Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

# Cloud-based multiclass anomaly detection and categorization using ensemble learning

Faisal Shahzad[1], Abdul Mannan[2], Abdul Rehman Javed[3,4], Ahmad S. Almadhor[5], Thar Baker[6] and Dhiya Al-Jumeily OBE[7*]

**Abstract**

The world of the Internet and networking is exposed to many cyber-attacks and threats. Over the years, machine learning models have progressed to be integrated into many scenarios to detect anomalies accurately. This paper proposes a novel approach named cloud-based anomaly detection (*CAD*) to detect cloud-based anomalies. *CAD* consist of two key blocks: ensemble machine learning (EML) model for binary anomaly classification and convolutional neural network long short-term memory (*CNN-LSTM*) for multiclass anomaly categorization. *CAD* is evaluated on a complex UNSW dataset to analyze the performance of binary anomaly detection and categorization of multiclass anomalies. Furthermore, the comparison of *CAD* with other machine learning conventional models and state-of-the-art studies have been presented. Experimental analysis shows that *CAD* outperforms other studies by achieving the highest accuracy of 97.06% for binary anomaly detection and 99.91% for multiclass anomaly detection.

**Keywords:** Cloud computing, Anomaly detection, Cyberattacks, Deep learning, Ensemble learning, Multiclass attack

## Introduction

Anomalies are the unusual patterns that do not conform to the usual patterns of data [1–4]. Anomaly detection is the detection of deviation or uncertainty in data. Doing so in the early stages can save the time and resources spent during the processing and decision taken after processing the data having anomalies.

Cyber security can save the data and digital systems from widespread critical security threats emerging from the Internet [5]. Cyber security can secure networks, and can protect data, applications and digital infrastructure from unauthorized access, attack, unauthorized modification and availability issues to [6–12] keeps Confidentiality, Integrity, and Availability (CIA) triad intact [13–15].

Network Intrusion Detection Systems (NIDS) detect signature and anomaly-based attacks. The primary target of NIDS is to provide robust automated detection capability to networked devices for efficient and effective protection. Network activities are compared in Signature-based attacks with the database of attack patterns to identify if an attempt is being made to compromise the network. Alerts are generated in case of detection of an attack [16]. Anomaly-based attacks detect the unknown attacks in network traffic by checking the variance in behavior from the baseline already identified [17]. There are many ways to detect anomaly intrusions, most of which comprise statistical methods and machine-learning techniques. A detailed review of machine learning methodologies for anomaly detection is presented in [18].

Statistical analysis methodologies can create a generic or benign profile of a particular activity. This analysis can detect and identify the deviating nature of a particular activity from a typical profile which can also be considered a cyber-attack or suspicious activity. Machine learning opens a new doorway to the detection technologies [19]. The use of federated learning is also

*Correspondence: d.aljumeily@ljmu.ac.uk

[7] School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, UK
Full list of author information is available at the end of the article

increasing in large-scale networks like smart transport infrastructure [20].

In NIDS, anomaly detection can be automated by utilizing machine learning classifiers. Numerous researchers utilized machine learning classification techniques to detect or identify different kinds of attacks [21–23]. However, these techniques produced low accuracy in anomaly detection. The novelty of this paper is proposing an ensemble learning-based approach to detect cloud-based anomalies accurately.

### Paper Contributions

The following are the research contributions to this article:

- We design an approach named *CAD* that comprises Convolutional Neural Network Long Short-Term Memory *CNN-LSTM* based customized deep learning model for network graph data based binary anomaly detection and multiclass anomaly categorization. *CNN-LSTM* applies CNN at the first layer, LSTM layer at the second layer till the second last layer, and the final dense layer at the output to detect anomalies.
- We also design an approach named *Ensemble Machine Learning (EML)* that combines conventional machine learning algorithms and results in detecting binary anomalies in networks.
- Analyzed the complex state-of-the-art dataset *UNSW-NB-15* [23] and highlighted the significant parameters to adjust for the performance enhancement of AI models.
- The results of the experiments demonstrate that *CAD* approach improves the anomaly classification rate and attains the maximum accuracy of 97.06% in case of binary anomaly detection with *EML* and 99.91% for multiclass anomaly detection with *CNN-LSTM* which outperforms other state-of-art studies.

### Paper Structure

The remainder of the manuscript is structured as follows. Related Work section sheds light on the state-of-the-art previous relevant research work. Network Preliminaries and Dataset section discusses the dataset used for this research. CAD for Anomaly Detection section states the proposed approach *CAD* and sub-methods for different anomaly detection. The experimental analysis in the form of proposed approaches is also part of this Results and Evaluations section. Finally, Discussion section summarizes this article's methodologies and signifies the intended future approach.

## Related Work

Numerous researchers have investigated and tested the network's security with machine learning techniques. The authors in [24] proposed a framework that uses the past behavior of nodes and machine learning techniques to improve the network's security. The information on the past behavior of the node's participant benefits its trustworthiness in the network. For this accomplishment, the datasets should be preprocessed properly to remove numerous irrelevant features and noisy data.

Faker *et al.* proposed two approaches that are composed of a Deep Feed-Forward Neural Network (DNN) and two ensemble techniques, Random Forest and Gradient Boosting Tree (GBT), to detect the network intrusions by training the model over UNSW NB15 and CICIDS2017 dataset [25]. Khan *et al.* proposed a novel approach based on a two-stage deep learning model and tested the model KDD99 and UNSW-NB15 datasets to prove the proficiency of the model [26]. Furthermore, trustworthiness is required to check for malicious participating nodes in the network, and past information can be used to identify if the user has been reliable with the network [27]. This proved to be an excellent approach to guarantee the trustworthiness of the environment.

Machine learning techniques can be utilized to automate NIDS in anomaly detection. Djibouti *et al.* proposed a k-nearest neighbour methodology for distance-based outlier detection to perform flow distribution probability (FDP) outlier detection [28]. Chapaneri *et al.* presented a comprehensive survey of machine learning approaches to prevent network intrusion attacks using UNSW-NB15, TUIDS, and NSLKDD datasets [29]. Bagui *et al.* examined machine learning techniques over the UNSW-NB15 dataset to test the capabilities of algorithms [30]. In 2015, Authors in [22, 23] introduced the UNSW-NB15 dataset, a hybrid of the normal modern attacks and the new synthesized attack activities of the network traffic. The authors Moustafa and Slay [23, 31, 32], also criticized that other datasets such as KDD'99 or NSL-KDD are limited, and these datasets did not cover the modern attacks in NIDS and proposed a new dataset UNSW-NB-15 that included different features from the KDD'99 dataset and only shared few standard features.

The authors in [21] also utilized the UNSW-NB-15 dataset and improved the results by using central points of attribute values in the preprocessing stage. The authors used the Apriori algorithm with Naive Bayes (NB) and Logistic Regression machine learning

classifiers. On the UNSW-NB15 dataset, numerous researchers have used machine learning techniques to evaluate the dataset's efficiency. In 2020, Mohanad Sarhan [33] experimented on the UNSW-NB15 dataset and achieved the highest accuracy of 99.25% with binary classification without reducing all unnecessary features. The authors also used multi-label classification on this dataset and achieved the weighted accuracy of 98.19% with an f-score of 98%. However, the whole dataset is vast, and all the previous research has used random sampling to train their respective models instead of using the original training and testing files given with the dataset. Similarly, another research [34] published recently in 2020 by J. Olamantanmi Mebawondu explains that if the features of datasets are reduced through an algorithmic procedure, then the dataset can be used in a real-time intrusion detection system. The author evaluated the dataset's efficiency by applying the Artificial Neural Network (Multi-layer perceptron) algorithm for anomaly classification and achieved an accuracy of 76.96%.

## Network Preliminaries and Dataset

The proposed models discussed in CAD for Anomaly Detection section use unprocessed network packets of the UNSW-NB 15 dataset generated by the IXIA PerfectStrom tool. The purpose of creating the UNSW-NB15 dataset is to build Artificial Intelligent models that observe the system's sophisticated real-time activities and real-time exploitation feedback. 100 GB of the raw traffic data was generated using the `tcpdump`, which includes Pcap files. This dataset comprises nine attacks, including Shellcode analysis, DoS, Exploits, Generic, Fuzzers, Reconnaissance, Backdoors, and Worms. UNSW-NB-15 dataset comprises two files: the training and testing files containing records of all types of attacks and regular traffic features. The training data file contains 82 and 332 records; in the testing file, there are 175 and 341 records. The dataset contains 45 features in training and testing files [35]. In the UNSW-NB15 dataset, features such as scrip, sport, strip, time, and time are missing in the training and testing dataset.

The optimum and maximum performance of an ML model can be achieved by performing preprocessing on the dataset. In the preprocessing stage of this research, not a number (NAN) values, identical instances were removed, and scaling was performed, which points to the re-scaling of arithmetic real-values to a fixed scope. Moreover, the first four columns of the dataset are also removed due to non-usability in identifying network intrusion detection. Those columns include source IP address, source port number, destination IP address, and destination port number. Due to the low variance of the dataset, MinMax scaling is applied for feature normalization, as mentioned in Eq. 1.

$$X_{norm} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{1}$$

The original value of the feature is denoted by $X_i$ that is subtracted from the minimum magnitude of that particular feature and then divided by the subtracted value from the maximum and minimum of the feature.

## CAD for Anomaly Detection

CAD comprises data analysis, preprocessing, feature reduction, classification of anomalies from regular files through machine learning and deep learning techniques, and then categorizing them. Figure 1 presents the proposed approach for classifying and categorizing network anomalies.

### Binary Anomaly Detection

We review and test different conventional machine learning classifiers. The most proficient combination has been chosen to be incorporated within an ensemble model upon the evaluation results. The evaluated models are as follows:

**Decision Tree (DT) classifier** works on the principle of supervised learning. DT's ability allows it to take continuous or series values and thus give a series of predicted results in a continuous manner [36]. DT performance is based on entropy as shown in Eq. 2 in which $p$ represents the probability and $E(S)$ represents the entropy of the specific entity. The less the entropy, the better the performance.

$$E(S) = \sum_{i=1}^{c} -p_1 log_2 p_i \tag{2}$$

We tune the attributes to optimize the model's performance for anomaly detection. We set various parameters to improve the performance of DT for anomaly detection are the following: criterion='gini', max_depth=10, random_state=0, splitter=best, min_samples_split=2, min_samples_leaf=1.

**Random Forest (RF) classifier** works on the theory of ensemble learning methodology for classification,

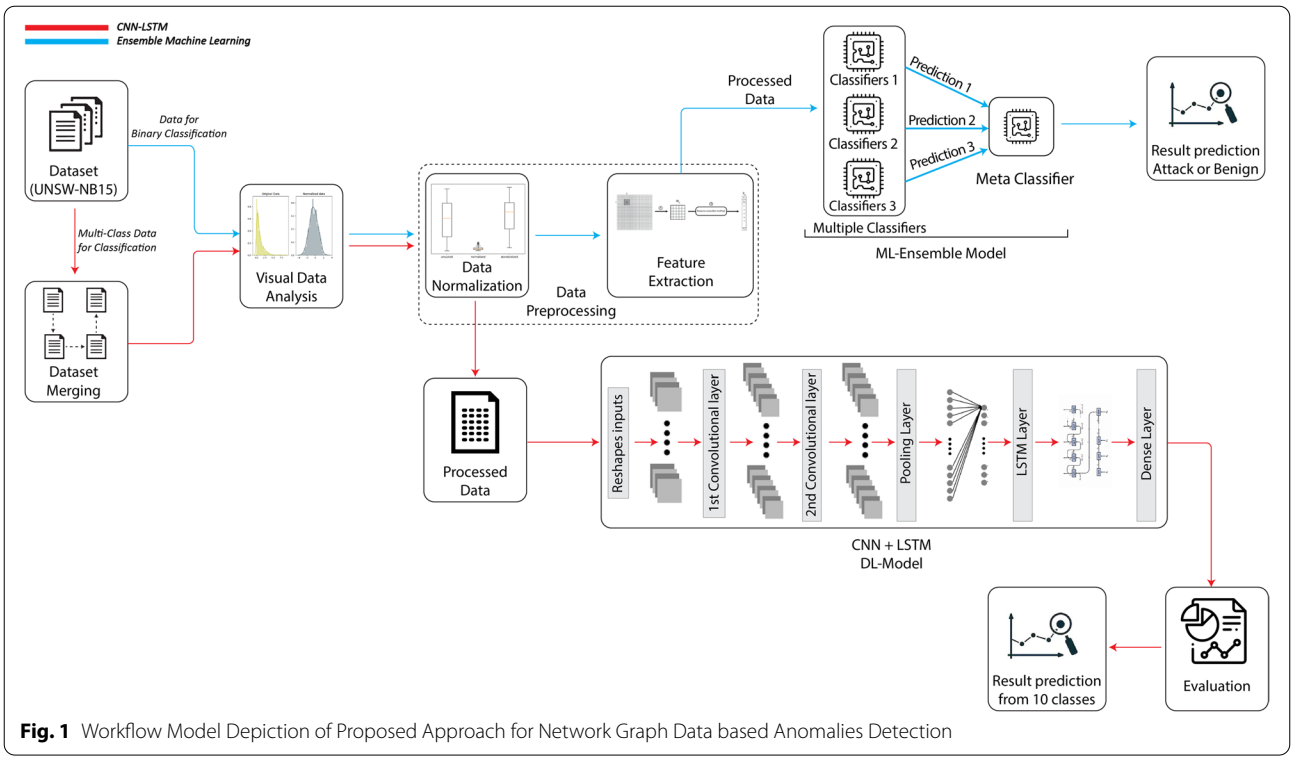


**Fig. 1** Workflow Model Depiction of Proposed Approach for Network Graph Data based Anomalies Detection

regression, and other similar tasks by developing multiple trees when training the model and resulting in the predicted class. The prediction calculation is done by taking the mode of the targeted classes from the independent trees [37].

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2 \tag{3}$$

In Eq. 3, the number of data points $i$ are denoted by $N$, where $y_i$ is the actual value of data point and $f_i$ denotes the value returned by the classifier. Following parameters are configured to tune the RF model: bootstrap = true, criterion ='gini', min-samples-leaf = 1, min-samples-split = 2, n-estimators = 100, and random-state = 0.

**Gradient Boosting (GB) classifier** is a combination of machine learning classifiers that integrate weaker models to create a more robust predictive machine learning model [38][1]. Gradient boosting is a technique that uses weak predictions and a decision tree format to build ensemble structure for better accuracy in regression and classification problems.

$$F_0(x) = arg_\gamma min \sum_{i=1}^{n} L(y_i, \gamma) \tag{4}$$

In Eq. 4, $F_0(x)$ is the constant function, $y$ is the observed value where the $\gamma$ is the real value in the loss function $L$. Following parameters are configured to tune the GB model: learning_rate=0.01, n_estimators=100, random_state=1, subsample=1, criterion=friedman_mse, max_depth=3, validation_fraction=0.1.

**Extreme Gradient Boosting (XGB) classifier** inherits most of the features from GB but for approximation, it uses the $2^{nd}$ order derivative [39][2].

$$F_m(x) \leftarrow F_{(m-1)}(x) + \gamma_x h_m(x) \tag{5}$$

Equation 4 is extended to Eq. 5 in which $m$ represents number of iterations, $h_m(x)$ is the match on the gradient that is result of each iteration, and $\gamma_m$ is the multiplicative factor. Following parameters are configured to tune the XGB model: max_depth = 10, objective = multi: softmax, num_class=2, n_gpus=1, sampling_method=uniform, tree_method = auto, max_bin =256.

[1] https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

[2] https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

**Logistic Regression classifier** is a statistical Learning technique categorized in supervised ML methods dedicated to classification tasks.

$$g(E(y)) = \alpha + \beta x1 + \gamma x2 \tag{6}$$

In Eq. 6, $g()$ is the function also known for linking, $E(y)$ is the possibility of required variable and $\alpha + \beta x1 + \gamma x2$ are for linear predictions. $\alpha$, $\beta$, $\gamma$ are the required variables which are predicted. The 'link' function combines the possibility of $E(y)$ with $\alpha + \beta x1 + \gamma x2$. Following parameters are configured to tune the LR model:: penalty=l2, fit_intercept=True, intercept_scaling=1, max_iter=100, multi_class=auto, solver = liblinear.

**Stochastic Gradient Descent (SGD) classifier** takes only one random point while varying the weights. It is more useful when working with a dataset of a larger size. [40].

$$\Theta_1 = \Theta_1 - \alpha(\frac{\sigma}{\sigma\Theta_1} C\widehat{y}_i - y_i) \tag{7}$$

Equation 7 represents the standard equation of SGD in which $\theta_1$ is the parameter, $\widehat{y}$ is the model, and where $y$ is the subject in the supervised dataset. Following parameters are configured to tune the SGD model: loss=hinge, penalty=l2, fit_intercept=True, max_iter=1000, learning_rate=optimal, early_stopping=False.

**Ridge classifier** interchanges the labeled data in the range of $[-1, 1]$. The model outputs the final prediction based on the highest value attained during prediction. Following parameters are configured to tune the RF model: normalize=False, fit_intercept=True, solver=auto.

$$\widehat{\beta}^{ridge} = arg_{\beta\epsilon\mathbb{R}_p} min \sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{8}$$

Equation 8 defines the standard equation of the ridge classifier. This equation has 2 segments. Before the addition sign, the first segment denotes the least square term or loss and the second part denotes the lambda of the summation of $\beta^2$ where $\beta$ is the coefficient. Several research types suggest utilizing ensemble methods to obtain better performance as final predictions [41, 42]. In this research, the following machine learning classifiers are used as 3 layers to form a meta classifier: 1) Stochastic Gradient Descent, 2) Logistic Regression, and 3) Ridge classifier to analyze the selected features of dataset UNSW and detection of anomalies.

**Ensemble Model:** Suppose $D$ denote the dataset containing instances $I = \{i_1, i_2, \ldots, i_n\}$. *CP* represents each classifier's confidence prediction, and *CT* represents the targeted confidence threshold which is set to evaluate the *CP* of each classifier. Suppose *PL* represents each classifier's predicted classifier and *ATL* denotes the All target classes. Here $sgd_{acc}$ denotes the accuracy score predicted by the SGD classifier. *rc* denotes the accuracy score of the ridge classifier. *lr* denotes the accuracy of logistic regression. The notation *IC* represents the instance of each class, whereas the number of classes as a sum is denoted by *ICC*, which is incremented upon a particular classifier's vote in favor of a class's prediction. Each prediction of three evaluated models on every instance *I* is the input given to the voting classifier for evaluated prediction as an anomaly reading or regular and then added in *IC*. The *ICC* confidence and *TL* are then estimated. The classifier casts their prediction to the vote counter. The value of ground truth is configured at 80% to perform the comparison between the certainties. When the prediction results yield the same number of evaluated prediction votes and arbitrary decisions, anyone can be selected as a classification result; if the *CL* value is more significant than the initialized threshold, the target class will be selected as a resultant label of that attribute. Following are the technical explanation of the proposed ensemble-based machine learning model as shown in Algorithm 1.

**Input:** $Reading \leftarrow CloudNetworkAttacksReadings$

**Output:** Benign, Anomalous

$i \leftarrow [Reading]$ {Current Reading}
$AR \leftarrow []$ {All Reading}
$CP \leftarrow []$ {Confidence Predictions}
$CT \leftarrow 80$ {Confidence Thresh hold}
$PC \leftarrow [Benign, Anomaly]$ {predict classes}
$C \leftarrow \phi$ {Confidence}
$ATL \leftarrow len(PL)$ {All target class labels}
$RC \leftarrow NULL$ {Record class}
$RCC \leftarrow NULL$ {Record class Count}
$sgd_ac \leftarrow \mathbf{SGD}(AI)$ {SGD classifier}
$lr_ac \leftarrow \mathbf{LR}(AI)$ {Logistic Regression}
$rc_ac \leftarrow \mathbf{RC}(AI)$ {Ridge classifier}
**for** *each i in I* **do**
    $AI \leftarrow AI++$
    $RC \leftarrow \mathbf{getClassification}(sgd_ac(i), lr_ac(i), rc_ac(i))$
    $RCC[PL] \leftarrow IC++$
    $(C, ATL) \leftarrow \mathbf{getHighestConfidenceLevel}(RCC, PL)$
    **if** $(C \geq CT)$ **then**
        $RCC \leftarrow PL$
    **end if**
**end for**
**return** $max(RCC)$

**Algorithm 1** Ensemble Machine Learning Classifier

**Table 1** Achieved Results (%) using binary classification of anomalies

| Algorithms | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Decision Tree classifier | 91.86 | 91.65 | 99.78 | 95.54 |
| XGBoost | 93.34 | 93.38 | 99.59 | 96.39 |
| Random Forest classifier | 93.57 | 93.59 | 99.63 | 96.52 |
| Gradient Boosting classifier | 94.40 | 94.72 | 99.36 | 96.99 |
| *Machine Ensemble* | 97.06 | 98.39 | 98.45 | 98.45 |

### Multicalass Anomaly Detection

The Multi-Layer structure of the CNN processes the input to manipulate for the desired outcome [43]. CNN requires less preprocessing in the initial phase than other classification techniques as it comprises a dense neural network based on multiple layers, as depicted in Fig. 1.

$$h_t = H(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \tag{9}$$

$$p_t = W_{hy}y_{t-1} + b_y \tag{10}$$

Equations 9 and 10 represents the LSTM based neural network core computations where $x_t$ denotes the input time series, $y_t$ denotes the output time series, $h_t$ indicates the hidden memory cells, $W$ indicates the weight matrices, and $b$ indicates the bias vectors. The hidden state of memory cells is calculated in the following Eqs. 11, 12, 13, 14, 15 where $i_t$ represents the input gate, $f_t$ represents the forget gate, $c_t$ represents cell state and $o_t$ represents the output gate. The cell state carries cumulative information of the sequence data from one time step to the next time step till the end of the sequence. Based on these gates, the hidden state is calculated. Cell state passes through a 'tanh' function reducing all feature values between -1 and 1, enabling it to decide on the labels.

$$i_t = \sigma(W_{ix}x_t + W_{hh}h_{t-1} + W_{ic}c_{t-1} + b_i) \tag{11}$$

$$f_t = \sigma(W_{fx}x_t + W_{hh}h_{t-1} + W_{fc}c_{t-1} + b_f) \tag{12}$$

$$c_t = f_t * c_{t-1} + i_t * g(W_{cx}x_t + W_{hh}h_{t-1} + W_{cc}c_{t-1} + b_c) \tag{13}$$

$$o_t = \sigma(W_{ox}x_t + W_{hh}h_{t-1} + W_{oc}c_{t-1} + b_o) \tag{14}$$

$$h_t = o_t * h(c_t) \tag{15}$$

Following configuration is made for CNN-LSTM to detect anomalies: two layers of 1-dimensional convolutional, filter size of each layer respectively $32 * 3$ and $64 * 3$, activation=relu, padding=causal, maxpooling layer with pool size 2, for LSTM layer recurrent_dropout=0.1, flatten layer use, Root Mean Square Propagation optimizer with learning rate=0.005 use, loss=binary_crossentropy, validation_split=0.33, batch_size=2048, epochs=8.

The working of whole multiclass CNN-based LSTM is given in Algorithm 2. Let $D$ represents the dataset which contains instance $I = \{i_1, i_2, \ldots, i_n\}$ and $LE$ represents the label encoding transformer function which changes the labels into 1-dimensional vectors, $V$. The mean, $\mu$, is subtracted from data for normalization and then normalizes the variance $\sigma$. Then data is converted to get a 2D matrix. The library NumPy is used for this operation. Then Gaussian variable is used to initialize the weights. $L$ denotes the total number of layers, $n$ denotes the total number of features, and $W$ denotes the matrix's weight. $x * y$ denotes the dimension of the generated weight matrix. $D_2$ denotes the 2D matrix that contains the dataset of the training file, which is

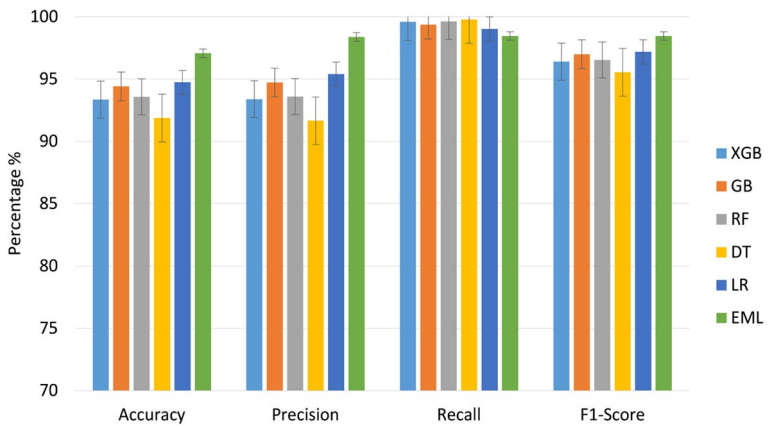


**Fig. 2** Performance Metrics Comparison of Machine Learning Algorithms

further processed into a 3D matrix, $D_3$. This procedure is supported by the reshape function to get the input ready for processing into the CNN model. Two states are used for feature extraction by applying $32 * 3$ and $64 * 3$ filters. The feature map $F$ is generated by the CNN model and converted into 1-dimensional vectors $V$ after applying a max-pooling layer. $V$ is a feature vector fed to the LSTM layer as input to the LSTM functionality model. This information is forwarded to the flatten layer, which converts into a 1-dimensional array. Flatten layer 1-dimensional array denoted by *flstm*. This 1-dimensional data pass to the dense layer as input to predict the target labels. Last train for eight epochs. At every epoch, the LSTM model learns its weights to improve its accuracy by updating weights. Actual loss, validation loss, actual accuracy, and validation accuracy are measured after every epoch.

**Input:** data ← Cloud Network Attacks Readings

**Output:** Normal, Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, Shellcode, Worms

$V \leftarrow LE(data)$ {Label Encoding}

$\mu \leftarrow 1/m * \sum_{i=1} X^{(i)}$ { Normalizing data }

$X \leftarrow X - \mu$

$\sigma^2 \leftarrow 1/m * \sum_{i=1} X^{(i)2}$

$X/ = \sigma^2$

$D2 \leftarrow np.array(Df)$ {Convergence of Matrix}

**for** $l$ *in* $range(1, len(L)$ **do** {Weight Initialization}

  $W[l] \leftarrow rand((m \times n)) * \sqrt{2/n[l-1]}$

**end for**

$D3 \leftarrow$ **ReshapeMatrix**$(D2)$ {CNN Model }

$D4 \leftarrow$ **STATE**$(D3)$

$F \leftarrow$ **STATE**$(D4)$

$V \leftarrow$ **MaxPooling**$(F)$ {Conversion of Vector}

$lstm \leftarrow$ **LSTM**$(V)$ {LSTM layer}

$flstm \leftarrow$ **Flatten**$(lstm)$ {Flatten layer}

$PC \leftarrow$ **PredictClass**$(flstm)$ {Dense layer}

**for** $i$ *in* $range(1, len(PC)$ **do**

  **if** $(PC[i] = y\_test[i])$ **then**

    **return PC[i]**

  **else**

    **return y_test[i]**

  **end if**

**end for**

**return** Output

**Algorithm 2** Multiclass CNN based LSTM

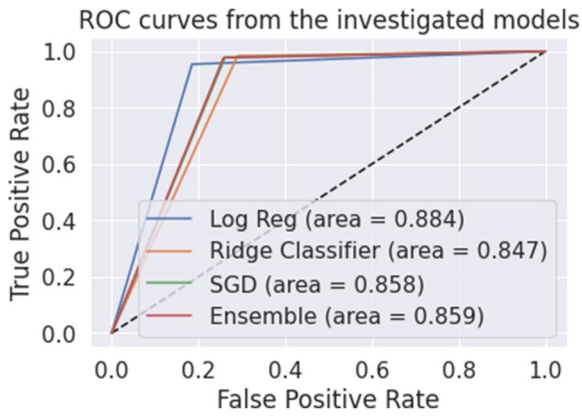


**Fig. 3** ROC curve of the Proposed Approach

## Results and Evaluations

We use the following computing environment for experiments. We use Windows 10 Professional 20H2 operating system, Intel(R) Core(TM)i7-6700HQ, 16*GB* RAM, NVIDIA GeForce 1060 GPU, CUDA 9.0 and Python 3.8 version.

### Binary Anomaly Detection

Table 1 provides an overview of the overall results achieved by each classifier for cloud-based anomaly detection. DT classifier is trained for cloud-based anomaly detection. DT achieved an accuracy of 91.86%, a Precision of 91.65%, a Recall of 99.78%, and an F1-score of 95.54%. Due to complex and high-dimensional data, DT does not provide promising performance to achieve an acceptable accuracy score. XGB method yielded an accuracy of 93.34%, a Precision of 93.38%, a Recall of 99.59%, and an F1-score of 96.39%. XGB also deals with irrelevant features without affecting prediction performance by implementing the decision tree with boosted gradient, and yet it yields better results from the decision tree. Applying the RF method yielded an accuracy of 93.57%, the Precision of 93.59%, a Recall of 99.63%, and an F1-score of 96.52%. Random forest is called a bagging algorithm which reduces the variance of data. In UNSW data, the variance is high. Thus, RF tends to improve accuracy, Precision, Recall, and F1-score more than others. Applying the GB method to training features achieved the accuracy of 94.40%, the Precision of 94.72%, Recall of 99.36%, and F1-score of 96.99% on binary classification.

UNSW dataset is used to train the EML, which contains 175, 341 features in the training set as shown in Table 1. The model is evaluated on a testing set containing 82, 332 features. Applying the ensemble method on stochastic gradient descent, logistic regression, and ridge classifier achieved the accuracy of 97.06%, the Precision of 98.39%, and a Recall of 98.45% F1-score of 98.45% on binary classification.

Figure 2 shows the graphical representation of performance metrics comparison between the evaluated machine learning algorithms and the proposed ensemble machine learning model, as discussed in the above sections.

Figure 3 shows the graphical representation of the Receiver Operating Characteristic (ROC) curve of the machine learning models used in the EML approach, including logistic regression, ridge classifier, and stochastic gradient descent. Combining these algorithms, the ensemble machine learning approach is also depicted in the graph in terms of ROC.

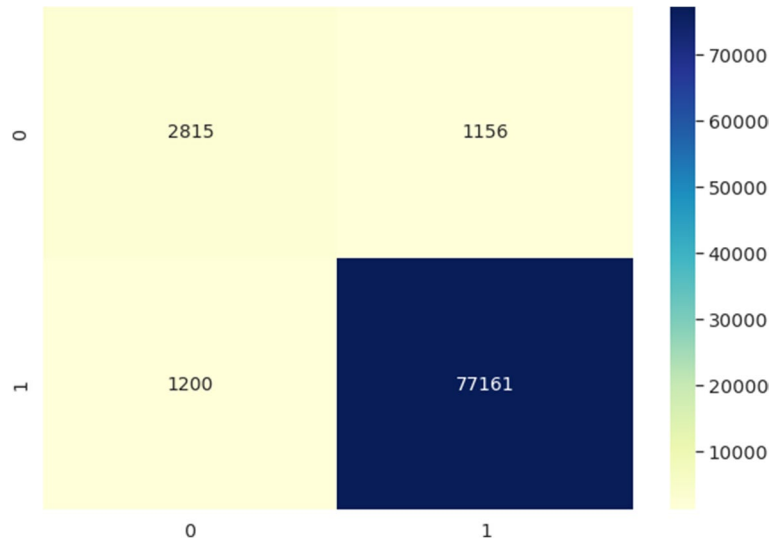


**Fig. 4** Confusion Matrix of *Machine Ensemble* Model

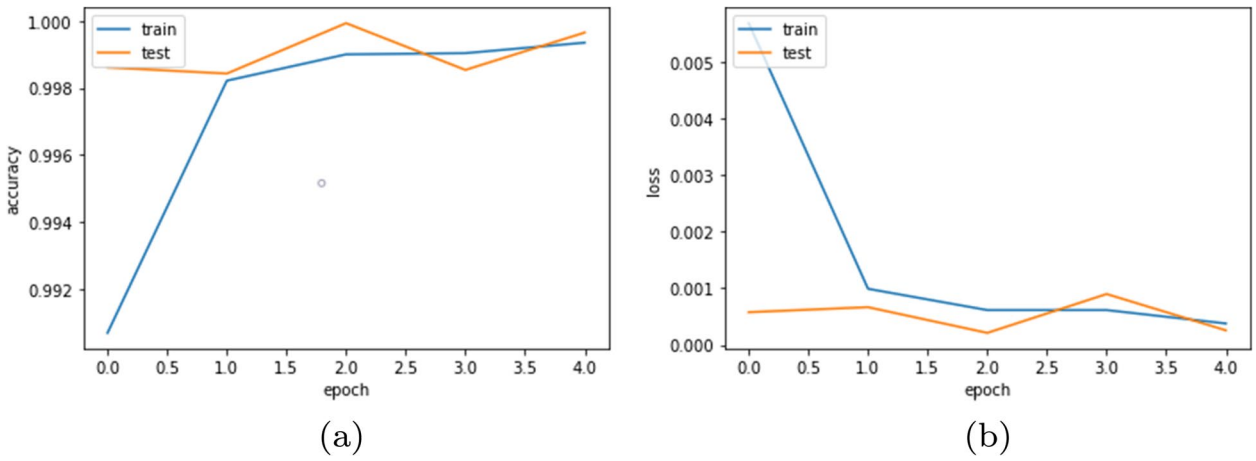


**Fig. 5** Graphical Representation of Performance Evaluation of *Deep CNN-LSTM*

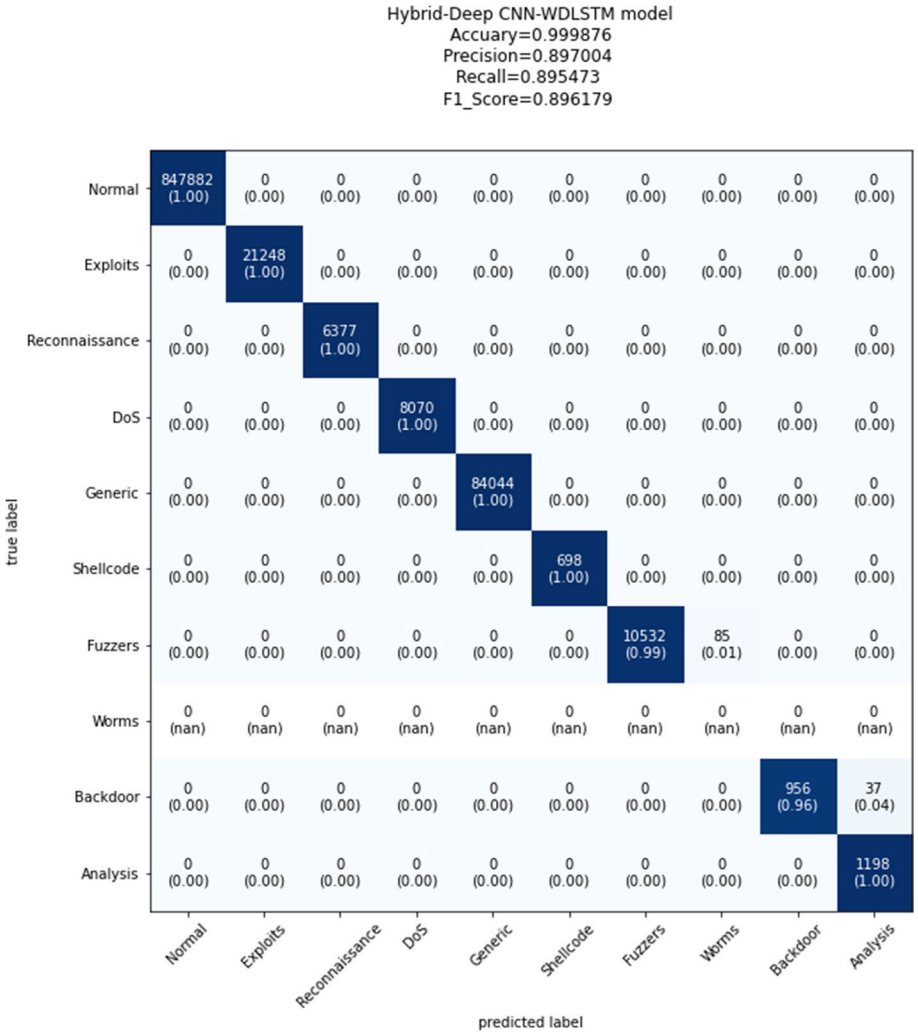


**Fig. 6** Confusion Matrix of *CNN-LSTM* Evaluation

**Table 2** Comparative Analysis

| Research | Algorithms | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| [44] | Random Forest | 97.49% | 97.75% | 93.53% | - |
| [45] | Extra Trees | 86.57% | - | - | - |
| [46] | J48 | 98.71% | - | - | - |
| [25] | DNN | 99.16% | - | - | - |
| [33] | Extra Trees | 99.25% | - | - | 92% |
| This study | *EML* | 97.06% | 98.39% | 98.45% | 98.45% |

Figure 4 represents the confusion matrix generated from evaluation of the our *Machine Ensemble*. It depicts instances considered Network-based anomalies or wrongly identified as other classes. There are 1258 instances wrongly identified as anomalous instances, while 1155 instances are wrongly identified as normal instances.

### Multi Class Anomaly Categorization

In this section, the evaluation of *Deep CNN-LSTM*, the second segment of the proposed approach, is discussed. Figure 5a, the accuracy of the model during the test and train phase over the epochs has been depicted. It can be seen that the model shows the highest accuracy near the second epoch. Figure 5b shows the loss trend over the successive epochs during the train and test phase.

The Fig. 6 depicts the confusion matrix of model *Deep CNN-WDLSTM*. The confusion matrix depicts the model accuracy while identifying the class of the type of attack. The confusion matrix shows that the model outcomes excellent performance in identifying the correct class of attacks.

## Comparative Analysis

In Table 2, the comparative analysis is presented that comprises various research results from the articles [25, 33, 44–46]. In 2018, the author of research [44] utilized a random forest classifier to achieve the accuracy of 97.49% with the Precision of 97.75%. Similarly in 2019, [46] and [25] presented their researches which improved the previous results, [46] used J48 decision tree algorithm to achieve 98.71% accuracy and [25] used DNN algorithm to achieve 99.16% accuracy. Over the years, many kinds of research were conducted in which recently, in 2020, [33] achieved 99.25% accuracy by utilizing the Extra Trees algorithm with an F1-score of 92%. In this research, we contributed to work on the original files of the dataset for classification purposes and achieved the accuracy of 97.06% with an F1-score of 98.45% by utilizing the *CAD* method.

Figure 7 shows the graphical representation of the performance metrics comparison of the proposed approaches: ensemble and machine learning and the CNN-LSTM model. The graph shows that the ensemble machine learning model performs better than the CNN-LSTM model in detecting the attack in the binary outcome category in terms of Precision, F1-score, and Recall. However, better accuracy is being observed by the CNN-LSTM model than in the EML model.

## Discussion

Network connectivity is one of the essential features of the digital world since its medium connects the world by all means. With this digital advancement, hackers have discovered vulnerabilities in the networking systems, thus exploiting them. However, AI-based cybersecurity solutions encounter those attacks with proficiency. In this research, the dataset UNSW-NB15 is analyzed by preprocessing, and feature extraction, and then the data is divided into training and testing samples. The selected ratio for training is 80%, and testing is 20% samples. This sample ratio has already been given in the original files of the UNSW-NB15 dataset. Then different machine learning classifiers are trained on

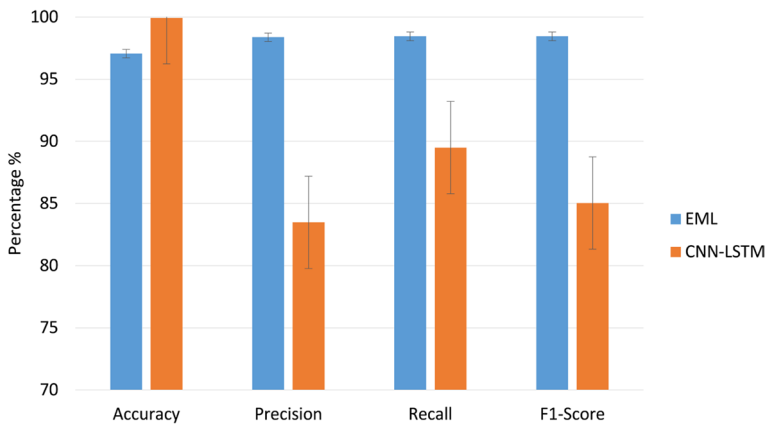


**Fig. 7** Performance Metrics Comparison of Proposed Approaches

the dataset. To improve the final prediction ensemble method is used to bring out the optimal results. In this experiment, *CAD* classification technique comprises the best combination of layers of SGO, Ridge, and Logistic regression machine learning algorithms to achieve the highest accuracy in final predictions. Our approach *CAD* achieves the highest accuracy of 97.06% with the F1-score of 98.45%. Other algorithms also achieved good results in which the Gradient Boosting classifier achieved 94.4% accuracy with an F1-score of 96.99%. Then XGboost and Random Forest classifier achieved 93% accuracy with 96% F1-score. In the end, the lowest accuracy in this experiment is achieved by the Decision Tree classifier, which is 91.86% with an F1-score of 95.54%. This approach comprises the use of original files from the dataset, which is the main contribution of this dataset.

## Conclusion and Future Work

In this paper, a novel AI-based technique is proposed, namely, *CAD* which is composed of an ensemble machine learning model and deep learning-based CNN-LSTM technique to efficiently detect and classify the anomalies by using the state-of-the-art dataset, UNSW-NB15. This dataset contains all sorts of critical attacks that are regarded as harmful to the systems. The proposed approach *CAD* achieved the highest accuracy of 97.06% with precision, recall, and F1-score of 98.38%, 98.45%, and 98.45%, respectively. In the future, we plan to combine the UNSW-NB15 dataset with other anomaly and signature-based datasets to filter the dataset to critical features to enhance the automated anomaly detection systems' performance. In order to keep pace with the advancement in computing [47], as well as respond to the matured offensive techniques effectively, well in time detection has become of utmost importance. The use of cloud computing made it possible to have numerous computing power available to the researchers [48] that can be researched and utilized for such integration of trained cloud-based models to provide anomaly detection as a service offering. For such a global anomaly detection mechanism, Federated learning can be utilized to fight various security events like spam detection, anomaly identification, behavioral-based security, and other network-based attacks. Explainable artificial intelligence (XAI) can be an excellent option to understand why a model made a decision [12]. Furthermore, fog computing and sever-less computing can be used to reduce the latency and improve privacy [49].

**Acknowledgements**
Not applicable.

**Authors' contributions**
Conceptualization, Abdul Mannan, Abdul Rehman Javed and Faisal Shahzad; Data curation, Faisal Shahzad; Formal analysis, Abdul Rehman Javed, Thar Baker, Dhiya Al-Jumeily OBE; Funding acquisition, Abdul Mannan, Dhiya Al-Jumeily OBE; Investigation, Abdul Rehman Javed; Methodology, Abdul Rehman Javed, Ahmad S. Almadhor, Thar Baker; Project administration, Thar Baker and Abdul Mannan; Resources, Faisal Shahzad and Abdul Mannan, Dhiya Al-Jumeily OBE; Software, Abdul Rehman Javed, Ahmad S. Almadhor; Supervision, Abdul Rehman Javed, and Thar Baker; Validation, Ahmad S. Almadhor and Faisal Shahzad; Visualization, Faisal Shahzad and Thar Baker; Writing â review & editing, Abdul Mannan and Faisal Shahzad. The author(s) read and approved the final manuscript.

**Funding**
This research received no external funding.

**Availability of data and materials**
Not applicable.

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

**Competing interests**
Not applicable.

**Author details**
[1]Department of Cyber Security, Air University, 44000 Islamabad, Pakistan. [2]National University of Computer and Emerging Sciences, 44000 Islamabad, Pakistan. [3]Department of Cyber Security, PAF Complex, E-9, Air University, Islamabad, Pakistan. [4]Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon. [5]College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia. [6]Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, United Arab Emirates. [7]School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, UK.

Received: 10 May 2022 Accepted: 3 September 2022
Published online: 03 November 2022

**References**

1. Mohiyuddin A, Javed AR, Chakraborty C, Rizwan M, Shabbir M, Nebhen J (2022) Secure cloud storage for medical iot data using adaptive neuro-fuzzy inference system. Int J Fuzzy Syst 24(2):1203–1215
2. Ahmad W, Rasool A, Javed AR, Baker T, Jalil Z (2021) Cyber security in iot-based cloud computing: A comprehensive survey. Electronics 11(1):16
3. Kiani R, Keshavarzi A, Bohlouli M (2020) Detection of thin boundaries between different types of anomalies in outlier detection using enhanced neural networks. Appl Artif Intell 34(5):345–377
4. Javed AR, Usman M, Rehman SU, Khan MU, Haghighi MS (2020) Anomaly detection in automated vehicles using multistage attention-based convolutional neural network. IEEE Trans Intell Transp Syst
5. Corallo A, Lazoi M, Lezzi M (2020) Cybersecurity in the context of industry 4.0: A structured classification of critical assets and business impacts. Comput Ind 114:103165
6. Telikani A, Gandomi AH, Choo KKR, Shen J (2021) A cost-sensitive deep learning-based approach for network traffic classification. IEEE Trans Netw Serv Manag 19(1):661–670
7. ur Rehman S, Khaliq M, Imtiaz SI, Rasool A, Shafiq M, Javed AR, Jalil Z, Bashir AK, (2021) Diddos: An approach for detection and identification of distributed denial of service (ddos) cyberattacks using gated recurrent units (gru). Futur Gener Comput Syst 118:453–466
8. Mittal M, Iwendi C, Khan S, Rehman Javed A (2021) Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive

clustering hierarchy protocol using levenberg-marquardt neural network and gated recurrent unit for intrusion detection system. Trans Emerg Telecommun Technol 32(6):e3997
9. Rehman A, Rehman SU, Khan M, Alazab M, Reddy T (2021) Canintelliids: detecting in-vehicle intrusion attacks on a controller area network using cnn and attention-based gru. IEEE Trans Netw Sci Eng
10. Imtiaz SI, ur Rehman S, Javed AR, Jalil Z, Liu X, Alnumay WS, (2021) Deepamd: Detection and identification of android malware using high-efficient deep artificial neural network. Futur Gener Comput Syst 115:844–856
11. Ahmed W, Shahzad F, Javed AR, Iqbal F, Ali L (2021) Whatsapp network forensics: Discovering the ip addresses of suspects. In: 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS). pp 1–7. https://doi.org/10.1109/NTMS49979.2021.9432677
12. Javed AR, Shahzad F, ur Rehman S, Zikria YB, Razzak I, Jalil Z, Xu G (2022) Future smart cities requirements, emerging technologies, applications, challenges, and future aspects. Cities 129:103794
13. Warkentin M, Orgeron C (2020) Using the security triad to assess blockchain technology in public sector applications. Int J Inf Manag 102090
14. Wang R, Ji W (2020) Computational intelligence for information security: A survey. IEEE Trans Emerg Top Comput Intell 4(5):616–629
15. Afzal S, Asim M, Javed AR, Beg MO, Baker T (2021) Urldeepdetect: A deep learning approach for detecting malicious urls using semantic vector models. J Netw Syst Manag 29(3):1–27
16. Song HM, Woo J, Kim HK (2020) In-vehicle network intrusion detection using deep convolutional neural network. Veh Commun 21:100198
17. Tahaei H, Afifi F, Asemi A, Zaki F, Anuar NB (2020) The rise of traffic classification in iot networks: A survey. J Netw Comput Appl 154:102538
18. Verma A, Ranga V (2020) Machine learning based intrusion detection systems for iot applications. Wirel Pers Commun 111(4):2287–2310
19. Shahzad F, Javed AR, Jalil Z, Iqbal F (2022) Cyber forensics with machine learning. In: Phung D, Webb GI, Sammut C (eds) Encyclopedia of Machine Learning and Data Science. Springer US, New York. https://doi.org/10.1007/978-1-4899-7502-7_987-1
20. Javed AR, Hassan MA, Shahzad F, Ahmed W, Singh S, Baker T, Gadekallu TR (2022) Integration of blockchain technology and federated learning in vehicular (iot) networks: A comprehensive survey. Sensors 22(12). https://doi.org/10.3390/s22124394. https://www.mdpi.com/1424-8220/22/12/4394
21. Mogal DG, Ghungrad SR, Bhusare BB (2017) Nids using machine learning classifiers on unsw-nb15 and kddcup99 datasets. Int J Adv Res Comput Commun Eng (IJARCCE) 6(4):533–537
22. Moustafa N, Slay J (2015) The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems. In: 2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS). IEEE, pp 25–31
23. Moustafa N, Slay J (2015) Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS). pp 1–6. https://doi.org/10.1109/MilCIS.2015.7348942
24. Chkirbene Z, Erbad A, Hamila R (2019) A combined decision for secure cloud computing based on machine learning and past information. In: 2019 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, pp 1–6
25. Faker O, Dogdu E (2019) Intrusion detection using big data and deep learning techniques. In: Proceedings of the 2019 ACM Southeast Conference. Association for Computing Machinery New York NY United States, Kennesaw, pp 86–93
26. Khan FA, Gumaei A, Derhab A, Hussain A (2019) A novel two-stage deep learning model for efficient network intrusion detection. IEEE Access 7:30373–30385
27. Zina C, Hasna M, Hamila R (2016) Hamdi N (2016) Location privacy preservation in secure crowdsourcing-based cooperative spectrum sensing. EURASIP J Wirel Commun Netw 1:1–11
28. Djenouri Y, Belhadi A, Lin JCW, Cano A (2019) Adapted k-nearest neighbors for detecting anomalies on spatio-temporal traffic flow. IEEE Access 7:10015–10027
29. Chapaneri R, Shah S (2019) A comprehensive survey of machine learning-based network intrusion detection. In: Smart Intelligent Computing and Applications. Springer, pp 345–356
30. Bagui S, Kalaimannan E, Bagui S, Nandi D, Pinto A (2019) Using machine learning techniques to identify rare cyber-attacks on the unsw-nb15 dataset. Secur Priv 2(6):e91
31. Elsayed MS, Le-Khac NA, Jurcut AD (2020) Insdn: A novel sdn intrusion dataset. IEEE Access 8:165263–165284
32. Kumar V, Sinha D, Das AK, Pandey SC, Goswami RT (2020) An integrated rule based intrusion detection system: analysis on unsw-nb15 data set and the real time online dataset. Clust Comput 23(2):1397–1418
33. Sarhan M, Layeghy S, Moustafa N, Portmann M (2020) Netflow datasets for machine learning-based network intrusion detection systems. arXiv preprint arXiv:2011.09144
34. Mebawondu JO, Alowolodu OD, Mebawondu JO, Adetunmbi AO (2020) Network intrusion detection system using supervised learning paradigm. Sci Afr 9:e00497
35. Janarthanan T, Zargari S (2017) Feature selection in unsw-nb15 and kddcup'99 datasets. In: 2017 IEEE 26th international symposium on industrial electronics (ISIE). IEEE, pp 1881–1886
36. Fletcher S, Islam MZ (2019) Decision tree classification with differential privacy: A survey. ACM Comput Surv (CSUR) 52(4):1–33
37. Resende PAA, Drummond AC (2018) A survey of random forest based methods for intrusion detection systems. ACM Comput Surv (CSUR) 51(3):1–36
38. Son J, Jung I, Park K, Han B (2015) Tracking-by-segmentation with online gradient boosting decision tree. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE Institute of Electrical and Electronics Engineers, Santiago, pp 3056–3064
39. Babajide Mustapha I, Saeed F (2016) Bioactive molecule prediction using extreme gradient boosting. Molecules 21(8):983
40. Netrapalli P (2019) Stochastic gradient descent and its variants in machine learning. J Indian Inst Sci 99(2):201–213
41. Abba SI, Linh NTT, Abdullahi J, Ali SIA, Pham QB, Abdulkadir RA, Costache R, Anh DT et al (2020) Hybrid machine learning ensemble techniques for modeling dissolved oxygen concentration. IEEE Access 8:157218–157237
42. Dong X, Yu Z, Cao W, Shi Y, Ma Q (2020) A survey on ensemble learning. Front Comput Sci 1–18
43. Li R, Pan Z, Wang Y, Wang P (2019) A convolutional neural network with mapping layers for hyperspectral image classification. IEEE Trans Geosci Remote Sens 58(5):3136–3147
44. Belouch M, El Hadaj S, Idhammad M (2018) Performance evaluation of intrusion detection based on machine learning using apache spark. Procedia Comput Sci 127:1–6
45. Idhammad M, Afdel K, Belouch M (2018) Semi-supervised machine learning approach for ddos detection. Appl Intell 48(10):3193–3208
46. Nawir M, Amir A, Yaakob N, Lynn OB (2019) Effective and efficient network anomaly detection system using machine learning algorithm. Bull Electr Eng Inform 8(1):46–51
47. Gill SS, Xu M, Ottaviani C, Patros P, Bahsoon R, Shaghaghi A, Golec M, Stankovski V, Wu H, Abraham A, Singh M, Mehta H, Ghosh SK, Baker T, Parlikad AK, Lutfiyya H, Kanhere SS, Sakellariou R, Dustdar S, Rana O, Brandic I, Uhlig S (2022) Ai for next generation computing: Emerging trends and future directions. Internet Things 19:100514. https://doi.org/10.1016/j.iot.2022.100514. https://www.sciencedirect.com/science/article/pii/S254266052200018X
48. Shahzad F, Iqbal W, Bokhari FS (2015) On the use of cryptdb for securing electronic health data in the cloud: A performance study. In: 2015 17th International Conference on E-health Networking, Application Services (HealthCom), pp 120–125. https://doi.org/10.1109/HealthCom.2015.7454484
49. Gill SS, Xu M, Ottaviani C, Patros P, Bahsoon R, Shaghaghi A, Golec M, Stankovski V, Wu H, Abraham A et al (2022) Ai for next generation computing: Emerging trends and future directions. Internet Things 19:100514

## Publisher's Note