Comparative Analysis of Multiple Machine Learning Models for Cloud-Based Intrusion Detection Systems

Habibun Nabi Hemel, Md Sakib Sadman Badhon, Debjoty Mitra
Department of Computer Science
BRAC University, Dhaka, Bangladesh
{habibun.nabi.hemel, sakib.sadman.badhon, debjoty.mitra}@g.bracu.ac.bd

***Abstract*—Buying and maintaining own server or infrastructure is quite hard and costly. Because of that, most small or large companies use cloud servers as their primary infrastructure. Using these cloud servers can reduce both costs and the extra hassle of maintenance. Not only that, these cloud services provide 100% uptime and protection from all intrusions. But as the AI is evolving, so are these attacks. Because of that, manually detecting these intrusions is getting quite hard for these cloud service providers. That is the problem this paper is trying to solve. This paper tries to implement machine learning models to detect cloud-based intrusions. Moreover, this paper provides a comparative analysis of multiple Machine Learning (ML) models to find which one is the best suitable for cloud-based intrusion detection systems.**

***Index Terms*—Machine learning, Data Analysis, Ter classification, K- nearest neighbors (KNN),Naive Bayes, SVM, Streamlit, F1, recall, Lable encoding, regression, Random Forest and Decision tree, LSTM, SMOTE, CatBoostClassifier.**

I. Introduction

Cloud computing plays an integral role in everyone's daily lives. As everything from newspapers to shopping becomes more popular online, more businesses are beginning to adopt cloud computing. Cloud computing provides several features, like resilience, scalability, and ubiquitous access. Because of that, it is becoming more popular among businesses of all sizes.

However, As more businesses adapting cloud platform, cyber threats are beginning to rise. SQL injection, distributed denial of service (DDoS) and other cloud-based attacks are becoming more popular in recent years, posing a significant threat to the businesses and their users.

In response to the security threats, this paper tries to propose a solution by implementing ML in the intrusion detection system. The authors have carefully tested popular ML models like Random Forest, KNN, Logistic Regression, AdaBoostClasifier, and Naive Baye to see which one is best for spotting the cloud-based intrusion. The result of this study revealed that Random Forest and a few other classification models performed best in this scenario. The authors have also tested this model in a real-life scenario by deploying it on a cloud server.

This research sets the stage for future improvements using advanced machine learning models. The main goal of this study is to make cloud system safer from evolving cyber threats.

A. Architecture Diagram

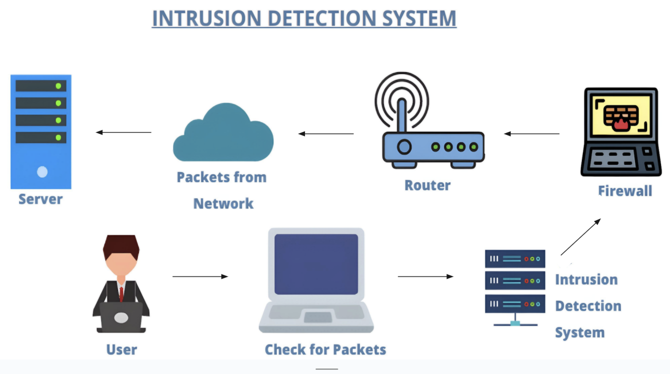


Fig. 1. Project Flow

The proposed intrusion detection system will start working when someone tries to visit the protected server. The detection system will check the visitor's request. If everything looks normal, the visitor will get access to the server. On the other hand, if the detection system thinks something is fishy, then it will block the request.

II. Literature Review

[1] introduces a system for detecting intrusion made specifically for cloud computing. It combines a multi-layer perceptron (MLP) network with an Artificial Bee-Colony (ABC) and fuzzy clustering algorithm. By using this ABC optimization method, the MLP can tell the difference between normal and abnormal network traffic, making the cloud more secure. The author of this paper tested this model on CloudSim with the NSL-KDD dataset. The result shows that this method works well.

On the other hand, [2] talks about keeping cloud computing safe from any kind of cyber threat. The author felt that people urgently needed tools to detect intruders in cloud systems. This paper suggests a unique plan called Intrusion Tolerance via Threshold Cryptography to make cloud systems stronger against attacks. The proposed system uses a special code to make sure the data stays safe. The author tested the plan using CloudSim and showed that it works well in finding and fixing intrusions.

On [3], the author talks about how cloud computing has changed the way to access networks and computer resources. He also mentioned the difficulties in keeping these services secure. To reduce these difficulties, the paper proposes a cloud-based intrusion detection model using machine learning. This study used a random forest model, which classified the intrusions with 98.3 percent accuracy.

The profile-based network intrusion detection and prevention system described in the paper [4] represents an innovative approach to safe cloud environments from both internal and external threats. By employing network profiling, the system creates individual profiles for each virtual machine within the cloud, which captures and analyzes the network activity. This approach enables the system to detect and identify potential risks specific to cloud users and their assigned virtual machines. By using these profiles, the system can effectively detect and prevent malicious activities, enhancing the overall security posture of cloud networks.

[5] introduces a network based intrusion detection model for the cloud, employing ensemble based ML and a voting scheme. The author of this paper uses CICID's 2017 dataset to train the model. The model performed very well in intrusion detection with minimal error. Results showcase an accuracy of 97.24 %.

These papers explore many innovative approaches for intrusion detection in cloud computing, leveraging techniques such as ensemble-based machine learning, profile-based network analysis, and ANN. As none of the research has compared different ML models to know which one works better in intrusion detection, this paper is trying to fill that gap.

III. Methodology

The process started with exploratory data analysis (EDA). In this step, the authors tried to focus on the important features. Removing null values and duplicate values is a crucial part of this study. Not only that, label encoding on categorical values also plays a big part, as most of the ML models cannot handle the categorical values, so converting the values to numerical values will be very helpful for those models.

Moreover, to address the data imbalance issue, the authors implemented the Synthetic Minority Oversampling Technique (SMOT), which uses synthetic data to make the class balanced. After finishing the EDA, the authors tested the performance of multiple machine learning models using LazyClassifier to identify the best-performing model. Finally, among all the models, the authors deployed the chosen model using Streamlit, enabling interactive access to our intrusion detection system.

A. Data Collection

The name of the dataset that is used for this study is "Intrusion Detection System." This binary classification dataset was originally from the US Air Force and collected from Kaggle. The dataset provides a comprehensive simulation of intrusions in the military network environment of the US Air Force LAN system. For each TCP/IP connection, 42 quantitative and qualitative features are obtained from normal and attack data. The dataset also contains 25192 data points, but as the data is collected from a real-world scenario, there is lots of noise in this dataset. To remove these noises from the dataset, the authors had to heavily preprocess the data.

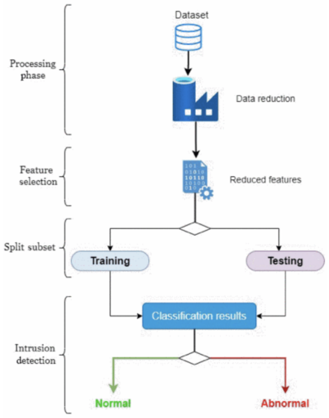


Fig. 2. Model Trainning Processes

B. Data Preprocessing

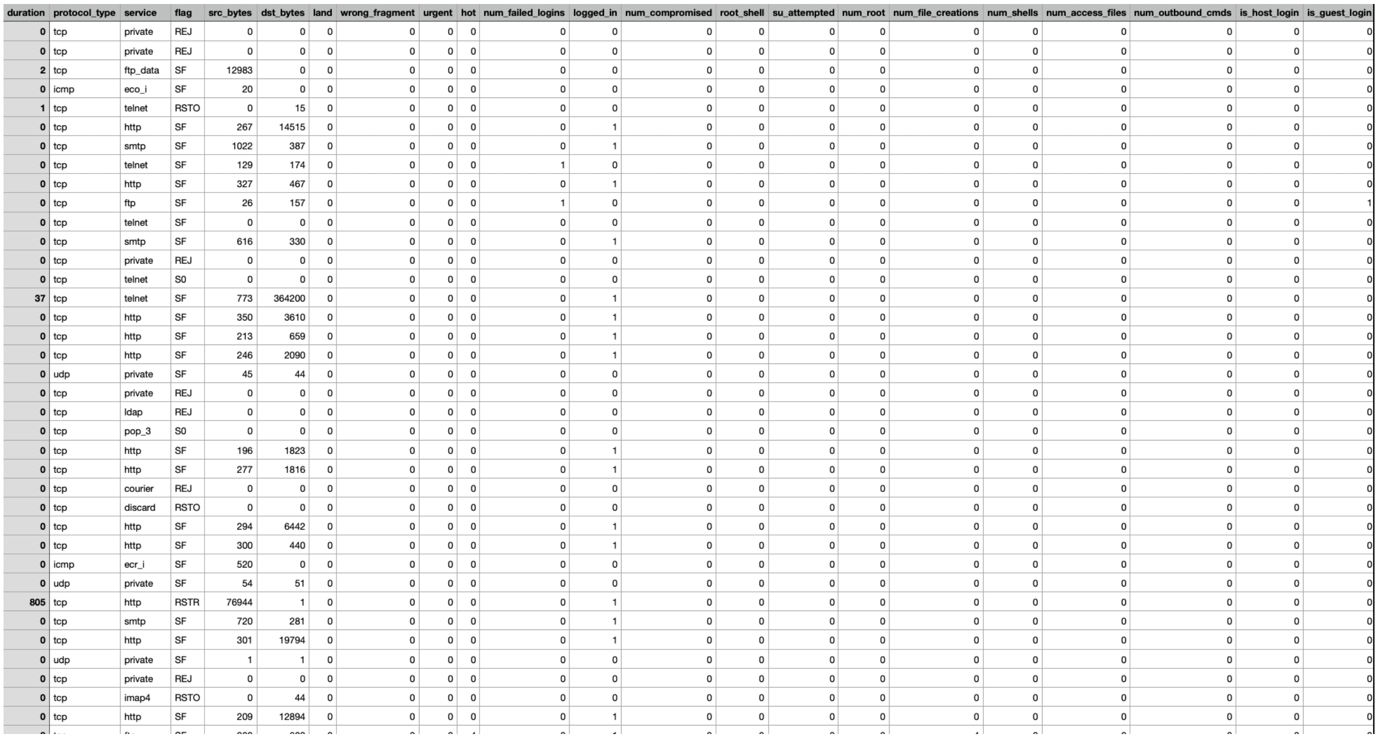
Data preprocessing is the most important thing to do to convert some random information into meaningful data. This preprocessing contains steps like data collection, data cleaning, data analysis, and data interpretation. Doing these processes step by step will yield some valuable insights and make the dataset more prepared for better decision-making.

C. Label encoding

To convert categorical features into numerical features for training the ML models, the authors have chosen label encoding as it suits the dataset more. There are a few reasons behind using this encoding. The first reason is the low number of unique values. As most of the features of the dataset are binary or ternary, choosing Lebel encoding would work perfectly in this scenario. Also, the dataset has a large dimension; using the second-best encoding option, which is one hot encoding, would further increase the dimension unnecessarily. However, alternatives like ordinal encoding (assigning unique integers based on order) and target encoding (replacing categories with mean target values) are also available.

D. Correlation Matrix

The correlation matrix also plays a vital role for choosing the right features. Correlation matrix tells the association between features and identifies multicollinearity, while heatmap offers straightforward data visualization of this correlation. In this study, the authors removed all the features which had 70% correlation in between them, because of that the features of this dataset dropped from 41 to 28.



duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_shell	su_attempted	num_root	num_file_creations	num_shells	num_access_files	num_outbound_cmds	is_host_login	is_guest_login
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	icmp	eco_i	SF	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	tcp	telnet	RSTO	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	267	14515	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	smtp	SF	1022	387	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	telnet	SF	129	174	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	327	467	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	ftp	SF	26	157	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
0	tcp	telnet	SF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	smtp	SF	616	330	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	telnet	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	tcp	telnet	SF	773	364200	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	350	3610	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	213	659	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	246	2090	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	udp	private	SF	45	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	ldap	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	pop_3	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	196	1823	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	277	1816	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	courier	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	discard	RSTO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	294	6442	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	300	440	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	icmp	ecr_i	SF	520	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	udp	private	SF	54	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
805	tcp	http	RSTR	76944	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	smtp	SF	720	281	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	301	19794	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	udp	private	SF	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	imap4	RSTO	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	209	12894	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Fig. 3. Dataset Sample

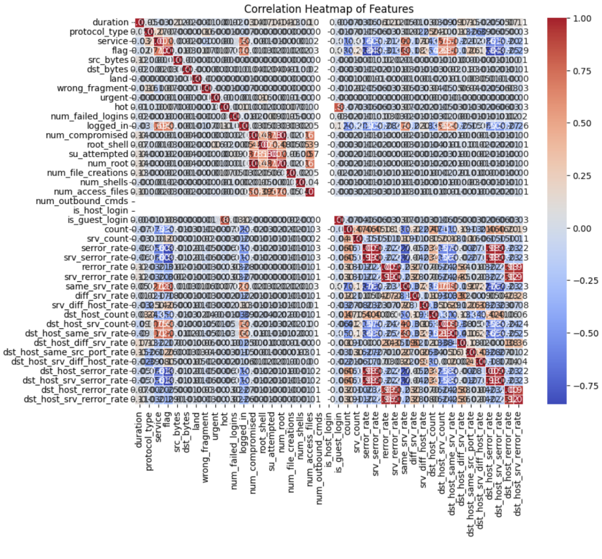


Fig. 4. Correlation Matrix

E. Data Cleaning and Handling Outliers

The feature selection process is a very important step as it removes unnecessary features from the dataset and narrows down the dimension of the dataset. The “Network Intrusion Dataset” originally had 42 features and 25192 data points in the beginning. After using CatBoostClassifier and Random Forest to find out the importance of every feature, the authors manage to find out that there are only 15 features that are actually useful for this task. By removing these unnecessary features, the dimension of the dataset shrinks, which reduces the time to train models. The selected features for the final training includes service, flag, src bytes, dst bytes, count, samesrv rate, diffsrv rate, dsthosts rvcount, dst hostsamesrv rate, dsthostsamesrcportrate, and class. The model was then trained on this new dataset to predict the label, indicating whether the connection is an intrusion or not.

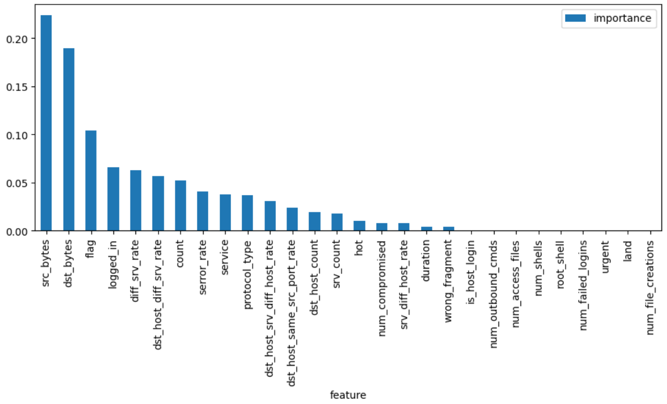


Fig. 5. Best Features

F. Imblanced Dataset

The target class of this dataset has two categories: ”normal” and ”anomaly”. This target class has a huge imbalance between them. As the data is taken from the real world scenario, most of the connection was normal. Not only that, other features of this dataset also indicate the imbalance issue among them. This imbalance may create biases in the prediction. Because of that the authors of this paper implemented “SMOT” to create synthetic data points to remove the imbalance.

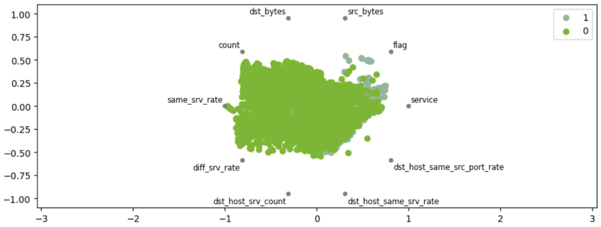


Fig. 6. Classes of Output

G. Model Training

The authors chose a bunch of ML models for the analysis and comparison. As the study conducted in a short time, choosing the right model was pretty crucial to save the time. To test the dataset on most of the ML models in a short time, the authors imported LazyClassifier and tested the accuracy and other performance metrics.

The dataset was split into 70–30 ratios. Then the dataset is tested on the following models: LGBMClassifier, RandomForestClassifier, DecisionTreeClassifier, ExtraTreesClassifier, XGBClassifier, BaggingClassifier, ExtraTreeClassifier, AdaBoostClassifier, KNeighborsClassifier,LabelPropagation, LabelSpreading, SVC, LogisticRegression, PassiveAggressiveClassifier, SGDClassifier, LinearSVC, CalibratedClassifierCV, RidgeClassifierCV, RidgeClassifier, LinearDiscriminantAnalysis, Perceptron, QuadraticDiscriminantAnalysis, NuSVC, NearestCentroid, GaussianNB, BernoulliNB, and DummyClassifier

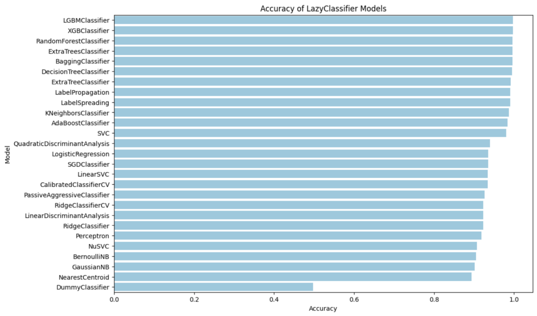


Fig. 7. Result Bar

a) Random Forest: As it can be seen from the figure, many ML models performed best to classify the data properly. Among them, the random forest was selected, saved, and deployed in the cloud.

Among the classification algorithms available, Random Forest stands out for its effectiveness. It comprises multiple decision trees, each determining the class label based on the feature variable. Through voting and ensemble, the final class label is determined, resulting in a more accurate prediction model.

Advantages:

- can balance error in data sets where classes are imbalance
- Large data with higher dimensionality can be handled easily additional assumptions are not required

Disadvantages:

- It does better job on classification problems rather than regression as it finds harder to produce continuous values rather than discrete one.

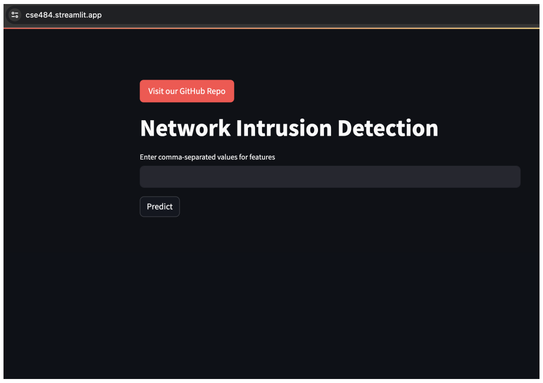

cse484.streamlit.app
Visit our GitHub Repo
Network Intrusion Detection
Enter comma-separated values for features
Predict

Fig. 8. Cloud Hosting

H. Deployment in Cloud

To ensure wider accessibility and usability of the detection system, the authors deployed it on Streamlit, a cloud-based website hosting platform that will allow users to interact with it through a GUI. Despite the time constraint preventing the development of a full-fledged website to give the whole project scenario of how an intrusion detection system will work between a user and a website, the authors provided a small demonstration of how the process will work. The link to the website can be found in the resources section. Through this website, users can input data and receive insights on network intrusion detection, distinguishing ”normal” and ”anomalous activities” based on the input.

IV. Result and Analysis

The final results of performing various ML models on the dataset are mentioned in the table (Fig. 9) below. It can be noticed that many models performed well, but random forest was selected because of its easy-to-understand mechanism. This model can perfectly predict the normal user and the anomaly now using random forest. Anyone can go to the website and test the abnormal request, then take the necessary steps to protect their website.

To select the random forest, the author considered F1 score, Precision,recall ,and accuracy. Combining all the four matrices random forest performed came out as one the best. Following graph shows the graph of the measurements.

A. F1 Score

The F1 score is the harmonic mean of precision and the recall score. It serves as a balance between those two metrics. A perfect F1 score of 1.00 suggests an ideal balance between precision and recall, which indicates that the model achieves optimal performance in terms of both identifying relevant instances.

B. Recall

The recall is also known as sensitivity and highlights. It measures the ability to identify all relevant instances of a class. A recall score of 1.00 implies that the model correctly identifies all instances of classes ”0” and ”1” among all instances that truly represent these classes. This indicates that there are no false negatives in the classification.

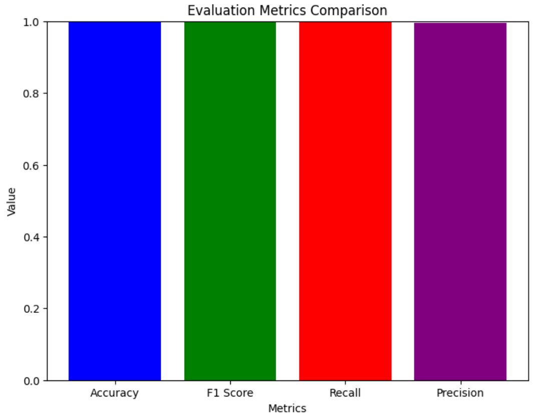


Fig. 9. Random Forest’s Score

V. Conclusion

After testing various ML models, LGBMClassifier and XGBClassifier RandomForestClassifier, ExtraTreesClassifier, and BaggingClassifier performed the best. Among these models, a random forest classifier was selected for the study. Reaching 100 percent accuracy indicates the possibility of overfitting. To fix this issue, more work needs to be done. The result of this research showed that ML models can effectively distinguish between normal and anomalies. Lastly, the dataset used in our analysis is outdated, and with the ever-evolving nature of cloud computing, to do further research, new data is needed. Therefore, there is ample scope for future work to develop more advanced security by creating and analyzing updated data.

VI. Resources

- Detection System - Website Link
- Detection System Code - Colab Link
- Dataset - Kaggle Link
- Website Deployment Code - Github Link

References

[1] B. Hajimirzaei and N. J. Navimipour, “Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm,” *ICT Express*, vol. 5, no. 1, pp. 56–59, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959517303405

[2] A. V. Reddy, K. S. Kumar, and V. H. Prasad, “Intrusion detection on cloud applications,” *International Journal of Computer Science and Mobile Computing*, vol. 2, pp. 1–7, 2013.

[3] H. Attou, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, “Cloud-based intrusion detection approach using machine learning techniques,” *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 311–320, 2023.

[4] M. Derfouf and M. Eleuldj, “Performance analysis of intrusion detection systems in the cloud computing,” in *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*. IEEE, 2017, pp. 1–8.

[5] P. Singh and V. Ranga, “Attack and intrusion detection in cloud computing using an ensemble learning approach,” *International Journal of Information Technology*, vol. 13, pp. 565–571, 2021.