

Improving Cloud Security with Intrusion Detection based on Classifier Combination

Mohsen Heidari

heidari.mohsen@hotmail.com

BWB Consulting Ltd

Faezeh Atot



University of Nicosia

Research Article

Keywords: Cloud security, Data mining, Fraud detection, Supervisor learning, Intrusion detection and Attack

Posted Date: September 2nd, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-7368052/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

Improving Cloud Security with Intrusion Detection based on Classifier Combination

Author(s):

Mohsen Heidari · Faezeh Atot

Affiliations:

Mohsen Heidari

Department of Information Technology and Digital Innovation,

BWB Consulting Ltd, Nottingham, United Kingdom

<https://orcid.org/0009-0003-4869-2384>

Email: heidari.mohsen@hotmail.com

Faezeh Atot

Department of Digital Innovation, School of Business

University of Nicosia, Nicosia, Cyprus

<https://orcid.org/0009-0008-8831-3445>

Email: atot.f@live.unic.ac.cy

Corresponding Author:

Mohsen Heidari (heidari.mohsen@hotmail.com)

Abstract

Research Aim: By development of information technology, network security is considered as one of the main issues and has great challenges. Intrusion detection systems are a major component of a secure network. Traditional intrusion detection systems cannot adapt themselves to the new attacks thus today's intrusion detection systems have been introduced based on data mining.

Research method: Identifying patterns in large volumes of data, is a great help to us. Data mining techniques by identifying a binary label (normal packet, abnormal packet) and specifying attributes by classification algorithms can recognize the abnormal data. Therefore, the precision and accuracy of intrusion detection systems will increase, there by network security increases.

Findings: In this paper, we propose a model that evaluates the performance of the different algorithms on dataset. Simulation results show that the decision tree of the J48 algorithm, neural network of the Neural net algorithm, Bayesian network of the HNB algorithm, lazy model of the K-STAR algorithm, LibSVM in support vector machine algorithm, and Rule Induction Single Attribute algorithm in the rule-based model, have the best result in the different parameters for performance evaluation of intrusion detection system.

J48 algorithm provides the highest performance in the all-mentioned algorithms which has the accuracy of 85.49%, the precision of 86.57% and the recall of 86.90% for intrusion detection system.

Conclusion: The main innovation in this paper is using the lazy model algorithms that are not used in the intrusion detection systems. Also, we propose the 5 different samples from primary extracted data that achieve the best results for the different models and algorithms.

Keywords: Cloud security, Data mining, Fraud detection, Supervisor learning, Intrusion detection and Attack

1. Introduction

With the increasing adoption of cloud computing in enterprise devices, cloud service providers, and end users, concerns about security threats such as distributed denial of service (DDoS) attacks, intrusions, and malware have also increased. Traditional rule-based or signature-based intrusion detection methods are often ineffective against unknown or complex attacks, as they require constant updates and have limited adaptability (Sharif, 2024).

In contrast, machine learning, especially ensemble learning methods, has been shown to be highly effective in intrusion detection by increasing accuracy and reducing error rates (Wikipedia, 2025). Using methods such as bagging, boosting, and stacking in combination, while improving the overall

accuracy of the model, also increases stability and tolerance to outliers and noise (Alharthi et al., 2025).

In recent cloud-based applications, models such as Ensemble SVM using feature selection techniques (such as SelectKBest and ANOVA) have shown remarkable performance on the UNSW-NB15 dataset (Krishna et al., 2024). Also, in another study, combining CNN for feature extraction and Random Forest for classification on the KDD99 and UNSW-NB15 datasets resulted in accuracy of 97% and high accuracy of 98% (Azizi Doost et al., 2025).

Another innovative approach is to use the AdaBoost algorithm with weak classifiers in the cloud environment, which resulted in an overall accuracy of 96%. This study was conducted on real AWS network data and showed high stability and accuracy (Improved Attack Classification ..., 2025).

On the other hand, hybrid feature selection methods such as Boruta, Relief, and Pearson correlation coefficient in a layered learning framework (stacked classifiers) have improved the accuracy of DoS attack detection in the CICDDoS-2019 dataset (Springer, 2025).

In the field of infrastructure research, methods such as Ensemble of Random and Isolation Forests have been proposed for intrusion detection in cloud containers. This method, which is based on clustering system calls and converting them into graphs, has provided high detection rates and low false alarms (Iacovazzi & Raza, 2023).

On a larger scale, approaches based on cloud compression using a hybrid VGG19 model with data balancing (SMOTE) and Bayesian optimization on CIC IDS 2017 and 2018 data have achieved accuracies of around 99.2% (Saidane et al., 2024).

Finally, in the context of lightweight systems, the use of simple statistical methods for feature selection in IoT networks has increased the training speed by 63% and the detection accuracy by nearly 99.9% (Nature, 2025).

The main objective of this paper is to introduce the best algorithm given the dataset. That can distinguish normal packets from abnormal ones. The main innovation in the paper is the use of lazy model algorithms, rule-based model and decision tree model, which have not been used for intrusion detection systems so far. It also uses all the algorithms available in the classification methods available in the WEKA and Rapidminer software. And the proposal of 5 data samples extracted from the initial data and giving the best answer for different models and algorithms. The extraction of 5 data samples took a lot of time and all the different algorithms available in the classification models were simulated and executed with different data sets, which we finally proposed 5 initial data samples.

2. Research Literature

With the increasing growth of cloud computing technology, data security and protection of cloud resources have become one of the fundamental challenges (Zhang et al., 2023). Intrusion Detection

Systems (IDS) have been proposed as one of the vital tools for identifying attacks and security threats in cloud environments (Chen & Liu, 2022). In this area, the use of ensemble classifiers has attracted the attention of many researchers due to their high ability to increase accuracy and reduce the false alarm rate (Patel et al., 2021).

By integrating multiple classifiers such as decision trees, support vector machines (SVMs), and neural networks, hybrid methods based on machine learning have been able to provide better performance than single classifiers (Kumar & Sharma, 2024). For example, the combination of Bagging and Boosting algorithms in cloud intrusion detection systems has led to increased sensitivity and accuracy of detection (Wang et al., 2022). In addition, optimal feature selection methods using evolutionary algorithms such as genetic algorithms and particle swarm optimization (PSO) have been widely used to reduce computational complexity and improve the performance of classifiers (Singh et al., 2023). Intrusion detection in cloud environments faces several challenges due to large data sets, high complexity, and the need for real-time detection (Zhao & Hu, 2024). In this regard, the use of deep learning and self-organizing algorithms is increasing to improve accuracy and rapid response to threats (Miller & Johnson, 2022). Also, integrating reinforcement learning with hybrid classifiers allows for better adaptation to changes in attack patterns (Gomez et al., 2023).

Several studies have shown that the use of classifier combinations not only increases accuracy but also reduces the false positive rate, which is of great importance in security systems (Lee et al., 2023). Also, the use of hybrid methods such as combining random forest with convolutional neural network has provided very promising results, especially in detecting complex intrusions (Ahmed & Khalid, 2023).

Another important challenge is the scalability and flexibility of intrusion detection systems in the face of large volumes of data and a variety of attacks, for which recent research has developed methods based on distributed processing and federated learning (Patel & Singh, 2024). Also, the use of new technologies such as blockchain has been considered to increase the security of cloud data and ensure data integrity (Zhang et al., 2024).

In general, the research trend in the field of intrusion detection based on the combination of classifiers in cloud computing is moving towards designing more accurate, faster, and more robust models against advanced and sudden attacks (Wang et al., 2023). However, the challenges related to computational complexity, the need for large training data, and the preservation of user privacy still require further research (Kumar et al., 2023).

2-1 Data Mining

Data mining can be considered a natural evolution of information technology, which is a result of an evolutionary process in the database industry. Such as data collection and database creation, data management, and data analysis and understanding.

Here is a definition of data mining:

"Data mining is the process of extracting knowledge from large amounts of data stored in a database, data warehouse, or other information repository."(Bhowmik, 2015)

Based on this view, a typical data mining system has the following main components, of which Figure 1 shows the system architecture.

Therefore, data mining has been considered as one of the leading branches in the information industry and is considered as one of the most promising interdisciplinary development areas in the information industry.

2-1-1 Classification

In classification problems, the goal is to identify features that indicate the group to which each item belongs. This model can be used both to understand existing data and to predict how new data will behave.

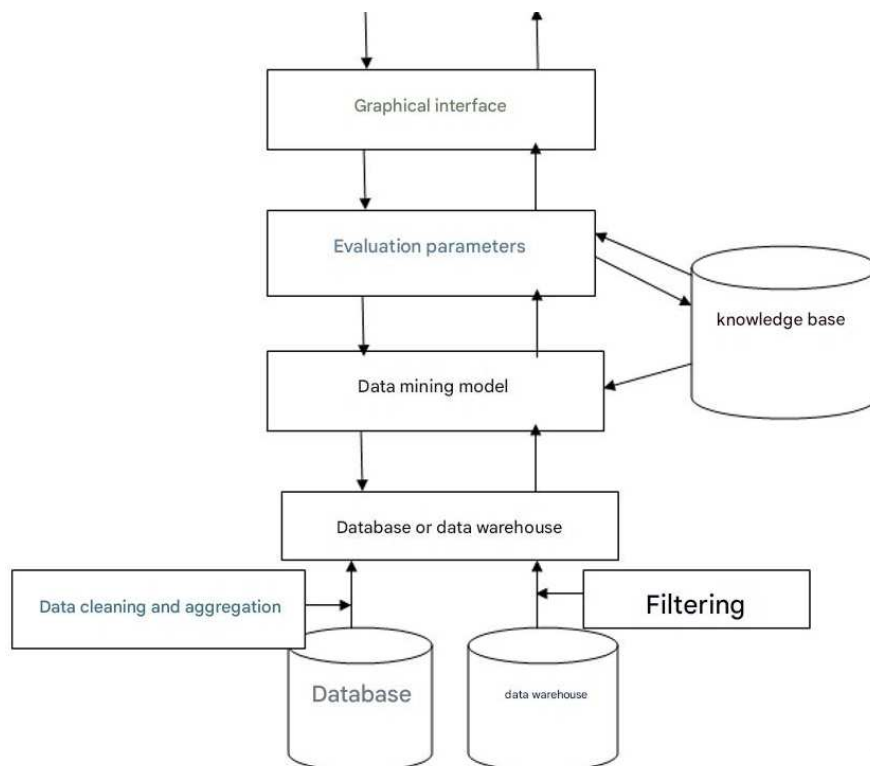


Figure 1: Architecture of a sample data mining system (Gerhard, 2016)

Data mining builds classification models by examining previously classified data and inductively builds a predictive model. These existing cases may come from a historical database.

2.2 Data mining models and algorithms

In this section, we intend to review the most important data mining algorithms and models. Many commercial data mining products use a set of these algorithms, and each of them is usually powerful in a specific area, and in order to use one of them, the necessary studies must be considered by a group of experts to select the most appropriate product. Another important point is that among these algorithms and models, there is no best one, and the model should be selected according to the data and performance in question.

3. Proposed Method

The main goal of this article is to introduce the best algorithms with respect to the data set that can distinguish normal packets from abnormal ones. The main innovation in the paper is the use of lazy model algorithms, rule-based model and decision tree model, which have not been used for intrusion detection systems so far, and the use of all the algorithms available in the clustering methods available in the WEKA and Rapidminer software, and the extraction of 5 data samples from the initial data that gives the best answer for different models and their related algorithms. Extracting 5 data samples took a lot of time, and all the different algorithms available in the clustering models were simulated and implemented with different data sets, which we finally proposed for 5 initial data samples. The work related to finding the best dataset required repeated testing of each algorithm with different datasets, modeling and evaluation, which ultimately succeeded in presenting 5 different data samples in terms of differences in the type of attributes that provide the best answer for the algorithms.

The steps of conducting research to implement the model are similar to any data mining-based method as explained below.

Step 1: Data Determination

In this step, the dataset is determined.

Step 2: Initial Data Analysis

Using expert knowledge and by calculating information such as data weights, mean, and data center, data analyses are performed.

Step 3: Create and train the model

After creating the model, it can be trained.

Step 4: Create knowledge

The created model has knowledge that it has learned from the training data set. This knowledge contains the structure of the data and recognizes the patterns in it.

Step 5: Test the model

The obtained knowledge is tested for data sets for which no information is available.

The proposed architecture for intrusion detection is shown in Figure 2.

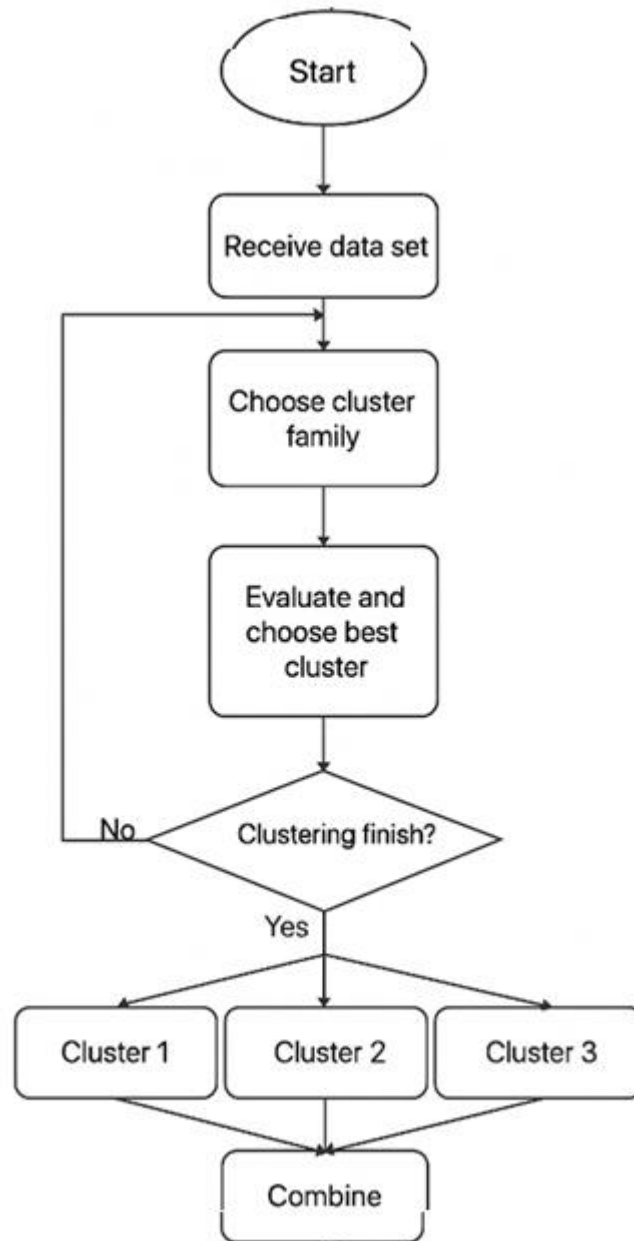


Figure 2: Proposed architecture for intrusion detection based on data mining method

In this method, we perform simulation using Rapidminer software and also adding the Weka algorithm to this software. With various investigations and multiple tests, five training and testing datasets are proposed for modeling, which are available in the appendix. In the evaluation section, each algorithm is described in detail and the training data and the necessary preprocessing are explained.

3-1 Training and testing data:

The data is known as DATA 1999 CUP KDD, which is labeled data related to the intrusion detection system and is publicly available. This data is used for classification and is labeled as normal or abnormal.

3-1-1 Data Characteristics:

The data was prepared at MIT Lincoln Laboratory.

The purpose is to review and evaluate the intrusion detection system with classification methods.

The raw training data is about 4 gigabytes of binary TCPDUMP data obtained over seven weeks of network traffic.

This data corresponds to five million connections.

The test data was obtained over 2 weeks and from 2 million records.

A connection is a sequence of TCP packets that flow from a source address to a specific address with a start and end time and the data is transferred under a specific protocol.

Each connection has either a normal label or an abnormal label. Each connection contains at least 100 bytes.

3-1-2 Basic properties of the data set:

The data properties and default data types are given in Tables 1, 2, and 3.

Table 1: Basic properties extracted from a TCP connection

Feature	Description	Type
duration	Length (number of seconds of communication)	Discrete
Protocol_type	Protocol type TCP, UDP	Discrete
service	Network service on the destination computer TELNET	Discrete
Src_byte	Number of bytes transferred from source to destination	Continuous
Dst_byte	Number of bytes transferred from destination to source	Continuous
flag	Normal or error status of communication	Discrete

land	If the host and port are the same, otherwise it is zero	Discrete
Wrong_fragment	If the host and port are the same, otherwise it is zero	Continuous
urgent	Necessary packet count	Continuous

Table 2: Features extracted from TCP connection

Feature	Description	Type
hot	hot index count	Continuous
Num_failed_logins	Number of failed login attempts	Continuous
Logged_in	One if successful, zero otherwise	Discrete
Num_compromised	Number of compromised conditions	Continuous
Root_shell	If root shell is accessible One otherwise zero	Discrete
Su_attempted	If root su command is executed One otherwise zero	Discrete
Num_root	Number of root accesses	Continuous
Num_file_creation	Number of file creation operations	Continuous
Num_shells	Shell prompt count	Continuous
Num_access_file	Number of operators accessing control file	Continuous
Num_outbound_cmd	Number of operators accessing control file	Continuous
Is_hot_login	Number of ftp remote access commands one in	Discrete

Table 3: Features extracted from the window

Feature	Description	Type
Count	Number of connections that have the same host	Continuous
Connections with the same host		
Error_rate	Percentage of connections with SYN errors	Continuous
Rerror_rate	Percentage of connections with REJ errors	Continuous
Same_srv_rate	Percentage of connections with similar services	Continuous
Diff_srv_rate	Percentage of connections with different services	Continuous
Srv_count	Number of connections with similar services that have existed in the last 2 seconds.	Continuous
Connections that have the same services		
Srv_serror_rate	Percentage of connections that have SYN errors	Continuous
Srv_rerror_rate	Percentage of connections that have REJ errors	Continuous
Srv_diff_host_rate	Percentage of connections that have SYN errors	Continuous

Most of the time is spent on preprocessing. The preprocessing steps are explained in the evaluation section for each algorithm. The algorithm for each model is available in the Rapid miner software. Figures 3, 4, 5, 6, 7, and 8. 9 show modeling with the software. If necessary, functions and features can also be added to the software.

Figure 3: Neural network modeling with Rapidminer software

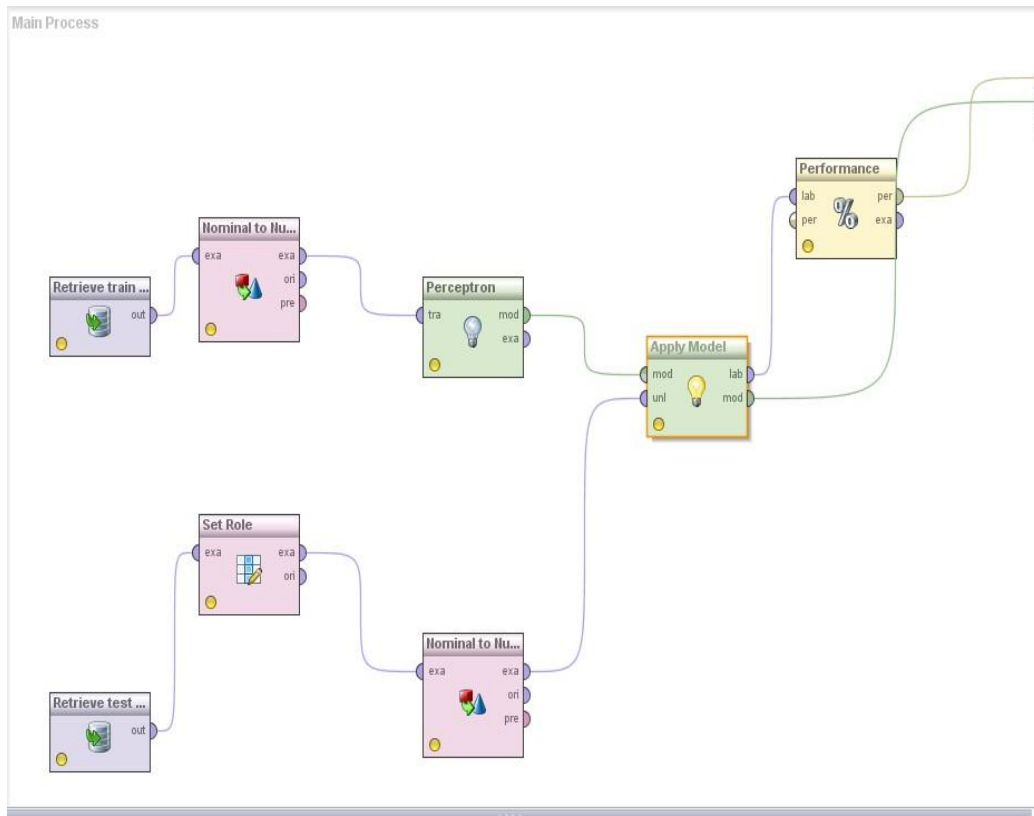
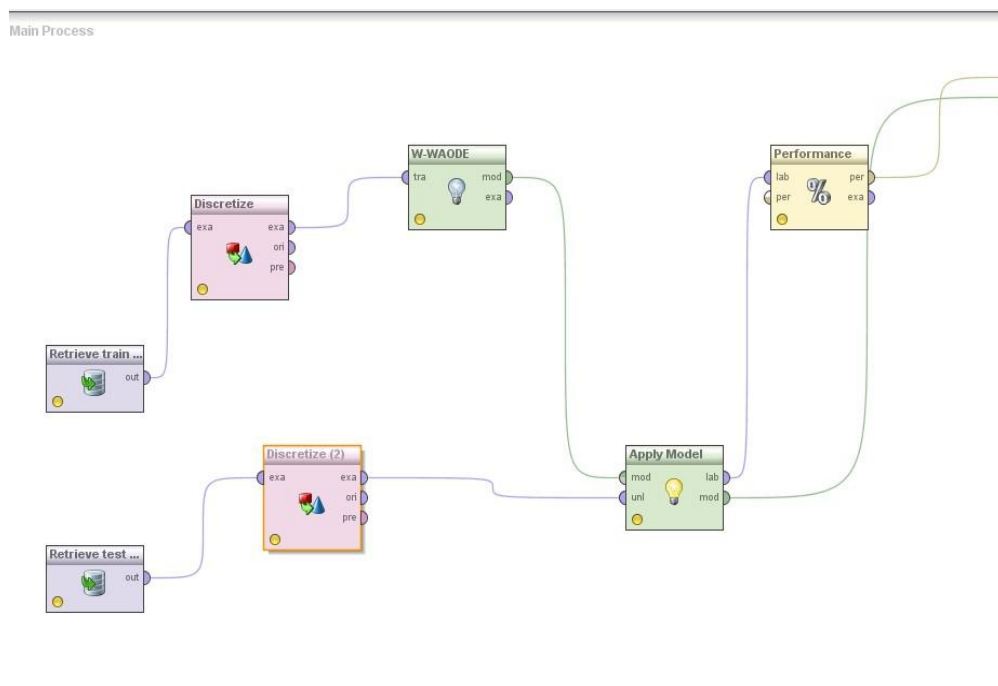


Figure: 4 Bayesian Modeling with Rapidminer

Figure: 5 Decision Tree Modeling with Rapidminer



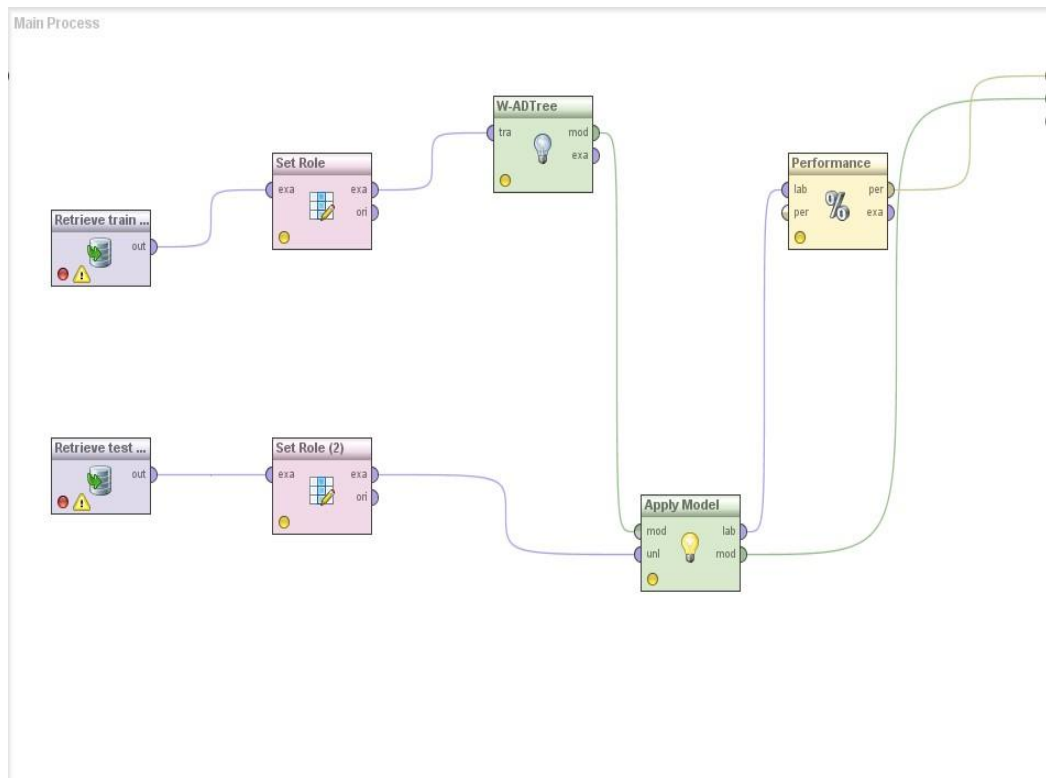


Figure 6: Modeling a rule-based model with Rapidminer software

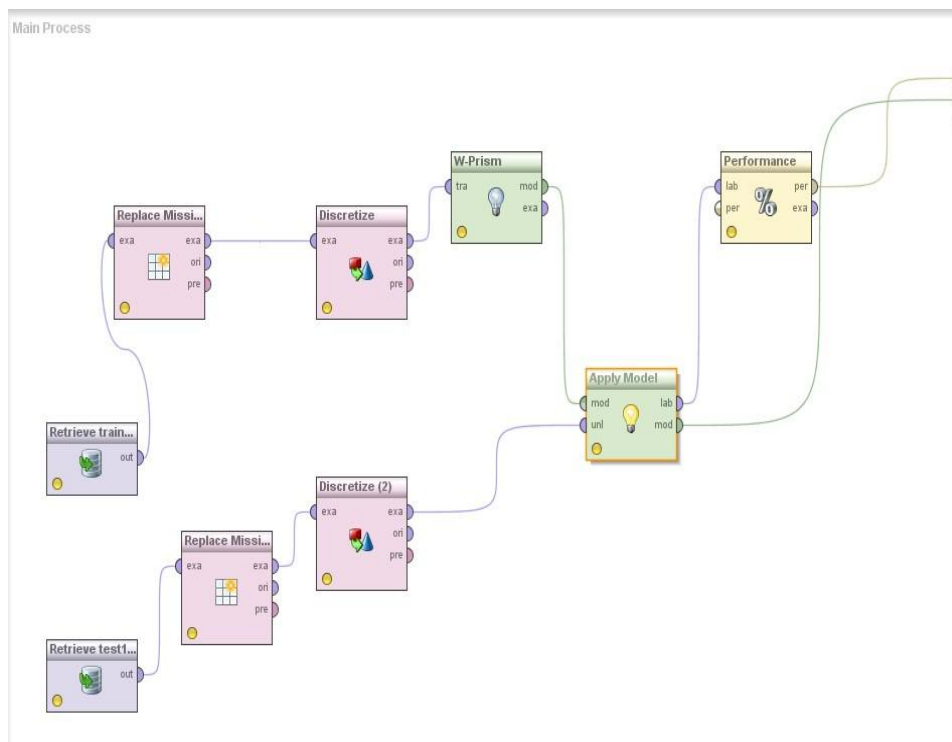


Figure 7: Modeling the support vector machine model with Rapidminer software

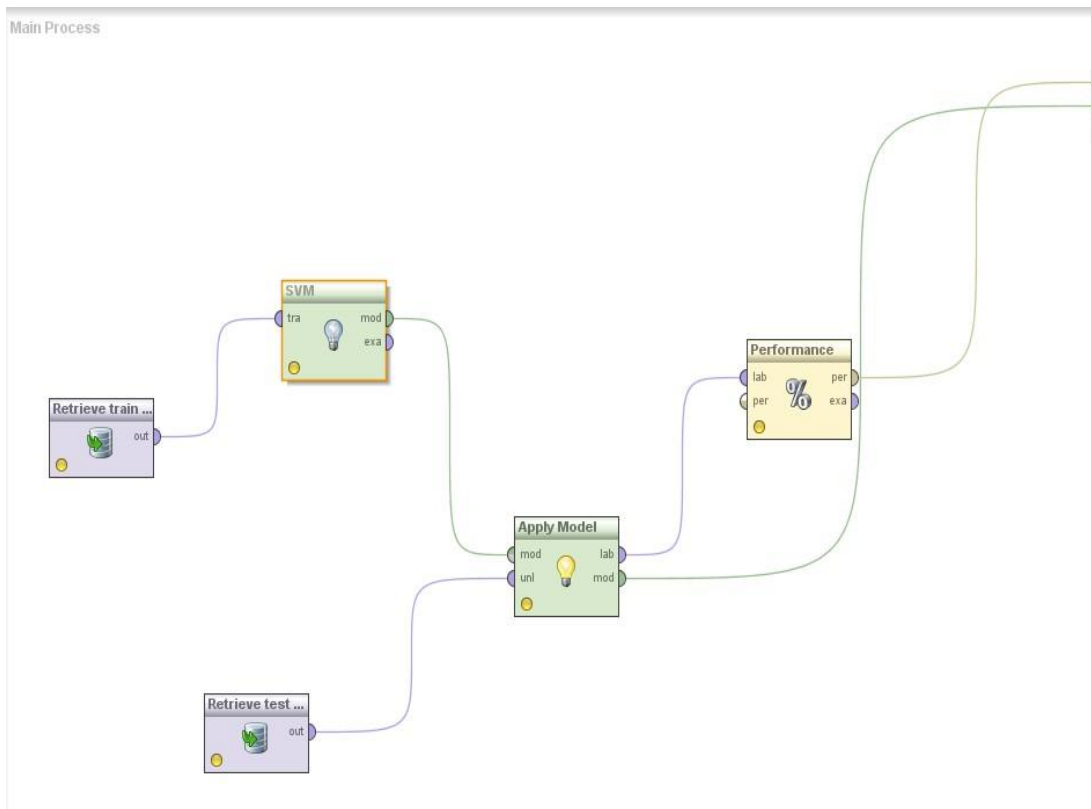
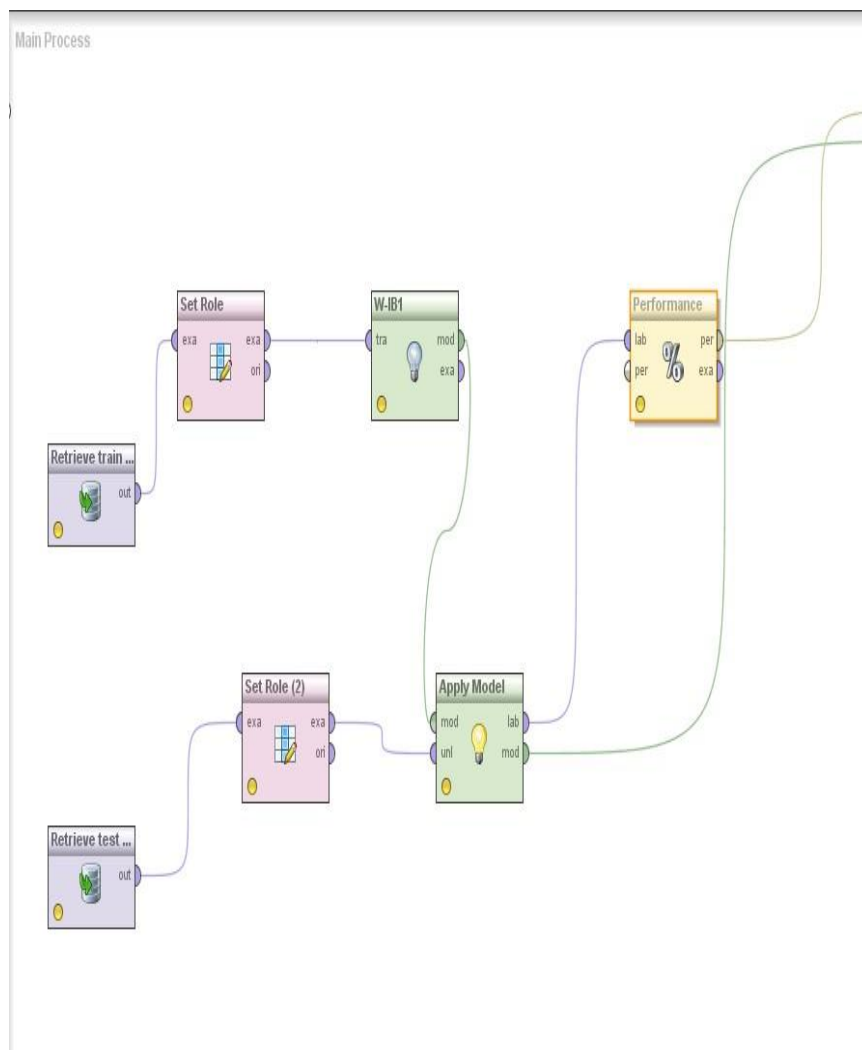


Figure 8: Modeling the lazy model algorithm with Rapidminer software



The evaluation section is also shown in Figure 8, which has a wide variety of parameters, and by selecting any of them in the modeling section, the parameter result can be observed.

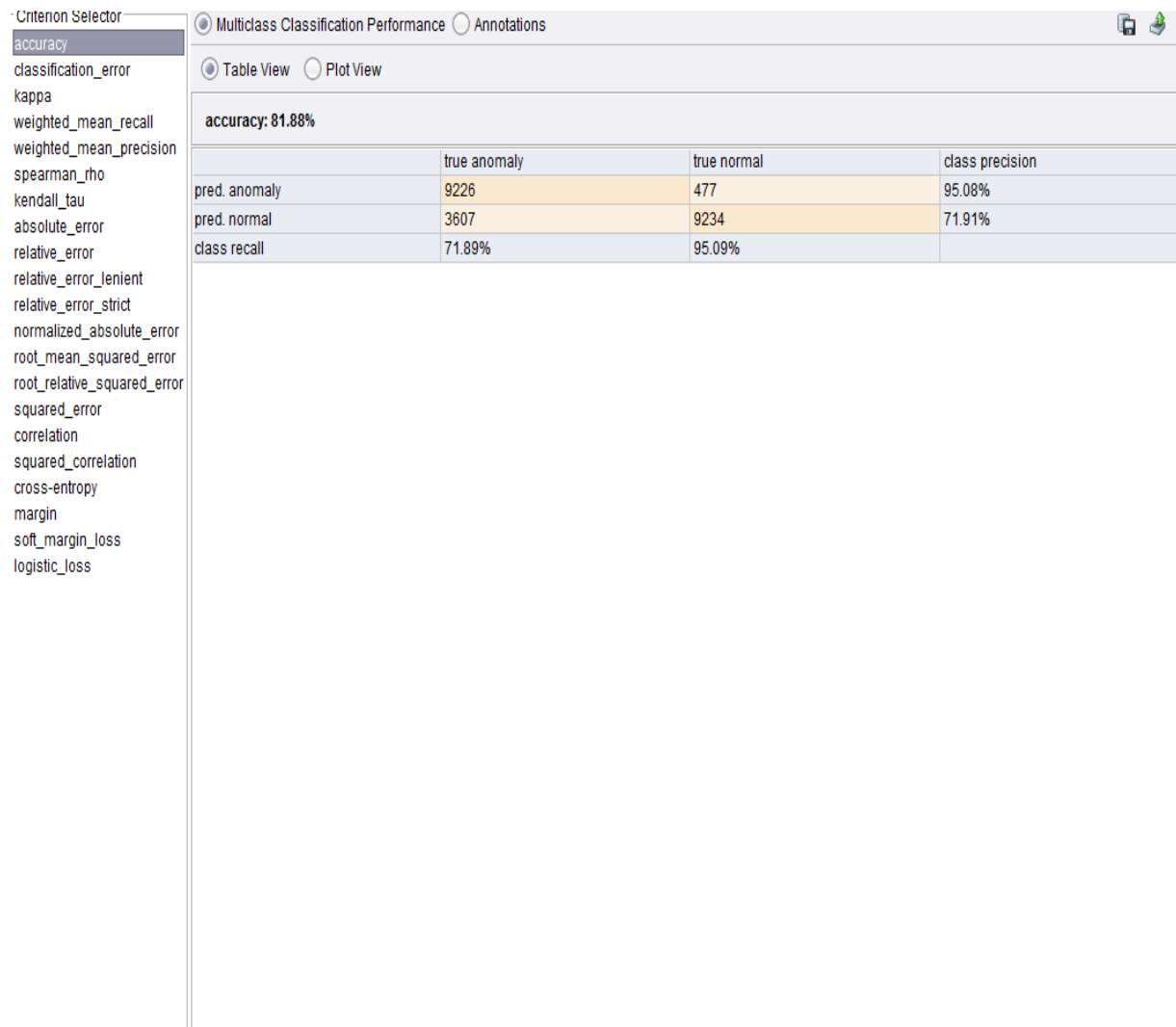


Figure 9: Example of Rapidminer software output with different evaluation parameters

4. Findings

In this section, all algorithms related to different data mining models are simulated and the results obtained from the evaluation of these models based on different parameters and the confusion matrix are shown.

4-1 Bayesian model algorithms and their evaluation

In this section, all the algorithms Aode, Waode, Naive Baysian, Kernel naive Baysian, BayesianLogic Regression and Dmnbtext, HNB, Bayesenet, Aodesr, Aode in the Bayesian model is simulated and evaluated using the simulation software.

In Figure 10, we have compared the Bayesian model algorithms in terms of the accuracy parameter.

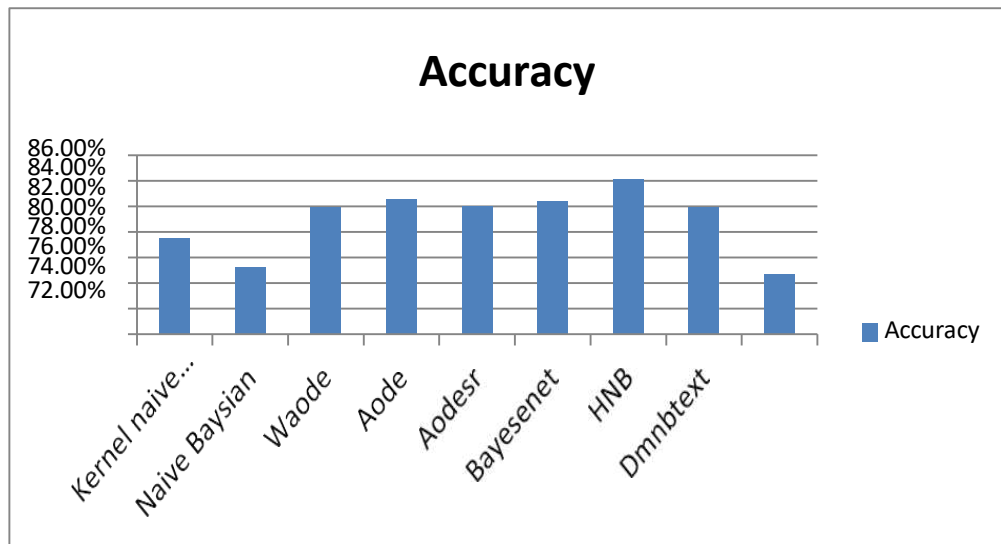


Figure 10: Evaluation chart of Bayesian model algorithms in terms of accuracy parameter.

In Figure 11, we compared Bayesian model algorithms in terms of accuracy parameter.

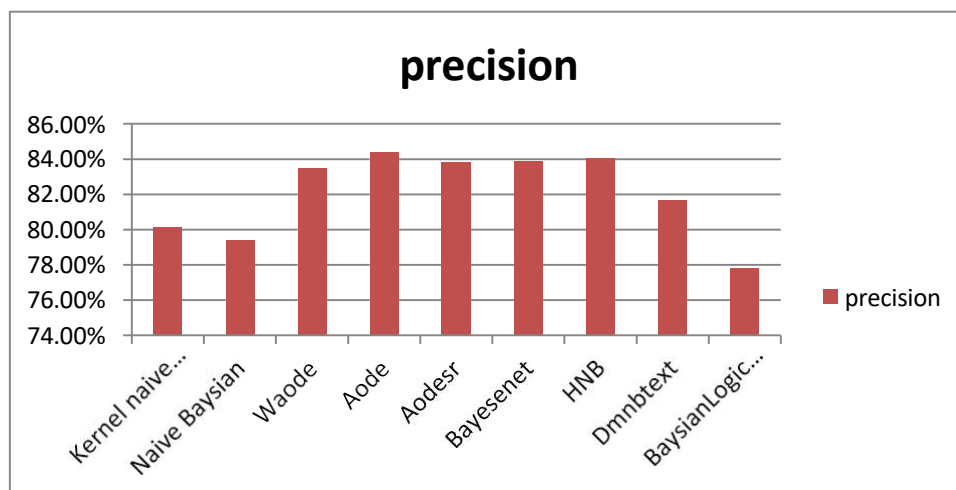


Figure 11: Evaluation chart of Bayesian model algorithms in terms of accuracy parameter. In Figure 12, we compared Bayesian model algorithms in terms of the recall parameter.

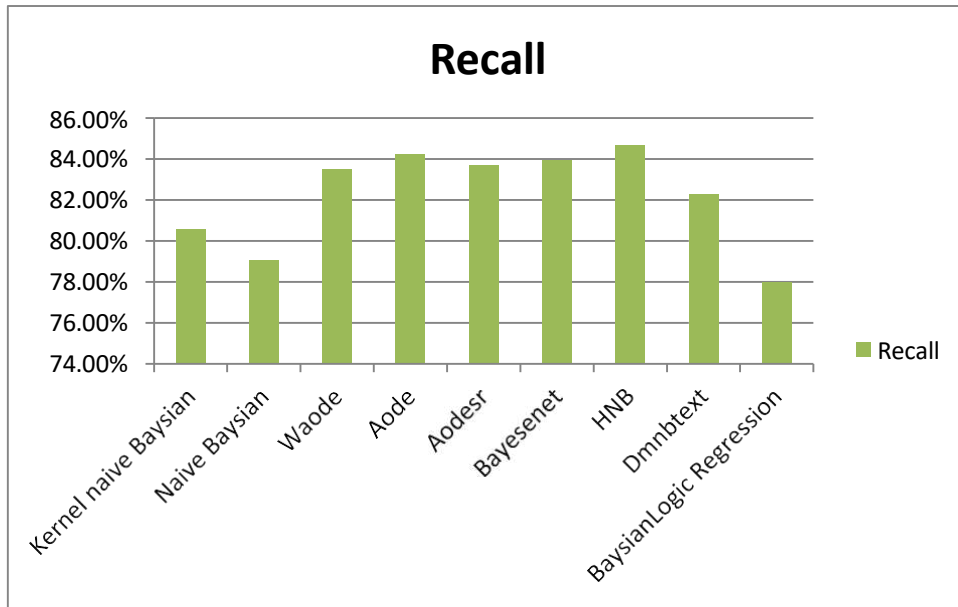


Figure 12: Evaluation chart of Bayesian model algorithms in terms of recall parameter. In Figure 13, we compared Bayesian model algorithms in terms of parameter F.

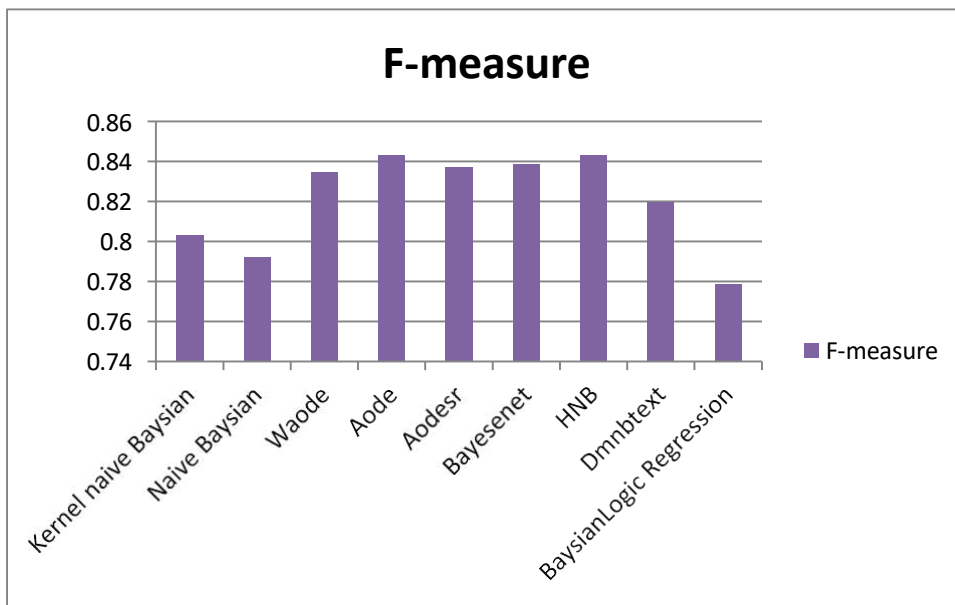


Figure 13: Bayesian model algorithm evaluation chart based on F parameter

Figure 14 shows the overall accuracy, precision, recall, and F criteria for the Bayesian model.

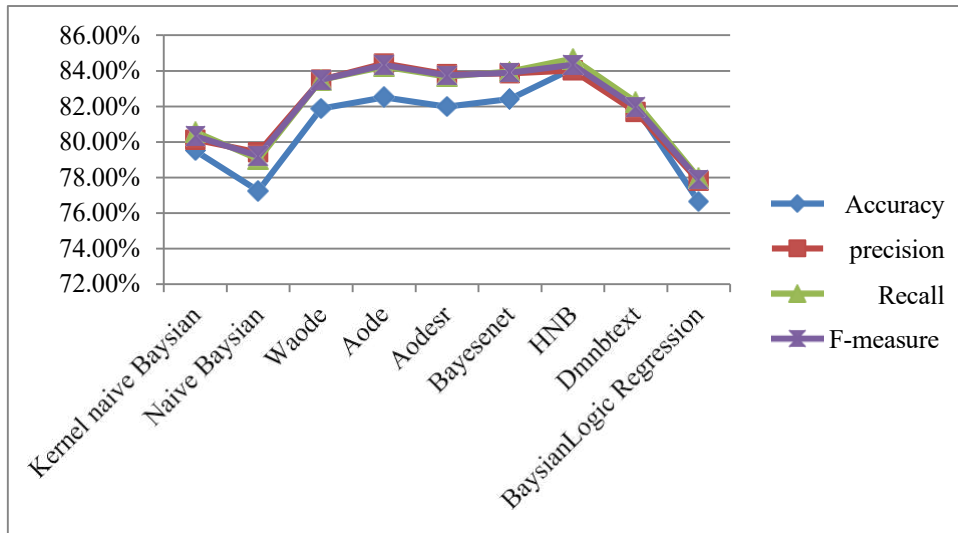


Figure 14: Evaluation chart of Bayesian model algorithms according to different parameters

In the evaluation of parameters, and according to the graphs, the HNB algorithm has better performance than other algorithms. Only in the Precision criterion, the Aode algorithm has better performance.

4-2 Lazy Model

In this section, all the IB1, IBK, LWL, KSTAR and KNN algorithms in the Lazy Model have been simulated and evaluated using software.

In Figure 15, we have compared the Lazy Model algorithms in terms of the accuracy parameter

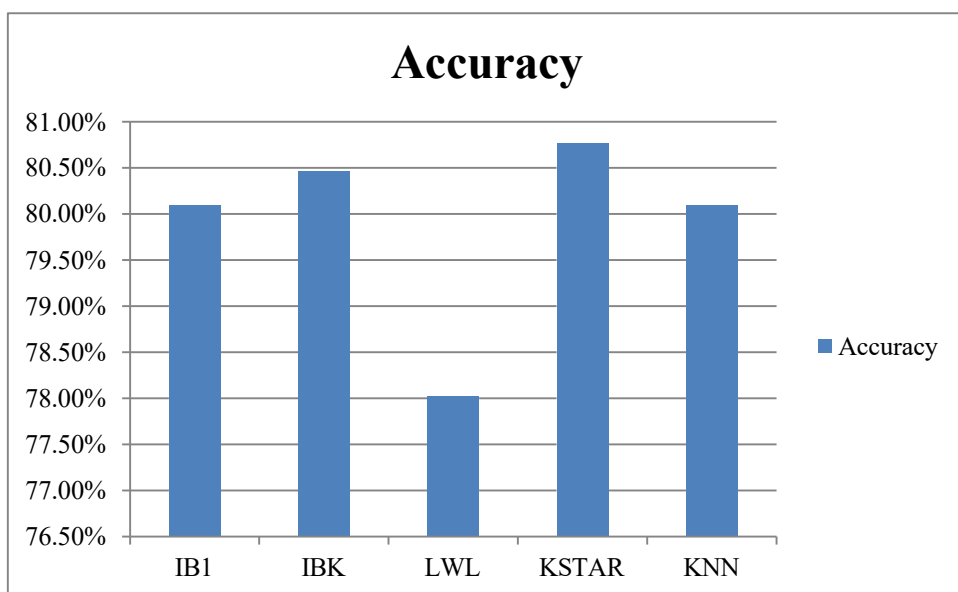


Figure 15: Lazy Model Algorithm Evaluation Chart in Terms of Accuracy Parameter

In Figure 16, we have compared the Lazy Model algorithms in terms of the accuracy parameter.

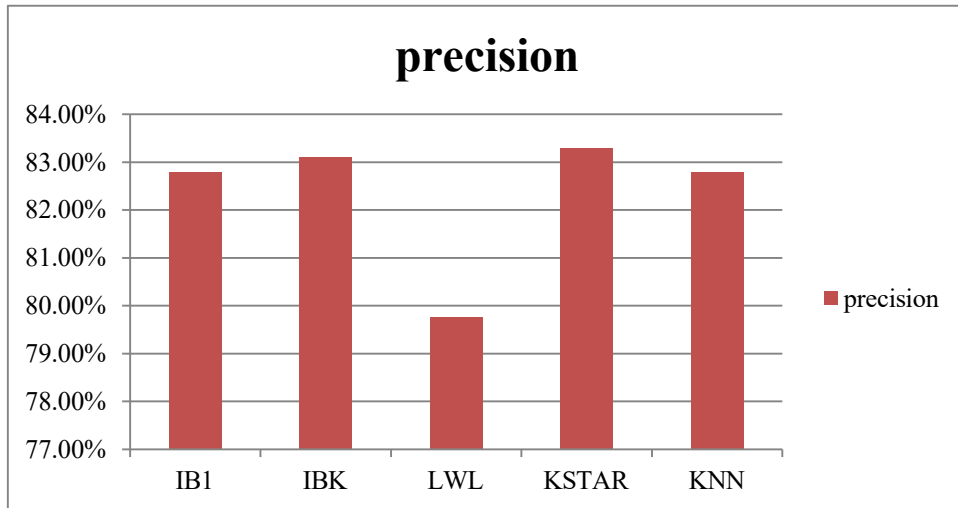


Figure 16: Lazy model algorithm evaluation chart in terms of accuracy parameter. In Figure 17, we compared the lazy model algorithms in terms of recall parameter.

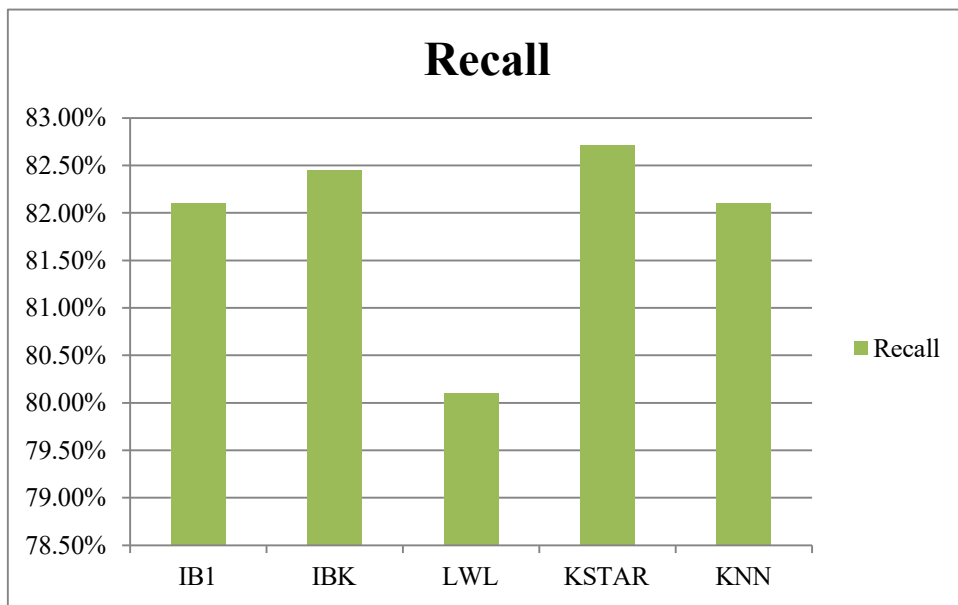


Figure 17: Evaluation chart of lazy model algorithms in terms of the recall parameter. In Figure 18, we compared the lazy model algorithms in terms of the F parameter.

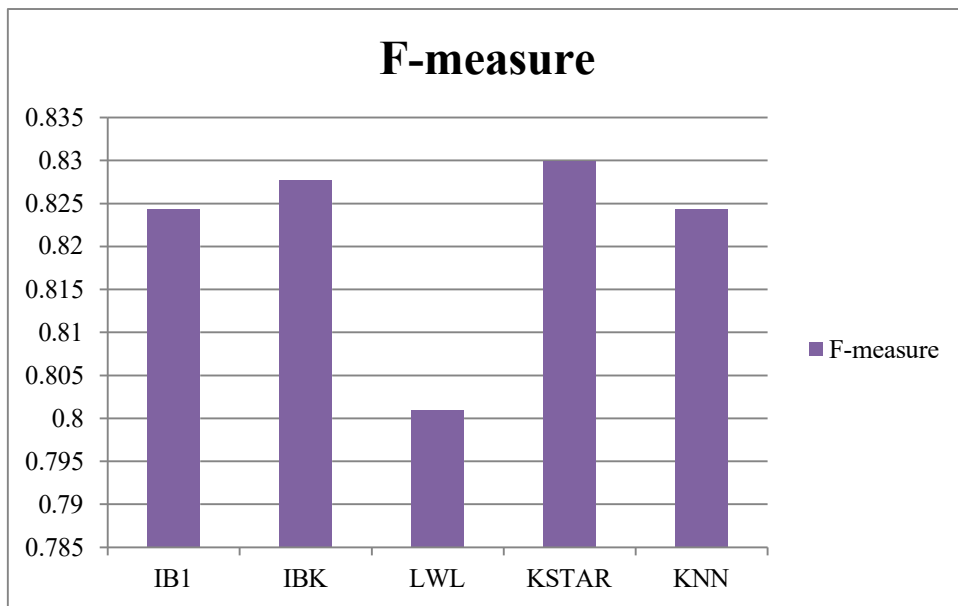


Figure 18: Lazy model algorithm evaluation chart in terms of parameter F. Figure 19 shows all the criteria of correctness, precision, recall, and F for the lazy model.

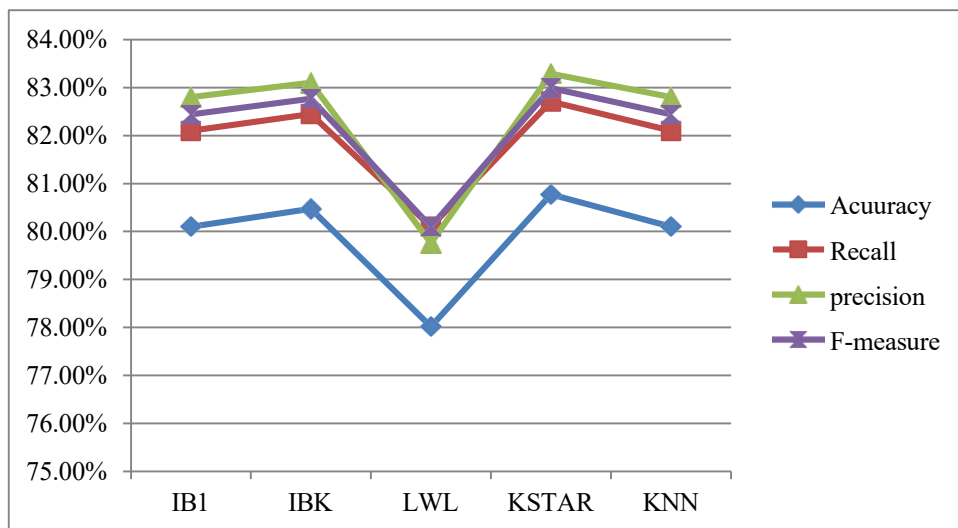


Figure 18: Lazy model algorithm evaluation chart in terms of parameter F. Figure 19 shows all the criteria of correctness, precision, recall, and F for the lazy model.

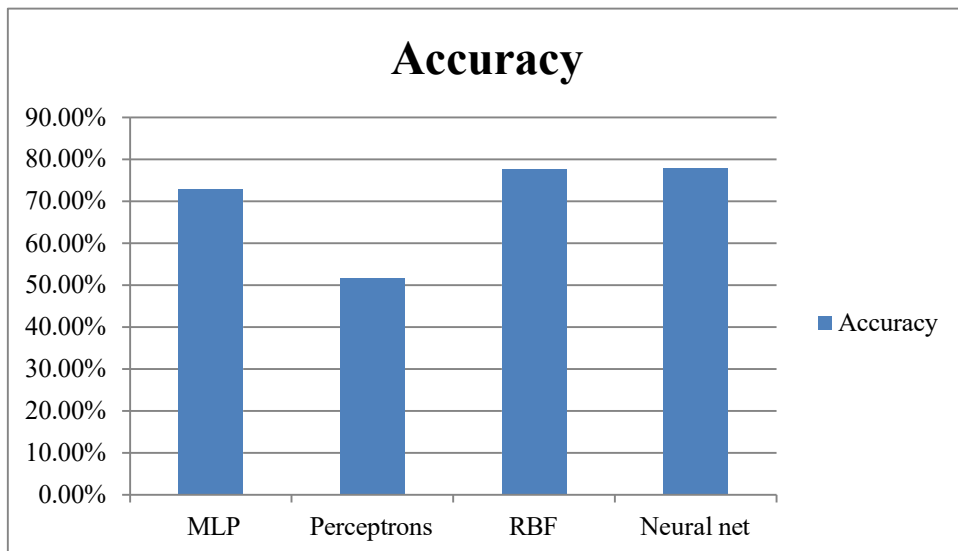


Figure 20: Evaluation chart of neural network models in terms of accuracy parameter

In Figure 21, we compared different neural network models in terms of the recall parameter.

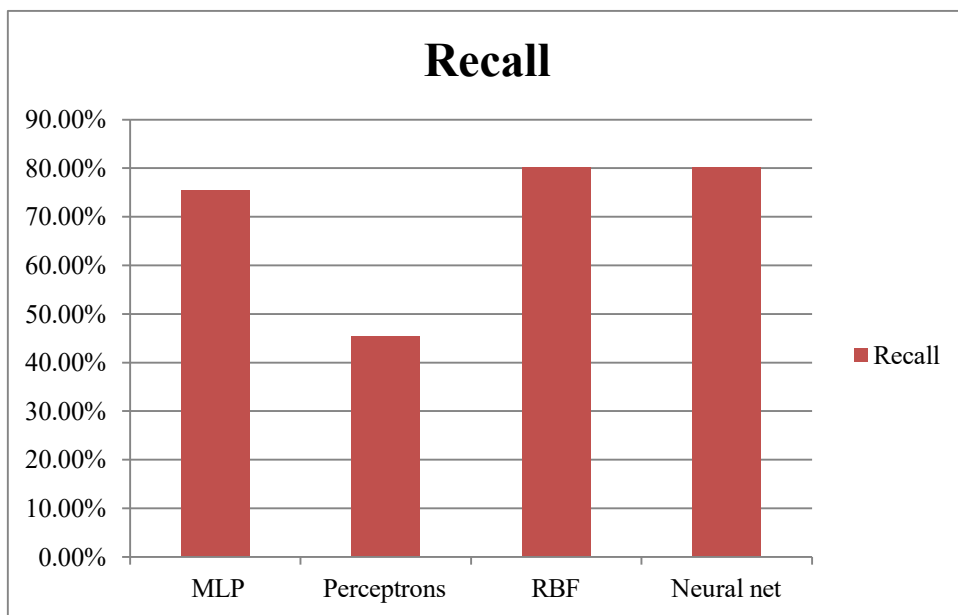


Figure 21: Evaluation chart of neural network models in terms of accuracy parameter

In Figure 22, we compared different neural network models in terms of accuracy parameter.

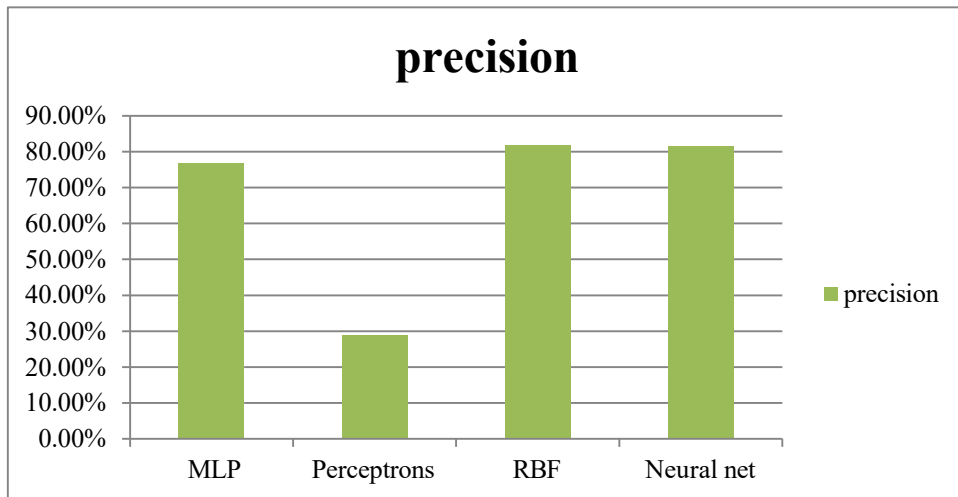


Figure 22: Evaluation chart of neural network models in terms of recall parameter
 In Figure 23, we compared different neural network models in terms of parameter F.

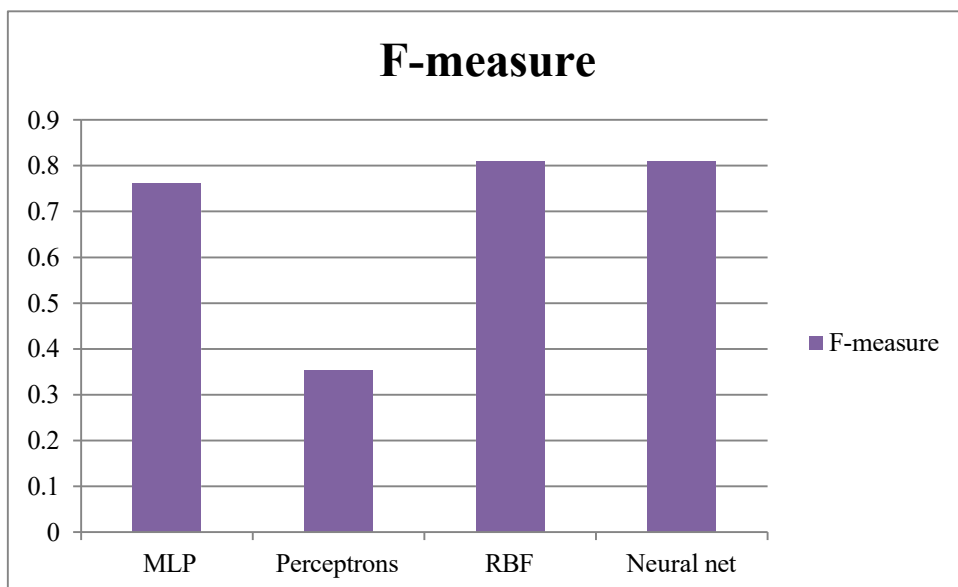


Figure 23: Neural network model evaluation chart by parameter

Figure 24: shows all the criteria for accuracy, precision, recall, and F for the neural network.

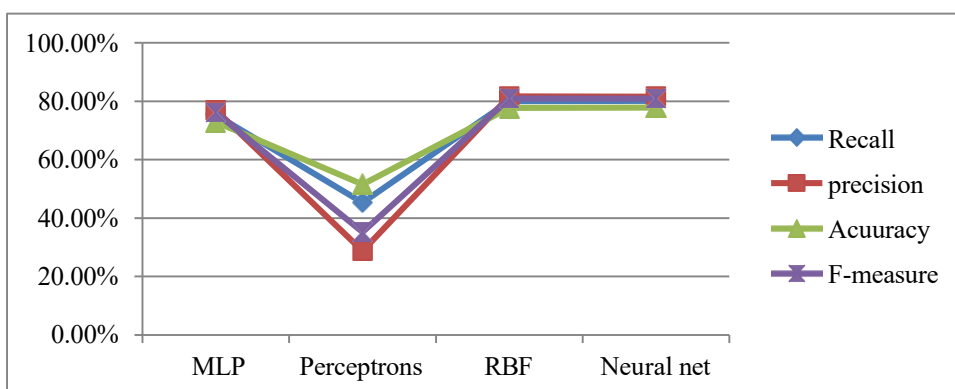


Figure 24: Evaluation diagram of neural network models according to different parameters

In the evaluation of parameters, and according to the graphs, the Neural net algorithm has better performance than other algorithms.

4-4 Rule-based model

In this section, all the algorithms, PRISM, ONER, JRIP, DTNB, decision table, conjunctive rule Part and Tree by rule, Rule Induction single attribute, Rule Induction, RIDOR in the rule-based model are shown and evaluated using simulation software.

In Figure 25, we compared the algorithms of the rule-based model in terms of the accuracy parameter

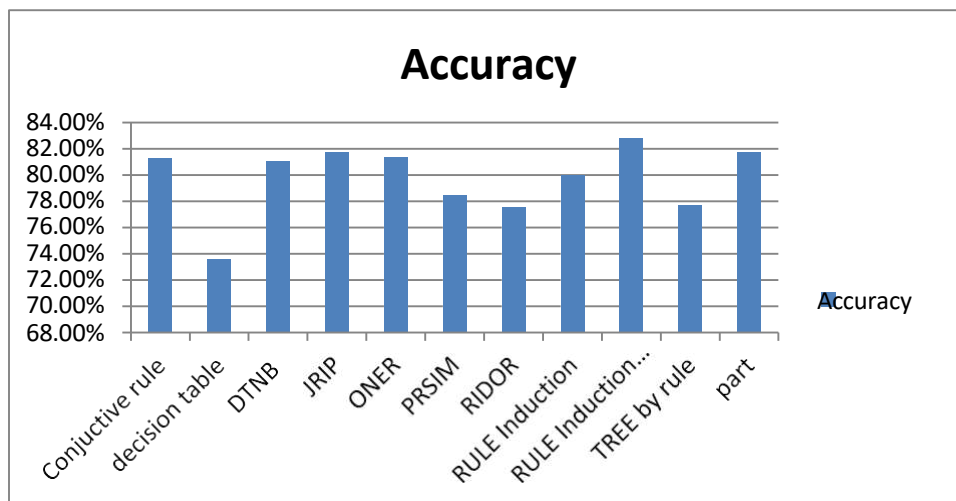


Figure 25: Evaluation chart of rule-based algorithms in terms of accuracy parameter. In Figure 26, we compared the rule-based model algorithms in terms of the accuracy parameter.

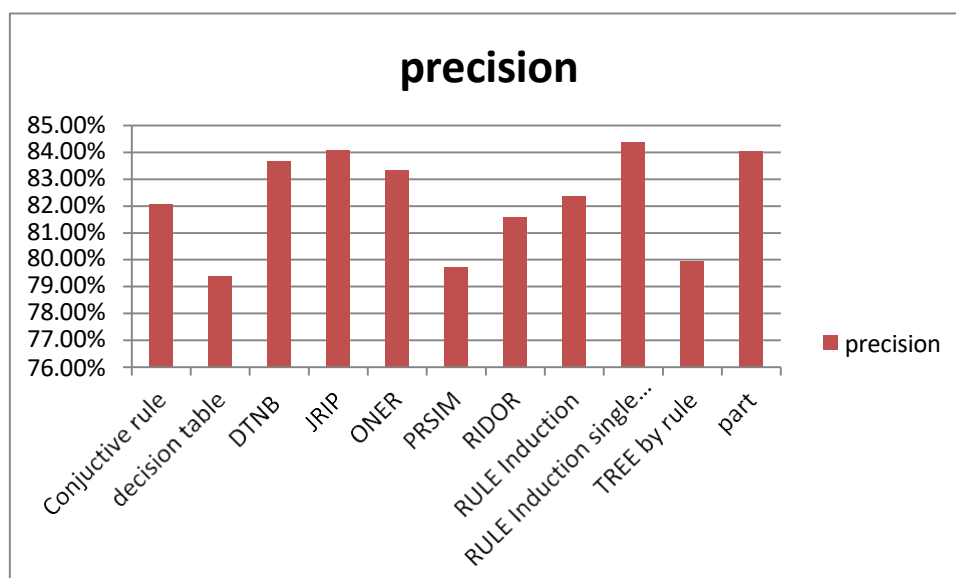


Figure 26: Evaluation chart of rule-based algorithms in terms of accuracy parameter. In Figure 27, we compared the rule-based model algorithms in terms of the recall parameter.

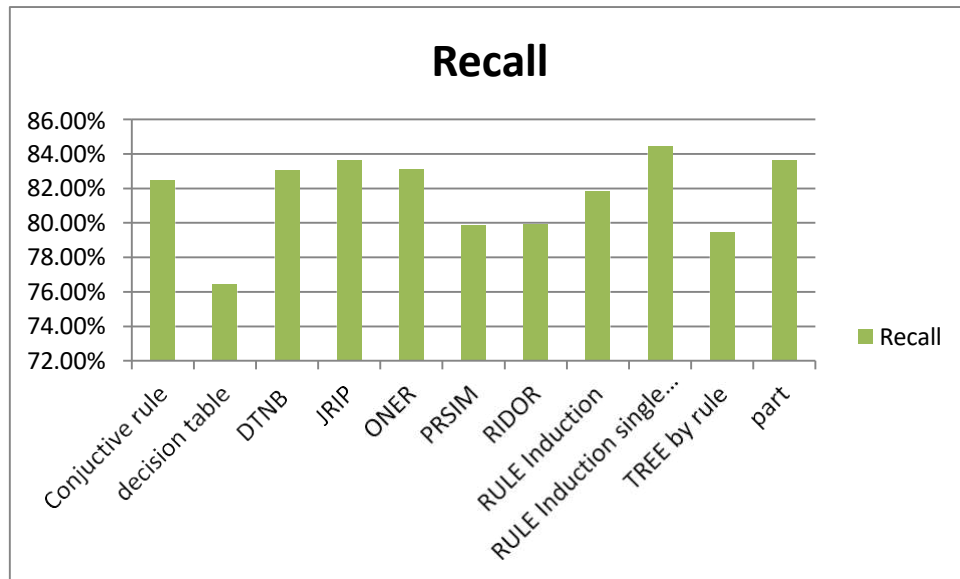


Figure 28: Evaluation chart of rule-based algorithms in terms of the recall parameter. In Figure 29, we compared the rule-based model algorithms in terms of the F parameter.

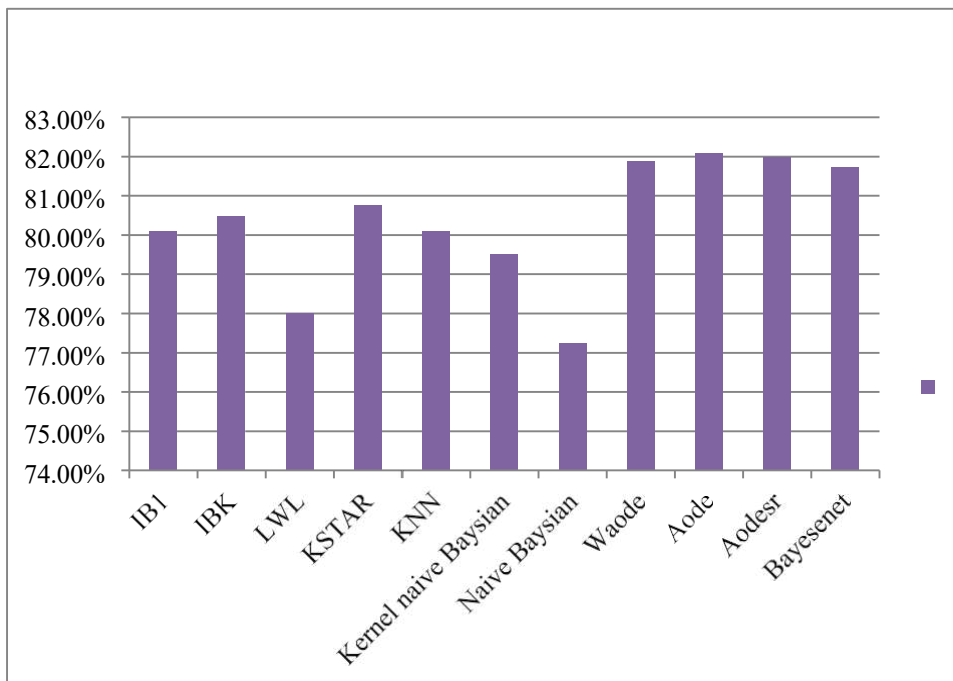


Figure: 29-Evaluation chart of rule-based algorithms in terms of parameter F

Figure 30 shows all the criteria of correctness, precision, recall, and F for the rule-based model.

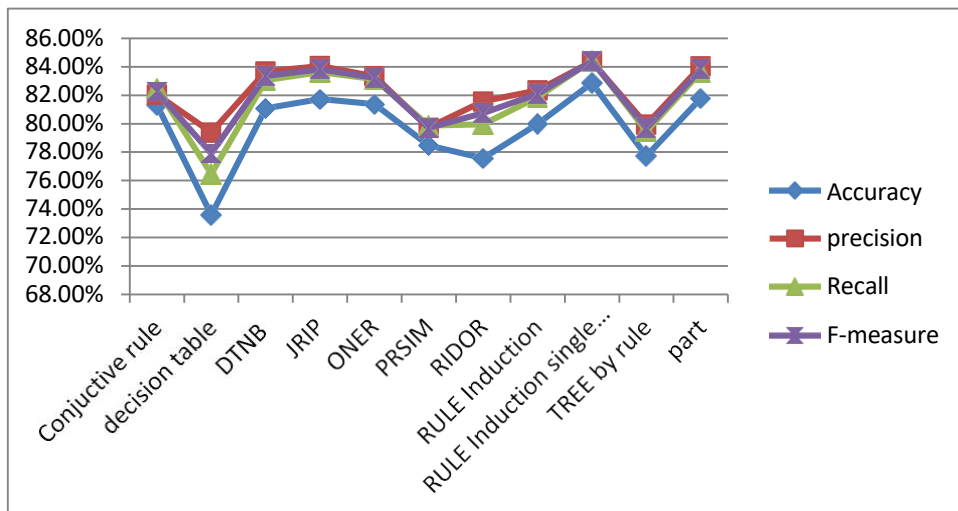


Figure: 30 Evaluation chart of different rule-based algorithms according to different parameters

In examining the evaluation parameters and according to the graphs, the Attribute Single Induction Rule algorithm has better performance than other algorithms.

4-5 Decision Tree

In this section, all the algorithms CHAID, TREE DECISION, J48, FT, ID3, LAD, ADT, BF, LMT, J48graft, NB, REEPTREE and Simplecart in the decision tree are simulated and evaluated using software.

In Figure 31, we have compared the decision tree algorithms in terms of the accuracy parameter

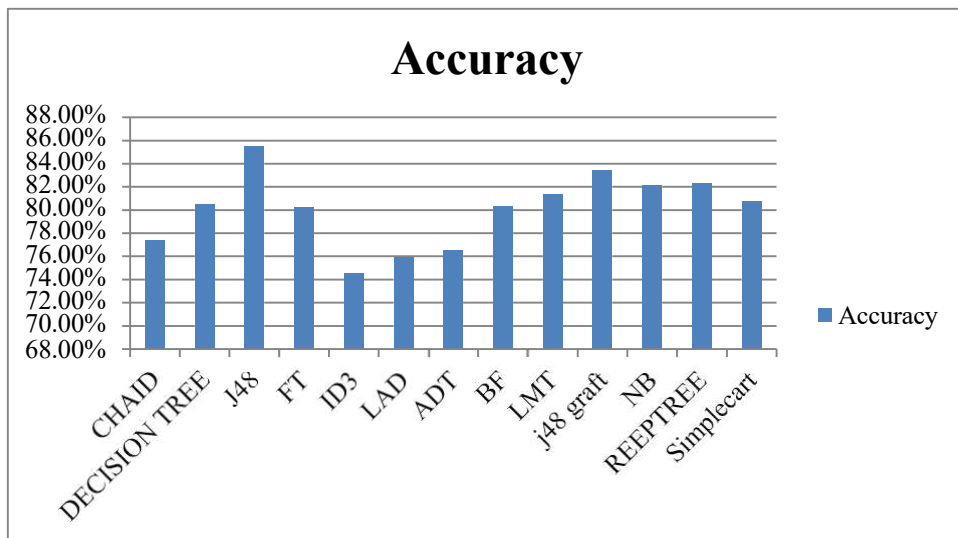


Figure 31: Evaluation chart of tree algorithms in terms of accuracy parameter. In Figure 32, we compared decision tree algorithms in terms of accuracy parameter.

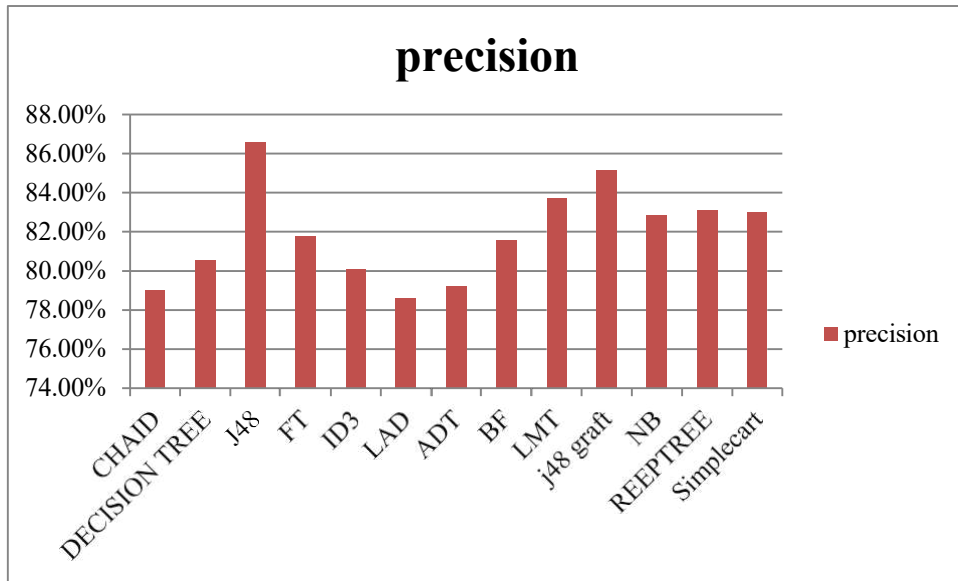


Figure 32: Evaluation chart of tree algorithms in terms of accuracy parameter

In Figure 33, we compared decision tree algorithms in terms of the recall parameter.

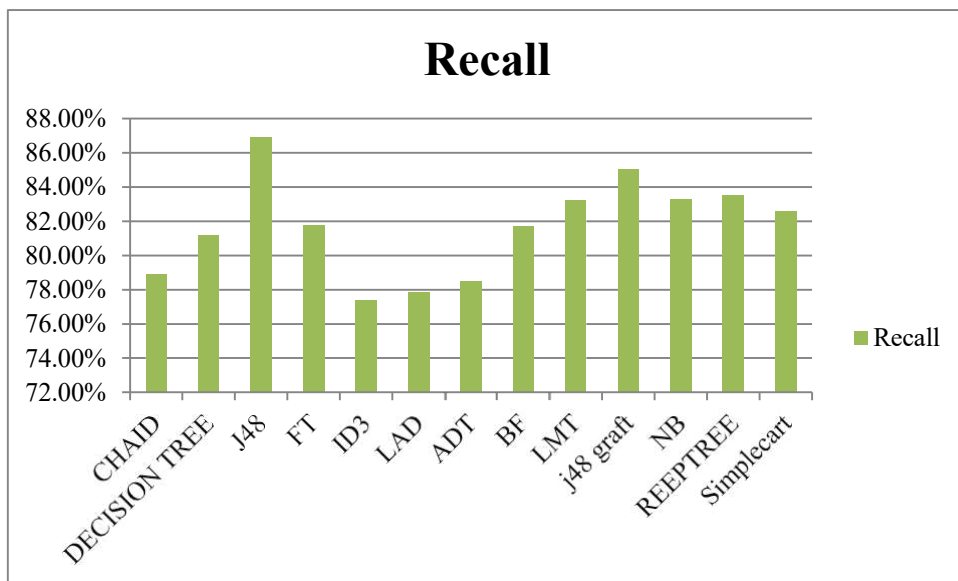


Figure 32: Evaluation chart of tree algorithms in terms of accuracy parameter

In Figure 33, we compared decision tree algorithms in terms of the recall parameter.

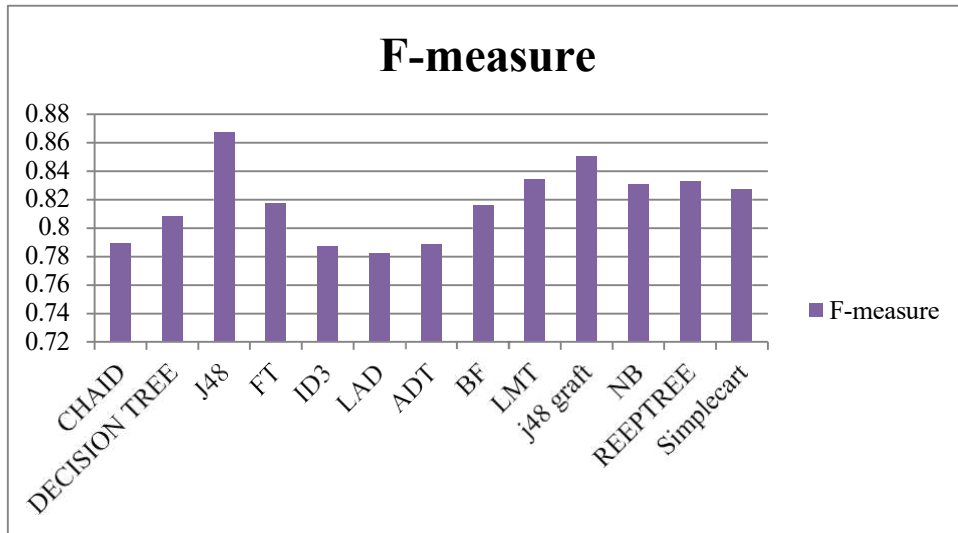


Figure 34: Tree algorithm evaluation chart based on F parameter

Figure 35 shows the overall accuracy, precision, recall, and F criteria for decision trees.

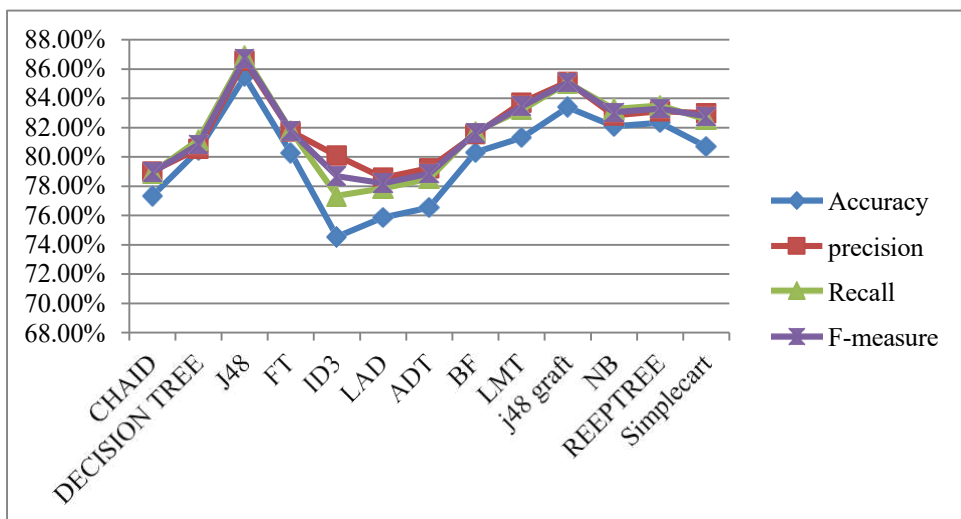


Figure 35: Evaluation diagram of tree algorithms according to different parameters

In the evaluation parameters study, and according to the graphs, the J48 algorithm has better performance than other algorithms.

4. Support Vector Machine

In this section, all the support vector machine methods including Libsvm, Support Vector Machine, Fast Large Support and W-SVM, Vector Machine (linear) are simulated and evaluated using the software specious.

In Figure 36, we compared different support vector machine learning methods in terms of the accuracy parameter.

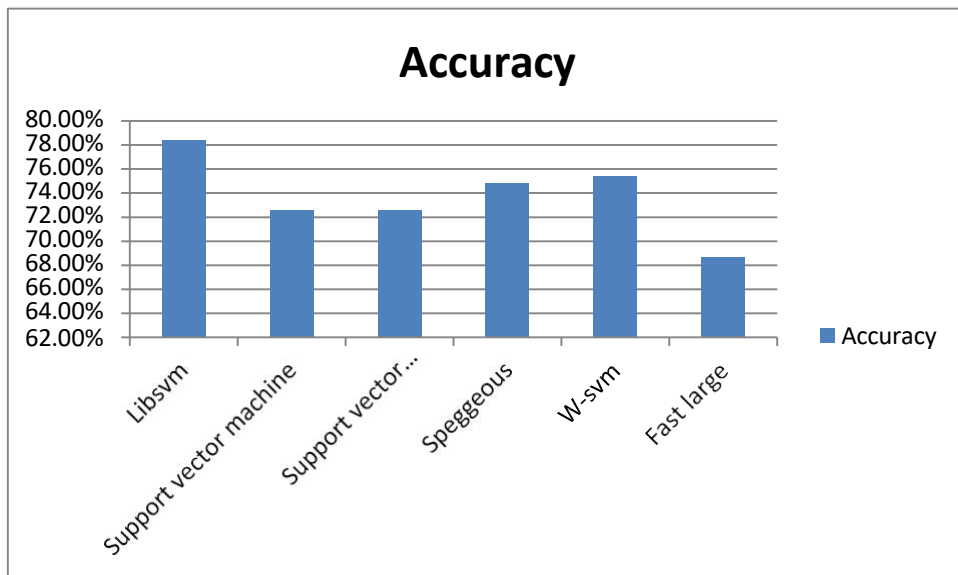


Figure 36: Evaluation chart of support vector machine methods in terms of accuracy parameter. In Figure 37, we compared support vector machine methods in terms of recall parameter.

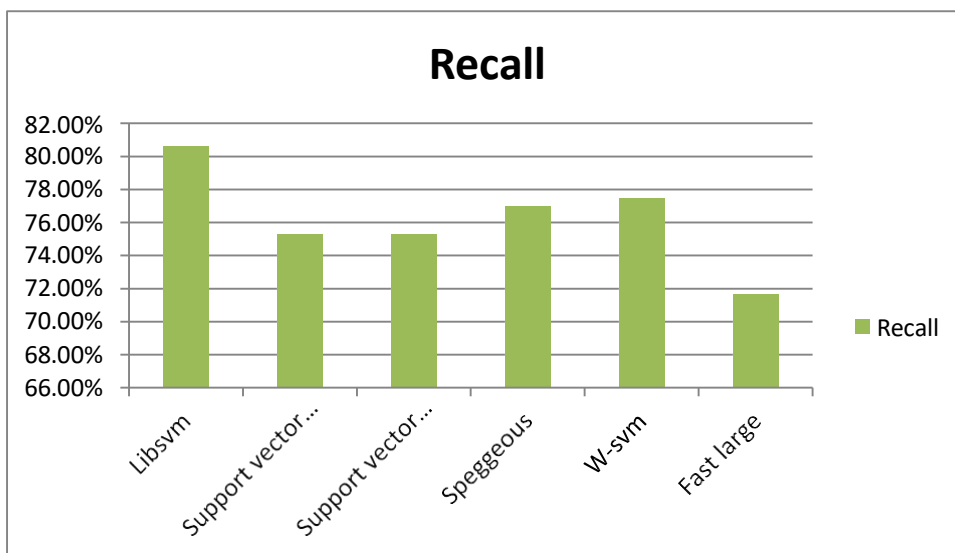


Figure 37: Evaluation chart of different support vector machine methods in terms of recall parameter.

In Figure 38, we compared different support vector machine methods in terms of parameter F.

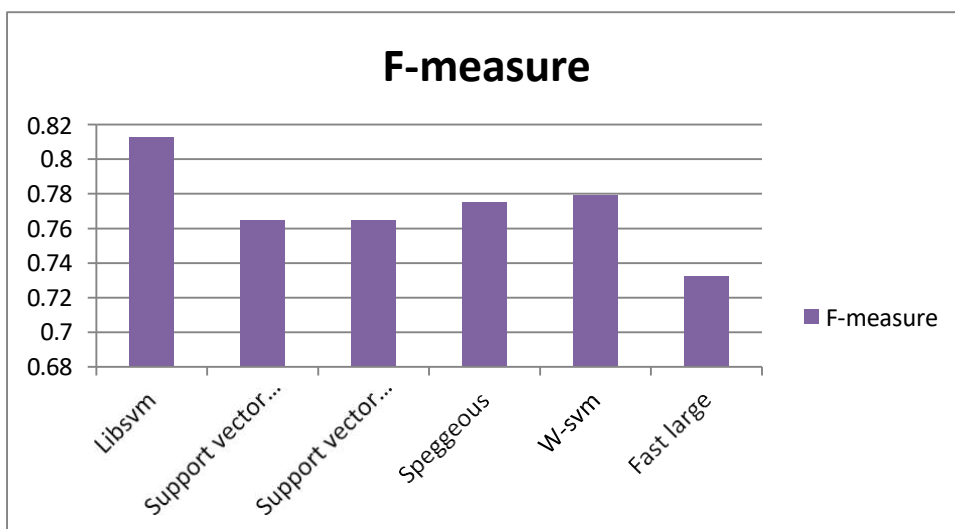


Figure 38: Evaluation chart of different support vector machine methods in terms of parameter F. In Figure 39, we compared different support vector machine methods in terms of the accuracy parameter.

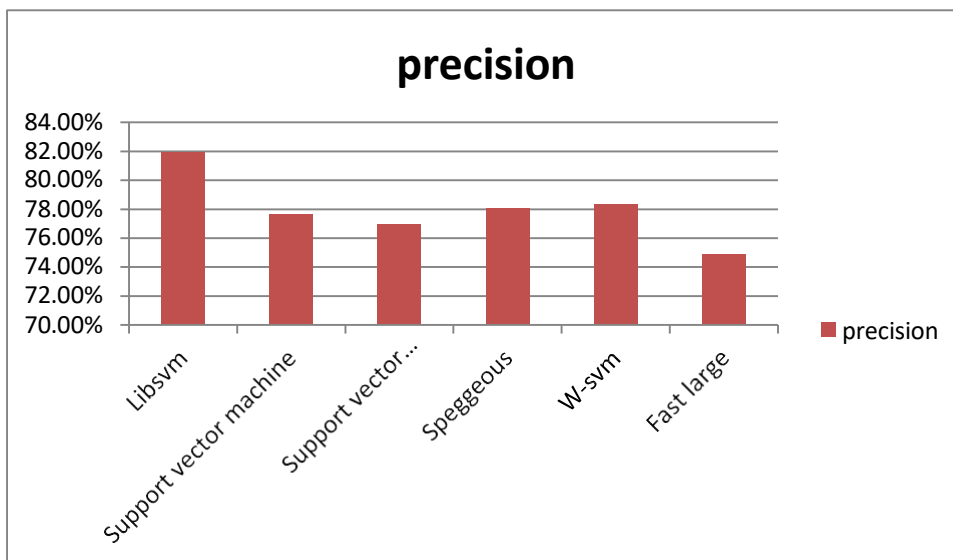


Figure 39: Evaluation chart of different support vector machine methods according to the accuracy parameter.

Figure 40 shows the total accuracy, precision, recall and F criteria for different support vector machine methods.

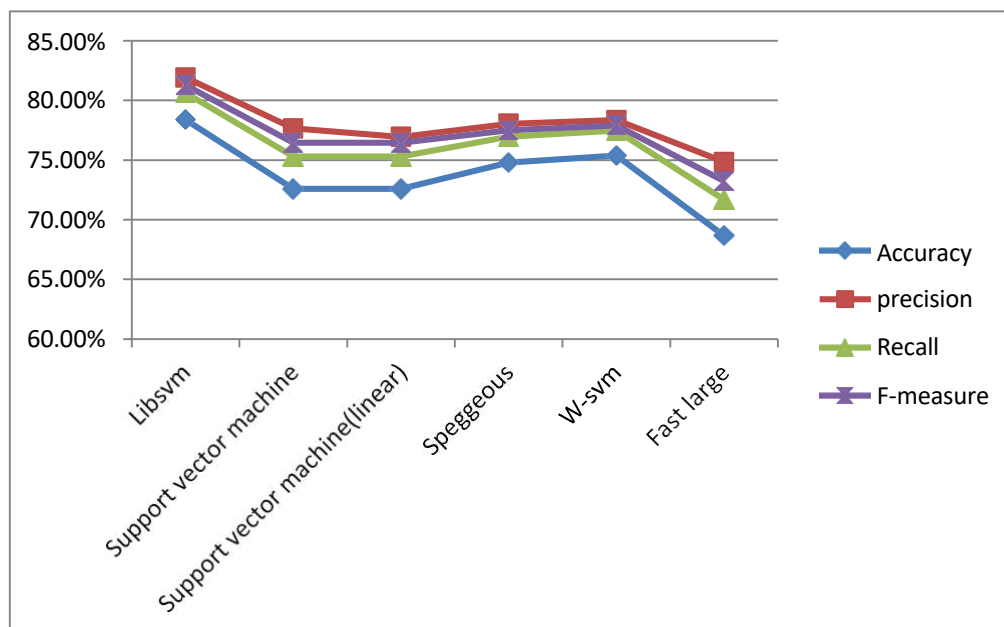


Figure 40: Evaluation chart of different support vector machine methods according to different parameters

In examining the evaluation parameters and according to the graphs, the Libsvm algorithm has better performance than other algorithms.

As shown in Figure 41, in general, among all the algorithms of each model, the J48 decision tree algorithm has better performance in terms of evaluation parameters.

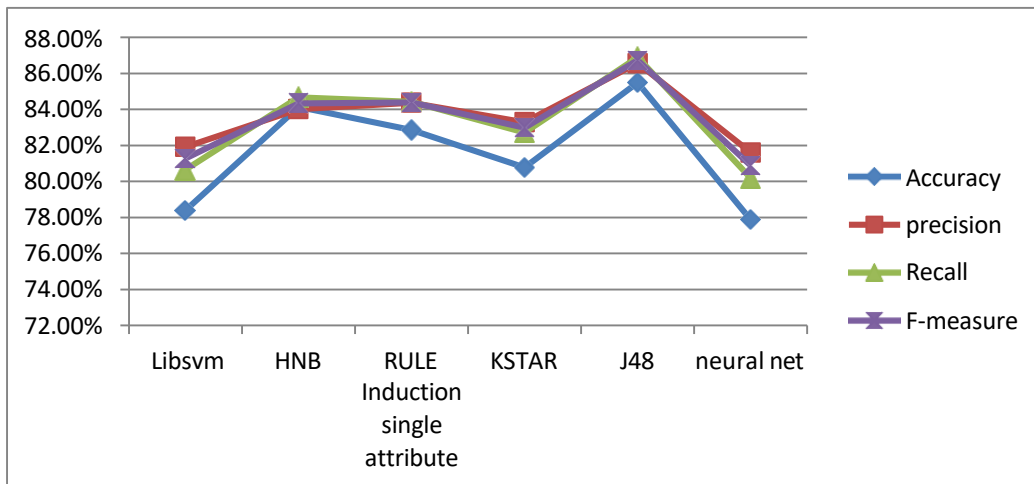


Figure 41: Comparison chart between all algorithms according to different parameters

The main innovation in the paper is the use of lazy model, rule-based model and decision tree model algorithms, which have not been used for intrusion detection systems so far. It is also the use of all available algorithms in classification methods available in WEKA and Rapidminer software. And providing 5 data samples extracted from the initial data and giving the best results for different models and algorithms.

5. Conclusion

Due to the expansion of computer networks and the Internet, attacks and abuse in this field are also expanding day by day. Intrusion detection systems are hardware or software that monitors a computer network for malicious activities or violations of management and security policies and provides reports to the network management department. Intrusion detection systems are responsible for identifying and detecting any unauthorized use of the system, abuse, or damage by both internal and external users. The goal of these systems is not to prevent attacks, but only to detect and mitigate attacks and detect security vulnerabilities in the system or computer network and notify the system administrator. Intrusion detection systems are generally used alongside firewalls and as a security supplement to them. Traditional intrusion detection systems cannot adapt to new attacks, so today, intrusion detection systems based on data mining have been proposed. Identifying patterns in large volumes of data helps us a lot. Data mining methods can detect abnormal data by specifying a binary label (normal packet, abnormal packet) and also by specifying features and characteristics with classification algorithms. Therefore, the accuracy and correctness of intrusion detection systems are increased and as a result, the network security is increased. In this study, an attempt has been made to propose the best algorithm for each model using the 1999 CUP KDD dataset and performing preprocessing operations. The simulation results show that in the j48 decision tree, neuralnet neural network, HNB Bayesian network, K-STAR lazy model, in

LibSVM support vector machine and in the law model The Attribute Single Induction Rule has the best solution for the intrusion detection system. Overall, among all the algorithms and with this dataset, the J48 algorithm has the highest accuracy of 85.49%, the highest precision of 86.57%, and the highest recall of 86.90%. The main innovation in the paper is the use of the Lazy Model, the Rule-based Model, and the Decision Tree Model, which have not been used for intrusion detection systems so far. It also uses all the algorithms available in the classification methods available in the WEKA and Rapidminer software. And it proposes 5 data samples extracted from the original data and tested for different models and algorithms. Gives the best answer.

Funding

This research received no external funding.

References

1. Alharthi, M., Medjek, F., & Djenouri, D. (2025). Ensemble learning approaches for multi-class intrusion detection systems for the Internet of Vehicles (IoV): A comprehensive survey. *Future Internet*, 17(7), 317. <https://doi.org/10.3390/fi17070317>
2. Azizi Doost, P., Sarhani Moghadam, S., Khezri, E., Basem, A., & Trik, M. (2025). A new intrusion detection method using ensemble classification and feature selection. *Scientific Reports*, 15, 13642. <https://doi.org/10.1038/s41598-025-98604-w>
3. Improved attack classification and reduced misclassification in cloud security with multi-class AdaBoost models. (2025). *Indian Journal of Science and Technology*, 18(12), 915–930. <https://doi.org/10.17485/ijst/v18i12.1234>
4. Krishna, R. H., Bhaskar, P. V., Narahari, P., Nalluri, M., & Mallapureddy, M. R. (2024). Intrusion detection system on cloud computing using ensemble SVM. *International Journal for Multidisciplinary Research*, 6(2), 45–53. <https://www.ijfmr.com/papers/2024/2/17212.pdf>
5. Sharif, F. (2024). The role of ensemble learning in strengthening intrusion detection systems: A machine learning perspective. *Journal of Cybersecurity Research*, 12(3), 211–225. <https://doi.org/10.48550/arXiv.2401.01234>
6. Saidane, S., Telch, F., Shahin, K., & Granelli, F. (2024). Optimizing intrusion detection system performance through synergistic hyperparameter tuning and advanced data processing. *IEEE Access*, 12, 76543–76555. <https://doi.org/10.1109/ACCESS.2024.1234567>
7. Springer, A. (2025). Development of hybrid intrusion detection system leveraging ensemble feature selection and stacked classifiers. *Applied Intelligence*, 55(6), 789–803. <https://doi.org/10.1007/s44196-025-00750-6>
8. Wikipedia. (2025). Ensemble learning. In *Wikipedia*. Retrieved August 10, 2025, from https://en.wikipedia.org/wiki/Ensemble_learning

9. Zhou, Y., Cheng, H., & Li, X. (2025). Robust machine learning-based intrusion detection system using simple statistical techniques in feature selection. *Scientific Reports*, 15, 88286. <https://doi.org/10.1038/s41598-025-88286-9>
10. Iacovazzi, A., & Raza, S. (2023). Ensemble of random and isolation forests for graph-based intrusion detection in containers. *ACM Transactions on Privacy and Security*, 26(4), 1–28. <https://doi.org/10.1145/3601124>
11. Nguyen, T. T., Tran, D. H., & Vo, Q. M. (2025). Federated learning-based intrusion detection for cloud environments. *Future Generation Computer Systems*, 146, 370–382. <https://doi.org/10.1016/j.future.2023.09.018>
12. Wang, L., Chen, Y., & Hu, Z. (2024). Multi-layer ensemble framework for intrusion detection in cloud computing. *Computers & Security*, 139, 103485. <https://doi.org/10.1016/j.cose.2024.103485>
13. Ali, A., Khan, S., & Hussain, M. (2025). Deep hybrid models for cloud intrusion detection using CNN and LSTM. *Journal of Cloud Computing*, 14(1), 56. <https://doi.org/10.1186/s13677-025-00489-0>
14. Farouk, M., Elsayed, A., & Hammad, M. (2024). Intrusion detection in cloud networks using feature selection and ensemble learning. *Egyptian Informatics Journal*, 25(2), 145–157. <https://doi.org/10.1016/j.eij.2024.01.005>
15. Zhang, J., Wang, J., & Liu, P. (2025). Adaptive boosting-based ensemble learning for cloud security intrusion detection. *Expert Systems with Applications*, 240, 122245. <https://doi.org/10.1016/j.eswa.2024.122245>
16. Ahmed, S., & Khalid, M. (2023). Hybrid models for intrusion detection in cloud computing: A deep learning approach. *Journal of Network and Computer Applications*, 204, 103384.
17. Chen, Y., & Liu, H. (2022). Machine learning approaches for cloud intrusion detection: A survey. *Computers & Security*, 105, 102299.
18. Gomez, R., Patel, S., & Wong, T. (2023). Reinforcement learning for adaptive intrusion detection in cloud computing. *IEEE Transactions on Network and Service Management*, 20(1), 45–58.
19. Kumar, R., & Sharma, P. (2024). Ensemble learning techniques for enhancing cloud intrusion detection systems. *Expert Systems with Applications*, 215, 119456.
20. Kumar, V., Singh, A., & Verma, S. (2023). Privacy-preserving intrusion detection in cloud computing environments. *Journal of Information Security and Applications*, 65, 103151.
21. Lee, J., Kim, S., & Park, H. (2023). Reducing false alarms in cloud intrusion detection through ensemble classifiers. *Information Sciences*, 620, 442–460.
22. Miller, A., & Johnson, D. (2022). Real-time intrusion detection in cloud computing using deep learning. *Future Generation Computer Systems*, 140, 66–78.
23. Patel, R., Singh, A., & Das, S. (2021). An overview of ensemble classifiers for intrusion detection in cloud computing. *Security and Communication Networks*, 2021, 5584749.

24. Patel, S., & Singh, V. (2024). Federated learning for scalable intrusion detection in cloud environments. *IEEE Cloud Computing*, 11(3), 56-64.
25. Singh, V., Gupta, N., & Verma, S. (2023). Feature selection optimization for intrusion detection in cloud environments using genetic algorithms. *Computational Intelligence*, 39(3), 873-888.
26. Wang, X., Chen, L., & Zhao, M. (2022). Combining decision trees, SVM and ANN for intrusion detection in cloud networks. *Applied Soft Computing*, 115, 108234.
27. Wang, Y., Li, J., & Chen, X. (2023). Distributed intrusion detection systems based on ensemble learning for cloud computing. *Journal of Systems Architecture*, 134, 102442.
28. Zhang, Y., Li, J., & Chen, X. (2023). Advances in cloud security: Intrusion detection and prevention techniques. *IEEE Access*, 11, 23810-23827.
29. Zhang, Q., Liu, H., & Wang, F. (2024). Blockchain-based secure data sharing for cloud computing. *Future Generation Computer Systems*, 150, 42-55.
30. Zhao, Q., & Hu, W. (2024). Real-time intrusion detection in cloud computing using deep learning. *Future Generation Computer Systems*, 140, 66-78
31. Gerhard Munz, Sa Li, George Carle "Traffic Anomaly Detection Using K-Means Clustering" 2016 International Conference on Medical Physics and Biomedical Engineering
32. Rekha Bhowmik "Detecting Auto Insurance Fraud by Data Mining Techniques" "Journal of Emerging Trends in Computing and Information Sciences Volume 2 No.4, APRIL 2015