IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 November 2023, accepted 5 January 2024, date of publication 10 January 2024, date of current version 24 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3352281

APPLIED RESEARCH

An Effective Classification of DDoS Attacks in a Distributed Network by Adopting Hierarchical Machine Learning and Hyperparameters Optimization Techniques

SANDEEP DASARI AND RAJESH KALURI

School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore 632014, India

Corresponding author: Rajesh Kaluri (rajesh.kaluri@vit.ac.in)

This work was supported by the Vellore Institute of Technology, Vellore, India.

ABSTRACT Data privacy is essential in the financial sector to protect client's sensitive information, prevent financial fraud, ensure regulatory compliance, and safeguard intellectual property. It has become a challenging task due to the increase in usage of the internet and digital transactions. In this scenario, DDoS attack is one of the major attacks that makes clients' privacy questionable. It requires effective and robust attack detection and prevention techniques. Machine Learning (ML) is the most effective approach for employing cyber attack detection systems. It paves the way for a new era where human and scientific communities will benefit. This paper presents a hierarchical ML-based hyperparameter-optimization approach for classifying intrusions in a network. CICIDS 2017 standard dataset was considered for this work. Initially, data was preprocessed with the min-max scaling and SMOTE methods. The LASSO approach was used for feature selection, given as input to the hierarchical ML algorithms: XGboost, LGBM, CatBoost, Random Forest (RF), and Decision Tree (DT). All these algorithms are pretrained with hyperparameters to enhance the effectiveness of algorithms. Models performance was assessed in terms of recall, precision, accuracy, and F1-score metrics. Evaluated approaches have shown that the LGBM algorithm gives a proven performance in classifying DDoS attacks with 99.77% of classification accuracy.

INDEX TERMS Machine learning, hyperparameter optimization, classification, cyberattacks, intrusion detection.

I. INTRODUCTION

Modern advancements in technology have motivated the progress of numerous domains like medicine, industries, communications, etc. Because of the digitization of several key essential service sectors such as insurance, power, mobile recharge, banking, and telephone bill payments, etc. It indirectly forces individuals to rely on online and mobile banking. The rise in digital users has led to growth in various services, and this association is growing more robust than ever.

The internet and data repositories have not only become the core of social and modern life, but they also store a considerable number of records concerning people's personal information and national security. In a network, when an attack or intrusion occurs, normal operations inevitably disrupt national and personal data security will be compromised. Consequently, network security has become increasingly important, and cyber security has attracted the attention of an increasing number of individuals [1]. One of the solutions in emerging security defense technologies is an intrusion detection system (IDS) [2].

DDoS is one of the attacks commonly observed in most networks. In this attack, by sending multiple requests to the server, the attackers place a heavy load of requests on the

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

user server. These enormous requests by the attacker overload the affected server's bandwidth, making it inaccessible to authorized users. Identification of DDoS attacks is necessary to provide the right assistance to authorized users. DDoS can be launched from various sources, which makes pinpointing the attack's origin difficult. IDS can assist in detecting and preventing DDoS attacks by monitoring network traffic and finding patterns that indicate any attack. It can also assist in determining the type of attack and providing recommendations on how to mitigate it. Furthermore, it helps organizations to meet legal obligations and protect sensitive data from unauthorized access. As a result, it is critical to design robust intrusion detection systems capable of detecting and preventing DDoS attacks in real-time. This study seeks to investigate the efficacy of different intrusion detection and mitigation strategies in detecting and mitigating DDoS attacks and to give insights into creating more resilient and efficient IDS systems. To support this, a case study is also included in this work.

Several research works contributed to detecting these attacks with better accuracy for different networks. Ransomware attacks hit the banking sector hard in the first half of 2021 [3]. During COVID-19, a four percent increase in business email compromise attacks occurred. Banks are becoming increasingly vulnerable to large-scale cyber attacks [4]. Because of the interconnectedness of the banks, a cyber attack on one bank could endanger the economic viability of another.

Cybercrimes have progressively grown over time as more people use mobile banking and the internet [5]. This led to a rise in fraudulent online activities such as credit card fraud, online identity theft, insurance fraud, banking fraud, and money laundering. DDoS approaches are challenging to test and implement because of the complex nature, toughness, and cost of current network infrastructure and protocols. Networks encounter obstacles differentiating between legal and malicious flows. ML has effective strategies to mitigate these attacks.

ML is a part of artificial intelligence, which helps the machine to learn intended tasks by itself from past experiences [6]. It includes mainly three different types of algorithms. The algorithm, which involves predicting dependent variables from a group of predictors, is supervised learning. On the other hand, an algorithm that does not involve a dependent variable for prediction is named as unsupervised learning. Reinforcement learning is an algorithm in which the machine makes certain decisions by learning from previous data and gives effective outcomes to make possible accurate decisions. To improve the prediction results more effectively, ensemble ML algorithms are developed. The main motto of this type of algorithm is to collect results from multiple models to make effective decisions rather than working as an individual model. The Xgboost, LGBM, and CatBoost algorithms come under this category.

ML models that have been properly trained will allow us to classify various malicious activities in traffic effectively. The purpose of inducing ML is varied [7]. They largely correspond to stakeholders' data objectives. These can be translated into enhanced results in medicine, boosting the study of the physical and life sciences, increasing manufacturing productivity, and giving businesses a competitive edge. For businesses to achieve a competitive edge, using data assets to produce value in addition to physical, financial, and human capital has become highly significant. The impact of ML on society cannot be understated, as it stands as the next frontier for intelligence, development, enhanced decision-making, and intrusion detection. Improving ML algorithms with hyperparameters optimizes resource utilizations and helps to achieve better performance results.

Hyperparameters are parameters that have been defined before the training phase and influence how the learning algorithm should work. These parameters are defined by the model designer. It also impacts the model's ability, training speed, regularization, and other conditions. It facilitates developing a trustworthy ML model and identifying the perfect set of hyperparameters, producing in an optimized model [8]. It intends to maximize generalization performance while minimizing error and achieving the best assessment metrics within the user-defined time budget, such as sensitivity, F1-score, accuracy, and specificity.

The following are main contributions of proposed work:

1) Proposal of a LASSO feature selection method using hierarchal-based ML models to analyze network attributes and attacks in a network.
2) Categorization of the working process into three modules: pre-processing, feature selection, and hyperparameter optimized tuning classification, which applies ML algorithms for features selected from the LASSO process.
3) Identification of the LGBM classifier as the best-performing classifier, achieving an accuracy of 99.77%.

A. PAPER ORGANISATION

This study employs tree-based methods (XGboost, LGBM, CatBoost, RF, and DT) to perform experiments on the CICIDS dataset. Section II presents the motivation for DDoS attack classification and the related works proposed in Section III. Current work is discussed in section IV, Outcomes in section V, case study in section VI, and conclusion and future works in section VII, respectively.

II. MOTIVATION

The initial DDoS intrusion was recognized by the Computer Incident Advisory Capability in the 20th century [9]. Microsoft stated that the DDoS attacks raised by about 25% between the first and fourth quarters of 2021 [10]. This serious DDoS incident has revealed the enormous danger related to DDoS attacks and has attracted the interest of

modern cybernetic societies. This attack has raised a vital argument about cyber security and its erratic behavior [11]. According to the NETSCOUT threat intelligence survey, there will be a significant rise in multi-vector DDoS attacks in the first half of 2020 [12]. As a result, there is a strong IDS need for current detection systems and DDoS attack prevention techniques.

Intrusion detection systems (IDS) are developed to investigate suspicious activities in a network. It enables one to know the actual intrusion detection on a network to reduce the influence of an attack. Attackers constantly develop new vulnerabilities and attack offensive methods to degrade the system. Acquiring user credentials that allow access to network resources and data is the primary objective of several attacks.

III. LITERATURE SURVEY

The privacy and integrity of network infrastructure and information systems are seriously threatened by DDoS attacks. It might be challenging to identify DDoS attacks before one can take measures to prevent them. Multiple strategies, such as machine learning techniques, are utilized in DDoS attack detection. ML approaches will detect these attacks, and the outcomes will be acceptable. Some of the major works are as follows.

A framework for detecting DDoS assaults based on the fog-to-node concept is suggested by Yahalom et al. [13]. Deep learning techniques are implemented using loT networks and analysis. They proposed a framework based on the NSAL-KDD, KDDCUP99, and ISCX datasets. Additionally, they compared the results with deep learning and shallow learning in diverse system environments. The proposed technique has obtained 99.2% accuracy with deep learning and 95.2% with shallow models. Khan et al. [14] implemented a two-staged IDS framework using a stacked autoencoder in combination with deep learning and soft-max. Initially, the network traffic is categorized into attacked or normal based on the probability score resultant; the result of the first step becomes an enhanced feature in the next phase to detect an attack. On the UNSW-NBI5 and KDD datasets, the accuracy of this work is 99.99% and 89.13%, respectively. Hameed et al. [15] presented a deep learning technique based on IDS to identify cyber threats. Yin et al. [16] proposed IDS based on RNN. The authors compared the results with the proposed algorithm and other conventional methods for classifying data outcomes, including J48, random forest, and naive Bayesian. The experimental evaluation of the KDD dataset has attained a resultant accuracy of 99.81%.

Diro and Chilamkurti in [17] worked on an LSTM-based fog-tithing network method for loT networks. Each loT node determines training and detection locally in the proposed system, while fog nodes coordinate with the cloud and other integrated nodes to compute and distribute model updates. This approach is assessed using the ISCX dataset with 15 iterations and 128 batch sizes. This methodology has achieved 99.91% accuracy and 98.22% for binary, and multi-class classification, respectively. Adaptive learning for malware detection was suggested by Su et al. [18] KDD dataset was used from the online database. KNN classifiers, RF, and DT are used. The findings revealed that ensemble models and DT produced good categorization outcomes during this research. The suggested technique has an accuracy rating of 85%.

Laghrissi et al. [19] implemented deep learning approaches for de-detecting violations in a network using principal component analysis applied for dimensionality reduction and Long Short-Term Memory for feature selection techniques. Hasan et al. [20] analyzed the interpretation of IoT system performance of ML algorithms used to find anomalies and forecast attacks. They compared the performance of DT, LR, RF, SVM, and ANN with respect to F1-score, precision, recall, and accuracy. They found that RF achieves better performance than other algorithms. Nagaraja et al. [21] worked on a combined deep-learning model for identifying intrusions. In this work, the authors merged two different deep learning models for CNN and LSTM classification from the RNN model. Operations are applied to the KDD dataset. They observed that the recommended strategies had an average accuracy of 85.14%.

Behal and Kumar [22] authors worked on five classes and 27 features. DDoS attacks were identified using ML models: SVM, decision tree, and MLP. It produces the best results since it cannot include additional types of modern attacks and examine the various features for feature selection. Hasan et al. [20] have done work on tracking the IDS effectiveness by using ML models. The CICDDoS2019 dataset was evaluated using ANN, SVM, gaussian naive bayes, KNN, bernoulli naive bayes, multinomial naive bayes, RF, LR, and decision tree approaches. The best accuracy results were evaluated using LR, KNN, and naive bayes. Cil et al. [23] applied deep neural networks to identify DDoS attacks by using a random selection of traffic recorded over the internet and worked on DNN models for feature extractions. The authors applied the CICIDS 2019 dataset, which includes a variety of 2019 DDoS attack types, using the features of deep learning.

Motylinski et al. [24] evaluated the results of KNN, RF, logistic regression, and SVM classifiers that are graphics processing unit accelerated, in addition to the preprocessing processes used to create the training data. Due to the use of GPU-based feature selection and training models, the training and estimation times were significantly reduced. Akash et al. [25] implemented ML methods. They made a model that takes botnet identification into account. Their algorithms focused on malicious communities of IoT devices that are trying to connect to a network that could be caused by a botnet.

In another study, Asadi [26] introduced SVM and autoencoder LSTM techniques, which are considered to identify IoT botnet attacks using cooperative game theory. Compared to prior research, the proposed method improved recall by 11.629% and accuracy by 11.624%. To identify the

twitter bot accounts by applying both fundamental and derived characteristics. Gera and Sinha [27] worked on the T-Bot identification framework. The machine learning model parameters were considerably optimized to achieve optimality value by calculating the automation score for suspicious bots in online trend-centric communities. The suggested T-Bot, which employs a novel centroid initialization technique, reduces the effort required to find bots in trend-centric datasets, especially when working with imbalanced datasets, to increase the security of IoT-enabled devices utilized for smart city network traffic.

An attack identification framework using a chi-square feature selection and multi-class support vector machine was introduced by Thaseen and Kumar [28] proposed model optimizes the kernel parameter values of the radial basis function using parameter tuning technology. The key idea is to build a new multi-class SVM for detecting intrusions to minimize train and test timings and improve the classification efficiency of malicious activities. Ikram et al. [29] implemented a strong anomaly detection model by combining various deep neural network models such as backpropagation networks, multilayer preceptors, and long short-term memory. To achieve greater accuracy, the model employs XGboost to incorporate the outcomes of each deep learning model.

Yin et al. [16] employedÂ RNN as one of the DL models to perform binary and multiclass classifications. The findings from experiments indicate that binary classification has greater results than multiclass classification. Furthermore, they discovered that hyperparameters such as hidden neurons and learning rate influence detection accuracy.

Cheon et al. [30] suggested model consists of the Word2vec embedding layer, which has been pre-trained using the given datasets, and the GRU-LM. Excluding Bi-GRU, the accuracy score acquired from them was used to analyze the efficacy of every model. The proposed model scored 87.3%, which was approximately threefold greater than the other models. This study identified DDoS attacks using an LGBM with an accuracy rate of 100%.

Mohmand et al. [31] recommended intrusion detection using ML methodologies, KDD dataset, and supervised models to balance the data for improved performance. A comparison analysis was implemented in this work using distinct algorithms for classification, and good results were obtained. Laurens D'hooge et al. [32] suggested an extensive review of machine learning models for malware detection. This work compares malware datasets obtained from various online sources by applying different techniques. Authors discovered that ML-supervised algorithmsÂ are extremely flexible to detect attacks, which lets them make better decisions quickly. Assault detection is simplified by balancing the AE and DNN parameters are modeled for this purpose. The author of this article describes how to decrease model complexity and generate a dense network with fewer nodes to avoid overfitting. Islam et al. [12] The suggested strategy was compared with ten contemporary finest techniques using performance measures such as accuracy rate, precision, F1-Score, and recall. Several assessments were run on the KDD and CICIDS datasets to validate the results. In this work, existing methods are outperformed by the proposed method.

An innovative methodology was proposed by Salem et al. [33] for digital transactions, which can be used to test network attacks. To detect fraud, scoring accuracy model levels for offline and real-time logs are combined. A framework for huge data processing is described as a technique for looking through huge transaction logs using spark, Kafka, and MPP Gbase. The experiment shows that their recommended methodology works well on a voluminous dataset of digital financial transactions. Gupta et al. [34] investigated cybercrime datasets and identified accessible problems using K-Means, J48 Prediction Tree, and influenced association classifier. K-means clustering is used in influenced association classification with the J48 technique, which chooses the initial centroids using K-means selection. It can explore the record and identify whether cybercrime has occurred. Financial institution cybercrime can be investigated better and more appropriately by combining data from J48, K-Means, and influenced Association Classifier.

Elsayed et al. [35] suggested DDoSNet, an IDS for DDoS intrusions in SDN systems that use RNN as an encoder for selecting features and softmax process at the resultant layer with the null routing approach. CICDDoS2019 was introduced to test the model for binary class classification on DDoS. Rai and Mandoria [36] used Deep Neural Network (DNN), Gradient Boosting Tree, and linear classifiers to detect a network breach and train a multilayer model comprised of all classifiers using the KDD dataset. Gradient Boosting DT ensembles LGBM, multilayer models, and the XGboost achieved better results in this work. Bakhareva et al. [37] for detecting attacks, applied Logistic Regression, CatBoost tree, Linear SVC, and LGBM models were applied. Employed Logistic Regression, CatBoost tree, LGBM, and Linear SVC for attacks detection. CatBoost outperforms in terms of all performance metrics. As a result, new opportunities for applying CatBoost to different domains where gradient-boosted decision trees can help to solve cyber-security concerns.

A comparative study for classifying network traffic was introduced by Su et al. [18]. To detect intrusions, they implemented ML classifiers. KDD and CICIDS datasets were considered. Results explored that when compared to other methods, SVM produces the best outcomes.

Alissa et al. [38] authors applied ML techniques for botnet detection in this work. XGBoost model, LR, and DT models were evaluated as part of the recommended methodology. The training and testing phases used the UNSW-NB15 dataset. In this research, the decision tree strategy outperformed with 94% test accuracy. Table 1 presents accuracy results with various algorithms and datasets.

TABLE 1. Recent works on attacks using ML.

S.No	Method	Dataset	Attack	Accuracy
1.	LSTM [39]	Mod bus/ TCP network traffic data	DDoS	98.99%
2.	LSTM [40]	KDD99	DoS	93.82%
3.	RNN [41]	KDD99	DoS	94.20%
4.	GRU [41]	KDD99	DoS	99.70%
5.	Deep model [42]	KDD cup 99, ISCX, NSLKDD	DoS	98.27%,
6.	Shadow model [42]	KDD cup 99, ISCX, NSLKDD	DoS	96.75%
7.	Restricted boltzmann mechanism [43]	KDD99	DoS	85.00%
8.	RTS-DELM-CSIDS [44]	KDD99	DoS	96.20%
9.	DBN+LR [17]	KDD99	DoS	97.90%
10.	DNN [45]	NSL-KDD (1 hidden layer)	DoS	98.27%
11.	DNN [45]	NSL-KDD (5 hidden layer)	DoS	78.05%
12.	Extra boost + RF [46]	UNSW-NB15 and IoTID20	Malicious traffic	98.02%
13.	RF and Multilayer pre-ceptor [47]	Application layer DDoS dataset	DDoS	99.05%

IV. PROPOSED METHODOLOGY

This section outlines the approach used to forecast the intrusions using CICIDS. Dataset, which can be accessed through the Kaggle public repository. This data was brought into existence by the University of New Brunswick for analyzing various intrusions and DDoS attacks. The dataset is created based on logs of the servers, which resemble different types of attacks and attributes regarding logs. These are observed during a particular period of time. The label column specifies which type of attack took place.

The goal was to implement XGboost, LGBM, CatBoost, RF, and DT algorithms. F1 score, accuracy, recall, and precision were taken into account to evaluate their effectiveness. Machine learning algorithms learn from data that includes several types of attributes. The features in the dataset substantially influence the training time and performance of a machine learning system. Features in the dataset that assist the machine learning model in acquiring knowledge. Unnecessary and redundant features slow down an algorithm's training time and impact the algorithm's performance.

A. FEATURE SELECTION USING MIN-MAX SCALING, SMOTE, AND LASSO

Feature scaling is an ML model that normalizes information collected with varying mean and variance. Thus, min-max scaling was employed to normalize the obtained data, which lies in the features in the range of [0, 1] and [1, −1]. The min-max scaler is represented by Equation 1

$$Y' = \frac{Y - \min(y)}{\max(y) - \min(y)} \tag{1}$$

SMOTE(Synthetic Minority Over-sampling Technique) was applied for data balancing. It is an effective oversampling technique that can help improve ML models' performance on imbalanced datasets. It reduces the influence of noisy samples and obtains a more precise estimation of the true decision surface. Feature selection refers to identifying the finest features for training the ML model. LASSO methodology was applied for this process.

$$Lasso(L) = LSq(loss) + \lambda * \sum |\beta_j| \tag{2}$$

LSq(loss): This standard linear regression loss strives to reduce the variation between target values and predictions. The regularization parameter lambda specifies the amount of penalty imposed on the true values of the coefficient (β_j). A higher λ value results in an effective normalized value by shrinking their coefficient values towards zero. $\sum |\beta_j|$ resembles the total of the regression coefficients' absolute values. The penalty term induces the model to maintain small coefficient values and effectively causes some coefficients to become zero.

DDoS attack characteristics include a high volume of traffic. It can evolve from multiple and variety of sources with different forms. The proposed LASSO approach helps to handle all the above-mentioned issues. LASSO

(Least Absolute Shrinkage and Selection Operator) is a machine-learning algorithm for feature selection and regularization. A regression analysis technique that selects variables and applies regularisation to improve the statistical model's interpretation and prediction accuracy. It helps to identify the most important features in the dataset that contribute to the prediction of the target variable. In addition, LASSO has unique capabilities like reducing overfitting, multicollinearity, interpretability, and high-dimensional data handling.

The proposed model is implemented with a hierarchical ML-based methodology for classifying attacks. Classification is a procedure to predict the class of an individual data attribute, a predictive classification modeling task for estimating the mapping function from the discrete input with the output discrete variables. Estimating the classification of a given data attribute using either proper or improper data is feasible. The classifier uses training information to determine how the input characteristic relates to the available classes. Important features of hierarchical models are discussed in the case study section. ML employs a classifier to discover patterns in available data before categorizing the data into several groups concerning their patterns. Table 2 gives details of the unique characteristics of chosen ML models.

XGboost, LGBM, CatBoost, RF, and DT are effective ML models that are relevant for classification. It is more efficient than other algorithms and performs best for classification problems. With respect to execution time, it is more efficient than other algorithms. Suggested models are more efficient at classifying intrusions. The performance of suggested approaches is evaluated using the accuracy of the outcomes. Results demonstrate that the implemented strategy outperforms existing methods in terms of accuracy. The proposed methodology includes loading from the repository, preprocessing the collected dataset, and processing resultants, which can be split into training and test partitions. Models are trained with hyperparameters to improve the performance of each algorithm. Resultants are given to XGboost, LGBM, CatBoost, DT, and RF for evaluating results. Figure 1 describes the procedure of the current work. It includes CICIDS-dataset loading from the repository, preprocessing the collected dataset with min-max scaling, and SMOTE analysis. LASSO was applied for feature selection, and processed resultants can divided into training and test partitions. Pretrained models with hyperparameters train the models here with LGBM, CatBoost, XGboost, RF, and DT to evaluate the results.

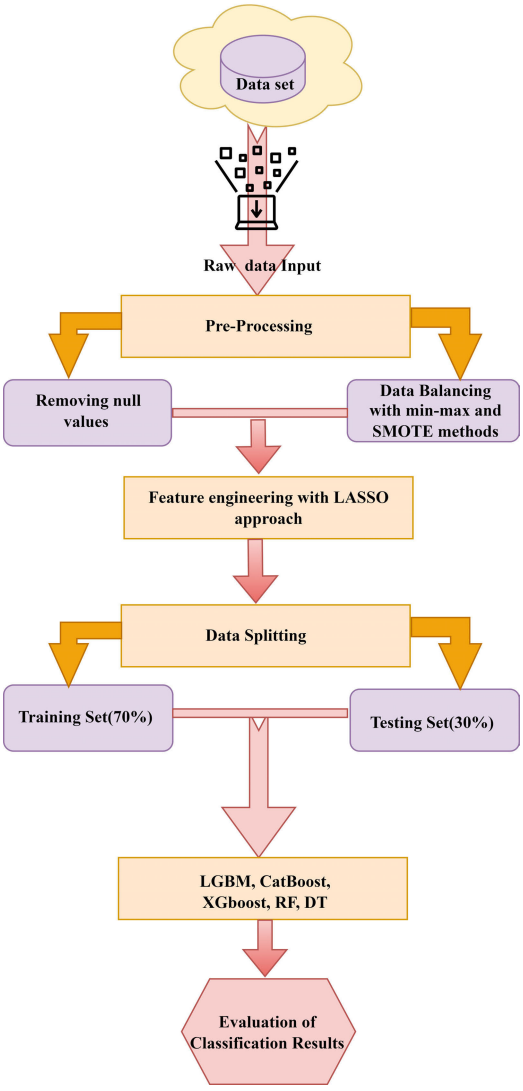

FIGURE 1. Proposed hierarchical IDS framework.

B. XGBOOST

XGboost: Gradient-boosted decision trees are implemented in XGboost. This technique successively generates decision trees, and weights are significant in this model. All independent variable values are given as weights and subsequently passed to the DT's for predicting results [48]. If the weight of variables predicted incorrectly by a tree increases, these variables are then supplied to the subsequent decision tree. At last various classifiers/predictors are combined to create a robust and precise model. Pseudocode is available in algorithm 1. It can address issues like classification, regression, rankings, and customized predictions. XGBoost is a supervised ML algorithm based on ensemble trees. It aims at optimizing a cost objective function containing a loss function (d) and a regularization term (β):

$$\Omega(\theta) = \underbrace{\sum_{m=1}^{n} b\left(x_i, \widehat{x_i}\right)}_{\text{Loss}} + \underbrace{\sum_{j=1}^{J} \beta\left(p_j\right)}_{\text{regularization}}, \tag{3}$$

where $\widehat{x_m}$ is the prediction values, n the instances values in the training set, J is the no of trees can be created and f_k is a tree from ensembled trees. The regularization expression can

be defined as:

$$\beta\left(f_u\right) = \gamma L + \frac{1}{2}\left[\alpha \sum_{i=1}^{L} |c_i| + \lambda \sum_{i=1}^{L} c_i^2\right], \tag{4}$$

where γ is the min loss split reduced value, λ is a regularization value given on weights and c is the weight included in each leaf. Let $f_u\left(x_i\right) = c_{q(x_i)}$, where q is in [1, L], where L is the no of leaf values. Greedy methodology is applied to choose the splitting value to improve the gain values.

Algorithm 1 Pseudocode for XGboost

1: Initialize the ensemble model E_i
2: **for** K = 1 to N trees **do**
3: Pseudo residuals should be calculated for each training sample: Set the pseudo-residuals for every sample used for training as the loss function's negative gradients concerning the estimates a (Negative_Gradients)
4: **end for**
5: Fit the pseudo-residuals to a regression tree: By using (X_train, Negative_Gradients), maximum_depth, maximum_features.
6: Update the E_i
7: Update the training predictions: For each training Sample: Calculate tree.predict() method, upgrade the training predictions by multiplying the learning rate by the current tree's prediction.
8: Generate predictions on the testing dataset using test_predictions, tree predict.

The following table summarizes the unique characteristics of DT, RF, LGBM, XGBoost, and CatBoost machine learning models for classification tasks:

TABLE 2. Unique Characteristics of DT, RF, LGBM, XGboost, and CatBoost.

S.No	ML model	characteristics
1.	Decision Tree	Simple, easy to interpret, and can handle both categorical and numerical data
2.	RF	Robust can handle missing and categorical and numerical data.
3.	LGBM	Efficient, fast, and accurate. Can handle large datasets, categorical features, and missing values.
4.	XGBoost	Fast, accurate, and can handle large datasets.
4.	CatBoost	Can handle categorical data and missing values and large datasets.

C. LIGHT GBM

Light GBM: It produces results very quickly, so it was preceded with the term 'Light'. It can handle massive amounts of data while using less memory. Gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) are two major classifications in LGBM. The best fit divides the tree leaf-wise, whereas other boosting methods split the tree level-wise. This approach minimizes loss compared to the level-wise algorithm when growing on the same leaf. Another reason for its popularity is its effectiveness on results. Pseudocode is available in algorithm 2. It also enables GPU learning. Therefore, data scientists frequently employ it to build data science applications.

$$\hat{S}_i(b) = \frac{1}{n}\left(\frac{\left(\sum_{y_j \in q_m} h_j + \frac{1-c}{d}\sum_{j_j \in p_m} h_j\right)^2}{n_m^i(b)} + \frac{\left(\sum_{y_j \in q_n} h_j + \frac{1-c}{d}\sum_{y_j \in p_n} h_j\right)^2}{n_n^i(b)}\right) \tag{5}$$

where $\hat{S}_i(b)$ is the estimated variance gain on the subset $q \cup p$, $q_l = \left\{y_i \in q : y_{ji} \leq b\right\}, q_n = \left\{y_j \in q : y_{ji} > b\right\}, p_m = \left\{y_j \in p : y_{ji} \leq b\right\}, p_n = \left\{y_j \in p : y_{ji} > b\right\}$, and the coefficients $\frac{1-c}{d}$ is applied to minimize the sum of the gradients over B back to the size of q^c. The estimate $\hat{S}_i(b)$ is applied on small instance subset values instead of the accurate $S_i(b)$ over all the occurrences that are used to identify the split point.

D. CATBOOST

CatBoost: This algorithm works with categorical features without applying feature encoders. It handles missing values and dynamically scales all columns to the same scaling, whereas other models require considerable column conversion. It also employs a cross-validation method to select the ideal hyperparameters for the given model. It supports both L1 and L2 regularization methods to reduce overfitting. CatBoost estimates the negative gradient of the loss function concerning the present estimations at every cycle of the algorithm and then applies a modified version of the gradient to the available predictions to update the predictions. Pseudocode is available in algorithm 3.

If $\sigma = (\sigma 1\ \sigma n)$ is the permutation, then $y\sigma q$, s is replaced with

$$\frac{\sum_{i=1}^{q-1}\left[y_{\sigma_i,s} = x_{\sigma_q,k}\right] M_{\sigma_j} + f \cdot q}{\sum_{i=1}^{q-1}\left[y_{\sigma_i,s} = x_{\sigma_q}, s\right] M_{\sigma_j} + f} \tag{6}$$

where q is a privious value, and f is the weight of the privious value. Meanwhile, the parametric value f > 0.

E. DECISION TREE

Decision Tree (DT): DT is one of the extensive forecast modeling methods used in various applications [49]. It is a common algorithmic strategy used to generate decision trees. It is one of the popularÂ methods of supervised learning [50]. The purpose is to create a prototype that employs the learning instructions from the easy selection of a tree to predict the value of an objective variable. It follows the concept of if-then

Algorithm 2 Pseudocode for Light GBM

1: 1: Initialize model parameters: learning_rate, num_leaves, max_depth, no of iterations
2: 2: Find the initial prediction value (at iteration 0) initial_predictionvalue = initialize_prediction(y)
3: 3: Iterate the specified number of iterations by using num_iterations
4: 4: Apply Hessian and gradient of the loss function. Hessians, gradients = computed value_gradients and hessians(y, initial_predictionvalue)
5: 5: Train a decision tree based on the gradients and Hessian values
6: 6: **if**objective == 'regression' **then**
7: 7: tree = train_regression_tree(lgbm_dataset, gradients, hessians, num_leaves, max_depth)
8: 8: objective == 'binary'
9: 9: tree = train_binary_tree(lgbm_dataset, gradients, hessians, num_leaves, max_depth)
10: 10: objective == 'multiclass'
11: 11: tree = train_multiclass_tree (lgbm_dataset, gradients, hessians, num_leaves, max_depth)
12: 12: raise ValueError("Invalid objective function.")
13: 13: **end if**
14: 14: Calculate the prediction value of decision tree tree_predictions = tree.predict(X)
15: 15: Update the ensemble and prediction value by appending to the decision tree
16: 16: ensemble.append(tree)
17: 17: initial_prediction=(update_ensemble_prediction (initial_prediction, tree_predictions, learning_rate))
18: 18: Return the trained ensemble of decision trees.

Algorithm 3 Pseudocode for CatBoost

1: Procedure CatBoost (X, y, params): Set F_0 as the starting estimate, which is commonly specified as the mean of y.
2: **for** m = 1 to pn_i **do**
3: Calculate the gradients of the loss function concerning
4: F_m-1 for each sample in 'X'. For every sample in X compute the loss functions to second order gradient concerning F_m-1.
5: **end for**
6: For each feature column j in X:
Compute the per-object gradients and hessians for feature f_j.
Based on the gradients and Hessians, customize the weights of each group in element e_j.
7: Using line search or alternative optimization approaches, determine the appropriate step size for F_m.
8: F_m should be updated for every instance in X using the best step size and per-object weight values.
9: Return the obtained final predictions
F_pn$_i$.

rule sentences. Classification of good behavior can be done without performing a lot of computation. Each split can be done based on a feature. If the feature is categorical, a divide is performed on the items that belong to a class. If the feature is continuous, the elements greater than the threshold are used to split the feature. The decision tree will select the most favorable variable at each split pseudocode available in algorithm 4.

Algorithm 4 Pseudocode for Decision Tree

1: To generate A classifiers:
2: Decision Tree (Node n, Data Partition D)
3: **if** K>0 **then**
4: Create ch children Ch_1, Ch_2 … Ch_k of n
5: splitting-criterion to partition D into D_1, D_2,… D_k
6: AOI-Method to D to find
7: use splitting-criterion of node n
8: Let K be the no of children of n
9: **for** i=1 to K **do**
10: Decision Tree (Ch_i, D_i)
11: **end for**
12: **end if**

The formula for Entropy is shown below:

$$D(T) = -pro_{(+)} \log pro_{(+)} - pro_{(-)} \log pro_{(-)}$$

Here $pro_+$is the probability of active classes, pro is the probability of nonactive classes, and T is the subset value of the train data.

F. RANDOM FOREST

Random Forest (RF): RF is one of the supervised ML classifiers. It can be employed in applications for both classification and regression. Data is trained using a range of techniques, like bagging. RF functions similarly to a decision tree and resolves a categorization problem. Here, a group of decision trees are used to perform classification, and the grouping of separate judgments results in the final classification. A tree in the ensemble makes a classification using a subset of the features in the total dataset. In many instances, the final classification that is created from the results of all such trees accurately depicts the training data patterns. Pseudocode for RF is available in algorithm 5.

Random forest: Then feature importance values from each tree are summed and normalized:

$$RandomForestX_x = \frac{\sum_y \text{ norm } X_{Xy}}{\sum_{y\in \text{ all features }, T\in \text{ all trees }} \text{ norm } X_{yT}} \tag{7}$$

RandomForest X_x is the priority of feature x calculated from all trees in the RF model, normX sub (xy) is the normalized feature importance for x in tree y. The RF was frequently employed for a wide range of operations since it effectively produces high performance. The development of numerous DT's during the training phase and the integration of their estimates via voting by the majority are two fundamental

Algorithm 5 Pseudocode for Random Forest

To create A classifiers:
for i=1 to A **do**
    To generate D, choose samples at random fromÂ the training data D using replacement.
    Generate a parent node N_i that contains D_i.
    Call BuildTree(T_i) BuildTree (T):
end for
if N comprises only instances of one class. **then**
    return
else
    Select x% of the possible dividing features in T at random.
    To split on, choose the feature F with the most information gained.
    Create st child nodes of $T, T_1, \ldots .T_st$, where F has st possible values($F_1, \ldots F_st$)
end if
for i=1 to F **do**
    place the values of T_i to D_i, where D_i represents all T instances that match F_i.
    Call BulidTree(T_i)
end for

components. The voting method reduces the likelihood of models overfitting.

V. RESULTS AND ANALYSIS

In this section, evaluated results concerning machine learning algorithms are discussed. Python programming was used to implement the system. Sklearn is the library used to retrieve data. It offers efficient ML and statistical modeling tools, such as categorization, regression, cluster analysis, and dimensionality reduction, to extract the algorithms and display the results.

Metrics considered for evaluating the ML model's performance are precision, recall, F1-score, accuracy values. Higher accuracy values result in greater ML model performance. The following formulas are used to calculate the accuracy value. The chosen metrics can be calculated using the equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 = \frac{2 * TP}{2 * TP + FP + FN} \tag{11}$$

All the above metrics can be determined by considering True Positive (TP), which indicates how many predictions made by the ML model are accurate. It is considered a true positive if the estimated value matches the actual value

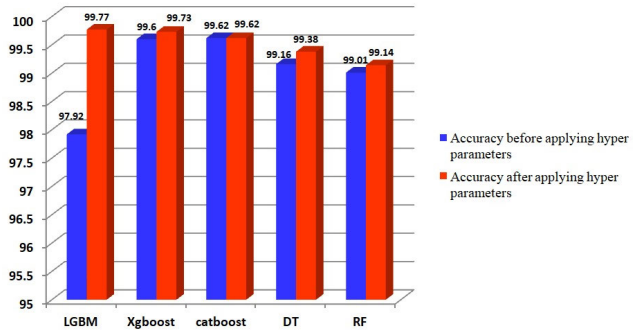

FIGURE 2. Classification accuracy evaluation using proposed ML models.

in the data set. If the actual value of the ML model's result variable is not equal to the predicted value, it is appended to the true negative (TN) values. If the attack is considered positive in the label but is absent from the true data set, this instance is taken into account as a false positive (FP), and vice versa for a false negative (FN). The ML models used for intrusion detection and classification are discussed here. The analysis of experiments was performed on hierarchical algorithms and optimization methods. Table 3

TABLE 3. Performance metrics evaluation on ML models using HPO.

Algorithm	Precision	Recall	F-1 Score	Accuracy
LGBM	100	100	100	99.77%
XGboost	100	100	100	99.73%
CatBoost	99	100	99	99.62%
DT	99	100	99	99.38%
RF	98	99	99	99.14%

gives precision, recall, F-1 score, and accuracy outcomes by applying HPO values. Table 4 describes the hyperparameters and their type, considered search space, and obtained the best search space values. LGBM has effectively performed with max_depth as −1, learning_rate as 0.05, num_leaves as 31, and n_estiamtors as 200. XGboost has shown its best outcome values with max_depth as 6, learning_rate as 0.1, and n_estiamtors as 300. CatBoost produced its results with depth as 8, learning_rate as 0.1, and iterations as 300. DT produced max_depth as 5 and min_samples_split as 2. RF given n_estimators as 200, max_depth as 4, and min_samples_split as 2. Table 5 presents the accuracy results before and after applying hyperparameters. It spots a light to show the performance enhancement, especially the LGBM algorithm, which has shown better accuracy than other models.

Figure 2 gives the classification accuracy of each ML model. It represents the accuracy results of the ML models on the dataset and illustrates that the LGBM model performs effectively. In the research observations, LGBM handled the large-scale data effectively, had a faster training speed, and had lower memory usage than other algorithms. Parallel and GPU learning aided in improving performance. Due to the presence of the above supporting functionalities in LGBM,

TABLE 4. Hyperparameter configuration space considered for ML models.

ML Model	Hyperparameter	Type	Search Space	Obtained Best Search Space Values
LGBM	max_depth	Discrete	(5, 10 -1)	-1
	learning_rate	Continuous	(0.01, 0.05, 0.1)	0.05
	num_leaves	Discrete	(31, 63, 127)	31
	n_estimators	Discrete	(100, 200, 300)	200
XGboost	max_depth	Discrete	(3, 6, 9)	6
	learning_rate	Continuous	(0.01,0.05,0.1)	0.1
	n_estimators	Discrete	(100, 200, 300)	300
CatBoost	depth	Discrete	(4, 6, 8)	8
	learning_rate	Continuous	(0.01, 0.05, 0.1)	0.1
	iterations	Discrete	(100, 200, 300)	300
DT	max_depth	Discrete	(4, 5, 10)	5
	min_samples_split	Discrete	(2, 5, 10)	2
RF	n_estimators	Discrete	(100, 200, 300)	200
	max_depth	Discrete	(4, 5, 10)	4
	min_samples_split	Discrete	(2, 5, 10)	2

TABLE 5. Accuracy of ML models before and after applying HPO.

Algorithm	Before HPO	After HPO
LGBM	97.77%	99.92%
CatBoost	99.60%	99.62%
XGboost	99.72%	99.73%
DT	99.16%	99.38%
RF	99.01%	99.14%

it has shown its proven performance compared with other algorithms. The accuracy of LGBM, XGboost, CatBoost, DT, and RF is 99.77%, 99.73%, 99.62%, and 99.14%, respectively. Outcomes show better performance results than the works done in [17], [44], and [45]. It also provides comparison information about before and after applying hyperparameters.

VI. CASE STUDY ON DDOS ATTACKS

In addition to the real-time scenarios discussed in the motivation section, the following are some more supporting incidents motivated for this work. In February 2020, Amazon Web Services reported that it had mitigated a massive DDoS attack. The attack was so intense that it saw incoming traffic at a rate of 2.3 terabits per second (Tbps), which is the largest ever recorded. AWS did not reveal the customers that were the focus of the attack [51]. GitHub is a widely used online code management platform on which millions of developers rely. Additionally, it became the target of a DDoS assault that sent packets at a pace of 126.9 million per second, reaching 1.3 Tbps [52]. In the complex web of international finance, the secure and efficient exchange of information is paramount. The current case study delves into a series of Distributed Denial of Service (DDoS) attacks that targeted both the SWIFT network and Banco de Chile, unraveling a disruptive narrative in the heart of the financial sector. SWIFT (Society for Worldwide Interbank Financial Telecommunication) is the backbone for secure and standardized messaging within the global financial community. Banco de Chile, a major financial institution, relies heavily on SWIFT for international transactions and communication with correspondent banks.

Implications Of DDoS attack on SWIFT and Banco de Chile: The SWIFT [53] and Banco de Chile [54] DDoS incidents highlight the vulnerability of critical financial infrastructure to targeted cyber attacks. The aftermath prompts a paradigm shift in cybersecurity strategies, emphasizing collaboration, information sharing, and continuous improvement to safeguard the integrity of international financial systems. The resilience demonstrated in the face of these challenges becomes a pivotal chapter in the ongoing evolution of cybersecurity within the global financial landscape. SWIFT and Banco de Chile, both financial firms services, experience intermittent disruptions and face a surge in malicious traffic. DDoS attack impacts the bank's online services, affecting customer transactions. News of the incidents spreads, impacting SWIFT's reputation for secure financial messaging–declining customer trust and confidence.

A. SOLUTION

In this scenario, the proposed hierarchical Machine Learning (ML) technique involves organizing data into a tree-like structure, where each node represents a cluster of data points. The clusters are then further divided into sub-clusters, and so on, until the individual data points are reached. It is a powerful tool for analyzing complex datasets and can be used in various applications. This approach is useful for a variety of reasons:

1) Interpretability: These models are often more interpretable than other types of models, as they provide a clear visual representation of the data structure.

2) Flexibility: Models can be used with many data types, including continuous, categorical, and binary data.

3) Scalability: Large datasets can be easily parallelized and distributed across multiple machines.

4) Accuracy: these models can capture more complex relationships between data points than other models.

5) Clustering: which is grouping similar data points together. This can be useful for various applications, such as anomaly detection, image and customer segmentation.

B. PRACTICAL IMPLICATIONS

1) Financial sector administrators and security teams can deploy this approach to enhance their DDoS mitigation strategies.

2) Continuous model retraining and feature refinement are essential to adapt to evolving attack methods.

VII. CONCLUSION

Network security is becoming more crucial in today's information-based environment, especially in financial sectors. With the advent of digital transformation in public life, the range of threats and attack patterns was growing rapidly. Cyberspace provides enormous benefits, potential, and significant challenges, too. It is crucial to develop advanced security systems that are capable of accurately detecting various sorts of intrusions. The current study provides effective hierarchical ML techniques for intrusion detection and classification to support this issue. Proposed research suggests that machine learning algorithms are capable of accurately predicting attacks. The LGBM model prediction accuracy performs more effectively than other ML models. Hyperparameters also played an influential role in improving classification accuracy. In future work, planning to add more ensembled ML and deep learning models along with various optimization methods to improve forecast accuracy.

TABLE 6. Abbreviation.

Abbreviation	Meaning
ML	Machine Learning
RF	Random Forest
DT	Decision Tree
IDS	Intrusion Detection System
DDoS	Distributed Denial of Service
LGBM	Light Gradient Boosting Machine
CatBoost	Categorical Boosting
XGboost	Extreme Gradient Boosting
HPO	Hyperparameter Optimization

ABBREVIATIONS

The following abbreviations shown in Table. 6 are used in this manuscript

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] L. Lv, W. Wang, Z. Zhang, and X. Liu, "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105648.

[2] K. Lakshmanna, R. Kaluri, N. Gundluru, Z. S. Alzamil, D. S. Rajput, A. A. Khan, M. A. Haq, and A. Alhussen, "A review on deep learning techniques for IoT data," *Electronics*, vol. 11, no. 10, p. 1604, May 2022.

[3] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, Jan. 2021.

[4] M. Khalid, S. Hameed, A. Qadir, S. A. Shah, and D. Draheim, "Towards SDN-based smart contract solution for IoT access control," *Comput. Commun.*, vol. 198, pp. 1–31, Jan. 2023.

[5] C. Iwendi, Z. Jalil, A. R. Javed, T. G. Reddy, R. Kaluri, G. Srivastava, and O. Jo, "KeySplitWatermark: Zero watermarking algorithm for software protection against cyber-attacks," *IEEE Access*, vol. 8, pp. 72650–72660, 2020.

[6] R. M. S. Priya, S. Bhattacharya, P. K. R. Maddikunta, S. R. K. Somayaji, K. Lakshmanna, R. Kaluri, A. Hussien, and T. R. Gadekallu, "Load balancing of energy cloud using wind driven and firefly algorithms in Internet of Everything," *J. Parallel Distrib. Comput.*, vol. 142, pp. 16–26, Aug. 2020.

[7] S. Siddiqui, S. Hameed, S. A. Shah, I. Ahmad, A. Aneiba, D. Draheim, and S. Dustdar, "Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects," *IEEE Access*, vol. 10, pp. 70850–70901, 2022.

[8] S. Agrawal, S. Sarkar, M. Alazab, P. K. R. Maddikunta, T. R. Gadekallu, and Q.-V. Pham, "Genetic CFL: Hyperparameter optimization in clustered federated learning," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–10, Nov. 2021.

[9] M. S. Khan, N. M. Khan, A. Khan, F. Aadil, M. Tahir, and M. Sardaraz, "A low-complexity, energy-efficient data securing model for wireless sensor network based on linearly complex voice encryption mechanism of GSM technology," *Int. J. Distrib. Sensor Netw.*, vol. 17, no. 5, May 2021, Art. no. 155014772110186.

[10] M. Sardaraz and M. Tahir, "SCA-NGS: Secure compression algorithm for next generation sequencing data using genetic operators and block sorting," *Sci. Prog.*, vol. 104, no. 2, pp. 1–18, Apr. 2021.

[11] S. Sambangi and L. Gondi, "A machine learning approach for DDoS (distributed denial of service) attack detection using multiple linear regression," *Proceedings*, vol. 63, no. 1, p. 51, 2020.

[12] U. Islam, A. Muhammad, R. Mansoor, M. S. Hossain, I. Ahmad, E. T. Eldin, J. A. Khan, A. U. Rehman, and M. Shafiq, "Detection of distributed denial of service (DDoS) attacks in IoT based monitoring system of banking sector using machine learning models," *Sustainability*, vol. 14, no. 14, p. 8374, Jul. 2022.

[13] R. Yahalom, A. Steren, Y. Nameri, M. Roytman, A. Porgador, and Y. Elovici, "Improving the effectiveness of intrusion detection systems for hierarchical data," *Knowl.-Based Syst.*, vol. 168, pp. 59–69, Mar. 2019.

[14] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "A novel two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[15] M. Hameed, F. Yang, M. I. Ghafoor, F. H. Jaskani, U. Islam, M. Fayaz, and G. Mehmood, "IOTA-based mobile crowd sensing: Detection of fake sensing using logit-boosted machine learning algorithms," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–15, Apr. 2022.

[16] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[17] A. Diro and N. Chilamkurti, "Leveraging LSTM networks for attack detection in fog-to-things communications," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 124–130, Sep. 2018.

[18] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020.

[19] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *J. Big Data*, vol. 8, no. 1, p. 65, Dec. 2021.

[20] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet Things*, vol. 7, Sep. 2019, Art. no. 100059.

[21] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty, and V. Sravan Kiran, "Similarity based feature transformation for network anomaly detection," *IEEE Access*, vol. 8, pp. 39184–39196, 2020.

[22] S. Behal and K. Kumar, "Trends in validation of DDoS research," *Proc. Comput. Sci.*, vol. 85, pp. 7–15, Jan. 2016.

[23] A. E. Cil, K. Yildiz, and A. Buldu, "Detection of DDoS attacks with feed forward based deep neural network model," *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114520.

[24] M. Motylinski, Á. MacDermott, F. Iqbal, and B. Shah, "A GPU-based machine learning approach for detection of botnet attacks," *Comput. Secur.*, vol. 123, Dec. 2022, Art. no. 102918.

[25] N. S. Akash, S. Rouf, S. Jahan, A. Chowdhury, and J. Uddin, "Botnet detection in IoT devices using random forest classifier with independent component analysis," *J. Inf. Commun. Technol.*, vol. 21, no. 2, pp. 201–232, 2022.

[26] M. Asadi, "Detecting IoT botnets based on the combination of cooperative game theory with deep and machine learning approaches," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 12, pp. 5547–5561, Dec. 2022.

[27] S. Gera and A. Sinha, "T-Bot: AI-based social media bot detection model for trend-centric Twitter network," *Social Netw. Anal. Mining*, vol. 12, no. 1, p. 76, Dec. 2022.

[28] I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, Oct. 2017.

[29] S. T. Ikram, A. K. Cherukuri, B. Poorva, P. S. Ushasree, Y. Zhang, X. Liu, and G. Li, "Anomaly detection using XGBoost ensemble of deep neural network models," *Cybern. Inf. Technol.*, vol. 21, no. 3, pp. 175–188, Sep. 2021.

[30] M. Cheon, H. Ha, O. Lee, and C. Mun, "A novel hybrid deep learning approach to code generation aimed at mitigating the real-time network attack in the mobile experiment via GRU-LM and Word2vec," *Mobile Inf. Syst.*, vol. 2022, pp. 1–11, Sep. 2022.

[31] Ismail, M. I. Mohmand, H. Hussain, A. A. Khan, U. Ullah, M. Zakarya, A. Ahmed, M. Raza, I. U. Rahman, and M. Haleem, "A machine learning-based classification and prediction technique for DDoS attacks," *IEEE Access*, vol. 10, pp. 21443–21454, 2022.

[32] L. D'Hooge, T. Wauters, B. Volckaert, and F. De Turck, "Classification hardness for supervised learners on 20 years of intrusion detection data," *IEEE Access*, vol. 7, pp. 167455–167469, 2019.

[33] O. Salem, K. Alsubhi, A. Shaafi, M. Gheryani, A. Mehaoua, and R. Boutaba, "Man-in-the-mddle attack mitigation in Internet of Medical Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 2053–2062, Mar. 2022.

[34] D. Gupta, M. Gupta, S. Bhatt, and A. S. Tosun, "Detecting anomalous user behavior in remote patient monitoring," in *Proc. IEEE 22nd Int. Conf. Inf. Reuse Integr. for Data Sci. (IRI)*, Aug. 2021, pp. 33–40.

[35] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "DDoSNet: A deep-learning model for detecting network attacks," in *Proc. IEEE 21st Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Aug. 2020, pp. 391–396.

[36] M. Rai and H. L. Mandoria, "Network intrusion detection: A comparative study using state-of-the-art machine learning methods," in *Proc. Int. Conf. Issues Challenges Intell. Comput. Techn. (ICICT)*, vol. 1, Sep. 2019, pp. 1–5.

[37] N. Bakhareva, A. Shukhman, A. Matveev, P. Polezhaev, Y. Ushakov, and L. Legashev, "Attack detection in enterprise networks by machine learning methods," in *Proc. Int. Russian Autom. Conf. (RusAutoCon)*, Sep. 2019, pp. 1–6.

[38] K. Alissa, T. Alyas, K. Zafar, Q. Abbas, N. Tabassum, and S. Sakib, "Botnet attack detection in IoT using machine learning," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–14, Oct. 2022.

[39] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8852–8859, Sep. 2020.

[40] R. C. Staudemeyer, "Applying long short-term memory recurrent neural networks to intrusion detection," *South Afr. Comput. J.*, vol. 56, no. 1, pp. 136–154, 2015.

[41] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluation of recurrent neural network and its variants for intrusion detection system (IDS)," *Int. J. Inf. Syst. Model. Des.*, vol. 8, no. 3, pp. 43–63, Jul. 2017.

[42] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Gener. Comput. Syst.*, vol. 82, pp. 761–768, May 2018.

[43] A. Dawoud, S. Shahristani, and C. Raun, "Deep learning and software-defined networks: Towards secure IoT architecture," *Internet Things*, vols. 3–4, pp. 82–89, Oct. 2018.

[44] A. Haider, M. A. Khan, A. Rehman, R. M. Ur, and H. S. Kim, "A real-time sequential deep extreme learning machine cybersecurity intrusion detection system," 2021.

[45] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[46] O. Olayemi Petinrin, F. Saeed, X. Li, F. Ghabban, and K.-C. Wong, "Malicious traffic detection in IoT and local networks using stacked ensemble classifier," *Comput., Mater. Continua*, vol. 71, no. 1, pp. 489–515, 2022.

[47] M. J. Awan, U. Farooq, H. M. A. Babar, A. Yasin, H. Nobanee, M. Hussain, O. Hakeem, and A. M. Zain, "Real-time DDoS attack detection system using big data approach," *Sustainability*, vol. 13, no. 19, p. 10743, Sep. 2021.

[48] S. Bhattacharya, S. S. R. Krishnan, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu, M. Alazab, and U. Tariq, "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," *Electronics*, vol. 9, no. 2, p. 219, Jan. 2020.

[49] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106432.

[50] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.

[51] A. Singh and B. B. Gupta, "Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms: Issues, challenges, and future research directions," *Int. J. Semantic Web Inf. Syst.*, vol. 18, no. 1, pp. 1–43, Apr. 2022.

[52] I. Balaban, "Denial-of-service attack," *Int. J. Inf. Secur. Cybercrime*, vol. 10, no. 1, pp. 59–64, 2021.

[53] *Bangladesh Bank Heist: 30m Stolen in Attacks on Swift System*. Accessed: 2016. [Online]. Available: www.reuters.com/article/us-cyber-heist-swift/bangladesh-bankheist- 30m-stolen-in-attacks-on-swift-system-idUSKCN0W10JG

[54] *Banco de Chile Says Hit by Cyber Attack, Internal Systems Down*. Accessed: 2018. [Online]. Available: www.reuters.com/article/us-chile-cyber-bancodechile/banco-de-chilesays-hit-by-cyber-attack-internal-systems-down-idUSKCN1J01QD

SANDEEP DASARI received the M.Tech. degree in computer science from the Rajeev Gandhi Memorial College of Engineering and Technology, India, in 2017. He is currently pursuing the Ph.D. degree with the School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, India. His research interests include machine learning and cyber security.

RAJESH KALURI received the B.Tech. degree in computer science and engineering (CSE) from JNTU, Hyderabad, the M.Tech. degree in CSE from ANU, Guntur, India, and the Ph.D. degree in computer vision from VIT, India. He was a Visiting Professor with the Guangdong University of Technology, China, in 2015 and 2016. He is currently an Associate Professor with the School of Computer Science Engineering and Information Systems, Vellore Institute of Technology. He has published research articles in various reputed international journals. His current research interests include the areas of computer vision and human–computer interaction.