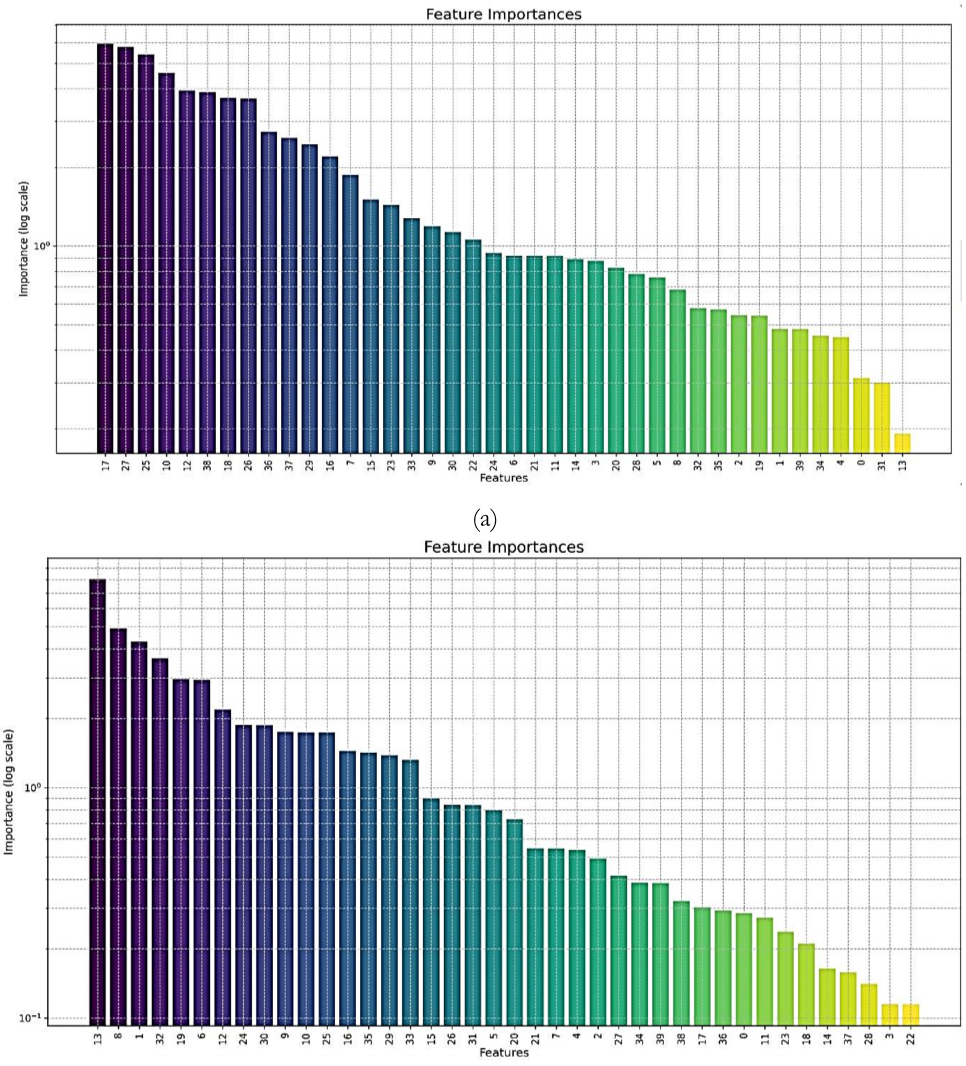
(b)

Figure 3. Features after CBLOF transformation (**a**) CICIDS2017; (**b**) CICIDS2018.

Once the potential outliers are recognized, XGBoost is utilized to categorize these outliers as either benign or malicious traffic. Renowned for its resilience and exceptional precision in managing extensive datasets, XGBoost incorporates the outlier scores from CBLOF as at-tributes to improve the classification process. The role of the XGBoost in enhancing the results is clear because the data is unbalanced (more benign traffic than DDoS attacks), and the presence of this disparity makes the prediction or expectation biased, encouraging the use of the hybrid model. Integrating these two methodologies enables the precise identification of established and new attack patterns, rendering the model especially efficient in dynamic and varied network settings. Cross-validation 5-fold was used on the training set to ensure that the model gives good results by testing it on multiple subsets of the data. The mean cross-validation accuracy was (0.99). SHAP explains the model's decisions and which features are driving the model's predictions for each dataset as shown in Fig.4.

The features on the graph's left side (displaying the highest bars) hold the utmost significance for the model's forecasts. These attributes exert the most substantial impact on the model's outputs. Conversely, the features on the right (with the lowest bars) have minimal impact on the model's predictions and may be less critical for the model's decision-making process.

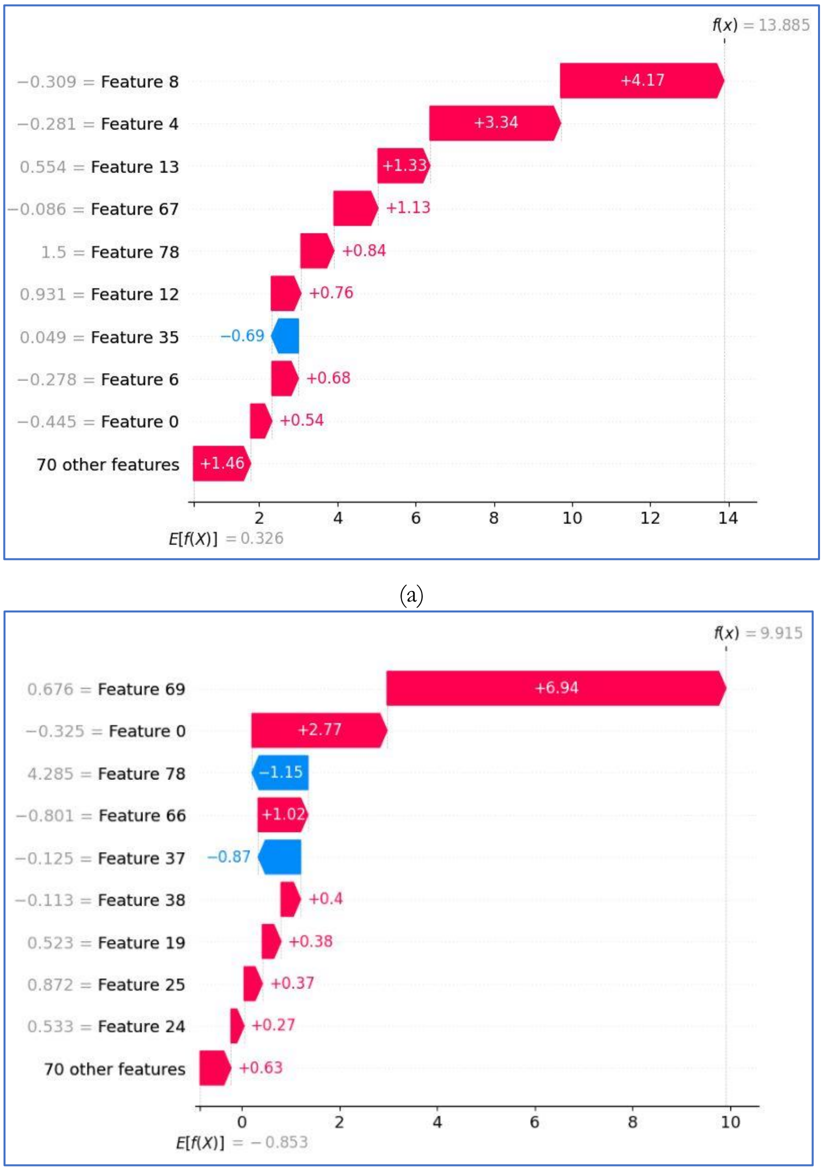
Fig.5 provides a visual representation of the model's process in arriving at its ultimate prediction, displaying each dataset's individual contributions of features. Each feature's influence is accurately measured, enabling observation of which features are responsible for the prediction and whether they have a positive or negative effect.



(a)

(b)

Figure 4. Features importance (**a**) CICIDS2017; (**b**) CICIDS2018.



(b)

Figure 5 The prediction of a single instance (**a**) CICIDS2017; (**b**) CICIDS2018.

Where the base value (E[f(X)] = 0.326, E[f(X)] =- 0.853) value is the average prediction that the model would make across the entire dataset if no specific feature values were known, it represents the model's baseline prediction. Positive contributions (Red Arrows) These features push the prediction higher than the base value, while negative contributions (Blue Arrows) these features pull the prediction lower than the base value. Table 3 shows a sample of feature descriptions that positively affect model prediction.

Table 3. Positive feature description

Feature Number	Description
8	Fwd Packet Length Mean
4	Total Length of Fwd Packets
0	Destination Port
6	Fwd Packet Length Max
12	Bwd Packet Length Mean
69	act_data_pkt_fwd
66	Subflow Bwd Bytes

5.2.1. Accuracy and Precision

The hybrid model demonstrates an exceptional accuracy of 99.99%, signifying its ability to accurately differentiate between attack and benign traffic in almost every scenario. This high accuracy level highlights the hybrid approach's effectiveness in managing diverse and intricate network traffic. The model attains a 100% level of precision, as shown in Fig.6, 7 indicating that all instances identified as attacks by the model are, in fact, attacks. This is essential in reducing incorrect identifications, guaranteeing that valid traffic is not erroneously categorized as malicious.

5.2.2. Recall and F1-Score

The model has achieved a recall rate of 100%, as shown in Fig.6, effectively identifying all genuine DDoS attacks in the dataset. This high recall value holds particular significance in a security setting, as a failure to detect an attack could result in considerable operational disruption and financial harm. The F1-score, as shown in Fig.7, effectively weighs precision and recall and is also at a 100% level. This indicates that the hybrid model is performing at an optimal level in identifying and accurately classifying DDoS attacks while maintaining an ideal balance in minimizing false positives and false negatives.

The proposed hybrid approach's performance metrics highlight its efficiency and dependability in identifying DDoS attacks within network traffic. By integrating CBLOF and XGBoost, the method not only improves accuracy but also guarantees thorough and accurate detection, rendering it highly suitable for real-time intrusion detection systems.

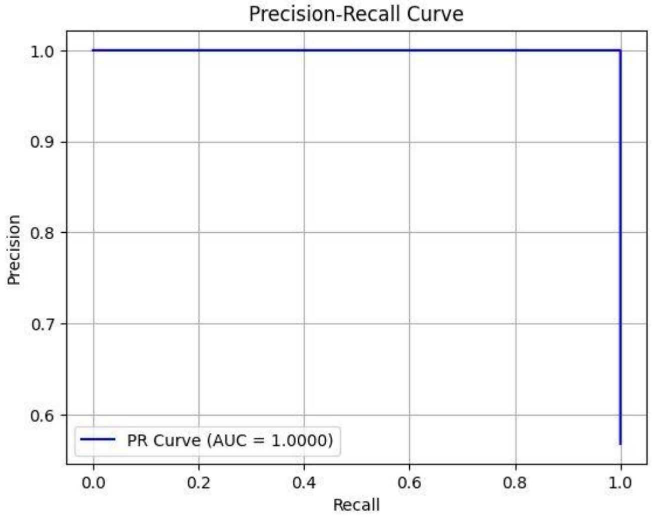


Figure 6. The hybrid approach precision, and recall

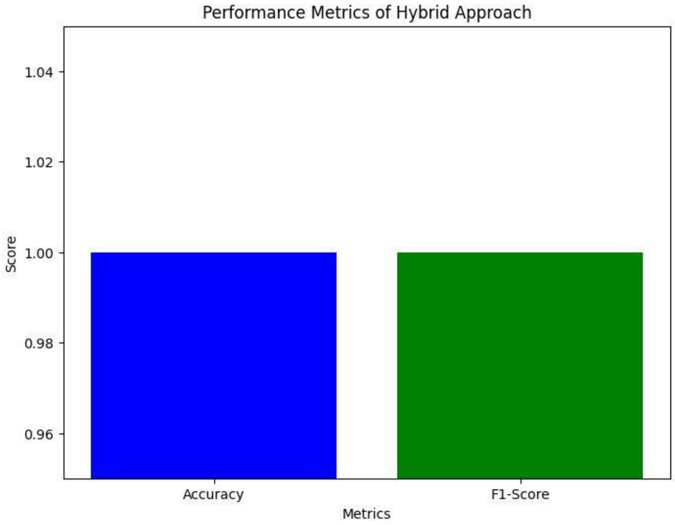


Figure 7. Performance Metrics of Hybrid Approach Accuracy and F1-Score

5.3. Performance Analysis

The hybrid method under consideration integrates the CBLOF with XGBoost. It underwent evaluation using the CICIDS2017 and CICIDS2018 datasets. The assessment primarily examines essential metrics such as accuracy, precision, recall, and F1-score. The work proposes a hybrid model, combining the CBLOF method with XGBoost to improve the accuracy in detecting DDoS attacks. For this purpose, an ablation experiment was performed to understand the importance of CBLOF in this hybrid framework. Further, based on the four different configurations, the effectiveness of the model was analyzed as follows:

1. IsoForest (isolation forest): The baseline unsupervised anomaly detection technique.
2. LOF (Local Outlier Factor): Another technique of a baseline identifies anomalies concerning local densities.
3. HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a hierarchical clustering method that extends DBSCAN and works with data of varying densities, often found in network traffic.
4. CBLOF (Clustering-Based Local Outlier Factor): the proffered model's most salient feature of engineering.

The important parameters used in each method are illustrated in Table 4.

Table 4. Methods parameters.

Method	Parameters
Isoforest	n_estimators=500, contamination=0.1, random_state=42
LOF	contamination=0.1
HDBSCAN	min_cluster_size=15, gen_min_span_tree=True
CBLOF	contamination=0.1, random_state=42, alpha=0.8, beta=5

The summary of these experiments is given in Table 5, comparing three methods in the way of important performance measures: accuracy, precision, recall, and the F1 score.

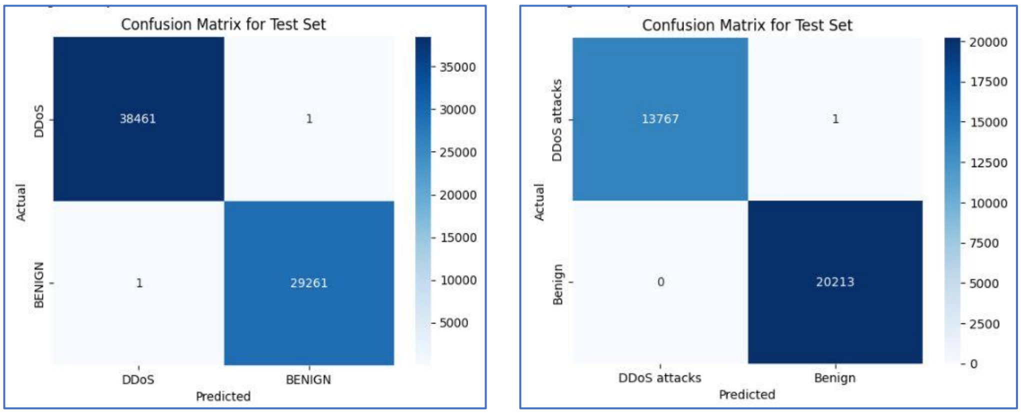
Table 5. Effect of various methods on detection performance.

Method	Accuracy	Precision	Recall	F1-score	Computational Time
Isoforest	64%	69%	64%	65%	60 sec.
LOF	87%	88%	87%	86%	16 Min
HDBSCAN	78%	74%	78%	71%	56 Min
CBLOF	99.99%	100%	100%	100%	40 sec.

The performing superior method is CBLOF because it has perfect performance measures and the fastest computation time; hence, if the task needs speed and proper accuracy, this will serve best. LOF provides very high accuracy with good detection performance but at the cost of much longer computing times. It is thus preferable when computing time is less of a problem. HDBSCAN has a medium level of performance. It is the second slowest and less suitable for application in real-time. The Isolation Forest ranks last for nearly all detection metrics and speeds most of the time.

CBLOF was expected to extract from network traffic those small differences other traditional anomaly detection methods would discard as noise. In fact, the ablation study shows that high detection performance can be achieved without using CBLOF, but this will not reach the very best levels accomplished by incorporating it into the working of the classifier. This confirms the hypothesis and indicates how important CBLOF is for further boosting the hybrid model.

The confusion matrix displayed in Fig.8 illustrates the efficacy of the combined CBLOF and XGBoost model in identifying DDoS attacks and legitimate network traffic. This matrix provides a summary of important performance indicators. Where instances of DDoS and BENGIN attacks were correctly identified, this confusion matrix indicates excellent model performance.



(a) (b)

Figure 8. Confusion Matrix of the Hybrid CBLOF + XGBoost Model (**a**) CICIDS2017; (**b** CICIDS2018.

Table 6 compares this study's work with similar studies focusing on DDoS detection using different machine learning techniques, highlighting the superior performance of the proposed CBLOF + XGBoost hybrid model.

Table 6. Comparison of DDoS Detection Methods and Their Performance Metrics

Ref	Method Used	Results
Proposed	CBLOF + XGBoost Hybrid	Accuracy: 99.99%, Precision: 100%, Recall: 100%, F1-score: 100%
[2]	Sparse Convolution Network + Bayesian Networks for DDoS detection	Maximum detection rate of 98.9%, 99.29% recognition accuracy for classifying normal/abnormal
[3]	Feature engineering with Grey Wolf Optimization + ML classifiers (SVM, RF)	Random Forest outperformed others; effective for detecting DDoS in SDN with high accuracy
[4]	Logistic Regression, CNN, XGBoost, Naive Bayes, AdaBoost, Random Forest	XGBoost achieved 99.9999% detection accuracy using SMOTE
[6]	XGBoost, Random Forest, ANN + SMOTE for classification	Achieved 99.99% and 100% accuracy
[8]	SVM classification + SNORT IPS integration for DDoS prevention	Achieved an average accuracy rate of 97%
[10]	Naive Bayes, SVM, Logistic Regression	Achieved 95.94% accuracy in DDoS detection
[12]	Random Forest, Decision Tree, AdaBoost, XGB, MLP, DNN	Random Forest classifier achieved 99.97%, and Decision Tree achieved 99.88%
[15]	XGBoost classifier for detection	Detection accuracy: 95.11% for UNSW-NB15, 99.92% for CICIDS2017
[18]	XGBoost for imbalanced multiclass IoT datasets	XGBoost achieved F1 scores of 99.9% and 99.87% on two datasets

As is evident from Table 6, the combined approach of CBLOF and XGBoost demonstrates a remarkable accuracy of 99.99%, comparable to or slightly surpassing other techniques, including Random Forest (99.97%) and XGBoost classifiers (99.92%). These findings underscore the effectiveness of the hybrid method in differentiating between authentic traffic and DDoS attacks, indicating its resilience and applicability for real-time use. Although deep learning techniques like Sparse Convolution Networks achieve a 98.9% accuracy rate [2], the hybrid CBLOF + XGBoost method outperforms them in both accuracy and F1-score. This indicates that the suggested hybrid approach is better equipped for detecting intricate attack patterns, particularly in network traffic characterized by significant variability.

To clarify the performance of the proposed (CBLOF + XGBoost) hybrid approach in terms of performance metrics like accuracy, precision, recall, and F1 score. Table 7 compares the proposed approach's performance with other similar studies.

Table 7. Performance Comparison of Various DDoS Detection Techniques.

Ref	Method Used	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
[2]	Sparse Convolution Network + Bayesian Networks	98.9	99.29	N/A	N/A
[3]	Feature Engineering with Grey Wolf Optimization + ML Classifiers (SVM, RF)	99.9	N/A	N/A	N/A
[4]	Logistic Regression, CNN, XGBoost, Naive Bayes, AdaBoost, RF	99.9999	N/A	N/A	N/A
[6]	XGBoost, RF, ANN + SMOTE for classification	99.99	100	100	N/A
[8]	SVM Classification + SNORT IPS Integration	97	N/A	N/A	N/A
[10]	Naive Bayes, SVM, Logistic Regression	95.94	N/A	N/A	N/A
[12]	RF, Decision Tree, AdaBoost, XGB, MLP, DNN	99.97 (RF), 99.88 (DT)	N/A	N/A	N/A
[15]	XGBoost Classifier for Detection	99.92	N/A	N/A	N/A
Proposed	CBLOF + XGBoost	99.99	100	100	100

By leveraging the outlier identification abilities of CBLOF and the proficient classification capabilities of XGBoost, the hybrid model demonstrates exceptional performance in detecting nuanced anomalies and their accurate classification. This combined capability surpasses models that depend solely on a single approach, such as the standalone XGBoost classifier, which exhibited slightly lower detection accuracy at 99.92%.

6. Conclusion and Future Directions

This study illustrates the effectiveness of combining CBLOF and XGBoost for detecting DDoS attacks in network traffic. The hybrid model performs significantly better than the individual models, achieving high accuracy, precision, recall, and F1 scores. By utilizing the clustering capabilities of CBLOF to identify outliers and the classification abilities of XGBoost, the proposed model not only identifies known attack patterns but also detects new anomalies in network traffic. This makes the hybrid approach a reliable solution for real-time DDoS detection. Despite its superior performance, the computational complexity of this method presents a challenge for deployment in real-time scenarios, necessitating further optimization to ensure its practicality in high-traffic environments. Subsequent research efforts should be directed toward resolving the computational constraints by enhancing the algorithm for real-time purposes. Furthermore, broadening the scope of the investigation to encompass alternative datasets and forms of attacks would contribute to establishing the model's versatility.

Funding: This research received no external funding.

Data Availability Statement: The dataset used is a public dataset that was published by [25], [26]. Direct access to its official repository is available at: https://www.unb.ca/cic/datasets.

Acknowledgments: Special thanks to my friend Nidaa Ghalib Ali for her unwavering support.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] P. Kumari and A. K. Jain, "A comprehensive study of DDoS attacks over IoT network and their countermeasures," *Comput. Secur.*, vol. 127, p. 103096, Apr. 2023, doi: 10.1016/j.cose.2023.103096.

[2] M. H. Ali *et al.*, "Threat Analysis and Distributed Denial of Service (DDoS) Attack Recognition in the Internet of Things (IoT)," *Electronics*, vol. 11, no. 3, p. 494, Feb. 2022, doi: 10.3390/electronics11030494.

[3] Z. Liu, Y. Wang, F. Feng, Y. Liu, Z. Li, and Y. Shan, "A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks," *Sensors*, vol. 23, no. 13, p. 6176, Jul. 2023, doi: 10.3390/s23136176.

[4] A. Golduzian, "Predict And Prevent DDOS Attacks Using Machine Learning and Statistical Algorithms," *arXiv*. 2023. doi: 10.48550/arXiv.2308.15674.

[5] D. R. I. M. Setiadi, S. Widiono, A. N. Safriandono, and S. Budi, "Phishing Website Detection Using Bidirectional Gated Recurrent Unit Model and Feature Selection," *J. Futur. Artif. Intell. Technol.*, vol. 2, no. 1, pp. 75–83, 2024, doi: 10.62411/faith.2024-15.

[6] M. A. Talukder *et al.*, "A dependable hybrid machine learning model for network intrusion detection," *J. Inf. Secur. Appl.*, vol. 72, p. 103405, Feb. 2023, doi: 10.1016/j.jisa.2022.103405.

[7] H. Karthikeyan and G. Usha, "Real-time DDoS flooding attack detection in intelligent transportation systems," *Comput. Electr. Eng.*, vol. 101, p. 107995, Jul. 2022, doi: 10.1016/j.compeleceng.2022.107995.

[8] R. Abubakar *et al.*, "An Effective Mechanism to Mitigate Real-Time DDoS Attack," *IEEE Access*, vol. 8, pp. 126215–126227, 2020, doi: 10.1109/ACCESS.2020.2995820.

[9] M. Snehi and A. Bhandari, "Vulnerability retrospection of security solutions for software-defined Cyber–Physical System against DDoS and IoT-DDoS attacks," *Comput. Sci. Rev.*, vol. 40, p. 100371, May 2021, doi: 10.1016/j.cosrev.2021.100371.

[10] S. J., "Assessing DDoS Detection Accuracy through Semi-Supervised Techniques," *Indian Sci. J. Res. Eng. Manag.*, vol. 08, no. 03, pp. 1–5, Mar. 2024, doi: 10.55041/IJSREM29861.

[11] A. Srivastava, S. Tiwari, D. Kumar, and N. Garg, "Finding of DDoS Attack in IoT-Based Networks Using Ensemble Technique," in *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, May 2024, pp. 1–4. doi: 10.1109/ISCS61804.2024.10581044.

[12] F. L. Becerra-Suarez, I. Fernández-Roman, and M. G. Forero, "Improvement of Distributed Denial of Service Attack Detection through Machine Learning and Data Processing," *Mathematics*, vol. 12, no. 9, p. 1294, Apr. 2024, doi: 10.3390/math12091294.

[13] J. Deng, L. Cheng, H. Yuan, K. Zheng, X. Li, and Q. Li, "An Online Detection System for LDoS attack Based on XGBoost," in 2023 IEEE Intl Conf on Parallel &amp; Distributed Processing with Applications, Big Data &amp; Cloud Computing, Sustainable Computing &amp; Communications, Social Computing &amp; Networking (ISPA/BDCloud/SocialCom/SustainCom), Dec. 2023, pp. 1083–1088. doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom59178.2023.00174.

[14] S. Ullah *et al.*, "TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT Networks," *Comput. Networks*, vol. 237, p. 110072, Dec. 2023, doi: 10.1016/j.comnet.2023.110072.

[15] W. Xu and Y. Fan, "Intrusion Detection Systems Based on Logarithmic Autoencoder and XGBoost," *Secur. Commun. Networks*, vol. 2022, pp. 1–8, Apr. 2022, doi: 10.1155/2022/9068724.

[16] S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective Intrusion Detection System Using XGBoost," *Information*, vol. 9, no. 7, p. 149, Nov. 2018, doi: 10.3390/info9070149.

[17] D. M. Sindika, M. R. Nicholaus, and N. B. Hamadi, "Improving Network Security: An Intrusion Detection System (IDS) Dataset from Higher Learning Institutions, Mbeya University of Science and Technology (MUST), Tanzania," *East African J. Inf. Technol.*, vol. 7, no. 1, pp. 23–38, Jan. 2024, doi: 10.37284/eajit.7.1.1679.

[18] T.-T.-H. Le, Y. E. Oktian, and H. Kim, "XGBoost for Imbalanced Multiclass Classification-Based Industrial Internet of Things Intrusion Detection Systems," *Sustainability*, vol. 14, no. 14, p. 8707, Jul. 2022, doi: 10.3390/su14148707.

[19] R. Devarajan and P. Rao, "An Efficient Intrusion Detection System by Using Behaviour Profiling and Statistical Approach Model," *Int. Arab J. Inf. Technol.*, vol. 18, no. 1, pp. 114–124, Dec. 2020, doi: 10.34028/iajit/18/1/13.

[20] P. TS and P. Shrinivasacharya, "Evaluating neural networks using Bi-Directional LSTM for network IDS (intrusion detection systems) in cyber security," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 448–454, Nov. 2021, doi: 10.1016/j.gltp.2021.08.017.

[21] M. Ozkan-Okay, R. Samet, O. Aslan, and D. Gupta, "A Comprehensive Systematic Literature Review on Intrusion Detection Systems," *IEEE Access*, vol. 9, pp. 157727–157760, 2021, doi: 10.1109/ACCESS.2021.3129336.

[22] M. R. Ayyagari, N. Kesswani, M. Kumar, and K. Kumar, "Intrusion detection techniques in network environment: a systematic review," *Wirel. Networks*, vol. 27, no. 2, pp. 1269–1285, Feb. 2021, doi: 10.1007/s11276-020-02529-3.

[23] L. Zheng, L. Chen, and Y. Wang, "A new unsupervised outlier detection method," *J. Intell. Fuzzy Syst.*, vol. 46, no. 1, pp. 1713–1734, Jan. 2024, doi: 10.3233/JIFS-236518.

[24] C. N. Obiora, A. N. Hasan, and A. Ali, "Predicting Solar Irradiance at Several Time Horizons Using Machine Learning Algorithms," *Sustainability*, vol. 15, no. 11, p. 8927, Jun. 2023, doi: 10.3390/su15118927.

[25] Canadian Institute for Cybersecurity, "Intrusion detection evaluation dataset (CIC-IDS2017)," *Canadian Institute for Cybersecurity*. https://www.unb.ca/cic/datasets/ids-2017.html

[26] Communications Security Establishment (CSE) and Canadian Institute for Cybersecurity (CIC), "CSE-CIC-IDS2018 on AWS," *Canadian Institute for Cybersecurity*. https://www.unb.ca/cic/datasets/ids-2018.html

[27] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116. doi: 10.5220/0006639801080116.