



**ISSN: 2454-9940**



**INTERNATIONAL JOURNAL OF APPLIED  
SCIENCE ENGINEERING AND MANAGEMENT**

**E-Mail :**  
**editor.ijasem@gmail.com**  
**editor@ijasem.org**

**[www.ijasem.org](http://www.ijasem.org)**

# CLOUD BASED INTRUSION DETECTION APPROACH USING MACHINE LEARNING TECHNIQUES

P. SIVA PRASAD, Assistant Professor, Dept of MCA, Chirala Engineering College, Chirala,  
[LakshmiPrasad8216@gmail.com](mailto:LakshmiPrasad8216@gmail.com)

PASUMARTHI VENKATA JANANI, PG Student - MCA, Dept of MCA, Chirala Engineering College, Chirala,  
[pasumarthivenkatajanani@gmail.com](mailto:pasumarthivenkatajanani@gmail.com)

**Abstract:** Cloud computing provides on-demand access to a broad range of network and computer resources, encompassing storage, data management services, computing power, applications, and more. Users can easily access and utilize these resources as needed. The project focuses on enhancing cloud security by implementing an intrusion detection model leveraging machine learning techniques. The primary aim is to monitor and analyze resources, services, and networks within the cloud environment to effectively detect and prevent cyber-attacks. The proposed intrusion detection model utilizes machine learning techniques, specifically emphasizing the use of the Random Forest (RF) algorithm. Random Forest is a powerful ensemble learning method that combines multiple decision trees to make more accurate predictions. Feature engineering is a critical aspect of the model development process. It involves selecting and optimizing relevant features from the dataset to feed into the machine learning model. Effective feature engineering contributes to the model's ability to discern patterns and identify potential attacks accurately. The model's implementation is aimed at improving cloud security by continuously monitoring cloud resources, services, and networks. By applying machine learning

algorithms, the model identifies unusual activities or patterns associated with cyber-attacks, thereby enhancing the overall security posture of the cloud infrastructure. The model's performance is evaluated and validated using two datasets: Bot-IoT and NSL-KDD. These datasets are common benchmarks in the field of intrusion detection. The model demonstrates high accuracy in detecting intrusions compared to recent related works, indicating its effectiveness and reliability in identifying potential security threats. The project's includes a Voting Classifier combination of RF + ADaBoost and Stacking Classifier with RF + MLP with LightGBM got 99% and 100% of accuracy for Kdd-Cup and Bot-IoT data respectively for enhanced cloud detection performance.

*Index terms* -cloud security; anomaly detection; features engineering; random forest.

## 1. INTRODUCTION

Cloud technologies allow practical access on demand to a shared network, storage, and resources and offer more choices regarding their service models[1]. These models are platform as a service (PaaS), software as a service (SaaS), and infrastructure as a service (IaaS)[2], used in one of the deployment

models private, public, and hybrid cloud[3]. The cloud provides services with high performance due to its characteristics[2] according to the National Institute of Standards and Technology[4]: network access, resource pooling, quickelasticity, and measured service.

Recently, the cloud suffers from many security problems like availability, data confidentiality, integrity, and control authorization. In addition, the Internet is used to facilitate access to the services offered by the cloud representing a major source of threats that can infect the cloud systems and resources[2]. Then enhancing cloud security becomes a primary challenge for cloud providers[5]. Therefore, several approaches such as firewall tools, data encryption algorithms, authentication protocols, and others have been developed to better secure cloud environments from various attacks[6]. However, traditional systems are not sufficient to secure cloud services from different limits[7]. Therefore, a set of intrusion detection approaches are proposed and applied to detect and prevent undesirable activities in realtime[8, 9].

In general, the detection methods are divided into misuse detection method which uses known attacks to detect intrusion and anomaly detection method which detect intrusion using unknown attack. The hybridmethod is obtained by combining the advantages of these two methods[10]. Despite of more solutions given to secure cloud environments, the recent intrusion detection systems (IDSs) are affected by various significant limitations[8], for example, huge amounts of analyzed data, real-time

detection, data quality, and others that aim to decrease the performance of detection models.

Nowadays, academic researchers show that intelligent learning methods[6, 11] such as machine learning (ML), deep learning (DL), and ensemble learning are useful in various areas[12, 13] and are able to perform network security[14–18]. Our main goal in this research work is to propose an anomaly detection approach based on random forest (RF) binary classifier and feature engineering is carried out based on a data visualization process aiming to reduce the number of used features and perform the proposed anomaly detection model. The evaluation performances of the model are implemented on NSL-KDD and BoT-IoT datasets. Then, the obtained outcomes demonstrate model performances.

## 2. LITERATURE SURVEY

The cloud computing exhibits, remarkable potential to provide cost effective, easy to manage, elastic, and powerful resources on the fly, over the Internet. The cloud computing, upsurges the capabilities of the hardware resources by optimal and shared utilization. The above mentioned features encourage the organizations and individual users to shift their applications and services to the cloud [1]. Even the critical infrastructure, for example, power generation and distribution plants are being migrated to the cloud computing paradigm. However, the services provided by third-party cloud service providers entail additional security threats. The migration of user's assets (data, applications etc.) outside the administrative control in a shared environment where numerous users are collocated escalates the security

concerns. This survey details the security issues that arise due to the very nature of cloud computing. Moreover, the survey presents the recent solutions presented in the literature to counter the security issues. Furthermore, a brief view of security vulnerabilities in the mobile cloud computing are also highlighted [18,30]. In the end, the discussion on the open issues and future research directions is also presented.

The cloud computing provides on demand services over the Internet with the help of a large amount of virtual storage. The main features of cloud computing is that the user does not have any setup of expensive computing infrastructure and the cost of its services is less. In the recent years, cloud computing integrates with the industry and many other areas, which has been encouraging the researcher to research on new related technologies [2]. Due to the availability of its services & scalability for computing processes individual users and organizations transfer their application, data and services to the cloud storage server. Regardless of its advantages, the transformation of local computing to remote computing has brought many security issues and challenges for both consumer and provider. Many cloud services are provided by the trusted third party which arises new security threats. The cloud provider provides its services through the Internet and uses many web technologies that arise new security issues [1,23,5,7,19]. This paper discussed about the basic features of the cloud computing, security issues, threats and their solutions. Additionally, the paper describes several key topics related to the cloud, namely cloud architecture framework, service and

deployment model, cloud technologies, cloud security concepts, threats, and attacks. The paper also discusses a lot of open research issues related to the cloud security.

Network security has been a very important problem. Intrusion detection systems have been widely used to protect network security. Various machine learning techniques have been applied to improve the performance of intrusion detection systems, among which ensemble learning has received a growing interest and is considered as an effective method [6]. Besides, the quality of training data is also an essential determinant that can greatly enhance the detection capability. Knowing that the marginal density ratios are the most powerful univariate classifiers. In this paper, we propose an effective intrusion detection framework based on SVM ensemble with feature augmentation. Specifically, the logarithm marginal density ratios transformation is implemented on the original features with the goal of obtaining new and better-quality transformed training data; SVM ensemble was then used to build the intrusion detection model. Experiment results show that our proposed method can achieve a good and robust performance, which possesses huge competitive advantages when compared to other existing methods in terms of accuracy, detection rate, false alarm rate and training speed. [6,24]

Cloud computing provides a framework for supporting end users easily attaching powerful services and applications through Internet. To give secure and reliable services in cloud computing environment is an important issue. Providing security

requires more than user authentication with passwords or digital certificates and confidentiality in data transmission, because it is vulnerable and prone to network intrusions that affect confidentiality, availability and integrity of Cloud resources and offered services [1,23,5,7,19]. To detect DoS attack and other network level malicious activities in Cloud, use of only traditional firewall is not an efficient solution. In this paper, we propose a cooperative and hybrid network intrusion detection system (CH-NIDS) to detect network attacks in the Cloud environment by monitoring network traffic, while maintaining performance and service quality [7]. In our NIDS framework, we use Snort as a signature based detection to detect known attacks, while for detecting network anomaly, we use Back-Propagation Neural network (BPN). By applying snort prior to the BPN classifier, BPN has to detect only unknown attacks. So, detection time is reduced. To solve the problem of slow convergence of BPN and being easy to fall into local optimum, we propose to optimize the parameters of it by using an optimization algorithm in order to ensure high detection rate, high accuracy, low false positives and low false negatives with affordable computational cost. In addition, in this framework, the IDSs operate in cooperative way to oppose the DoS and DDoS attacks by sharing alerts stored in central log [32,47]. In this way, unknown attacks that were detected by any IDS can easily be detected by others IDSs. This also helps to reduce computational cost for detecting intrusions at others IDS, and improve detection rate in overall the Cloud environment.

Cyber-attacks are becoming more sophisticated and thereby presenting increasing challenges in accurately detecting intrusions. Failure to prevent the intrusions could degrade the credibility of security services, e.g. data confidentiality, integrity, and availability. Numerous intrusion detection methods have been proposed in the literature to tackle computer security threats, which can be broadly classified into Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS). This survey paper [8] presents a taxonomy of contemporary IDS, a comprehensive review of notable recent works, and an overview of the datasets commonly used for evaluation purposes [22,29]. It also presents evasion techniques used by attackers to avoid detection and discusses future research challenges to counter such techniques so as to make computer systems more secure.

### 3. METHODOLOGY

#### i) Proposed Work:

The Random Forest machine learning algorithm, known for its accuracy and robustness, is harnessed alongside strategic feature engineering. This combination is utilized to create a sophisticated intrusion detection system for cloud environments, aiming to substantially enhance security. The approach focuses on accurately identifying potential threats and abnormal patterns, contributing to an efficient and reliable solution that strengthens overall cloud security measures. And also included the combination of a Voting Classifier, incorporating Random Forest (RF) and ADaBoost, achieves an impressive 99% accuracy for the Kdd-Cup dataset.

Additionally, the Stacking Classifier, integrating Random Forest (RF), Multi-Layer Perceptron (MLP), and LightGBM, attains an outstanding 100% accuracy for the Bot-IoT dataset [28,29,39]. These ensemble models showcase the project's commitment to robust and high-performing intrusion detection in cloud environments. The user-friendly Flask framework with SQLite integration ensures practical usability, offering a seamless experience for user testing while maintaining data security in cybersecurity applications.

**ii) System Architecture:**

It begins with dataset exploration and data preprocessing, followed by the crucial steps of train-test split and model training. The core architecture involves the implementation of ensemble techniques, specifically the Stacking Classifier and the Voting Classifier extensions, designed to enhance the overall intrusion detection performance [24]. These classifiers demonstrate their efficacy through robust model evaluations, achieving notable accuracies of 99% and 100% respectively. The architecture prioritizes the versatility of the models, ensuring effective detection across diverse datasets, and emphasizes practical usability through a user-friendly interface facilitated by the Flask framework and SQLite integration. This unified system architecture positions the project as a sophisticated and adaptable solution for cloud-based intrusion detection using machine learning techniques.

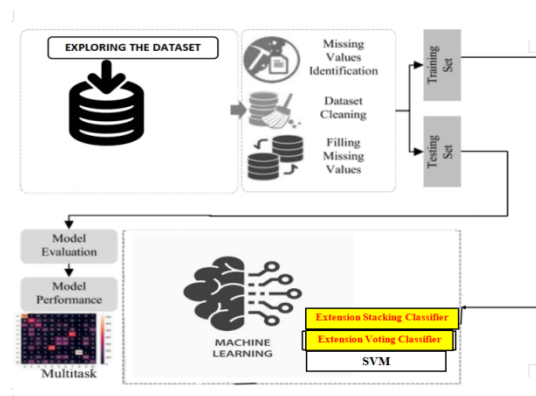


Fig 1 Proposed architecture

**iii) Dataset collection:**

**KDD CUP DATASET**

The KDD-CUP (Knowledge Discovery and Data Mining Cup) dataset [35,26] is a widely used dataset for intrusion detection system research. In the context of a cloud-based intrusion detection approach, the KDD-CUP dataset serves as a foundational dataset for training and evaluating machine learning models to detect intrusions and cyber-attacks. It allows the development of models that can analyze network traffic and detect abnormal or malicious patterns, crucial for securing cloud-based environments.

```
data = pd.read_csv("archive/kdd_train.csv")
data.head()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot
0	0	tcp	ftp_data	SF	491	0	0	0	0	0
1	0	udp	other	SF	146	0	0	0	0	0
2	0	tcp	private	S0	0	0	0	0	0	0
3	0	tcp	http	SF	232	8153	0	0	0	0
4	0	tcp	http	SF	199	420	0	0	0	0

5 rows x 42 columns

Fig 2 KDD-CUP dataset

## BOT IOT DATASET

The BOT-IoT dataset is a specialized dataset focusing on IoT (Internet of Things) security. In the context of a cloud-based intrusion detection approach utilizing machine learning, the BOT-IoT dataset [46] is highly relevant for training and evaluating models tailored to detect intrusions in IoT devices and networks. As IoT devices are often integrated with cloud platforms, understanding and mitigating IoT-based attacks are critical for overall cloud-based intrusion detection.

```
data = pd.read_csv("data_1.csv")
data.head()
```

	pkSeqID	stime	flgs	proto	saddr	sport	daddr	dport
0	1	1.526344e+09	e	arp	192.168.100.1	NaN	192.168.100.3	NaN
1	2	1.526344e+09	e	tcp	192.168.100.7	139	192.168.100.4	36390
2	3	1.526344e+09	e	udp	192.168.100.149	51838	27.124.125.250	123
3	4	1.526344e+09	e	arp	192.168.100.4	NaN	192.168.100.7	NaN
4	5	1.526344e+09	e	udp	192.168.100.27	58999	192.168.100.1	53

5 rows x 35 columns

Fig 3BOT-IOT dataset

### iv) Data Processing:

Data processing involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts

of data, including big data, into meaningful insights for quality management and decision-making.

### v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

### vi) Algorithms:

**Random Forest (RF)**, Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the classes (classification) of the individual trees. It's effective for intrusion detection due to its ability to handle a large number of features, deal with overfitting, and provide high accuracy.

```

from sklearn.ensemble import RandomForestClassifier

# instantiate the model
rf = RandomForestClassifier(random_state=40)

# fit the model
rf.fit(X_train, y_train)

#predicting the target value from the model for the samples
y_pred = rf.predict(X_test)

rf_acc = accuracy_score(y_pred, y_test)
rf_prec = precision_score(y_pred, y_test)
rf_rec = recall_score(y_pred, y_test)
rf_f1 = f1_score(y_pred, y_test)
rf_aucroc = roc_auc_score(y_test, rf.predict_proba(X_test)[: , 1])
rf_mcc = matthews_corrcoef(y_pred, y_test)

storeResults('Random Forest', rf_acc, rf_prec, rf_rec, rf_f1, rf_aucroc, rf_mcc)

```

Fig 4 Random forest

**Decision Tree (DT)** Decision Trees are a type of supervised learning model that makes decisions based on asking a series of questions related to the features in the dataset. It splits the data into subsets based on the feature values to create a tree-like structure, aiding in intrusion detection by understanding decision rules [28].

```

from sklearn.tree import DecisionTreeClassifier

# instantiate the model
tree = DecisionTreeClassifier(max_depth=30)

# fit the model
tree.fit(X_train, y_train)

#predicting the target value from the model for the samples
y_pred = tree.predict(X_test)

dt_acc = accuracy_score(y_pred, y_test)
dt_prec = precision_score(y_pred, y_test)
dt_rec = recall_score(y_pred, y_test)
dt_f1 = f1_score(y_pred, y_test)
dt_aucroc = roc_auc_score(y_test, tree.predict_proba(X_test)[: , 1])
dt_mcc = matthews_corrcoef(y_pred, y_test)

storeResults('Decision Tree Classifier', dt_acc, dt_prec, dt_rec, dt_f1, dt_aucroc, dt_mcc)

```

Fig 5 Decision tree

**Support Vector Machine (SVM)** SVM is a powerful supervised learning algorithm used for classification tasks. It creates a hyperplane or a set of hyperplanes

in a high-dimensional space to separate different classes. SVM is effective in intrusion detection for its ability to handle complex data relationships and non-linearity.

```

from sklearn.svm import SVC

# instantiate the model
svm = SVC(probability=True)

# fit the model
svm.fit(X_train, y_train)

#predicting the target value from the model for the samples
y_pred = svm.predict(X_test)

svc_acc = accuracy_score(y_pred, y_test)
svc_prec = precision_score(y_pred, y_test)
svc_rec = recall_score(y_pred, y_test)
svc_f1 = f1_score(y_pred, y_test)
svc_aucroc = roc_auc_score(y_test, svm.predict_proba(X_test)[: , 1])
svc_mcc = matthews_corrcoef(y_pred, y_test)

storeResults('Support Vector Machine', svc_acc, svc_prec, svc_rec, svc_f1, svc_aucroc, svc_mcc)

```

Fig 6 SVM

**Naive Bayes** Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that features are independent of each other, even though this assumption may not always hold. Naive Bayes is commonly used in intrusion detection due to its simplicity and speed, particularly with text-based data [28].

```

from sklearn.naive_bayes import GaussianNB

# instantiate the model
nb = GaussianNB()

# fit the model
nb.fit(X_train, y_train)

#predicting the target value from the model for the samples
y_pred = nb.predict(X_test)

nb_acc = accuracy_score(y_pred, y_test)
nb_prec = precision_score(y_pred, y_test)
nb_rec = recall_score(y_pred, y_test)
nb_f1 = f1_score(y_pred, y_test)
nb_aucroc = roc_auc_score(y_test, nb.predict_proba(X_test)[: , 1])
nb_mcc = matthews_corrcoef(y_pred, y_test)

storeResults('Naive Bayes', nb_acc, nb_prec, nb_rec, nb_f1, nb_aucroc, nb_mcc)

```

Fig 7 Naive bayes



**Deep Learning (DL)** Deep Learning involves neural networks with multiple layers (deep neural networks). DL models, like multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs), can learn complex patterns in the data, making them effective for intrusion detection where features might be intricate.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Conv1D
from tensorflow.keras.layers import MaxPooling1D

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 42)

X_train=X_train.values
X_test=X_test.values

X_train = X_train.reshape(-1, X_train.shape[1],1)
X_test = X_test.reshape(-1, X_test.shape[1],1)

Y_train=to_categorical(y_train)
Y_test=to_categorical(y_test)
```

Fig 8 Deep learning

**Long Short-Term Memory (LSTM)** LSTM is a specialized type of recurrent neural network (RNN) designed to model sequences and time-dependent data. LSTM is valuable for intrusion detection, especially in handling sequences of events or network activities, allowing the model to capture long-term dependencies effectively [33].

```
from keras.models import Sequential
from keras.layers import Dense, LSTM
from keras.layers import Dropout
from keras import regularizers
import tensorflow as tf

# define a function to build the keras model
def create_model(input_shape):
    # create model
    d = 0.25
    model = Sequential()

    model.add(LSTM(32, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(LSTM(64, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(LSTM(128, input_shape=input_shape, activation='relu', return_sequences=True))
    model.add(Dropout(d))

    model.add(LSTM(256, input_shape=input_shape, activation='relu', return_sequences=False))
    model.add(Dropout(d))

    model.add(Dense(32, kernel_initializer='uniform', activation='relu'))
    model.add(Dense(1, kernel_initializer='uniform', activation='linear'))

    # compile model
    adam = tf.keras.optimizers.Adam(learning_rate=0.001, decay=0.00001)
    model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

model = create_model(input_shape=(30,1))
print(model.summary())
```

Fig 9 LSTM

### Stacking Classifier (RF + MLP with Light GBM)

The Stacking Classifier extension combines the predictive power of Random Forest (RF) and Multi-Layer Perceptron (MLP) with Light Gradient Boosting Machine (LightGBM). RF, an ensemble of decision trees, excels at capturing complex patterns, while MLP with LightGBM introduces diverse learning techniques. The Stacking Classifier intelligently merges their outputs, leveraging the strengths of each base classifier to enhance overall intrusion detection performance, especially in cloud environments with diverse cyber threats.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from lightgbm import LGBMClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import StackingClassifier

estimators = [('rf', RandomForestClassifier(n_estimators=1000)), ('mlp', MLPClassifier(random_state=1, max_iter=3000))]
clf = StackingClassifier(estimators=estimators, final_estimator=LGBMClassifier(n_estimators=1000))

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)

vot_acc = accuracy_score(y_pred, y_test)
vot_prec = precision_score(y_pred, y_test)
vot_rec = recall_score(y_pred, y_test)
vot_f1 = f1_score(y_pred, y_test)
vot_aucroc = roc_auc_score(y_test, clf.predict_proba(X_test)[:,: , 1])
vot_mcc = matthews_corrcoef(y_pred, y_test)

storeResults('Stacking Classifier', vot_acc, vot_prec, vot_rec, vot_f1, vot_aucroc, vot_mcc)
```

Fig 10 Stacking classifier

### Voting Classifier (RF + AdaBoost)

The Voting Classifier extension integrates the capabilities of Random Forest (RF) and AdaBoost to create a robust intrusion detection model. RF excels in capturing intricate patterns through decision trees, while AdaBoost adapts by adjusting weights to prioritize the correct classification of previously misclassified instances. This combination ensures a strong ensemble model that leverages the strengths of both classifiers, achieving high accuracy and reliability in identifying potential intrusions in cloud-based systems. The versatility of this ensemble makes it well-suited for handling various types of cyber threats, contributing to the effectiveness of the overall intrusion detection approach in the project.

#### RF + AdaBoost

```

from sklearn.ensemble import RandomForestClassifier, VotingClassifier, AdaBoostClassifier
clf1 = AdaBoostClassifier(n_estimators=100, random_state=0)
clf2 = RandomForestClassifier(n_estimators=50, random_state=1)

eclf1 = VotingClassifier(estimators=[('r1', clf1), ('r2', clf2)], voting='soft')
eclf1.fit(X_train, y_train)
y_pred = eclf1.predict(X_test)

vot_acc = accuracy_score(y_pred, y_test)
vot_prec = precision_score(y_pred, y_test)
vot_rec = recall_score(y_pred, y_test)
vot_f1 = f1_score(y_pred, y_test)
vot_auroc = roc_auc_score(y_test, eclf1.predict_proba(X_test)[:, 1])
vot_mcc = matthews_corrcoef(y_pred, y_test)

storeResults('RF + AdaBoost', vot_acc, vot_prec, vot_rec, vot_f1, vot_auroc, vot_mcc)
    
```

Fig 11 Voting classifier

## 4. EXPERIMENTAL RESULTS

**Precision:** Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives / (True positives + False positives) = TP / (TP + FP)

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

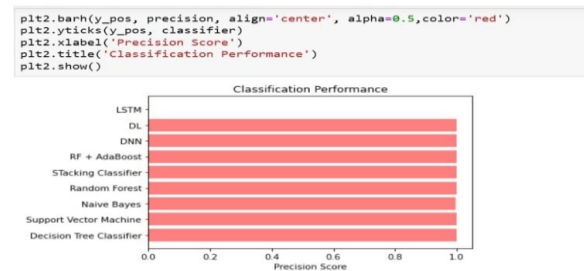


Fig 6 Precision comparison graph

**Recall:** Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

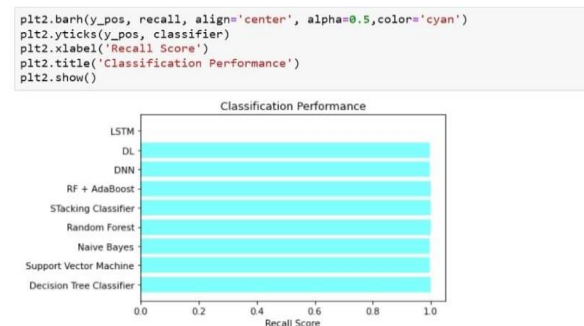


Fig 7 Recall comparison graph

**Accuracy:** Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

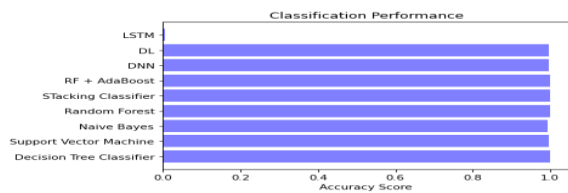


Fig 8 Accuracy graph

**F1 Score:** The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$F1\ Score = 2 * \frac{Recall \times Precision}{Recall + Precision} * 100$$

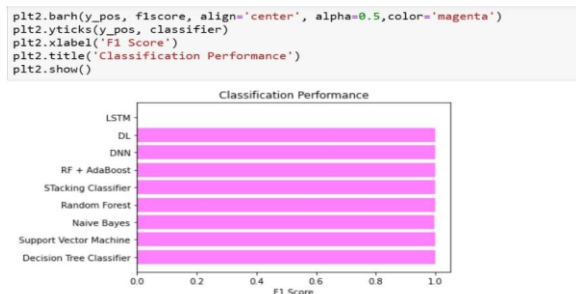


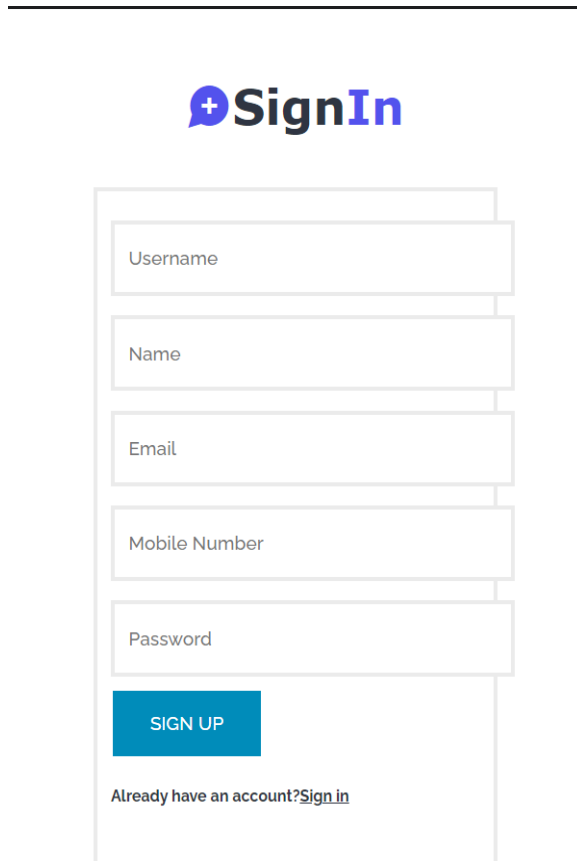
Fig 9 F1Score

ML Model	Accuracy	Precision	Recall	F1-Score
Decision Tree Classifier	1.000	1.000	1.000	1.000
Support Vector Machine	0.996	1.000	0.996	0.998
Naive Bayes	0.993	0.997	0.996	0.997
Random Forest	1.000	1.000	1.000	1.000
Extension Stacking Classifier	1.000	1.000	1.000	1.000
Extension RF + AdaBoost	1.000	1.000	1.000	1.000
DNN	0.996	1.000	0.996	0.998
DL	0.995	1.000	0.995	0.998
LSTM	0.005	0.000	0.000	0.000

Fig 10 Performance Evaluation



Fig 11 Home page



**+SignIn**

Username

Name

Email

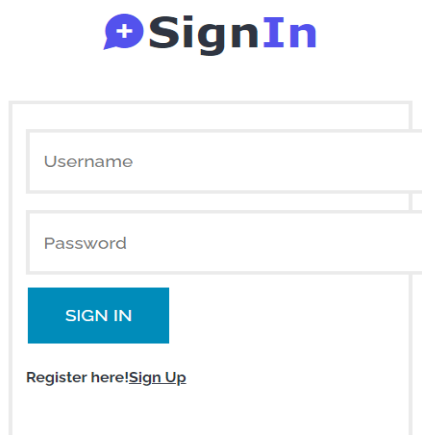
Mobile Number

Password

**SIGN UP**

Already have an account?[Sign in](#)

Fig 12 Signin page



**+SignIn**

Username

Password

**SIGN IN**

Register here![Sign Up](#)

Fig 13 Login page

dst\_host\_same\_src\_port\_rate

dst\_host\_srv\_diff\_host\_rate

dst\_host\_serror\_rate

dst\_host\_srv\_serror\_rate

dst\_host\_rerror\_rate

**PREDICT**

Fig 14 User input

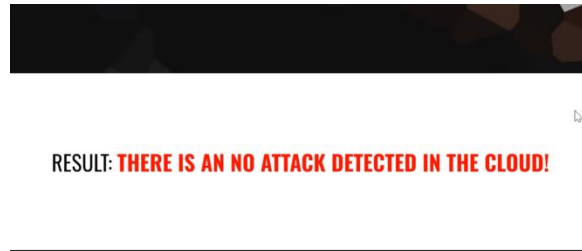


Fig 15 Predict result for given input

## 5. CONCLUSION

The intrusion detection model built for the cloud, employing Random Forest (RF) and feature engineering, excels in accuracy, precision, and recall. It demonstrates superior performance in detecting abnormal activities within the cloud environment compared to recent works. This highlights the effectiveness and reliability of the proposed approach. Random Forest (RF) [26,29] is a pivotal component of the model, contributing to its success. RF is effective in handling outlier data, providing robustness in abnormal activity detection. Its simplicity in parameter establishment and automatic creation of variable importance and accuracy metrics make it an efficient choice, enhancing the overall performance of the intrusion detection model. The project extends accuracy through ensemble techniques like Voting Classifier and Stacking Classifier. Integration of a user-friendly Flask interface with secure authentication improves the testing experience, emphasizing practical usability in cybersecurity applications.

## 6. FUTURE SCOPE

Future work aims to enhance the recall rate, especially using the NSL-KDD dataset, by

integrating deep learning (DL) and ensemble learning techniques [27]. Deep learning models can capture complex patterns, potentially improving the system's ability to detect intrusions. Ensemble techniques, on the other hand, combine multiple models to boost prediction accuracy, further enhancing the overall performance of the intrusion detection system. Future systems will focus on understanding user and system behavior through behavioral analysis. This approach is crucial for accurate anomaly detection, enabling the identification of abnormal patterns and potential security threats. Analyzing behaviors helps in creating a baseline for normal activities, making it easier to detect deviations that could signify security breaches. The research will strive to develop intrusion detection systems capable of efficiently scaling with the growing complexity and volume of cloud data. Optimizing resources for efficient performance and cost-effectiveness will be a priority, ensuring the system can handle the increased data load and adapt to evolving cloud infrastructures while maintaining cost-efficiency. Ensemble learning techniques will be leveraged to combine multiple models, harnessing their collective strength to make more accurate predictions. By integrating ensemble learning, the intrusion detection system can enhance its overall performance, achieving higher accuracy and reliability in identifying potential security threats in the cloud.

## REFERENCES

- [1] M. Ali, S. U. Khan, and A. V. Vasilakos, Security in cloud computing: Opportunities and challenges, *Information Sciences*, vol. 35, pp. 357–383, 2015.

- [2] A. Singh and K. Chatterjee, Cloud security issues and challenges: A survey, *Journal of Network and Computer Applications*, vol. 79, pp. 88–115, 2017.
- [3] P. S. Gowr and N. Kumar, Cloud computing security: A survey, *International Journal of Engineering and Technology*, vol. 7, no. 2, pp. 355–357, 2018.
- [4] A. Verma and S. Kaushal, Cloud computing security issues and challenges: A survey, in *Proc. First International Conference on Advances in Computing and Communications*, Kochi, India, 2011, pp. 445–454.
- [5] H. Alloussi, F. Laila, and A. Sekkaki, L'état de l'art de la sécurité dans le cloud computing: Problèmes et solutions de la sécurité en cloud computing, presented at *Workshop on Innovation and New Trends in Information Systems*, Mohamadia, Maroc, 2012.
- [6] J. Gu, L. Wang, H. Wang, and S. Wang, A novel approach to intrusion detection using SVM ensemble with feature augmentation, *Computers and Security*, vol. 86, pp. 53–62, 2019.
- [7] Z. Chiba, N. Abghour, K. Moussaid, A. E. Omri, and M. Rida, A cooperative and hybrid network intrusion detection framework in cloud computing based snort and optimized back propagation neural network, *Procedia Computer Science*, vol. 83, pp. 1200–1206, 2016.
- [8] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, Survey of intrusion detection systems: Techniques, datasets and challenges, *Cybersecurity*, vol. 2, p. 20, 2019.
- [9] A. Guezzaz, A. Asimi, Y. Asimi, Z. Tbatou, and Y. Sadqi, A global intrusion detection system using PcapSockS sniffer and multilayer perceptron classifier, *International Journal of Network Security*, vol. 21, no. 3, pp. 438–450, 2019.
- [10] A. Guezzaz, S. Benkirane, M. Azrour, and S. Khurram, A reliable network intrusion detection approach using decision tree with enhanced data quality, *Security and Communication Networks*, vol. 2021, p. 1230593, 2021.
- [11] B. A. Tama and K. H. Rhee, HFSTE: Hybrid feature selections and tree-based classifiers ensemble for intrusion detection system, *IEICE Trans. Inf. Syst.*, vol. E100.D, no. 8, pp. 1729–1737, 2017.
- [12] M. Azrour, J. Mabrouki, G. Fattah, A. Guezzaz, and F. Aziz, Machine learning algorithms for efficient water quality prediction, *Modeling Earth Systems and Environment*, vol. 8, pp. 2793–2801, 2022.
- [13] M. Azrour, Y. Farhaoui, M. Ouanan, and A. Guezzaz, SPIT detection in telephony over IP using K-means algorithm, *Procedia Computer Science*, vol. 148, pp. 542–551, 2019.
- [14] M. Azrour, M. Ouanan, Y. Farhaoui, and A. Guezzaz, Security analysis of Ye et al. authentication protocol for internet of things, in *Proc. International Conference on Big Data and Smart Digital Environment*, Casablanca, Morocco, 2018, pp. 67–74.

- [15] M. Azrou, J. Mabrouki, A. Guezzaz, and A. Kanwal, Internet of things security: Challenges and key issues, *Security and Communication Networks*, vol. 2021, p. 5533843, 2021.
- [16] A. Guezzaz, S. Benkirane, and M. Azrou, A novel anomaly network intrusion detection system for internet of things security, in *IoT and Smart Devices for Sustainable Environment*, M. Azrou, A. Irshad, and R. Chaganti, eds. Cham, Switzerland: Springer, 2022, pp. 129–138.
- [17] A. Guezzaz, A. Asimi, M. Azrou, Z. Tbatou, and Y. Asimi, A multilayer perceptron classifier for monitoring network traffic, in *Proc. 3rd International Conference on Big Data and Networks Technologies*, Leuven, Belgium, 2019, pp. 262–270.
- [18] S. Benkirane, Road safety against sybil attacks based on RSU collaboration in VANET environment, in *Proc. 5th International Conference on Mobile, Secure, and Programmable Networking*, Mohammedia, Morocco, 2019, pp. 163–172.
- [19] Q. Zhang, L. Cheng, and R. Boutaba, Cloud computing: State-of-the-art and research challenges, *J. Internet Serv. Appl.*, vol. 1, pp. 7–18, 2010.
- [20] M. K. Srinivasan, K. Sarukesi, P. Rodrigues, M. S. Manoj, and P. Revathy, State-of-the-art cloud computing security taxonomies: A classification of security challenges in the present cloud computing environment, in *Proc. 2012 International Conference on Advances in Computing, Communications and Informatics*, Chennai, India, 2012, pp. 470–476.
- [21] A. L. Buczak and E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
- [22] A. Alshammari and A. Aldribi, Apply machine learning techniques to detect malicious network traffic in cloud computing, *Journal of Big Data*, vol. 8, p. 90, 2021.
- [23] A. Geron, *Hands-On Machine Learning with Scikit-Learn & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2017.
- [24] N. Chand, P. Mishra, C. R. Krishna, E. S. Pilli, and M. C. Govil, A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection, in *Proc. 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)*, Dehradun, India, 2016, pp. 1–6.
- [25] A. B. Nassif, M. A. Talib, Q. Nasir, H. Albadani, and F. M. Dakalbab, Machine learning for cloud security: A systematic review, *IEEE Access*, vol. 9, pp. 20717–20735, 2021.
- [26] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, A survey of deep learning-based network anomaly detection, *Cluster Comput.*, vol. 22, pp. 949–961, 2017.
- [27] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, Deep learning for cyber security intrusion detection: Approaches, datasets, and

comparative study, *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[28] V. Kanimozhi and T. P. Jacob, Calibration of various optimized machine learning classifiers in network intrusion detection system on the realistic cyber dataset CSE-CICIDS2018 using cloud computing, *International Journal of Engineering Applied Sciences and Technology*, vol. 4, no. 6, pp. 209–213, 2019.

[29] L. Zhou, X. Ouyang, H. Ying, L. Han, Y. Cheng, and T. Zhang, Cyber-attack classification in smart grid via deep neural network, in *Proc. 2nd International Conference on Computer Science and Application Engineering*, Hohhot, China, 2018, pp. 1–5.

[30] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, Deep learning approach for network intrusion detection in software defined networking, in *Proc. 2016 International Conference on Wireless Networks and Mobile Communications*, Fez, Morocco, 2016, pp. 258–263.

[31] L. Zhang, L. Shi, N. Kaja, and D. Ma, A two-stage deep learning approach for can intrusion detection, in *Proc. 2018 Ground Vehicle Syst. Eng. Technol. Symp. (GVSETS)*, Novi, MI, USA, 2018, pp. 1–11.

[32] A. Mishra, B. B. Gupta, D. Perakovic, F. J. G. Penalvo, and C. H. Hsu, Classification based machine learning for detection of DDoS attack in cloud computing, in *Proc. 2021 IEEE International*

*Conference on Consumer Electronics*, Las Vegas, NV, USA, 2021, pp. 1–4.

[33] F. Jiang, Y. Fu, B. B. Gupta, Y. Liang, S. Rho, F. Lou, F. Meng, and Z. Tian, Deep learning based multichannel intelligent attack detection for data security, *IEEE Transactions on Sustainable Computing*, vol. 5, no. 2, pp. 204–212, 2018.

[34] A. N. Khan, M. Y. Fan, A. Malik, and R. A. Memon, Learning from privacy preserved encrypted data on cloud through supervised and unsupervised machine learning, in *Proc. 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies*, Sukkur, Pakistan, 2019, pp. 1–5.

[35] S. Potluri and C. Diedrich, Accelerated deep neural networks for enhanced intrusion detection system, in *Proc. 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation*, Berlin, Germany, 2016, pp. 1–8.

[36] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, Long short term memory recurrent neural network classifier for intrusion detection, in *Proc. 2016 International Conference on Platform Technology and Service*, Jeju, Republic of Korea, 2016, pp. 1–5.

[37] J. Zhang, Anomaly detecting and ranking of the cloud computing platform by multi-view learning, *Multimedia Tools and Applications*, vol. 78, pp. 30923–30942, 2019.

[38] F. B. Ahmad, A. Nawaz, T. Ali, A. A. Kiani, and G. Mustafa, Securing cloud data: A machine learning based data categorization approach for cloud



computing, <http://doi.org/10.21203/rs.3.rs-1315357/v1>, 2022.

[39] A. Mubarakali, K. Srinivasan, R. Mukhalid, S. C. Jaganathan, and N. Marina, Security challenges in Internet of things: Distributed denial of service attack detection using support vector machine-based expert systems, *Computational Intelligence*, vol. 36, no. 4, pp. 1580–1592, 2020.

[40] N. M. Abdulkareem and A. M. Abdulazeed, Machine learning classification based on random forest algorithm: A review, *International Journal of Science and Business*, vol. 5, no. 2, pp. 128–142, 2021.

[41] L. Breiman, Random forests, *Machine Learning*, vol. 45, pp. 5–32, 2001.

[42] I. Reis, D. Baron, and S. Shahaf, Probabilistic random forest: A machine learning algorithm for noisy data sets, *The Astronomical Journal*, vol. 157, no. 1, p. 16, 2018.

[43] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, Random forests and decision trees, *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 5, pp. 272–278, 2012.

[44] B. O. Yigin, O. Algin, and G. Saygili, Comparison of morphometric parameters in prediction of hydrocephalus using random forests, *Computers in Biology and Medicine*, vol. 116, p. 103547, 2020.

[45] A. Sarica, A. Cerasa, and A. Quattrone, Random forest algorithm for the classification of

neuroimaging data in alzheimer's disease: A systematic review, *Frontiers in Aging Neuroscience*, vol. 9, p. 329, 2017.

[46] A. Devarakonda, N. Sharma, P. Saha, and S. Ramya, Network intrusion detection: A comparative study of four classifiers using the NSL-KDD and KDD'99 datasets, *Journal of Physics: Conference Series*, vol. 2161, p. 012043, 2022.

[47] M. Zeeshan, Q. Riaz, M. A. Bilal, M. K. Shahzad, H. Jabeen, S. A. Haider, and A. Rahim, Protocol-based deep intrusion detection for DoS and DDoS attacks using UNSWNB15 and Bot-IoT datasets, *IEEE Access*, vol. 10, pp. 2269–2283, 2021.

[48] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, CorrAUC: A malicious Bot-IoT traffic detection method in IoT network using machine-learning techniques, *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3242–3254, 2021.

[49] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, Selection of effective machine learning algorithm and BotIoT attacks traffic identification for Internet of things in smart city, *Future Generation Computer Systems*, vol. 107, pp. 433–442, 2020.

[50] M. Hossin and M. N. Sulaiman, A review on evaluation metrics for data classification evaluations, *International Journal of Data Mining & Knowledge Management Process*, doi: 10.5121/ijdkp.2015.5201.