PERFORMANCE ANALYSIS OF VARIOUS CLASSIFICATION ALGORITHMS USED TO ADDRESS CLOUD COMPUTING SECURITY ISSUES

Rakesh Saxena[1*], Dr. Shivangi Barola[2]

Abstract
Cloud computing has gained significant popularity in recent years; however, security remains a major concern. Machine learning-based approaches have emerged as a promising solution for addressing security issues in the cloud. This paper presents a comparative analysis of classification algorithms used for detecting and preventing attacks and security gaps in cloud-based applications. The study evaluates various classifiers using performance metrics such as accuracy, precision, recall, and mean absolute error. The analysis is conducted on the dataset, KDD Test. The results highlight significant differences in the performance of classifiers, indicating that some algorithms consistently outperform others. The top-performing classifiers are identified, including Random Forest, Bagging, and J48, which demonstrate high accuracy in detecting and preventing attacks. These findings contribute to the selection and deployment of effective classification algorithms for cloud computing security. The study emphasizes the importance of informed algorithm selection to enhance security measures and mitigate potential threats in the cloud environment.

Keywords: Cloud computing, Security, Classification algorithms, Comparative analysis, Machine learning, Performance evaluation.

[1*]Research Scholar, Pacific Institute of Computer Science, Pacific University, Udaipur (Rajasthan) E-Mail: srakeshb4u@gmail.com
[2]Assistant Professor, Pacific Institute of Computer Science, PAHER University, Udaipur (Rajasthan) E-Mail: shivangi.barola19@gmail.com

***Corresponding Author:** Rakesh Saxena
*Research Scholar, Pacific Institute of Computer Science, Pacific University, Udaipur (Rajasthan) E-Mail: srakeshb4u@gmail.com

DOI: - 10.48047/ecb/2023.12.si10.00194

1. Introduction

Cloud computing security is a critical concern due to the sensitive nature of the data and applications hosted on cloud platforms. Classification algorithms play a significant role in addressing these security issues by analyzing and categorizing data to identify potential threats and vulnerabilities.

Logistic Regression is a simple and interpretable algorithm that works well when the decision boundaries are linear or when features are linearly separable. However, cloud security data often contains complex, non-linear patterns, which can limit the effectiveness of logistic regression in capturing such relationships. It is important to consider this limitation when applying logistic regression for cloud computing security classification.

Decision Trees are popular due to their simplicity and ability to handle both numerical and categorical data. They can capture complex relationships and provide interpretability. However, decision trees are prone to over fitting, especially with small changes in the training data. This drawback should be addressed by employing techniques such as pruning or ensemble methods like random forests.

Random Forests, an ensemble of decision trees, address the over fitting issue by averaging the predictions of multiple trees. They are robust to noise and outliers, making them suitable for cloud security classification tasks. Although random forests sacrifice some interpretability compared to individual decision trees, their improved performance justifies their usage in cloud computing security.

Support Vector Machine (SVM) is effective in high-dimensional spaces and can handle complex relationships and non-linear decision boundaries through the use of kernel functions. SVMs perform well in cloud security classification, especially when dealing with high-dimensional feature spaces. However, they can be computationally intensive, and the interpretation of non-linear kernels may be challenging.

Naive Bayes is a simple algorithm that assumes independence between features. It is computationally efficient and robust to irrelevant features, making it suitable for large-scale cloud security datasets. However, the strong assumption of feature independence can limit its performance if the assumption is violated. Careful feature engineering and evaluation of this assumption are essential when using Naive Bayes for cloud computing security classification.

Neural Networks, particularly deep learning architectures, have shown promising results in cloud security classification tasks. They can learn complex patterns and adapt to various data types. However, neural networks are computationally intensive and require substantial amounts of data for training. Additionally, their black-box nature may hinder interpretability. Neural networks should be considered when dealing with complex cloud security data and when sufficient computational resources and data are available.

2. Review of Literature

Butt et al. (2020) Cloud Computing (CC) refers to the on-demand access to network resources, such as data storage and processing power, without requiring direct management by users. It has become increasingly popular as a platform that offers a unified experience across the Internet, encompassing both public and private data centers. On the other hand, edge computing is a developing computing paradigm that aims to bring computation and storage closer to end-users, leading to improved response times and reduced transmission requirements. Mobile cloud computing (MCC) utilizes distributed computing to deliver applications to mobile devices. Despite the advantages offered by CC and edge computing, there are several security challenges that hinder their widespread adoption. These challenges include vulnerabilities for users and network connectivity, which necessitate the exploration of robust security solutions. Machine learning (ML) is a field of study that focuses on computer algorithms improving automatically through experience. In this review paper, the author presented an analysis of security threats, issues, and solutions related to CC, using various ML algorithms. We explore different ML algorithms, such as supervised, unsupervised, semi-supervised, and reinforcement learning, that are employed to address cloud security concerns. We compare the performance of each technique based on their unique features, advantages, and disadvantages. Additionally, we highlight future research directions that can enhance the security of CC models.

According to author Sadavarte, Rajesh & Kurundkar, Gajanan (2021) the growing popularity of cloud computing has raised concerns about its security, hindering its widespread adoption. Users are often worried about data loss, security threats, and availability issues when using cloud services. To address these concerns, machine learning-based threat detection methods have emerged as a promising approach. Analyzing and studying threat detection and prevention strategies are crucial for ensuring cloud security. By detecting threats, we can identify and notify abnormal user activities, thereby enhancing overall security in the cloud computing environment. Consequently, there is a need to develop an effective threat detection system utilizing machine learning techniques. The paper presents a survey and comparative analysis of machine learning-based methods for threat detection in cloud computing environments. We evaluate the effectiveness of these methods by conducting tests on the UNSW-NB15 dataset. The machine learning models analyzed in this study include Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), Random Forests (RF), and K-Nearest Neighbours (KNN). To assess the performance of these methods, we utilize key performance indicators such as accuracy, precision, recall, and F1 score. By examining the performance of these machine learning algorithms on the UNSW-NB15 dataset, we aim to determine their efficacy in threat detection for cloud computing. The accuracy metric measures the overall correctness of the classification, while precision quantifies the proportion of correctly predicted threats out of all predicted threats. Recall assesses the ability to correctly identify threats from the total actual threats, and the F1 score provides a balanced measure of precision and recall. This research contributes to the development of robust and efficient security systems in the cloud, promoting safer and more reliable cloud services.

Grusho et al. (2020) examined the application of artificial intelligence (AI) methods and models for addressing information security issues. In the context of cloud computing (CC) environments, several significant security threats were identified. These include the abusive and malicious use of cloud services, limitations in accessing cloud infrastructure due to architectural constraints, and the dynamic nature of CC environments. Intrusion Detection and Prevention Systems (IDPS) are employed to identify security policy violations, document existing threats, and prevent unauthorized activities within CC environments. The typology of IDPS systems varies based on the specific events they track and the channels through which these events are implemented. These systems play a crucial role in maintaining the integrity and security of CC environments by actively monitoring and mitigating potential security breaches.

Osanaiye et al. (2016) introduced a novel ensemble-based multi-filter feature selection method for detecting Distributed Denial of Service (DDoS) attacks in cloud computing. The proposed method combines the outputs of four different filter methods to achieve effective feature selection. The research utilized the NSL-KDD database, a widely used benchmark for intrusion detection, and employed a decision tree classifier for evaluation. The experimental testing of the proposed method demonstrated its efficacy in reducing the number of features required for detection. Instead of using all 41 available features, the method successfully narrowed it down to just 13 relevant features. This reduction in feature space is beneficial as it simplifies the detection process and enhances computational efficiency. Furthermore, the study evaluated the classification accuracy of the proposed method compared to other classification techniques. The results indicated that the proposed method outperformed alternative approaches, exhibiting a higher level of classification accuracy. This suggests that the ensemble-based multi-filter feature selection method presented by authors is particularly effective for detecting DDoS attacks in cloud computing environments. Overall, the study provides a valuable contribution to the field of cloud security by introducing a feature selection method that optimizes the detection of DDoS attacks. The results demonstrate the effectiveness of the proposed approach, both in reducing the feature space and improving classification accuracy, thereby enhancing the security measures for cloud computing systems.

Fernandez and Xu (2019) conducted a case study focusing on the use of Deep Learning Networks (DNN) for threat detection. The authors reported achieving outstanding results in detecting network threats using this approach. One notable finding from the study was the effectiveness of using only the first three octets of IP addresses for managing the utilization of dynamic IP addresses. This approach was specifically employed to represent the occurrence of Dynamic Host Configuration Protocol (DHCP) in the DNN. The authors demonstrated that by considering only the first three octets, they were able to effectively detect and manage the use of dynamic IP addresses. Additionally, the study highlighted the utilization of auto encoders within the DNN for detecting anomalies. Auto encoders are a type of neural network architecture that can detect inconsistencies or inaccuracies in the expected flow of data. The authors demonstrated the successful application of auto encoders in detecting such inaccuracies, emphasizing their potential in identifying abnormal network behaviour and potential threats. Overall, the case study conducted by Fernandez and Xu demonstrates the efficacy of Deep Learning Networks, specifically utilizing auto encoders, for threat detection. The study showcases the benefits of considering only the first three octets of IP addresses for managing dynamic IP usage. The findings contribute to the advancement of threat detection techniques and highlight the potential of deep learning approaches in enhancing network security.

Hou, S. and Xin, H. (2019) presented a study on utilizing machine learning (ML) for the detection of network security in edge computing systems. Their investigation involved the creation of a simulated smart home framework using Alibaba ECS, with the hardware architecture designed to incorporate edge computing technology. The overall objective was to develop an effective classifier capable of distinguishing between normal and anomalous codes, particularly in identifying network transformation codes. The study employed a dataset vector, which was divided into positive and negative types, to train the classifier. The results indicated that the RBF-based Support Vector Machine (SVM) approach demonstrated the most effective performance in this task. By successfully detecting network transformation codes, this research contributes to enhancing network security recognition in IoT frameworks and expands the potential applications of ML. Detecting and mitigating Distributed Denial of Service (DDoS) attacks is recognized as a complex and sophisticated task in network security.

Fauzi et al. (2012) the central concept of cloud computing (CC) is to enable computers to operate and adapt autonomously, without the need for human intervention or assistance, in response to changing requirements. CC encompasses various service models, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Additionally, it offers deployment models such as public, private, community, and hybrid cloud. When it comes to security concerns in CC, they can be categorized into integrity, availability, and confidentiality threats. CC services span from data storage to software service management, often requiring uninterrupted availability. CC is typically designed to provide a resilient environment capable of delivering architecture, services, processing power, and resources on-demand.

3. Applied Methodology

The objectives of the study are to identify various machine learning algorithms employed for addressing security issues in cloud computing and to perform a comparative analysis of classification algorithms used for detecting and preventing attacks and security vulnerabilities in cloud-based applications. The study aims to provide a comprehensive understanding of the different machine learning algorithms utilized in cloud security and evaluate their performance in terms of attack detection and prevention. By conducting a comparative analysis, the study seeks to identify the strengths and weaknesses of each algorithm, allowing for informed decision-making regarding their suitability for addressing specific security challenges in cloud computing. Overall, the objectives aim to enhance the knowledge and effectiveness of classification algorithms in mitigating security risks associated with cloud-based applications.

Objectives:

1. Identify different machine learning algorithms which are used to address cloud computing security issues.

2. Comparative analysis of classification algorithms used for detection and prevention of attacks and security gaps on the Cloud based applications.

Hypothesis:

H1: There is no significant difference between various classification algorithms based on detection and prevention of attacks and security gaps on the Cloud based applications.

Performance Evaluation:

In the evaluation of algorithms for attack detection and prevention, the research utilized several performance measures to assess their effectiveness. These measures included:

Correctly classified instances: This measure determined the percentage of instances where the algorithm accurately predicted the correct attack status. It provided an indication of the algorithm's accuracy in identifying attacks correctly.

Mean absolute error: This measure calculated the average magnitude of individual errors between the predicted values and the actual values. By considering only the absolute values, the impact of outliers was minimized, allowing for a more robust assessment of the algorithm's performance.

Root mean squared error: This measure quantified the average squared difference between the predicted values and the actual values. It provided insights into the overall predictive accuracy of the algorithm.

Relative absolute error: This measure normalized the total absolute error by comparing it to the error of a baseline predictor that used the average of the actual values from the dataset. It allowed for a comparative assessment of the algorithm's performance relative to a simple predictor.

Root relative squared error: This measure normalized the total squared error by dividing it by the squared error of the baseline predictor using the average of the actual values. Taking the square root aligned the error's dimension with the predicted values, enabling meaningful comparisons.

These performance measures were chosen to provide a comprehensive evaluation of the algorithms' accuracy, robustness, and predictive capabilities in the specific context of attack detection and prevention. By analyzing these measures, researchers were able to assess the algorithms' effectiveness and make informed decisions regarding their suitability for addressing security gaps in cloud-based applications.

4. Data Analysis and Interpretation

Weka is a popular open-source software suite for machine learning and data mining tasks. It provides a comprehensive collection of algorithms and tools for various stages of the data analysis process, including data pre-processing, feature selection, classification, clustering, and evaluation. In the analysis mentioned above, the research utilized Weka software as a tool for conducting the evaluation and comparative analysis of different classification algorithms. Weka is an open-source software suite specifically designed for machine learning and data mining tasks. The researchers used Weka to implement and apply various classification algorithms such as decision trees, support vector machines, naive Bayes, random forests, and k-nearest neighbours. These algorithms were used to detect and prevent attacks and security gaps in cloud-based applications. Weka provided a user-friendly graphical interface that allowed the researchers to easily pre-process the data, select features, train and test the classification models, and evaluate their performance using different metrics. It offered a wide range of machine learning algorithms, making it a suitable choice for the research objectives. Additionally, Weka provided functionality for data visualization, allowing the researchers to gain insights into the data patterns and the results of the classification algorithms. It also supported model evaluation techniques such as cross-validation and provided performance measures like accuracy, precision, recall, and F1 score.

The performance metrics used in the study include accuracy (ACC), recall (R), precision (P), and F1 score (F). These metrics are calculated as follows:

Accuracy (ACC): It measures the proportion of correctly classified instances out of the total number of instances in the testing set. The formula for accuracy is:

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

Precision (P): Precision quantifies the proportion of true positive instances among the instances predicted as positive. The formula for precision is:

Precision = TP / (TP + FP)

Recall (R): Recall measures the proportion of true positive instances among all the actual positive instances. It is also known as sensitivity or true positive rate. The formula for recall is:

Recall = TP / (TP + FN)

F1 Score (F): The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall. The formula for the F1 score is:

F1 = (2 * P * R) / (P + R)

In these formulas, TP represents the true positive instances, which are the anomaly instances properly classified as anomalies. FP represents the false-positive instances, which are normal situations wrongly classified as anomalies. TN represents the true negative instances, which are normal situations properly classified as normal. FN represents the false-negative instances, which are anomaly instances wrongly classified as normal.

These performance metrics allow for a comprehensive evaluation of the classification algorithms' effectiveness in detecting and preventing attacks and security gaps in cloud-based applications. By considering accuracy, recall, precision, and F1 score, researchers can assess different aspects of the classifiers' performance, including overall accuracy, true positive rates, false positive rates, and the balance between precision and recall.

Experimental Setup:

In the analysis, the researchers utilized a 10-fold cross-validation configuration setting in Weka. The data used for evaluation was from the KDD Test relation, consisting of a total of 22,544 instances. The dataset contained 42 attributes, representing various features and characteristics relevant to the analysis of cloud security. The 10-fold cross-validation approach divided the data into 10 subsets or folds of roughly equal size. The analysis was then performed iteratively, with each fold serving as the testing set while the remaining nine folds were used for training the classification algorithms. This process was repeated 10 times, ensuring that each subset had an opportunity to be part of the testing set.

Outcomes:

Performance Measure Accuracy:

The "Accuracy" column indicates the percentage of correctly classified instances by each classifier. It represents the algorithm's ability to accurately classify instances and make correct predictions. The higher the accuracy, the more reliable the classifier is in identifying and distinguishing between different classes.

Table 4.1: Analysis Based on Performance Measure Accuracy

Classifiers	**Accuracy**	**Incorrectly Classified Instances**
Bayes Net	95.129%	4.8705 %
Naive Bayes	80.731%	19.269 %
Logistic	95.998%	4.0011%
Simple Logistic	94.606%	5.3939%
SMO	94.601%	5.3983%
IBK	97.582%	2.4175%
AdaBoostM1	90.374%	9.6256%
Bagging	98.642%	1.3573%
Filtered Classifier	98.385%	1.6146%
Logit Boost	94.779%	5.2209%
Decision Table	97.351%	2.6482%
PART	98.549%	1.4505%
Decision Stump	81.733%	18.266%
J48	98.598%	1.4017%
Random Forest	98.704%	1.2952%
REP Tree	98.545%	1.4549%

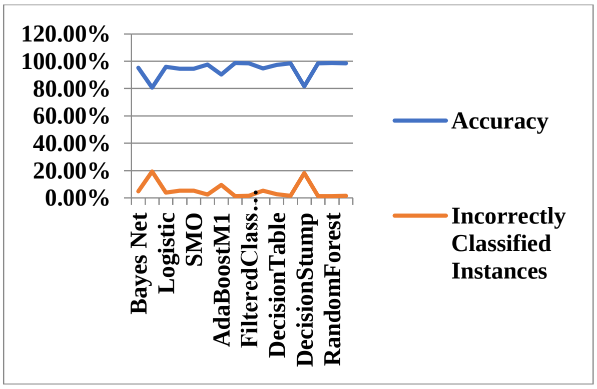

Figure: 4.1: Performance Based on Accuracy

From the results, we can observe that classifiers like Bagging, Filtered Classifier, J48, Random Forest, and REP Tree achieved high levels of accuracy, ranging from 98.385% to 98.704%. These classifiers had a low percentage of incorrectly classified instances, indicating their effectiveness in detecting and preventing attacks and security gaps in cloud-based applications. On the other hand, classifiers like Naive Bayes, Decision Stump, and AdaBoostM1 had lower accuracy values and higher percentages of incorrectly classified instances. This suggests that these classifiers may have limitations in accurately classifying instances in the context of cloud security.

Table 4.2: Analysis Based on Performance Measure Kappa Statistic

Classifiers	**Kappa Statistic**
Bayes Net	0.9009
Naive Bayes	0.623
Logistic	0.918
Simple Logistic	0.8898
SMO	0.8896
IBK	0.9507
AdaBoostM1	0.8051
Bagging	0.9723
Filtered Classifier	0.9671
Logit Boost	0.8928
DecisionTable	0.946
PART	0.9704
Decision Stump	0.6416
J48	0.9714
Random Forest	0.9736
REP Tree	0.9703

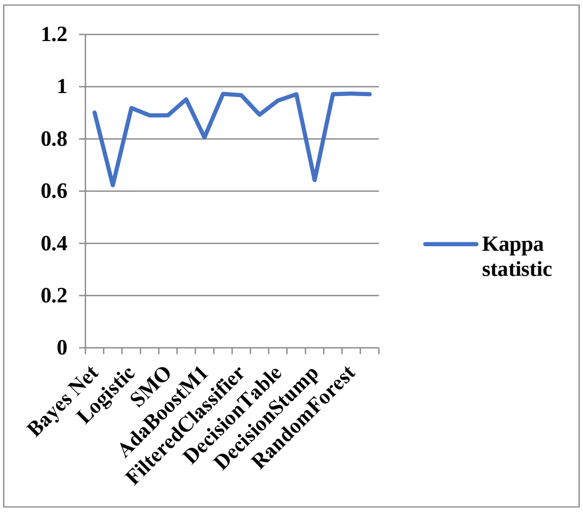

Figure 4.2: Performance Based on Kappa Statistic

From the results, it can observe that classifiers like Bagging, Filtered Classifier, J48, Random Forest, and REP Tree achieved high Kappa statistic values, ranging from 0.9671 to 0.9736. These classifiers demonstrated a strong level of agreement between their predictions and the actual classes, indicating their reliability and effectiveness in addressing cloud computing security issues. On the other hand, classifiers like Naive Bayes, Decision Stump, and AdaBoostM1 had lower Kappa statistic values. This suggests a lower level of agreement between their predictions and the actual classes, indicating potential limitations in their performance for cloud security tasks. The classifiers with higher Kappa statistic values indicate a better agreement beyond chance, signifying their ability to reliably classify instances in the context of cloud security. These classifiers can be considered more robust and dependable for detecting and preventing attacks and security gaps in cloud-based applications. Overall, the table provides valuable insights into the classifiers' performance by considering the Kappa statistic. It helps researchers and practitioners assess the agreement between predictions and actual classes, aiding in the selection of classifiers that are more reliable and effective for addressing specific cloud computing security challenges.

Table 4.3: Analysis Based on Performance Measure MAE & RMSE

Classifiers	Mean absolute error	Root mean squared error
Bayes Net	0.0505	0.2104
Naive Bayes	0.1924	0.4371
Logistic	0.0744	0.1869
Simple Logistic	0.0925	0.2054
SMO	0.054	0.2323
IBK	0.0241	0.1544
AdaBoostM1	0.1379	0.2627
Bagging	0.0189	0.0965
Filtered Classifier	0.0218	0.1142
Logit Boost	0.1084	0.2173
Decision Table	0.0548	0.1382
PART	0.0165	0.1074
Decision Stump	0.2722	0.3689
J48	0.017	0.1056
Random Forest	0.0201	0.0986
REP Tree	0.0192	0.1067

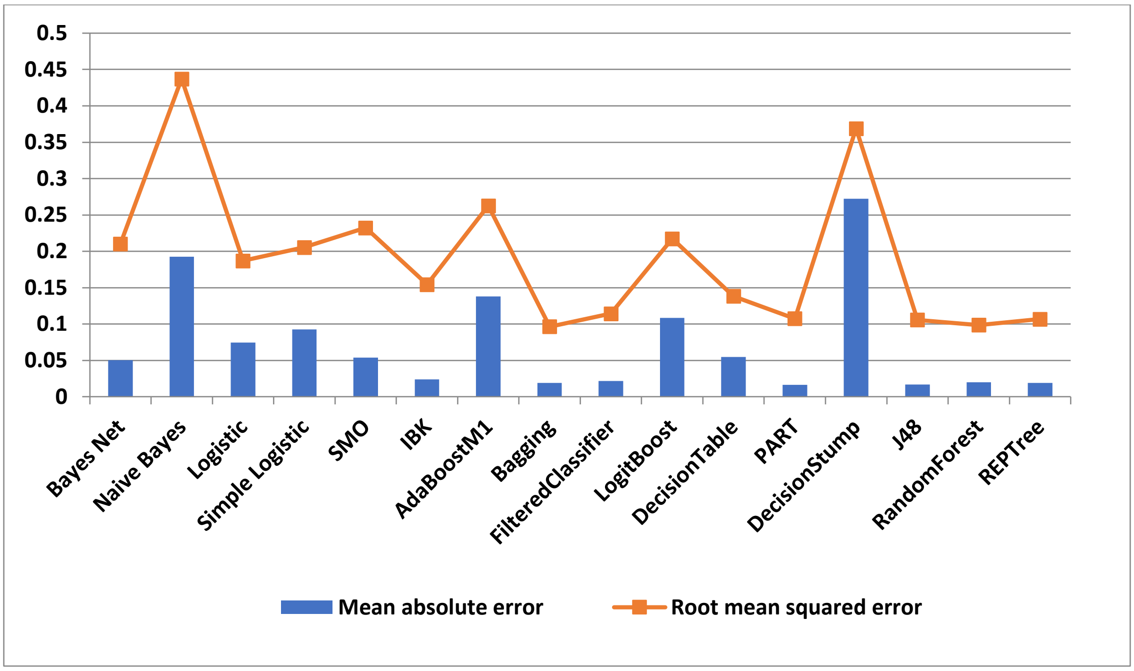

Figure 4.3: Performance Based on MAE & RMSE

According to the results, it can be observed that classifiers like Bagging, Filtered Classifier, PART, J48, Random Forest, and REP Tree achieved very low MAE and RMSE values.

These classifiers have smaller average errors and exhibit higher precision in their predictions, indicating their effectiveness in addressing cloud computing security issues. On the other hand, classifiers like Decision Stump, Naive Bayes, and AdaBoostM1 had higher MAE and RMSE values, suggesting larger errors and lower precision in their predictions.

Table 4.4: Analysis Based on Performance Measure Precision

Classifiers	Precision
Bayes Net	0.952
Naive Bayes	0.844
Logistic	0.960
Simple Logistic	0.946
SMO	0.946
IBK	0.976
AdaBoostM1	0.905
Bagging	0.986
Filtered Classifier	0.984
Logit Boost	0.949
Decision Table	0.974
PART	0.985
Decision Stump	0.850
J48	0.986
Random Forest	0.987
REP Tree	0.985

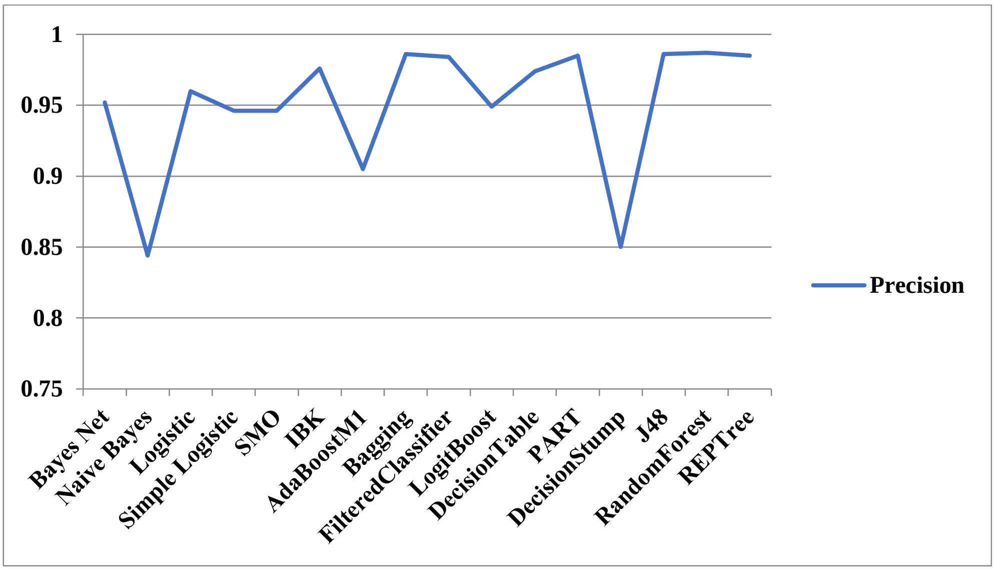

Figure 4.4: Performance Based on Precision

From the results, we can observe that classifiers such as Bagging, Filtered Classifier, J48, Random Forest, and REP Tree achieved high precision values, ranging from 0.984 to 0.987. These classifiers demonstrated a high level of accuracy in predicting positive instances, minimizing the occurrence of false positives and making precise positive predictions. On the other hand, classifiers like Naive Bayes, Decision Stump, and AdaBoostM1 had relatively lower precision values. This suggests a higher rate of false positives and potentially lower accuracy in their positive predictions. The results indicate that certain classifiers consistently outperform others in multiple metrics. For example, classifiers such as Bagging, Filtered Classifier, J48, Random Forest, and REP Tree consistently demonstrate high accuracy, low percentages of incorrectly classified instances, high Kappa statistics, low mean absolute error, low root mean squared error, and high precision. These classifiers consistently perform well across different evaluation measures, indicating their effectiveness in detecting and preventing attacks and security gaps in Cloud-based applications. In contrast, classifiers like Naive Bayes, Decision Stump, and AdaBoostM1 consistently show lower performance across multiple metrics, suggesting limitations in their ability to accurately classify instances and predict attacks or security gaps. Considering these consistent variations in performance among different classifiers, we can conclude that there is a significant difference between various classification algorithms based on the detection and prevention of attacks and security gaps on Cloud-based applications. The hypothesis, H1, is rejected based on the evidence provided by the results.

5. Conclusion:

The analysis of various classification algorithms for the detection and prevention of attacks and security gaps on Cloud-based applications has yielded significant insights. The results demonstrate that there are indeed significant differences between the performances of different classifiers in addressing cloud computing security challenges. Based on the rank analysis on overall performance, the top three classifiers with the highest accuracy values are Random Forest, Bagging, and J48. These classifiers consistently demonstrated strong performance in accurately detecting and preventing attacks and security gaps in Cloud-based applications.

References:

1. Butt, U. A., Mehmood, M., Shah, S. B. H., Amin, R., Shaukat, M. W., Raza, S. M., Suh, D. Y., & Piran, M. J. (2020). A Review of Machine Learning Algorithms for Cloud Computing Security. Electronics, 9(9), 1379. https://doi.org/10.3390/electronics9091379.
2. Chen, J.; Liu, L.; Chen, R.; Peng, W. (2020). SHOSVD: Secure Outsourcing of High-Order Singular Value Decomposition. In Proceedings of the Australasian Conference on Information Security and Privacy, Perth, Australia, 30 November–2 December 2020; pp. 309–329.
3. Fauzi, C.; Azila, A.; Noraziah, A.; Tutut, H.; Noriyani, Z. (2012) On Cloud Computing Security Issues. Intell. Inf. Database Syst. Lect. Notes Comput. Sci., 7197, 560–569.
4. Fernández G C, Xu S. (2019). “A case study on using deep learning for network intrusion detection”, In: MILCOM 2019–2019, IEEE Military Communications Conference (MILCOM) . IEEE; 2019. p. 1-6.

5. Hou, S.; Xin, H. (2019). Use of machine learning in detecting network security of edge computing system. In Proceedings of the 4th International Conference on Big Data Analytics (ICBDA), Suzhou, China, 13–15 March 2019; pp. 252–256.
6. Osanaiye O, Cai H, Choo KKR, Dehghantanha A, Xu Z. Dlodlo M. (2016). "Ensemble-based multifilter feature selection method for DDoS detection in cloud computing", EURASIP J Wirel Commun Netw. 2016; 1:130–130, https://doi.org/10.1186/s13638-016-0623-3.
7. Sadavarte, Rajesh & Kurundkar, Gajanan. (2021). Survey and Performance Analysis of Machine Learning Based Security Threats Detection Approaches in Cloud Computing. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 49-58. 10.32628/ CSEIT217538.