International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 13, Issue 2 (Mar.-Apr. 2024), PP. 138-149

A cloud based method for finding intrusions using machine learning

[1.] Chelamalasetti Tejasri Venkata Bhavani, *B.Tech, Department of CSE, DNR COLLEGE OF ENGINEERING AND TECHNOLOGY,tejasri.chelamalasetti@gmail.com*
[2.] Nangana Syam Babu, *B.Tech, Department of CSE, DNR COLLEGE OF ENGINEERING AND TECHNOLOGY,syamnangana2001@gmail.com*
[3.] Matta Geetanjali, *B.Tech, Department of CSE, DNR COLLEGE OF ENGINEERING AND TECHNOLOGY,mgeetanjali2002@gmail.com*
[4.] PotturiSai Madhav Varma, *B.Tech, Department of CSE, DNR COLLEGE OF ENGINEERING AND TECHNOLOGY,Saimadhavvarma1234@gmail.com*
[5.] Dr. A Rama Murthy, *M.Tech, PhD.,Professor, Department of CSE, DNR COLLEGE OF ENGINEERING AND TECHNOLOGY,ram111.sai@gmail.com*

Abstract: *Storage, data management services, processing power, apps, and a plethora of other network and computer resources are all at your fingertips with cloud computing. These materials are readily available for users to use whenever they need them. Improving cloud security via the deployment of a machine learning-based intrusion detection model is the main objective of the project. To successfully identify and avoid cyber-attacks, the main objective is to track and examine cloud-based resources, services, and networks. With a focus on the Random Forest (RF) method, the suggested intrusion detection model makes use of machine learning approaches. Combining several decision trees into one strong ensemble learning approach, Random Forest improves prediction accuracy. An important part of creating a model is feature engineering. It entails picking out the best characteristics from the dataset and making them the input for the machine learning model. The capacity of the model to recognize patterns and reliably detect possible threats is enhanced by well-engineered features. Continuous monitoring of cloud resources, services, and networks is the goal of implementing the concept to improve cloud security. The program improves the cloud infrastructure's security by spotting suspicious trends linked to cyberattacks using machine learning methods. Two datasets, Bot-IoT and NSL-KDD, are used to assess and verify the model's performance. When it comes to intrusion detection, these datasets are often used as standards. When compared to other recent efforts in the same field, the model's high intrusion detection accuracy shows that it is useful and reliable in spotting possible security risks.The project's Voting Classifier, which combines RF and ADaBoost, and the Stacking Classifier, which combines RF and MLP with LightGBM, achieved 99% and 100% accuracy for Kdd-Cup and Bot-IoT data, respectively, in terms of improved cloud detection results.*

Anomaly detection, features engineering, random forest, and cloud security are index words.

I. INTRODUCTION

With cloud computing, users have greater flexibility in choosing their service model and practical, on-demand access to pooled storage, networks, and resources [1]. These models are used in one of the deployment methods for private, public, or hybrid clouds, and they are platform as a service (PaaS), software as a service (SaaS), or infrastructure as a service (IaaS) [2]. According to the National Institute of Standards and Technology, the following features of the cloud enable it to deliver high-performance services: network access, resource pooling, quickelasticity, and measurable service.

Many security issues, including those pertaining to availability, data confidentiality, integrity, and control authorization, have recently plagued the cloud. Another big source of vulnerabilities that might infect cloud systems and resources is the Internet, which is utilized to access cloud services [2]. Providers of cloud services therefore have the formidable task of bolstering cloud security[5]. In order to make cloud systems more secure and resistant to different types of assaults, several methods have been created, including firewall technologies, data encryption techniques, authentication protocols, and others [6]. When it comes to protecting cloud services from various constraints, however, conventional methods fall short[7]. Thus, in order to identify and stop malicious actions in real-time, a suite of intrusion detection methods is suggested and implemented[8, 9].

In general, there are two types of detection methods: those that employ known attacks to identify incursion and those that utilize unknown assaults to identify anomalies. By merging their best features, these

two approaches form the hybrid technique [10]. Recent intrusion detection systems (IDSs) have a number of major drawbacks[8] that make them less effective than their predecessors, even though there are more solutions available for safe cloud environments. These include issues with data quality, real-time detection, and massive amounts of analyzed data.

Machine learning (ML), deep learning (DL), and ensemble learning are current intelligent learning approaches that academics have shown to be effective in a number of domains[12,13] and capable of implementing network security[14–18]. Our primary objective in this study is to provide a method for anomaly detection that makes use of the random forest (RF) binary classifier. To do this, we conduct feature engineering using a data visualization process with the goal of reducing the amount of features that are included in the model. The NSL-KDD and BoT-IoT datasets are used to evaluate the model's performance. Afterwards, the results show how well the model worked.

II. LITERATURE SURVEY

The incredible promise of cloud computing lies in its ability to make available, via the Internet, powerful, elastic, easily managed, and inexpensive resources whenever needed. Cloud computing maximizes the potential of hardware resources via their shared and efficient use. Organizations and individual users are being encouraged to move their apps and services to the cloud by the benefits listed above [1]. People are moving even the most important parts of society to the cloud, including power plants and other crucial infrastructure. Nevertheless, there are extra security risks associated with services offered by third-party cloud providers. In a shared environment with many users colocated, security issues are heightened when user assets (data, apps, etc.) are moved outside of administrative control. The inherent vulnerabilities of cloud computing are detailed in this overview. Additionally, the study summarizes the most up-to-date approaches to security problems as discussed in the literature. In addition, we provide a high-level overview of the security holes in mobile cloud computing [18,30]. Discussion of outstanding questions and potential avenues for further study is included at the conclusion as well.

By using a vast quantity of virtual storage, cloud computing enables the provision of on-demand services over the Internet. The primary benefit of cloud computing is the reduced service cost and the elimination of the need for the customer to set up costly computer equipment. Researchers have been encouraged to explore new related technologies by the increasing integration of cloud computing with many industries and other fields in recent years [2]. Organizations and individuals alike move their data, applications, and services to the cloud because of the ease and scalability of its services and processing needs. Despite its benefits, moving computers from on-premises to off-premises has introduced several security risks and difficulties for both customers and service providers. Trusted third parties provide many cloud services, which opens the door to new security risks. There are new security concerns that have arisen since the cloud provider offers its services over the Internet and makes use of several web technologies [1,23,5,7,19]. The article covered the fundamentals of cloud computing, including its characteristics, security concerns, risks, and potential remedies. Cloud security principles, risks, and attacks are also covered in the paper, along with cloud technologies, a model for services and deployment, and a framework for cloud architecture. Additionally, the study delves into several unanswered questions about cloud security research.

The issue of network security has become crucial. A large number of organizations have begun using intrusion detection systems to keep their networks safe. Ensemble learning is one of many machine learning approaches that have been used to enhance intrusion detection system performance, and it is often regarded as a successful method [6]. There is little doubt that high-quality training data is a key component in improving detection performance. With the knowledge that the best univariate classifiers are marginal density ratios. Here, we provide a powerful intrusion detection system that uses a support vector machine (SVM) ensemble enhanced with features. In particular, the initial features were modified using the logarithm of marginal density ratios in order to acquire fresh, higher-quality training data. The intrusion detection model was then built using a support vector machine ensemble. When compared to other approaches in the market, our suggested approach outperforms them in terms of accuracy, detection rate, false alarm rate, and training time, according to the experimental data. (6, 24)

The concept of "the cloud" refers to a system that allows users to access robust services and applications over the Internet. It is critical to provide trustworthy services in a cloud computing setting. Because it is susceptible to network intrusions that impact the availability, confidentiality, and integrity of Cloud resources and offered services, providing security requires more than just user authentication with passwords or digital certificates and confidentiality in data transmission [1,23,5,7,19]. Using a conventional firewall alone is insufficient to identify denial-of-service attacks and other network-level malicious activity in the cloud. We provide a cooperative and hybrid network intrusion detection system (CH-NIDS) to monitor network traffic in the cloud and identify network intrusions, all while keeping service quality and performance high [7]. Our NIDS architecture employs Snort, a signature-based detection system, to identify known threats, and a Back-

Propagation Neural Network (BPN), a network anomaly detector. To ensure that BPN only detects unknown threats, snort is applied before the BPN classifier. The result is a shorter detection time. We suggest optimizing BPN's parameters using an optimization algorithm to fix its slow convergence and avoid falling into local optimums. This will guarantee a high detection rate, accuracy, and low false positive and negative rates while keeping the computational cost low. Additionally, the IDSs in this architecture work together to counter DoS and DDoS assaults by exchanging alarms recorded in a central log [32,47]. This makes it easy for other IDSs to pick up on unknown assaults that one IDS picked up on. Overall, this improves the Cloud environment's detection rate and lowers the computational cost of detecting intrusions at other IDS.

Accurately identifying intrusions is growing more challenging as cyber-attacks become more sophisticated. If the attacks are not prevented, the credibility of security services, including data availability, integrity, and confidentiality, might be compromised. Both Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS) are two of the many intrusion detection systems that have been suggested in the literature to address computer security risks. An introduction of the datasets often utilized for assessment reasons, a taxonomy of modern IDS, and a thorough analysis of important recent publications are all provided in this survey study [8]. In order to make computer systems more safe, it also analyzes future research challenges to fight evasion strategies employed by attackers and outlines these solutions.

III. METHODOLOGY

i) Proposed Work:

Strategic feature engineering and the well-known Random Forest technique for machine learning are both used. In order to greatly improve security in cloud settings, this combination is used to build an advanced intrusion detection system. An effective and trustworthy solution that fortifies cloud security measures as a whole is the goal of the method, which centers on precisely detecting any dangers and unusual patterns. The Voting Classifier that incorporates ADaBoost and Random Forest (RF) achieves a remarkable 99% accuracy on the Kdd-Cup dataset. In addition, the Bot-IoT dataset achieves an astounding 100% accuracy with the help of the Stacking Classifier, which combines Random Forest (RF), Multi-Layer Perceptron (MLP), and LightGBM [28,29,39]. The project's dedication to effective and reliable intrusion detection in cloud systems is shown by these ensemble models. With the SQLite connection and the user-friendly Flask framework, cybersecurity apps may test their users' experiences with ease and keep their data secure.

ii) The system architecture starts with exploring the dataset and preparing the data. Then comes the train-test split and training the model, which are critical processes. In order to improve the overall performance of intrusion detection, the basic architecture incorporates ensemble approaches, including the Stacking Classifier and the Voting Classifier extensions [24]. Through rigorous model assessments, these classifiers prove their effectiveness, attaining remarkable accuracies of 99% and 100%, respectively. The design places an emphasis on practical usage via an easy-to-use interface made possible by the Flask framework and SQLite integration, and it prioritizes the models' adaptability to ensure successful detection across varied datasets. The project is positioned as an advanced and flexible solution for cloud-based intrusion detection employing machine learning methods, thanks to its unified system architecture.

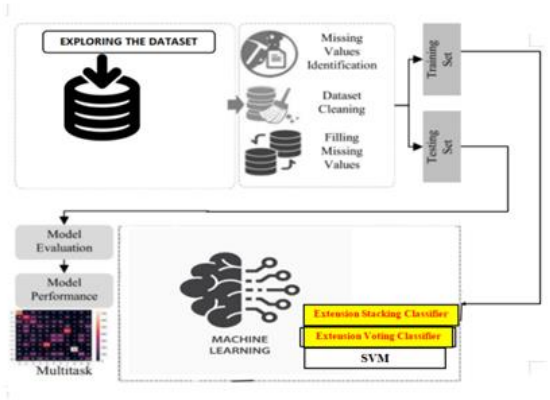

Fig 1 Proposed architecture

iii) Dataset collection:

KDD CUP DATASET

Research on intrusion detection systems makes extensive use of the Knowledge Discovery and Data Mining Cup (KDD-CUP) dataset [35,26]. To train and evaluate machine learning models to identify intrusions and cyber-attacks, the KDD-CUP dataset is used as a foundational dataset in a cloud-based intrusion detection strategy. It paves the way for the creation of algorithms that can safeguard cloud-based systems by analyzing network traffic and identifying suspicious or harmful trends.

```
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 × 42 columns

Fig 2 KDD-CUP dataset

BOT IOT DATASET

Internet of Things (IoT) security is the primary emphasis of the BOT-IoT dataset. For training and assessing models designed to identify intrusions in IoT devices and networks, the BOT-IoT dataset [46] is very useful within the context of a cloud-based intrusion detection method that uses machine learning. For cloud-based intrusion detection as a whole, it is crucial to comprehend and counteract IoT-based threats because of how often cloud platforms connect with IoT devices.

```
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 × 35 columns

Fig 3BOT-IOT dataset

iv) Data Processing:

Processing data entails making sense of raw data for companies. Collecting, organizing, cleaning, validating, analyzing, and transforming data into understandable representations like graphs or papers are all part of data processing. There are three main ways that data may be processed: mechanically, electronically, or by hand. Improving the usefulness of data and making decisions easier are the goals. Companies may then use this information to make better strategic choices and enhance their operations. Software development and other forms of automated data processing are crucial here. Quality management and decision-making may benefit from its ability to transform massive data sets, particularly big data, into actionable insights.
v) Feature selection refers to the process of identifying which characteristics are most relevant, consistent, and free of duplication before building a model. With the proliferation of datasets comes the need to systematically reduce their sizes. The primary objective of feature selection is to decrease the computational cost of modeling while simultaneously improving the performance of a predictive model.

An essential part of feature engineering is feature selection, which entails picking out the most relevant characteristics to feed into ML algorithms. By removing superfluous or unimportant characteristics and keeping just the most important ones, feature selection strategies help to decrease the amount of input variables used by machine learning models. Rather than relying on the machine learning model to prioritize features, it is recommended to undertake feature selection beforehand.

the sixth section, algorithms:

In order to train a model, Random Forest builds an ensemble of decision trees and then uses the mean of their classifications to produce an output class. Its excellent accuracy, capacity to manage overfitting, and handling of a huge number of features make it a useful intrusion detection tool.

```
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_auroc = 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_auroc,rf_mcc)
```

Fig 4 Random forest

Decision Tree (DT) As a supervised learning model, Decision Trees evaluate dataset characteristics by asking a set of predefined questions. In order to help with intrusion detection, it uses decision rules to divide the data into subsets according to the feature values and creates a tree-like structure [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_auroc = 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_auroc,dt_mcc)
```

Fig 5 Decision tree

Support Vector Machine (SVM)

```
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_auroc = 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_auroc,svc_mcc)
```

Fig 6 SVM

Naive Bayes The Naive Bayes method uses Bayes' theorem as its foundation for probabilistic categorization. It takes for granted that characteristics are unrelated to one another, which isn't always the case. The simplicity and speed of Naive Bayes make it a popular choice for intrusion detection, especially when dealing 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_auroc = 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_auroc,nb_mcc)
```

Fig 7 Naïve bayes

Deep Learning (DL) Neural networks with many layers are used in deep learning. For intrusion detection tasks involving potentially complicated characteristics, DL models such as multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) are invaluable.

```
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_2, 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 an RNN subset that excels at modeling sequences and data that changes over time. When dealing with events or network activity in sequences, LSTM is useful for intrusion detection because it enables the model to efficiently capture long-term relationships [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='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    #model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

model = create_model(input_shape=(19,1))
print(model.summary())
```

Fig 9 LSTM

Stacking Classifier (RF + MLP with Light GBM)
Stacking Classifier is an add-on that merges Light Gradient Boosting Machine (LightGBM) with Random Forest (RF), Multi-Layer Perceptron (MLP), and other powerful prediction methods. While MLP with LightGBM brings varied learning strategies, RF, an ensemble of decision trees, is great at capturing complicated patterns. Stacking Classifiers are great for improving intrusion detection effectiveness in cloud systems with varied cyber threats because they intelligently combine their outputs, exploiting the characteristics of each base classifier.

```
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_auroc = 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_auroc,vot_mcc)
```

Fig 10 Stacking classifier

Voting Classifier (RF + AdaBoost)

The Voting Classifier add-on builds a strong intrusion detection model by combining the features of AdaBoost and Random Forest (RF). While AdaBoost learns from mistakes and adjusts weights to emphasize accurate classification, RF is great at capturing complex patterns using decision trees. By combining their respective capabilities, the two classifiers form a powerful ensemble model that can reliably and accurately detect possible intrusions in cloud-based systems. As a whole, the project's intrusion detection strategy benefits from this ensemble's adaptability, which makes it ideal for dealing with different kinds of cyber threats.

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=[('lr', clf1), ('rf', 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

IV. EXPERIMENTAL RESULTS

Precision: The accuracy rate, or precision, is the percentage of true positives relative to the total number of occurrences or samples. Consequently, the following is the formula for determining the accuracy: Preciseness is TP divided by (TP plus FP), which is the sum of true positives and false positives.

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

```
plt2.barh(y_pos, precision, align='center', alpha=0.5,color='red')
plt2.yticks(y_pos, classifier)
plt2.xlabel('Precision Score')
plt2.title('Classification Performance')
plt2.show()
```

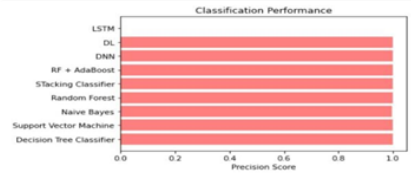

Fig 6 Precision comparison graph

Recall:

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

```
plt2.barh(y_pos, recall, align='center', alpha=0.5,color='cyan')
plt2.yticks(y_pos, classifier)
plt2.xlabel('Recall Score')
plt2.title('Classification Performance')
plt2.show()
```

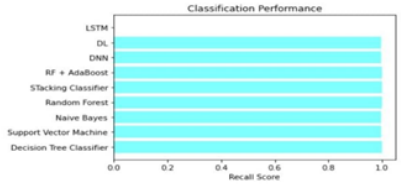

Fig 7 Recall comparison graph

Accuracy:

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

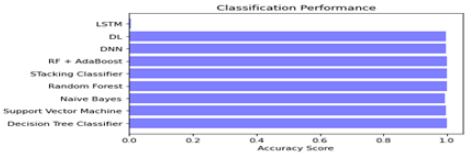

Fig 8 Accuracy graph

F1 Score:

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

```
plt2.barh(y_pos, f1score, align='center', alpha=0.5,color='magenta')
plt2.yticks(y_pos, classifier)
plt2.xlabel('F1 Score')
plt2.title('Classification Performance')
plt2.show()
```

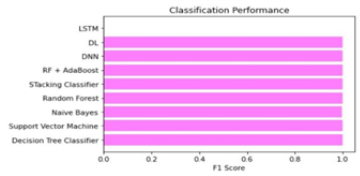

Fig 9 F1Score

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

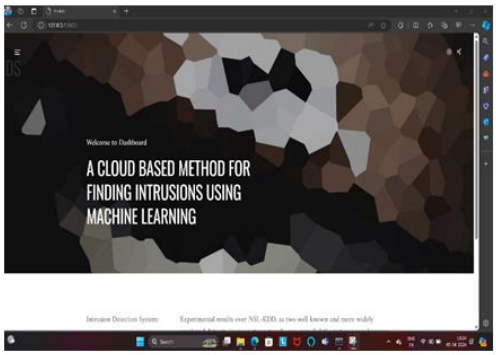

Fig 11 Home page

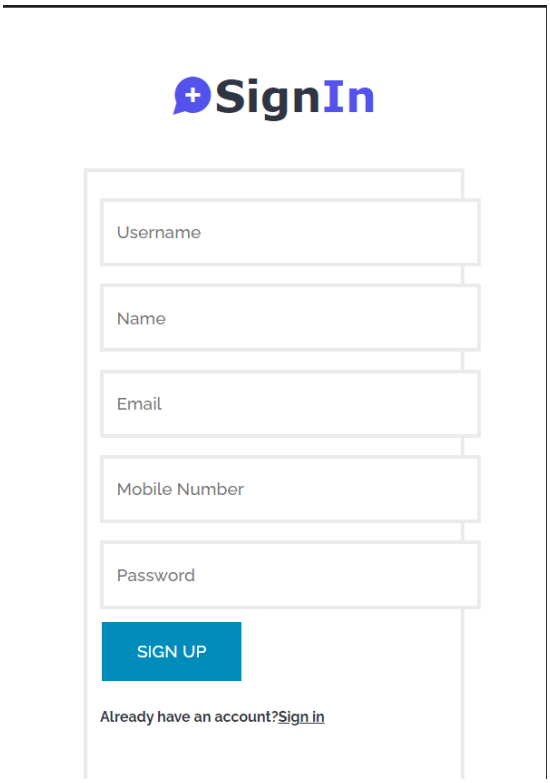

Fig 12 Signin page

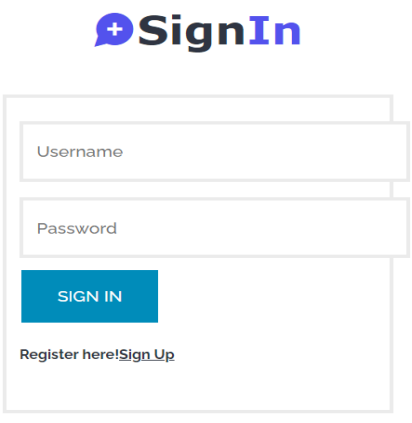

Fig 13 Login page

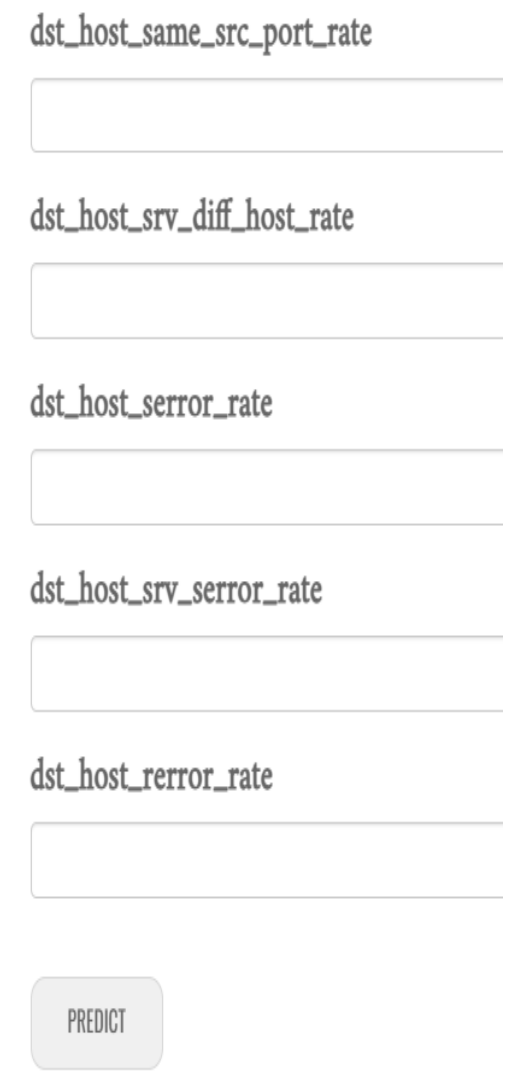

Fig 14User input

Fig 15 Predict result for given input

V. CONCLUSION

Using Random Forest (RF) and feature engineering, the cloud-based intrusion detection model achieves outstanding recall, accuracy, and precision. Compared to previous studies, it finds more instances of suspicious behavior in the cloud. This proves that the suggested method is both efficient and trustworthy. A key component that contributes to the model's effectiveness is Random Forest (RF) [26,29]. RF offers resilience in aberrant activity identification and is useful in managing outlier data. Improving the intrusion detection model's overall performance, it is an efficient option due to its simple parameter construction and automated production of variable significance and accuracy metrics. Voting Classifier and Stacking Classifier are two examples of the ensemble methods used to improve accuracy in this research. Highlighting practical usability in cybersecurity applications, the testing experience is improved by integrating a user-friendly Flask interface with secure

authentication.

8. IN THE LONG TERM

By combining deep learning (DL) with ensemble learning methods, future research hopes to improve the recall rate, particularly on the NSL-KDD dataset [27]. An improvement in the system's intrusion detection capabilities may be possible with the help of deep learning models, which can grasp intricate patterns. To improve the intrusion detection system's overall performance, ensemble approaches integrate many models to raise prediction accuracy. The goal of behavioral analysis in future systems will be to comprehend how users and the system interact with one another. Accurate anomaly detection relies on this method, which helps find suspicious patterns and possible security risks. By establishing a standard for typical actions, behavioral analysis makes it simpler to spot changes that could indicate security lapses. With the increasing complexity and amount of cloud data, the study will aim to build intrusion detection systems that can effectively scale. Making sure the system can manage the growing data load and adapt to changing cloud infrastructures while keeping costs down will be our top concern, so we can optimize resources for optimal performance and cost-effectiveness. By combining many models using ensemble learning approaches, we may increase the predictive power of each individual model. The intrusion detection system may improve its overall performance by using ensemble learning. This ensures that it identifies cloud security risks more accurately and reliably.

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 ComputerApplications, 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'etat de l'art de la ´ securit ´ e dans le cloud computing: Probl ´ emes et solutions ` de la securit ´ e 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. Azrour, 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. Azrour, A novel anomaly network intrusion detection system for internet of things security, in IoT and Smart Devices for Sustainable Environment, M. Azrour, A. Irshad, and R. Chaganti, eds. Cham, Switzerland: Springer, 2022, pp. 129–138.

[17]. A. Guezzaz, A. Asimi, M. Azrour, 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 Plateform 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. Abdulazeez, Machine learning classification based on radom 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 data-sets, 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.