Measurement: Sensors 25 (2023) 100612

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods

K. Samunnisa [a,*], G. Sunil Vijaya Kumar [b], K. Madhavi [c]

[a] Research Scholar, Department of Computer Science and Engineering, JNTUA, Anantapur, A.P, India
[b] Professor of CSE & Dean - CSE and Allied Departments, Lords Institute of Engineering & Technology, Hyderabad, Telangana, India
[c] Professor & HOD of CSE Department, JNTUA College of Engineering, Anantapur, A.P, India

ARTICLE INFO

Keywords:
Distributed cloud computing
Intrusion detection system
NSL-KDD
K-means clustering
Gaussian Mixture Model
Random Forest

ABSTRACT

Cloud Computing is popular nowadays due to its storage and data access services. Security and privacy are prime concerns when network threats increase. Cloud computing offers organizations and enterprises a scalable, flexible, and cost-effective infrastructure to store data on the Web. An anomaly-based IDS implementation protects the integrity of the data in a database by identifying and quarantining records when something appears to have changed unexpectedly. Machine learning based clustering and classification methods are used for anomaly based IDS attack classification and scalability in advanced networking environments. Machine learning is a fast, efficient, and adaptable approach to develop intrusion detection models that can deal with emerging threats, i.e., known and unknown attacks (including zero-day attacks). This paper proposes an efficient Hybrid clustering and classification models for implementing an anomaly-based IDS for malicious attack type classifications such as normal (no intrusion), DoS, Probe, U2R, and R2L using threshold-based functions, and the results are tested with two different threshold values (e), 0.01 & 0.5. The experiments have been performed on two tested datasets, namely, NSL-KDD and KDDcup99. Detection rate, False alarm ratio, and accuracy have been used to study the performance of the proposed methodology. After applying the proposed approach, the K-means with random forest has been shown at two different threshold values to have a better classification accuracy, detection rate, and false alarm rate of 99.85%, 99.78% and 0.09% on the NSL-KDD dataset and 98.27%, 98.12% and 2.08% respectively on the KDDcup99 dataset.

1. Introduction

Cloud network-based Intrusion Detection Systems (IDS) use anomaly-based methods to secure cloud-based applications. In a cloud network, there are many types of attacks on service applications, such as state and protocol attacks, volumetric Denial-of-Service (DoS) attacks [1], and encrypted or malicious input attacks. Injecting intrusions or threats into the system's network compromises its security and confidentiality. A common defense against attacks is known as an Intrusion Detection System (IDS), which will detect suspicious activities and intrusions before any damage is done. For example, an Intrusion Detection System (IDS) is used in cloud infrastructure as an early-warning system against intrusion and its consequences. IDS in cloud infrastructures present challenges such as false positives and the high cost of deploying large IDS systems. There are two common types of IDS: network-based and host-based, which detect and respond to intrusions [2]. Anomaly detection techniques have the ability to identify previously unseen forms of attack. The lack of automatic tuning and the prevalence of false positives are two major issues. In order to detect attacks in large-scale, distributed multi-cloud environments, a number of complicated rules must be configured [3].

Clustering and classification methods are highly recommended for use in intrusion detection. In the last few years, there has been significant development of clustering and classification techniques that can automatically detect new attacks without human intervention. This is why it makes sense to use machine learning to create IDSs that can detect previously unseen threats. The effectiveness of these systems is highly reliant on accurate model tuning and a method for monitoring how attacks are evolving over time. NSL-KDD [4] provides the mechanism for clustering and classification that can be incorporated into IDS to enable the automated discovery of previously unseen threats. i.e., Denial of Service attacks (DoS),R2L, U2R (User to Root Attack), probe, normal

* Corresponding author.
E-mail addresses: samunnisa14@gmail.com (K. Samunnisa), sunilvkg@gmail.com (G.S.V. Kumar), kasamadhavivenkat@gmail.com (K. Madhavi).

https://doi.org/10.1016/j.measen.2022.100612
Received 29 September 2022; Received in revised form 7 November 2022; Accepted 28 November 2022
Available online 29 November 2022

[5]. The primary contribution of this study is to built an intrusion Detection System utilizing hybrid clustering and classification approaches, tested with two alternative threshold values, and evaluated on two benchmark datasets to handle anomaly detection problems in a distributed cloud computing environment.

The key contribution of this research paper is.

1. Identify the types of intrusions in KDDCup99 and NSL-KDD datasets and divide them into training and test datasets for comparative study and evaluation.
2. We propose an anomaly detection approach based on a hybrid clustering and classification model and evaluate its efficiency using two different threshold values to detect intrusions from the two datasets based on detection rate, false alarm ratio, accuracy, F1Score and AUC.
3. Measure the performance of supervised and ensemble supervised learning approaches for detecting individual intrusion attacks.

The rest of the paper is organized as follows: Section II gives a brief background of the study regarding the types of clustering algorithms and their differences in the intrusion detection system. Section III discusses the methodologies that cover the KDDCup99 and NSL-KDD datasets and explains the proposed model; Section IV presents the analysis by comparisons along with results of the proposed work. Finally, Section V concludes the paper.

2. Background

Intrusion Detection Systems detect attacks from outside the system (IDS). IDS are critical for detecting a wide range of attack vectors. Intrusion detection systems (IDS) are primarily concerned with detecting intrusions, which can be viewed as a classification problem. DOS, probe, U2R, R2L, and normal are just a few of the many attack types that can be applied to IDS. IDS detection mechanisms are classified into two types: signature-based and behavior-based [6]. However, there are some drawbacks to signature-based techniques. It detects predefined attacks, also known as known attacks, and has a low false positive rate. Because there are no patterns available, it cannot effectively identify unknown attacks. Maintaining continuously updated attacks is a time-consuming process, and it cannot detect or identify zero-day attacks. Anomaly-based IDS can detect known and unknown attacks and also help identify zero-day attacks [7], but it requires time to tune and has a high false positive rate. Intelligent solutions are required as the number of new attacks and their complexity grows. To address the aforementioned issues, we proposed machine learning-based clustering and classification techniques by which an IDS classification technique constructs the model from the entire labeled data set. Similarly, IDS clustering assists in locating unlabeled data within clusters and does not require labeled data for training.

This paper reviews state-of-the-art machine learning strategies for cloud and network security. In our proposed work, an anomaly-based intrusion detection system is identified using the approaches hybrid clustering and classification, and comparisons are made with existing methodologies to highlight the importance of our proposed approaches in improving cloud and network security [8]. This paper's major emphasis is to analyze features roles in clustering and classification approaches for anomaly-based intrusion detection systems. K-means (Centroid approach) and all other approaches, including distribution-based approaches, distance-based approaches, and DBSCAN are reviewed and compared in the context of intrusions detection. The centroid-based method is a standard clustering technique that can handle both numerical and categorical features.

Similarly, we can use DBSCAN or density-link-based approaches for the density-based approach. Then the dense region is marked as an anomaly in intrusion detection. When it comes to modeling and analyzing data, a distribution-based approach begins with the premise that there must be some predetermined number of distributions in a given data set [9]. Gaussian Mixture Models (GMM) are ideal for representing these different distributions. The mixture of the Gaussian approach can incorporate one or more components to better describe a given data set for clustering analysis. This is a mixture of components which could be Gaussians or some other probabilistic density model, assumed to have finite means and variances and combined component densities that are arbitrary probability density functions.

GMM finds prototypical applications in the clustering process in the unsupervised subfield of machine learning. Segmentation analysis and identifying similarities and differences between dataset observations are typical applications of unsupervised techniques like clustering. Unlike supervised learning models, which are often used to make predictions, this method does not need input from an individual to establish the meaning of the data other than using the data to determine similarities and differences between observations. So, unsupervised machine learning techniques are used in intrusion detection. A monitoring system is set up to identify threats that invade computer systems in an attempt to find intruders and malicious activities.

2.1. K-means VS Gaussian Mixture Model (GMM)

GMM is primarily concerned with calculating weights whereas K-means focuses on placing the centroids (centers of mass) of clusters, as opposed to finding the weights, which are groups of nodes that are connected through short paths.

The K-means seeks to minimize the squared Euclidean distance, whereas the Gaussian Mixture Model (GMM) optimizes the information gain (IG) or log likelihood ratio of each Gaussian component.

The K-means and GMM, which typically assume spherical clusters with uniform cluster probabilities, can be used with non-spherical clusters with varying probabilities by clustering the whole dataset into a set of equally probable cluster centers (i.e, rather than using a different number of clusters), and then re-optimizing the cluster means and covariance matrices using the Gaussian Mixture Model algorithm.

3. Related work

In the past few years, there has been much research into IDS using machine learning methods. Support Vector Machines (SVM) were used to detect anomalies in the KDD dataset by the authors of [10]. The authors of [11] used deep learning based artificial neural networks to construct IDS models for anomaly detection on the same dataset, and the results of the model demonstrate its ability to detect intrusions with high detection accuracy and low false alarm rate, and indicate its superiority in comparison with state-of-the-art methods. The authors of [12] employed cascading classifiers to identify and categorize outliers in KDD datasets even though they were not distributed uniformly. The use of decision trees and random forest (RF) for anomaly detection was proposed in Ref. [13]. In Ref. [14], a decision tree classifier is created for trustworthy intrusion detection. Experimental analysis of two datasets demonstrates the proposed model's ability to produce reliable results. When compared to other models, this strategy offers many benefits in terms of Accuracy (ACC), Detection Rate (DR), and False Alarm Rate (FAR). Multiple machine learning methods can be combined into a single hybrid strategy, as suggested in Ref. [15]. The results prove that the hybrid methods outperform the individual models. Those interested in reading more about machine learning strategies for IDS can do so by consulting the surveys found in Ref. [16]. In recent years, researchers have proposed a number of machine learning strategies for IDS that address one or more of the issues raised above that make machine learning algorithms useful to IDS.

The authors of [17] employed a four-layered classification strategy to identify four distinct forms of attacks in the KDD dataset. Both the overall error and the misclassification error were determined to be relatively low in the specified method. Simplifying the method by

lowering the number of characteristics in the original dataset was also recommended by the authors to increase accuracy and reduce the complexity. The authors did not report mistakes in labeling that happened when one type of attack was mistakenly labeled as another type of attack.

The same dataset and various supervised, unsupervised, and outlier learning techniques were used because some attacks were misclassified, the overall accuracy was below that of the work presented in [18]. Anomaly detection and classification models constructed with machine learning have a widespread application in the KDD dataset. Four distinct types of attacks with immensely different traffic patterns are represented in the KDD dataset. In Ref. [19], KDD is used to classify attack types; the approach showed a low misclassification error. However, these models may struggle in modern multi-cloud environments, which features dynamic attacks and closely related attacks. The KDDcup99 dataset is also getting old, so it may not accurately interpret how networks are used today [20]. According to Ref. [21], SVM is a method that is used in data mining to extract predicted data. The author used the KDDCUP '99 IDS database for classification, which is based on neural networks, and then the author got an accuracy rate of 90% on the training data set and the author also added a 10-fold cross validation experiment, which gave an accuracy rate of 80% on the test set. The literature on IDS includes several classifiers and clustering methods, including unsupervised cluster analysis techniques. Due to the inaccurate classification of some attacks, the overall accuracy of attack detection approaches was lower. We recommend machine learning based hybrid models for handling the above issues in order to increase the detection accuracy of IDS.

4. Methodology

4.1. Dataset

In intrusion detection, the benchmark datasets KDDCup99 [22] and NSL-KDD [23] represents five classes of attacks i.e., normal (no intrusion), DoS (distributed denial-of-service attack), Probe (web application profiling), U2R (User-to-Resource), and R2L (Resource-to-User request) are the different types.

1) Normal: Networks with no intrusions.
2) Denial of Service (DoS): is a method of bringing down a network by overwhelming it with traffic.
3) R2L: Intrusion by a remote system without permission
4) U2R: An unauthorized party tries to log in to a predefined user account.
5) Probe: Probe attacks represent a port scanning type of intrusion which is used to collect the availability and types of applications that are running on a system.

KDDCup99: There have been many incursion datasets for cloud networks, but the KDDCup99 was the first to be produced in 1999. The dataset was developed to enhance IDS performance. In particular, this dataset would allow Cybersecurity researchers to better train algorithms to detect when an intrusion occurs in a network by providing quantitative and qualitative information on the state of the network. In the context of the two-way classification, the class distribution is shown in Fig. 2. A total of 494,021 samples and 43 features in the KDDCup99 dataset, including 395,214 samples from the training data, 49,408 samples from the test data, and 49,399 validation samples as shown in Fig. 1. The data set for our system requires loading and feature extraction. The KDDcup99Train+.txt training set and the KDDcup99Test+.txt evaluation set are used for this purpose.

NSL-KDD dataset: Fig. 2 shows that the NSL-KDD dataset has a total of 160,367 samples with 43 features, including 125,973 training data samples, 22,544 test data samples, and 11,850 validation samples. The data set for our system requires loading and feature extraction. The files KDDTrain+.txt (a training set) and KDDTest+.txt (a test set) are read for this purpose.

4.2. Proposed model

A network intrusion occurs when a malicious entity uses a distributed cloud network to perform actions outside of its permissions and capabilities. Network intrusion detection software identifies these malicious operations to protect the network, notifies users, and prevents them in the future. An anomaly-based IDS implementation protects the integrity of the data in a database by identifying and quarantining records when something appears to have changed unexpectedly. Machine learning based clustering and classification methods are used for anomaly based IDS attack classification and scalability in advanced networking environments. Intrusion detection systems perform three major functions: log and event analysis, pattern matching, and threshold evaluation. This paper focuses on implementing an anomaly-based IDS for malicious attack classifications such as normal (no intrusion), DoS, Probe, U2R, and R2L using hybrid clustering and classification models on benchmark datasets, NSL-KDD [23] and KDD Cup 99 [22] with reference to the threshold-based functions, and the results are tested on two different threshold values (e), 0.01 & 0.5. Compared to traditional IDS models, the proposed hybrid model improves overall accuracy, the detection rate, and the false alarm ratio. The flow of the proposed model is presented in Fig. 3.

4.2.1. Empirical Data Analysis

Empirical Data Analysis concerns with the generation and processing of data for statistical analysis. Data analysis encompasses the principles and techniques for summarizing and analyzing numerical data that is either observational or experimental in nature. Section 4.2 provides a

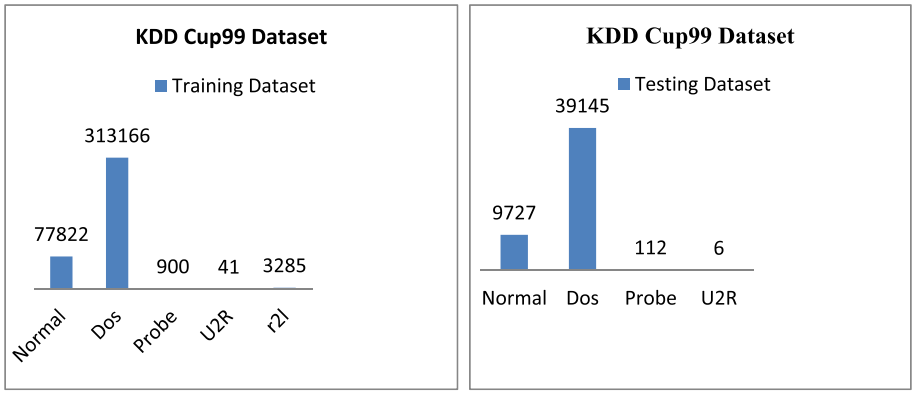

Fig. 1. Training and testing on KDD Cup 99 datasets.

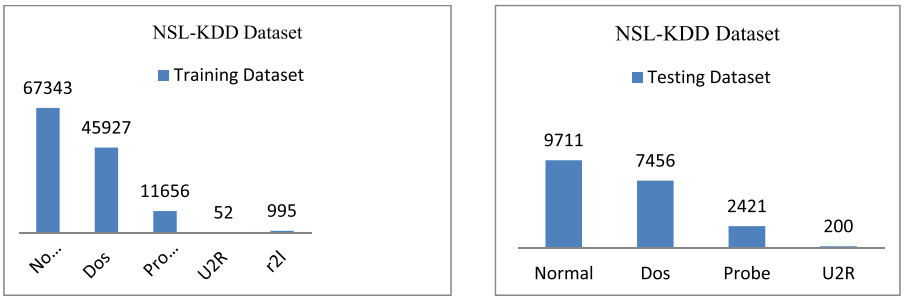

Fig. 2. Proposed benchmark dataset distributions.

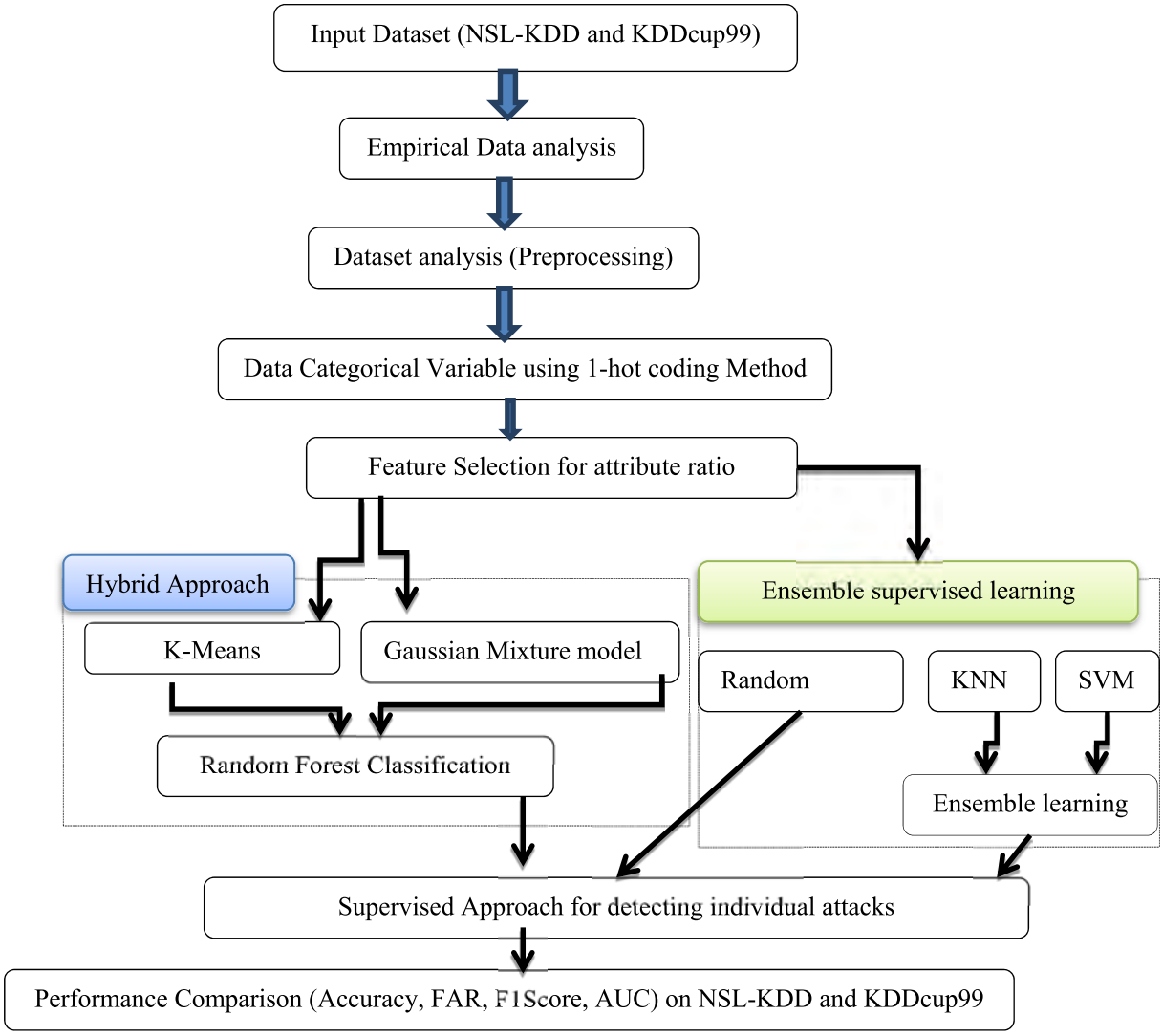

Fig. 3. Proposed Model flow diagram.

more in-depth analysis (See Figs. 4–7).

4.2.2. Data preprocessing

It is all about cleaning and transforming the data before fitting your model on it, such as noise removal, feature extraction, and data filtering. After preprocessing, the features and the labels in our dataset will be changed from their original shape such that they are suitable for model fitting.

4.2.3. Data categorical variable using 1-hot coding method

To categorize data using the 1-hot coding method a response variable is used which is a categorical variable with K possible outcomes, c1; c2; c3 ..,ck, and the original categorical variables can be represented by l values, l1; l2; l3,..,lk. We have to make a table for the original categorical variables l1; l2; l3,..,lk. Once the original categorical variables are converted into a data frame, we can build one-hot encoded columns by taking the l values of each original categorical variable l1; l2; l3,.., lk and

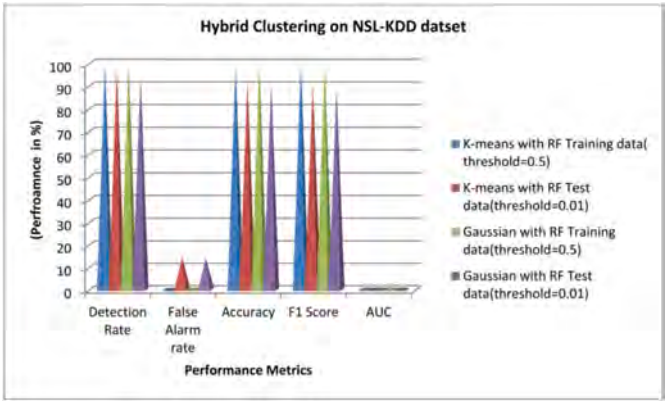

Fig. 4. Hybrid Clustering on NSL KDD dataset.

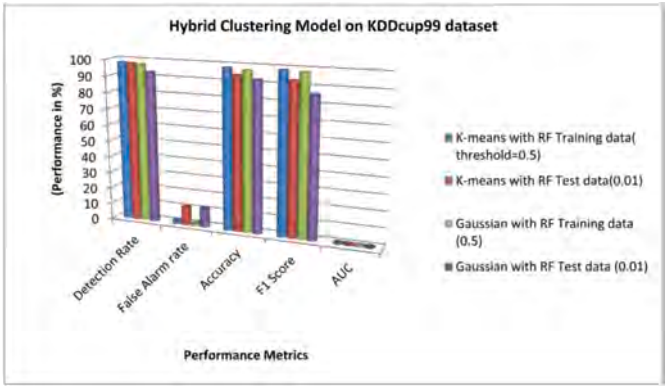

Fig. 5. Hybrid Clustering Model on KDDcup99 dataset.

storing them using their respective labels c1; c2; c3 .., ck and making a separate column for each categorical value for the new columns and adding 0 in the rest of the columns. After creating a data frame using the above method, we have to set new columns as 1-hot encoding using a function named "one_hot_encode".

4.2.4. Feature Selection using Attribute Ratio

Feature Selection using Attribute Ratio The general idea here is that the features that have high variance (unexplained information) are less helpful for the model and therefore should not be considered as strong features with low variance (explained information) should be considered as a strong feature that can help the model better. When dealing with imbalanced classes, certain attributes can be used as a binary feature to indicate which class they belong to. To help with this problem, we can use Attribute Ratio and ROC curves, which are built on top of the Attribute Selection process, and how they interact with the machine learning model when predicting data. We can see that precision, recall, and F1 score increases, when the Attribute Selection process filters out low variance attributes and uses only the high variance attributes.

4.2.5. Hybrid approach K-means and random forest (RF)

Anomaly learning for cyber-attack detection has greatly improved. The anomaly approach also produces a large number of false positives. To maintain accuracy and detection rate while lowering false alarms, we proposed hybrid learning methods. We used K-Means clustering as a pre-classification component in the hybrid learning approach to group data instances by behavior. Random Forests then classified the clusters into attack classes. We found that misclassified data could be reclassified.

4.2.6. K-means clustering

K-means, a type of centroid-based model, is an iterative unsupervised ML algorithm. Assuming that x_{in} stands for a processed dataset wherein $x_{in} \in R^{n_s \times d_e}$ includes n_s number of samples (network flows) each having d_e number of features. The objective is to divide the network flows into $K = 2$ clusters such that the distance between a network flow and center of its cluster is minimized. K-Means Clustering Network intrusion class labels are divided into four main classes, which are DoS, Probe, U2R, and R2L. The primary objective of K-Means clustering is to divide and classify data into benign and malicious instances. With K-Means clustering techniques, the input data set is divided into k-clusters with the help of an initial value, or seed points that is used to determine the cluster centroids. Centroids are the averages of the numbers that make up each cluster. To divide the data into three groups, we settled on $k = 2$. (C1, C2). When K-means is applied on medium or large resized dataset, it minimizes the Intra cluster distance and maximize the Inter Cluster-distance, but number of clusters "K" are predefined.

Algorithm 1. For the models' Training and Testing phases, the K-Means algorithm operates as follows:

Step 1 : Trainig

Step 2 : Initialize cluster Centroids $c1, c2 \in R^{d_e}$ *randomly*

Step 3 : Re*peat*

Step 4 : *for* every i do

Step 5 : $w_{ik} = \begin{cases} 1 & \text{if } k = \arg\ \min \left\|x_i - c_j\right\|^2 \\ 0 & \text{otherwise} \end{cases}$

Step 6 : *end for*

Step 7 : *for each* k do

Step 8 : $C_k = \frac{\sum_{i=1}^{n_s} w_{ik} x_i}{\sum_{i=1}^{n} w_{ik}}$

Step 9 : end for

Step 10 : Untill Conevrgence

Step 11 : Testing or Detection for a given network flow x

Step 12 : Calculate the distance : $d_k = \|x - c_k\|^2, K \in \{1, 2\}$

Step 13 : *if* argmin(d_k) == 1 then

Step 14 : then x belongs to benign cluster

Step 15 : else

Step 16 : the x belongs to malicious Cluster

Step 17 : end if

As shown in Algorithm 1, the operation of updating cluster centers based on network flows and then distributing flows to clusters based on the updated centers is repeated until there is no longer any fluctuation in cluster centers. The expense of repeating this operation in terms of processing resources increases considerably as the size of the dataset expands. To expedite the learning process, we use parallel computing

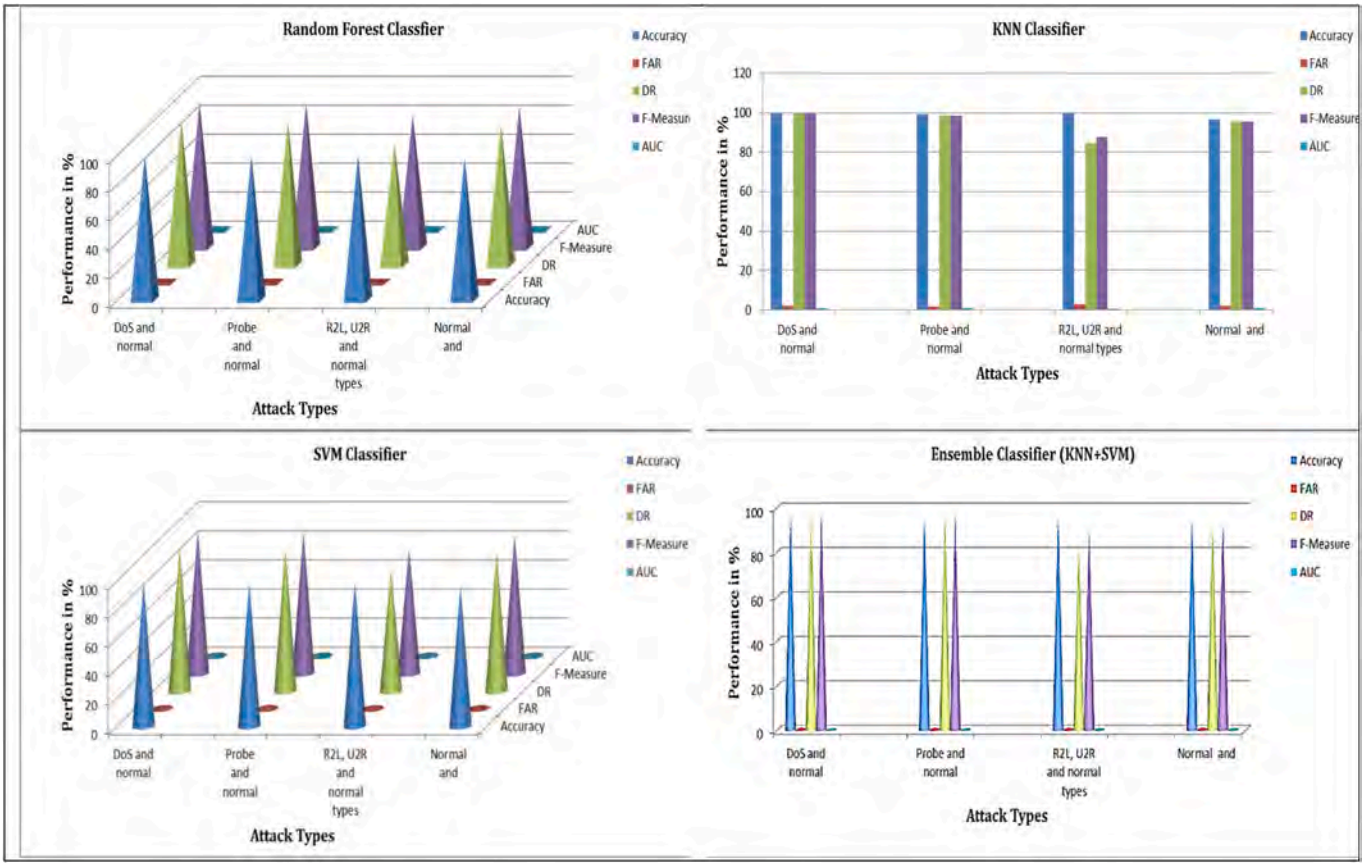

Fig. 6. Analyzing the performance of individual attacks classification with supervised learning classifiers and their hybrid models on the NSL-KDD benchmark dataset.

concepts to construct the K-means method. Furthermore, as indicated in Algorithm 1, the distance between a network traffic sample and the cluster centroid is evaluated while testing models or detecting DoS, Probe, U2R, and R2L attacks. The network flow sample is regarded as safe if there is minimal separation between it and the benign cluster. If it doesn't, it's labeled as unsafe. Function for characterizing the cluster's contents is shown in Table 4.

4.2.7. A Gaussian mixture model (GMM)

It is a probabilistic model used for clustering or classification purposes where each object (example) is classified into one of "K" subsets or clusters, based on probability distribution for each class represented as a mixture of several Gaussian distributions This model generates a number of Gaussian distributions with parameters (the mean and covariance for each mixture) that are optimized during the fitting of the model. In the model, each data point is a mixture of the K Gaussian distributions and each distribution is specified by mean μ and covariance σ2 used in the context of a classification task, GMM learns the mean vector and covariance matrix for each mixture from a training set As an example, given that the data are distributed according to a mixture of two Gaussians, the resulting probability distribution for each class will look like the one shown below in equation (1) [24].

$$N\left(\mu,\sum\right)=\frac{1}{(2\pi)^{d/2}\sqrt{|\sum|}}\exp\left(-\frac{1}{2}\left(x-\mu\right)^{T}\sum^{-1}\left(x-\mu\right)\right) \tag{1}$$

Where $\mu = Mean$

$\sum$ = Covar*aiance* Matrix of the Gaussian

d = Number of features in the dataset

x = No of datapoints

Variance-Covariance Matrix: Covariance is one way to measure the link between two variables. Whether or not a given set of variables is connected to another is irrelevant. So, the variance-covariance matrix is a measure of how these variables are related to each other in the same way that the standard deviation is. It just gives a better, more accurate answer when we have more dimensions.

$$V=\left[\frac{\sum x_1^2}{N}\ \frac{\sum x_1x_2}{N}\ldots\ldots\frac{\sum x_1x_c}{N}\ \frac{\sum x_2x_1}{N}\ \frac{\sum x_2^2}{N}\ldots\ldots\frac{\sum x_2x_c}{N}\ \frac{\sum x_cx_1}{N}\ldots\frac{\sum x_cx_2}{N}\ldots\ldots\frac{\sum x_c^2}{N}\right] \tag{2}$$

Where, V = c x c variance-covariance matrix.

N = the number of scores in each of the c datasets.

X_i = is a deviation score from the i^{th} dataset

$\frac{X_i^2}{N}$ = is the variance of element from the i^{th} dataset

$\frac{X_iX_j}{N}$ = is the covariance for the elements from i^{th} and j^{th} datasets and

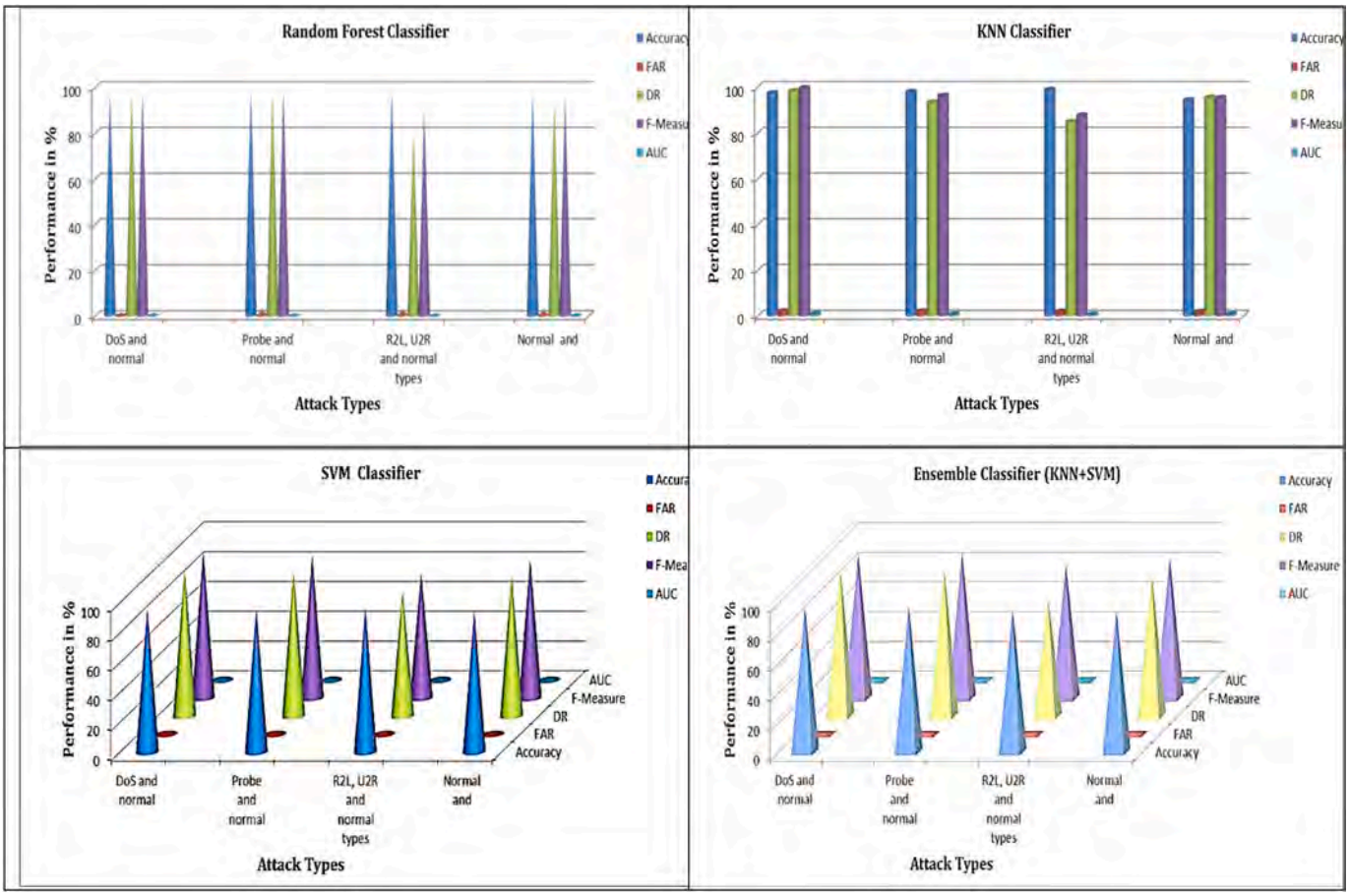

Fig. 7. Analyzing the performance of individual attacks classification with supervised learning classifiers and their hybrid models on the KDD cup99 benchmark dataset.

the probability given in a mixture of K Gaussian where K is a number of distributions:

$$p(x) = \sum_{j=1}^{k} w_j . N\left(x \middle| \mu_j, \sum_j\right) \tag{3}$$

Where w_j is the prior probability of the j^{th} Gaussian

$$\sum_{j=1}^{k} w_j = 1 \text{ and } 0 \leq w_j \leq 1$$

The probability value X for a given X data point is calculated by multiplying the d-dimensional probability distribution function by W, the prior probability of each of our Gaussians. Multiple bell curves would result from plotting multiple Gaussian distributions. We would want a continuous curve made up of several different bell shapes. Once we have that massive continuous curve, we can use it to determine the likelihood that a given data point belongs to a particular class. We want to maximize the likelihood that X belongs to a certain class or locate a class that this data point X is most likely to be a part of. Therefore, we need to get the highest likelihood estimate of X (the data point for which we want to forecast the probability). The k-means algorithm, with which it shares many similarities, is a good example. The same optimization technique can also be used in the expectation maximization method.

4.2.8. Random forest (RF) classifier

The Random Forest algorithm is a mix of decision trees, bagging and boosting, as well as bootstrapping. Due to its ability to maintain unbalanced datasets, estimate missing data, process adequately large datasets without dimensionality reduction, and save computational time by realizing the parallel operation of decision trees. Random Forest is a better model choice than other classification techniques for intrusion detection systems. The classifier has three different steps: Step 1: Data Collection and Preprocessing: Step 2: Training the Algorithm Using a Decision Tree Step 3: Final Classification Step Prediction.

4.3. Significance of the proposed work

In this paper, we propose a hybrid clustering and classification based IDS attack detection and type classification in a large scale cloud distributed network, with the following benefits:

- The proposed method uses cluster-based attack detection, which reduces false alarm rates.
- The proposed method can find attacks in real time without any false positives and with a low amount of computation.
- The proposed method improves the rate of attack detection by building on what worked well in previous cluster-based attack detection methods. Our model can also be used in the future to make other attack detection algorithms and systems work better.

5. Result and analysis

The evaluation of the system was conducted using the following experimental setup [25]: Using Python, Scikit-learn, and PySpark, the Jupyter Notebook is used to process and evaluate the NSL-KDD and KDDcup 99 IDS benchmark datasets (see Table 1).

Table:1
Notations and descriptions.

Notations	Descriptions
x_{in}	Processed dataset
$x_{in} \in R^{n_s \times d_e}$	n_s Number of samples, d_e number of features
K = 2	Number of predefined clusters (C1, C2).
μ	Mean
σ2	Covariance
X	No of data points
d	No of features in the dataset
$\sum$	Covariance matrix of the Gaussian
V = c x c	variance-covariance matrix
N	No of score in the dataset
X_i	deviation score from the i^{th} dataset
K	number of distributions:
w_j	prior probability of the j^{th} Gaussian
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
DR	Detection Rate
e	Threshold value

5.1. Empirical Data Analysis

The study and interpretation of data using empirical evidence is called "empirical analysis." Instead of abstract ideas or theoretical models, the empirical strategy is based on real information from the field, as shown in Tables 2 and 3.

5.2. Data preparation

A Vector Assembler is a programed that combines a list of columns into a single vector column. Then, for indexing categorical (binary) characteristics, the Vector Indexer is utilized. Indexing category characteristics enables algorithms to treat them correctly, resulting in improved performance. The training dataset is split into two parts: 80% training and 20% cross-validation.

5.3. Metrics to evaluate the results

The following performance metrics were computed and are used for evaluation.

True Positive (TP): has classified the intrusion as attack correctly.

True Negative (TN): No attacks have taken place and no detection is made.

False Positive (FP): An event signaling to produce an alarm when no attack has taken place.

False Negative (FN): When no alarm is raised when an attack has taken place.

Accuracy is the total number of connections that have been correctly identified and classified as normal and attack connections.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

False Alarm Rate is the same as the False Positive Rate

False Alarm Rate = FP / (FP + TP)

The Detection Rate is the ratio of the total number of attacks to the number of times they were identified.

Detection Rate (DR) = TN / (TN + FN)

Precision: Precision is given the highest priority and regarded as a measure of success. Precision is represented as the number of successfully identified attacks as a percentage of all records.

Precision = TP / (TP + FP)

Recall is equivalent to the detection rate and is also referred to as sensitivity.

The Area Under the Curve (AUC) represents the overall model performance. The AUC measures how well a model separates out anomalous behavior from attacks.

F1 score is the harmonic mean of precision and recall. F1 = (2 x (Precision x Recall))/(Precision + Recall).

5.4. Implementation of KMeans clustering with Random Forest Classifiers

Random Forest Classifiers were used to train each cluster data

Table 4
Function for characterizing the clusters' contents.

cluster_labels2	attack	normal	Count
0	6659	46,448	53,107
1	9125	2266	11,391
2	626	61	687
3	27,742	101	27,843
4	2670	5073	7743
5	1	0	1
6	0	24	24
7	2	42	44

Table 2
Empirical Data Analysis on Training and Test data - NSL KDD Dataset.

Training				Testing			
Labels2	Count	Labels5	count	Labels2	Count	Labels5	count
Normal	67,343	normal	67,343	Normal	9711	normal	9711
Attack	58,630	Dos	45,927	Attack	12,833	Dos	7458
		Probe	11,656			Probe	2754
		R2L	995			R2L	2421
		U2R	52			U2R	200

Table 3
Empirical Data Analysis on Training and Test data - KDD cup99 Data set.

Training				Testing			
Labels2	Count	Labels5	count	Labels2	count	Labels5	count
Normal	77,822	normal	77,822	Normal	19,454	normal	19,454
Attack	317,392	Dos	313,166	Attack	474,567	Dos	78,290
		Probe	900			Probe	224
		R2L	41			R2L	21
		U2R	3285			U2R	920

Table 5
Confusion matrix for the hybrid model (K-Means + RF) on NSL-KDD training data.

	Actual Values	
Predicted Values	Normal	Attack
Normal	13,316 (TP)	12 (FP)
Attack	26 (FN)	11,779 (TN)

Table 6
Results of Confusion Matrix for K-Means clustering with Random Forest Classifiers with thresholde = 0.5

Measure	Value	Formula
Sensitivity	0.9981	TPR = TP/(TP + FN)
Specificity	0.999	SPC = TN/(FP + TN)
Precision	0.9991	PPV = TP/(TP + FP)
Detection Rate	0.9978	NPV = TN/(TN + FN)
False Alarm rate	0.0009	FDR = FP/(FP + TP)
Accuracy	0.9985	ACC = (TP + TN)/(TP + TN + FP + FN)
F1 Score	0.9986	F1 = 2TP/(2 TP + FP + FN)
AUC	0.998	

Table 7
Confusion matrix for the hybrid model (K-Means + RF) on NSL-KDD test data.

	Actual Values	
Predicted Values	Normal	Attack
Normal	8262 (TP)	1449 (FP)
Attack	182 (FN)	12,651 (TN)

Table 8
Description of the clusters' contents.

cluster_labels2	attack	Normal	count
0	22,895	0	22,895
1	7548	19,846	27,394
2	30	26,900	26,930
3	4024	0	4024
4	4154	2212	6366
5	1136	1007	2143
6	5479	0	5479
7	1559	4050	5609

Table 9
Confusion matrix for the hybrid model (GMM + RF) on NSL-KDD training data.

	Actual Values	
Predicted Values	Normal	Attack
Normal	13,322 (TP)	6 (FP)
Attack	36 (FN)	11,769 (TN)

separately. The clusters are based only on the numerical properties. Since Random Forest provides probabilities, the detection rate for novel attacks may be enhanced by modifying the threshold. Classification using Random Forest is then used to train the model.

Table 10
Confusion matrix for the hybrid model (GMM + RF) on NSL-KDD test data.

	Actual Values	
Predicted Values	Normal	Attack
Normal	8340 (TP)	1371 (FP)
Attack	727 (FN)	12,106 (TN)

Table 11
Depicting the performance of the suggested Hybrid approach on the NSL-KDD dataset using IDS.

Metrics	K-means with RF Training data (threshold = 0.5)	K-means with RF Test data (threshold = 0.01)	GMM with RF Training data (threshold = 0.5)	GMM with RF Test data (threshold = 0.01)
Detection Rate	99.78	98.58	99.78	94.33
False Alarm rate	0.09	14.52	0.05	14.12
Accuracy	99.85	92.77	99.84	90.69
F1 Score	99.86	91.02	99.84	88.83
AUC	0.998	0.918	0.998	0.901

5.4.1. KMeans clustering

Table 6provides the Confusion Matrix for the Hybrid Model known as K-Means Clustering with Random Forest Classifiers at e = 0.5. We discovered that k-means provided a 99.78% detection rate and a 0.09% false alarm rate over the training data (see Table 7) (see Table 5).

5.5. Gaussian Mixture clustering with Random Forest Classifiers on NSL-KDD dataset

The concept behind this method is to first use Gaussian Mixture to divide the data into distinct groups, and then use those groups to tailor the Random Forest classifiers that are trained. The clustering that is generated by a Gaussian Mixture is distinct from that generated by a KMeans algorithm; therefore, it is possible that the two methods outputs are merged to enhance performance (see Tables 8 and 9).

Table 11 shows that on training data, the confusion matrix for K-Means clustering with RF at e = 0.5 has a DR of 99.78% and a FAR of 0.9% (see Table 10). The threshold was set at 0.01 on test data. A probability threshold of 0.01 makes sense because the test data comes from a different distribution and is vulnerable to unknown attack types (0.99 for normal connections). This method detects 98–99% of threats with a 14–15% FAR. There are also confusion matrix results for GMM with RF Classifiers, with a DR of 99.7% of training data and a FAR of 0.5%. The threshold was set at 0.01 at test data. This method has a detection rate of 94.33% and a FAR of 14.12%.

Table 12 shows that on training data, the confusion matrix for Gaussian Mixture clustering with RF at e = 0.5 has a DR of 98.12% and a FAR of 2.08%. The threshold (e) was set at 0.01 on test data. A probability threshold of 0.01 makes sense because the test data comes from a different distribution and is vulnerable to unknown attack types (0.99 for normal connections). This method detects 97.87% of threats with 11.42% FAR. There are also confusion matrix results for GMM with RF Classifiers, with a DR of 97.39% of training data and a FAR of 1.46%. The threshold (e) was set at 0.01 at test data. This method has a DR of 93.21% and a FAR of 11.98%.

Table 12
On KDDcup99, a table compares potential methods and metrics.

Metrics	K-means with RF Training data (threshold = 0.5)	K-means with RF Test data (0.01)	GMM with RF Training data (0.5)	GMM with RF Test data (0.01)
Detection Rate	98.12	97.87	97.39	93.21
False Alarm Rate	2.08	11.42	1.46	11.98
Accuracy	98.27	94.22	97.26	92.34
F1 Score	98.68	93.12	98.21	86.23
AUC	0.948	0.901	0.912	0.880

Table 13
Individual attacks over proposed benchmark datasets NSL-KDD and KDDcup99.

Individual Attacks	NSL-KDD dataset		KDDcup99 dataset	
	Total Count	Individual count	Total Count	Individual count
DoS and normal	90,750	Normal: 54,015 DoS: 36,735	488,732	Normal: 97,276 DoS: 391,456
Probe and normal	63,286	Normal: 54,015 Probe: 9271	98,400	Normal: 97,276 Probe: 1124
R2L, U2R and normal types	54,834	Normal: 54,015 R2L: 782 U2R: 37	101,543	Normal: 97,276 R2L: 62 U2R: 4205
Normal and Attacks	160,367	Normal:77,054 Attack:83,313	889,235	Normal: 97,276 Attack:791,959

Table: 14
Analyzing the performance of individual attacks classification with supervised learning classifiers and their hybrid models on the NSL-KDD benchmark dataset.

Classifiers	Attacks	Accuracy	FAR	DR	F-Measure	AUC
Random Forest	DoS and normal	99.98	1.19	99.66	99.78	0.9415
	Probe and normal	99.66	1.14	99.34	99.37	0.895
	R2L, U2R and normal types	99.76	0.99	83.43	91.70	0.7535
	Normal and Attacks	98.07	1.13	97.01	97.25	0.946
KNN	DoS and normal	99.71	2.10	99.66	99.67	0.9381
	Probe and normal	99.07	1.87	98.50	98.55	0.874
	R2L, U2R and normal types	99.70	2.78	84.83	87.75	0.7211
	Normal and Attacks	96.73	1.99	95.48	95.38	0.913
SVM	DoS and normal	99.37	1.78	99.45	98.99	0.9231
	Probe and normal	98.45	2.75	98.36	97.61	0.852
	R2L, U2R and normal types	99.65	2.31	83.98	85.91	0.7343
	Normal and Attacks	96.79	2.09	96.26	95.52	0.919
Hybrid Classifier (KNN + SVM)	DoS and normal	99.80	1.29	99.70	99.77	0.9401
	Probe and normal	99.28	1.17	98.95	98.90	0.882
	R2L, U2R and normal types	99.76	1.12	86.14	90.64	0.7521
	Normal and Attacks	97.21	1.23	96.43	96.02	0.939

5.6. Classification of individual attacks using hybrid classifiers on benchmark datasets

There are five distinct categories in both the NSL KDD and the KDD cup99 datasets: normal, DoS, probe, r2l, and u2r. Oversampling may be the reason for the inequitable distribution of data across the five categories, with normal and DoS having far more samples than r2l and u2r. As a result of experimenting with various machine learning classification methods and hybrid models. Random forest came out on top with 99.98% accuracy on the NSL-KDD dataset and 98.21% accuracy on the KDDcup99 dataset. Tables 13 and 14 shows that evaluating supervised learning classifiers and their hybrid models on the NSL-KDD and KDD cup99 benchmark dataset to determine the classification precision on specific attacks (see Table 15).

From the above Table 14, Highest Accuracy is classified among the DoS and normal attacks with 99.98%, having an AUC of 0.941. The

Table: 15
Analyzing the performance of individual attacks classification with supervised learning classifiers and their hybrid models on the KDDcup99 benchmark dataset.

Classifiers	Attacks	Accuracy	FAR	DR	F-Measure	AUC
Random Forest	DoS and normal	98.21	1.11	99.33	98.21	0.951
	Probe and normal	99.76	1.92	99.10	99.08	0.901
	R2L, U2R and normal types	99.15	2.16	80.23	90.20	0.721
	Normal andAttacks	98.01	1.87	96.01	95.32	0.946
KNN	DoS and normal	97.45	2.18	98.36	99.67	0.938
	Probe and normal	97.99	2.10	93.40	96.32	0.811
	R2L, U2R and normal types	98.90	1.98	84.83	87.75	0.721
	Normal and Attacks	94.33	1.87	95.48	95.38	0.913
SVM	DoS and normal	97.22	2.18	99.45	99.27	0.912
	Probe and normal	96.87	2.09	98.36	96.66	0.852
	R2L, U2R and normal types	97.32	2.01	84.10	84.56	0.710
	Normal and Attacks	95.10	1.97	95.20	93.21	0.880
Hybrid Classifier (KNN + SVM)	DoS and normal	98.10	1.19	99.70	98.10	0.932
	Probe and normal	99.34	2.01	98.95	98.90	0.852
	R2L, U2R and normal types	97.21	1.90	79.14	90.64	0.761
	Normal and Attacks	95.41	2.02	95.19	95.11	0.912

Table 16
The comparison table of the existing approaches with the proposed approaches.

Author	Techniques	Dataset	DT (%)	FAR (%)	Accuracy (%)
H. P. Vinutha and B. Poornima [26]	K-means with canopy	NSL-KDD	83.12	NA	93.51
Gambo, M. K., & Yasin, A [27].	Simple K-means + Random Forest	NSL-KDD	NA	0.14	99.98
Bangui, H., Ge, M., & Buhnova, B [28].	Random Forest	CICIDS2017	NA	NA	96.93
Bhati, B·S et al. [29]	Weight Extraction Algorithm	KDDcup99	NA	NA	98.13
Gogoi et al. [30]	Subspace based incremental clustering	KDDcup99	NA	NA	97.57
Our Proposed model	K-means + Random Forest	NSL-KDD	99.78	0.09	99.85
Our Proposed model	GMM + Random Forest	KDDcup99	98.12	2.08	98.27

Table 14 shows the comparison of different classifiers on identifying the Individual attacks classification with the performance metrics such as accuracy rate, FAR, DR, F-measure and AUC. As the table results show that random forest classifier performs better among the other classifiers with the highest accuracy detection with a range of (98.7–99.98)% and similarly low FAR is identified with a range of (0.99–1.19) %, DR range is (83.43–99.66)% and AUC ranges between (0.74–0.94) % The suggested approach was compared to others, and it was found that the false alarm rate was lowest when using the random forest classifier.

From the above Table 14, Highest Accuracy is classified among the DoS and normal attacks with 98.21%, and recorded AUC is 0.951. The Table 14 shows the comparison of different classifiers on identifying the Individual attacks classification with the performance metrics such as accuracy rate, FAR, DR, F-measure and AUC. As the table results show that random forest classifier performs better among the other classifiers with the highest accuracy detection with a range of (98.01–99.76)% and similarly Low FAR is identified with range of (0.99–1.19) %, DR varies from (80.23–9933)% and AUC range is within (0.74–0.94) %. And the comparison reveals that the random forest classifier reduced the false alert rate.

Our proposed approach is being compared with existing approaches that have been developed in the past. It addresses the challenges with the existing clustering and classification models. It improves the detection rate and accuracy and reduces the false alarm rate when compared with traditional ones. Table 16 shows the comparison of different approaches in terms of detection rate, accuracy rate, and false alarm rates.

6. Conclusion

From the empirical results and analyses, it can be concluded that the proposed model is efficient enough in detecting various attack types on a cloud environment. The NSL-KDD and KDD99 IDS benchmark datasets were used in an experiment to evaluate the GMM and K-Means clustering methods in conjunction with the RF Classifier. The goal of this research is to confirm and find malicious attacks on the cloud network in real time. This will help the network to stable and safe even though these attacks happen often. In developing an effective IDS with a low false alarm rate, our proposed model shows the result based on scaled threshold points such as 0.5 to 001, i.e., where the threshold is set to 0.5 K-Means clustering with an RF gain detection rate of 99.78% and a false alarm rate of 0.9%. The threshold was set at 0.01 on test data. The detection rate was 98–99% with a 14–15% false alarm rate. Similarly, for GMM with RF Classifiers, with a DR of 99.7% of training data and a FAR of 0.5%, The threshold was set at 0.01 for the test data. This method has a DR of 94.33% and a FAR of 14.12%. KDDcup99 has also been implemented using the same model. Furthermore, this paper implemented a supervised hybrid classifier method for identifying and categorizing individual attacks; the results demonstrated that the random forest classifier performed best for identifying and labeling DoS and normal attacks with high accuracy, low FAR, high DR, and high AUC. Future research will be conducted on more refined methods of modeling network traffic and attack behavior that best represents the parameters of individual attacks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Song Han, Miao Xie, Hsiao-hwa Chen, Yun Ling, Intrusion detection in cyber-physical systems: techniques and challenges, Sys. J., IEEE 8 (2014) 1049–1059, https://doi.org/10.1109/JSYST.2013.2257594.

[2] Ankit Thakkar, Ritika Lohiya, A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions, Artif. Intell. Rev. 55 (2022), https://doi.org/10.1007/s10462-021-10037-9.

[3] Guo Pu, Lijuan Wang, Jun Shen, Fang Dong, A hybrid unsupervised clustering-based anomaly detection method, Tsinghua Sci. Technol. 26 (2021) 146–153, https://doi.org/10.26599/TST.2019.9010051.

[4] Gopal Kushwah, Virender Ranga, Optimized extreme learning machine for detecting DDoS attacks in cloud computing, Comput. Secur. (2021), https://doi.org/10.1016/j.cose.2021.102260.

[5] Bhoopesh Bhati, C.S. Rai, Balamurugan Balamurugan, Fadi Al-Turjman, An intrusion detection scheme based on the ensemble of discriminant classifiers, Comput. Electr. Eng. 86 (2020) 106742, https://doi.org/10.1016/j.compeleceng.2020.106742.

[6] Wenjuan Li, Steven Tug, Weizhi Meng, Yu Wang, Designing collaborative blockchained signature-based intrusion detection in IoT environments, Future Generat. Comput. Syst. 96 (2019), https://doi.org/10.1016/j.future.2019.02.064.

[7] Tommaso Zoppi, Andrea Ceccarelli, Lorenzo Salani, Andrea Bondavalli, On the educated selection of unsupervised algorithms via attacks and anomaly classes, J. Inf. Secur. Appl. 52 (2020) 102474, https://doi.org/10.1016/j.jisa.2020.102474.

[8] Wafa Nabi, Agam Bansal, Bo Xu, Applications of artificial intelligence and machine learning approaches in echocardiography, Echocardiography 38 (2021), https://doi.org/10.1111/echo.15048.

[9] Caihui Lan, Haifeng Li, Caifen Wang, Analysis of the comments on “dentity-based distributed provable data possession in multicloud storage, 1-1, IEEE Trans. Serv. Comput. (2017), https://doi.org/10.1109/TSC.2017.2778250.

[10] Sandip Sonawane, Rule based learning intrusion detection system using KDD and NSL KDD dataset, 04, Prestige International Journal of Management & IT - Sanchayan (2015) 135–145, https://doi.org/10.37922/PIJMIT.2015.V04i02.009.

[11] Zouhair Chiba, Noreddine Abghour, Khalid Moussaid, Amina omri, Mohamed Rida, Intelligent approach to build A deep neural network based IDS for cloud environment using combination of machine learning algorithms, Comput. Secur. 86 (2019), https://doi.org/10.1016/j.cose.2019.06.013.

[12] Rajesh Thomas, Deepa Pavithran, A Survey of Intrusion Detection Models Based on NSL-KDD Data Set, 2018, pp. 286–291, https://doi.org/10.1109/CTIT.2018.8649498.

[13] Erxue Min, Jun Long, Qiang Liu, Jianjing Cui, Wei Chen, TR-IDS: anomaly-based intrusion detection through text-convolutional neural network and random forest, 2018, Secur. Commun. Network. (2018) 1–9, https://doi.org/10.1155/2018/4943509.

[14] Azidine Guezzaz, Said Benkirane, Mourade Azrour, Shahzada Khurram, A reliable network intrusion detection approach using decision tree with enhanced data quality, 2021, Secur. Commun. Network. (2021), https://doi.org/10.1155/2021/1230593.

[15] T. Arvind, A survey on building an effective intrusion detection system (IDS) using machine learning techniques, challenges and datasets, Int. J. Res. Appl. Sci. Eng. Technol. 8 (2020) 1473–1478, https://doi.org/10.22214/ijraset.2020.30598.

[16] Ananya Devarakonda, Nilesh Sharma, Prita Saha, Ramya Lokesh, Network intrusion detection: a comparative study of four classifiers using the NSL-KDD and KDD’99 datasets, J. Phys. Conf. 2161 (2022), 012043, https://doi.org/10.1088/1742-6596/2161/1/012043.

[17] Khalid Al-Gethami, Mousa AL-Akhras, Mohammed Alawairdhi, Empirical evaluation of noise influence on supervised machine learning algorithms using intrusion detection datasets, 2021, Secur. Commun. Network. (2021) 1–28, https://doi.org/10.1155/2021/8836057.

[18] Chao Liu, Zhaojun Gu, Jialiang Wang, A hybrid intrusion detection system based on scalable K-Means+ random forest and deep learning, IEEE Access 9 (2021) 75729–75740, https://doi.org/10.1109/ACCESS.2021.3082147.

[19] Srishti Sharma, Yogita Gigras, Rita Chhikara, Anuradha Dhull, Analysis of NSL KDD dataset using classification algorithms for intrusion detection system, Recent Pat. Eng. 12 (2018), https://doi.org/10.2174/1872212112666180402122150.

[20] Norberto Garcia, Tomas Alcaniz, Aurora González Vidal, Jorge Bernal Bernabe, Diego Rivera, Antonio Skarmeta, Distributed real-time SlowDoS attacks detection over encrypted traffic using Artificial Intelligence, J. Netw. Comput. Appl. 173 (2021), 102871, https://doi.org/10.1016/j.jnca.2020.102871.

[21] Sandip Sonawane, Rule based learning intrusion detection system using KDD and NSL KDD dataset, 04, Prestige International Journal of Management & IT - Sanchayan (2015) 135–145, https://doi.org/10.37922/PIJMIT.2015.V04i02.009.

[22] Mahbod M.Tavallaee, Ebrahim Bagheri, Wei Lu, Ali Ghorbani, A Detailed Analysis of the KDD CUP 99 Data Set, IEEE Symposium. Computational Intelligence for Security and Defense Applications, 2009, https://doi.org/10.1109/CISDA.2009.5356528. CISDA. 2.

[23] NSL-KDD | datasets | research | Canadian institute for cybersecurity | UNB, Online]. Available, http://www.unb.ca/cic/datasets/nsl.html, 2017. (Accessed 4 May 2022).

[24] Chin-Shiuh Shieh, Wan-Wei Lin, Thanh-Tuan Nguyen, Chi-Hong Chen, Mong-Fong Horng, Denis Miu, Detection of unknown DDoS attacks with deep learning and Gaussian mixture model, Appl. Sci. 11 (2021) 5213, https://doi.org/10.3390/app11115213.

[25] Smirti Dwibedi, Medha Pujari, Weiqing Sun, A Comparative Study on Contemporary Intrusion Detection Datasets for Machine Learning Research, 2020, https://doi.org/10.1109/ISI49825.2020.9280519.
[26] H.P. Vinutha, B. Poornima, Analysis of NSL-KDD dataset using K-means and canopy clustering algorithms based on distance metrics, Studies in Computational Intelligence (2018) 193–200, https://doi.org/10.1007/978-981-10-8797-4_21.
[27] Muhammed Gambo, Azman Yasin, Hybrid approach for intrusion detection model using combination of K-means clustering algorithm and random forest classification, Int. J. Eng. Sci. 6 (2017) 93–97, https://doi.org/10.9790/1813-0601029397.
[28] Hind Bangui, Mouzhi Ge, Barbora Buhnova, A hybrid machine learning model for intrusion detection in VANET, Computing 104 (2022), https://doi.org/10.1007/s00607-021-01001-0.
[29] Bhoopesh Bhati, C.S. Rai, Balamurugan Balamurugan, Fadi Al-Turjman, An intrusion detection scheme based on the ensemble of discriminant classifiers, Comput. Electr. Eng. 86 (2020), 106742, https://doi.org/10.1016/j.compeleceng.2020.106742.
[30] Gianluigi Folino, Clara Pizzuti, Giandomenico Spezzano, An ensemble-based evolutionary framework for coping with distributed intrusion detection, Genet. Program. Evolvable Mach. 11 (2010) 131–146, https://doi.org/10.1007/s10710-010-9101-6.