JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 08 Issue 01
2024

Research Article
https://doi.org/10.56979/801/2024

Assessing the Effectiveness of Ensemble Learning Models for Hepatitis C Detection through Advanced Machine Learning Techniques

Fatima Zafar[1], Syed Muhammad Junaid Zaidi[1], Muazzam Ali[1*], M U Hashmi[1], Muhammad Azam[2], and Suman Arshad[1]

[1]Deparment of Basic Sciences, Superior University, Lahore, 54000, Pakistan.
[2]Deparment of Computer Science, Superior University, Lahore, 54000, Pakistan.
[*]Corresponding Author: Muazzam Ali. Email: muazzamali@superior.edu.pk

Received: June 11, 2024 **Accepted:** September 29, 2024

Abstract: This paper investigates the potential benefits of utilizing advanced methods of machine learning to enhance Hepatitis C diagnosis tools. We used a publically available dataset to test different ensemble learning techniques, such as Grid Search and Random Search to optimize the parameters of Random Forest, Gradient Boosting, Bagging, XGBoost, and stacking. We evaluated the performance of the model using Cohen's Kappa, F1 score, accuracy, precision, and recall. With 92.37% accuracy, 83.85% precision, and a 70.17% F1 score, XGBoost with Random Search demonstrated the best performance. The results show that medical diagnostics can be improved and that methods of ensemble learning are useful for early Hepatitis C identification.

Keywords: Ensemble Methods; Hepatitis C Diagnosis; Random Forest Algorithm; Boosting Techniques; Bagging Approach; Hyperparameters Optimization; Grid Search Technique, Randomized Search Method; Evaluation Metrics for Classification; Machine Learning Applications in Healthcare.

1. Introduction

Around 71 million individuals worldwide are afflicted with hepatitis C, a severe global health concern. This virus causes harm to the liver, and if untreated, it can result in serious diseases like fibrosis and liver cancer. Early detection is essential because early symptoms are sometimes absent and many cases remain untreated until the condition has progressed. Ordinary blood testing may prove costly and lengthy for medical purposes, primarily for regions with limited funds [3-5]. Due to algorithms for learning (Machine learning (ML)), and the use of patient data investigation, hepatitis C is being identified faster, thoroughly, and cheaply. Improved results for patients in both evaluation and therapy may arise from initial infection risk prediction using deep learning algorithms. Numerous studies indicate that computerized learning is able to diagnosing Hepatitis C. Artificial neural networks, or ANNs for short, were used by in [6] attain 89% precision speeding up detection and eliminated the demand of several blood draws [7]. Support vector machines, or SVMs, were utilized to reach 87% reliability, and choice trees were applied [8] [9] to get 86% success. Another successful method is using one of multiple models in collaborative learning. With Random Forestry and gradient booster, a precision of 90% [10]. In [11] the XGBoost to achieve 91% efficiency. More valid and precise predictions, stack and wrapping increased the confirmation of the virus [12]. Despite prior work shows that data mining has potential for diagnosing illnesses, a few investigations looked at different group techniques for viral prediction. The majority of study focusses on single models to avoid combining models and improving efficiency with the hyper parameter modifications [13-16]. Additionally, advanced tuning techniques like Grid searches and Randomized Search tend to be ignored when conducting research [17-19]. This research analyses

the effectiveness of different group methods (Random Forest, who are currently Boosting Gradient, Bagging, which is a process called and Stacking) with optimum the hyperparameters with the aim to achieve this aim. The goals are to evaluate the consequences of a tuned hyper parameter and to determine the best ensemble plan for rapid virus diagnosis. The research we conducted provides thorough analyses of learning ensemble methods, which helps build a wealth of literature related to the diagnosis of hepatitis C [20-22]. Professional a hyper parameter altering techniques like Graph Search as well as Stochastic Searching were used to optimist the performance of the model. By a 70.17% F1 grade, 83.85% exactness, and 92.37% precision, we realized that the XGBoost together with Unexpected Search produced the highest quality results. This demonstrates the importance of a hyper parameter change and collaboration in developing accurate and reliable tools for diagnosing [23-25]. Our study offers a helpful basis to enhance diagnosis precision in medical services using data mining.

2. Materials and Methods

2.1. Dataset Explanation

The database used for the present research contains patient files with basic and medical data like age, and functioning of the liver test results. The purpose of this continuous classification issue is to predict if hepatitis C virus has occurred or not. You split the dataset into two sections—80% to train the models using machine learning and 20% for testing—in order ensure a correct evaluation.

2.2. Preprocessing Techniques

2.2.1. Preprocessing and Hyper parameter Tuning Methodology

A number of methods are used to process the hepatitis C infection dataset in this study to ensure it would be appropriate for models developed using machine learning. That meant that information had neatly structured and available for examination by collective methods for learning. Following the initial download of the data set, the other preprocessing phases finished.

2.2.2. Handling Missing Values

Whenever there lacks values, artificial intelligence methods may fail as well. To fix this, we eliminated any rows without empty content. This simple method ensured the model predictions were built on done, accurate information devoid of interference from features that were missing.

2.2.3. Feature Selection

It has been determined that the intended varying, which determines the presence of hepatitis C is present, is the "Category" row. Not-so-important entries that supplied no beneficial data, such "Unidentified: 0," were also eliminated. For choosing the traits (X) that were utilised for conditioning a model, the goal item (y), particularly displayed a liver cancer C assessment, was taken out.

2.2.4. Encoding Categorical Variables

Because many learning methods the need quantitative inputs, they converted category characteristics into numerical form. For categorical columns like “gender," we utilized single-hot encoding to translate binary values like Male and Female. The target value then label-encoded in order to convert categories into byte for identification. Afterwards, 20% of the dataset was put aside for testing, while the remaining 80% was applied for developing the algorithms. According to this section, we can assess the models on before unknown data, letting us to estimate their efficiency better.

2.3. Hyper parameter Tuning

To enhance the efficiency of predictive models, a hyper parameter altering is required for identifying the best parameter values. In this research, we used combined Grid Search along with Stochastic Search methods to optimize hyperparameters that obtaining the combination of hyper parameter combinations that produce the best results across every system is the purpose of these methods.

2.3.1. Grid Search

Searching for Grids methodically examines at all possible combinations of set hyperparameters that with each model, they set up a variety of settings, such the number of estimation tools, train rate, and tree thickness.

Then, the grid search process looked at every potential combination to find the one that functioned most effectively.

2.3.2 Random Search

Unique Search selects an unexpected assortment of parameters to test, as compared to Panel Research's thorough effort on every possible combination. This improves its effectiveness and rapidity through allowing it to handle a larger range of potential hyperparameters that with less processing power. In this work, Random Search was implemented to improve model performance while conserving funds.

2.4. Selected Ensemble Techniques

2.4.1. Random Forest

The Random Forest technique produces an abundance of tree of choices during development. Every tree is generated using an arbitrary portion of the data, and the predicted outcome is the sum of the entire trees' estimates (for regression) or a single most common estimate (for classifications). The method improves the reliability of the predictive model and avoids excess fitting by bringing together result data from many distinct trees. The random forest method performs excellently with unpredictable data that is a strong fit for large datasets with many modeling characteristics [1].

.

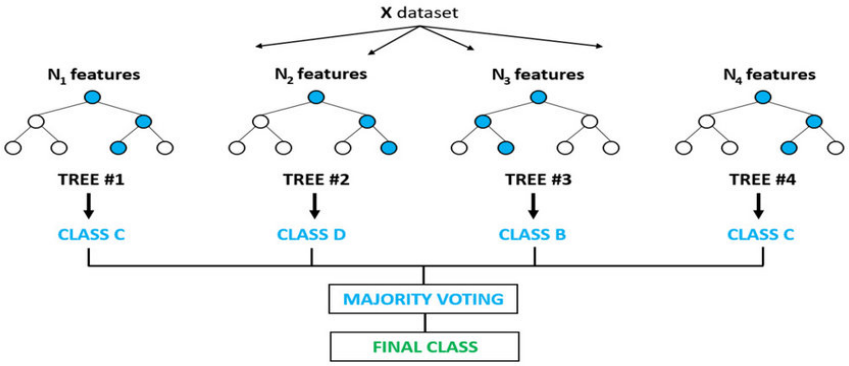

Figure 1. RF classes

2.4.2. Gradient Boosting

Gradient boost is a modelling method that builds ones one after another, with each model seeking to work on the flaws of earlier models. This method can be especially helpful for reducing errors where high accuracy is required or where collections are unbalanced. The gradient booster produces a series of tree-like decisions, every of whom focusses on repairing the errors made in the prior one. This ends up in an extraordinarily exact and sturdy model, given its disadvantage the process may demand greater over some other methods [2, 3].

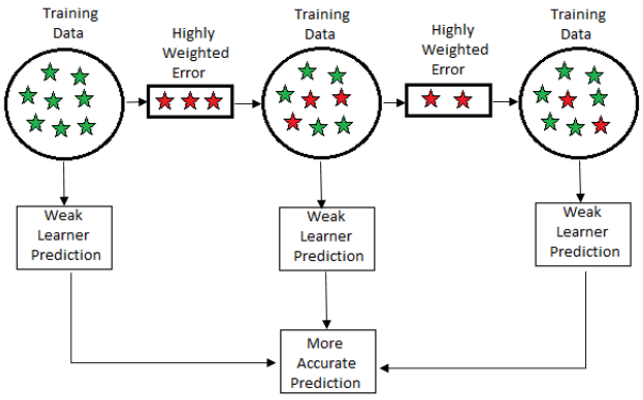

Figure 2. Gradient boost predictor

2.4.3. Bagging

A technique called "packing," or "Bootstrap, also combining," generates multiple algorithms by learning each of them on an individual portion of the data. To arrive at a final forecasting, it picks its most frequent projection for classification tasks or averages the forecasts for regression-related tasks. Carrying helps keep the algorithm form over fitting the data through decreased predicted variations. If seeking an extra reliable model—especially since it shifts a lot—this methodology can be very valuable.

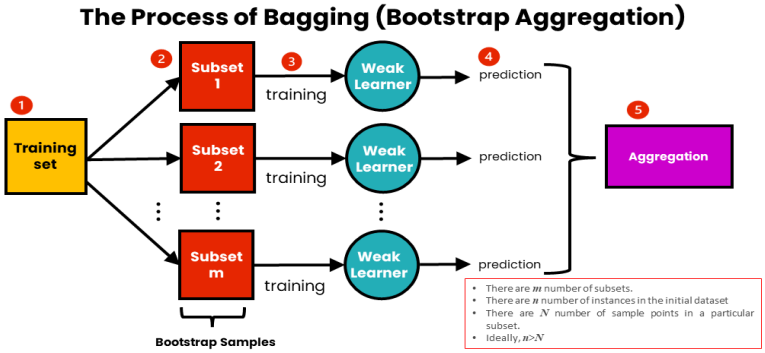

Figure 3. Bagging Process

2.4.4. XGBoost

Extreme Grade Boosting, or the XGBoost is a more rapid and more successful form of gradient boost. It utilises unique methods which prevent excessive fitting and produce accurate projections. XGBoost speed lies in its capacity to deal with large datasets in tandem. Its outstanding efficiency and ability to handle difficult classification tasks make it prominent in neural network tournaments [26, 27].

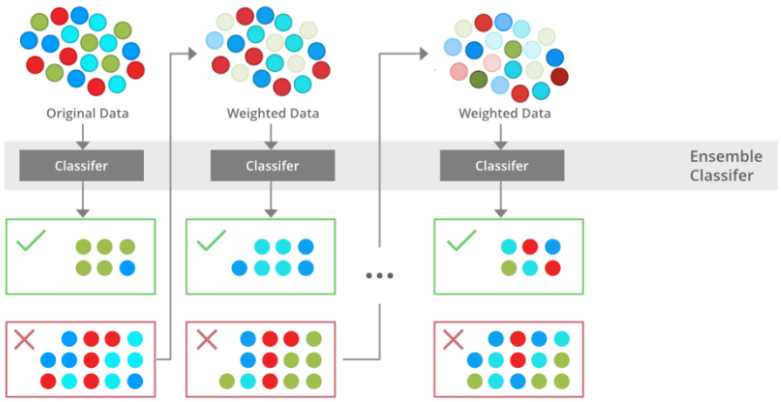

Figure 4. Ensemble Classifier

2.4.5 Stacking

To get the ultimate anticipate, the projections from different theories, namely Random Forests, gradient booster, and the case of X are mixed together using a procedure termed piling. From the predictions of these models, a less essential model like the logistic regression is used to calculate the endpoint. By utilising each predictor's abilities stacking raises correctness and improves the average validity of results versus using only one version [28-30].

2.5. Evaluation Metrics

2.5.1. Accuracy

The basic accuracy metric indicates what percentage of each of the guesses were true. In order to provide a broad sense of how effectively the composite conduct detected those with Hepatitis C, repeatability was assessed in this research.

2.5.2. Precision

The number of instances of the optimistic forecasts which manifest is measured by clarity. In instances like that, where the price of inaccurate diagnoses is substantial, it's important that we avoid misunderstanding patients in good health with the viral infection hepatitis C. An grew error reduction in the model can be observed by higher levels of accuracy.

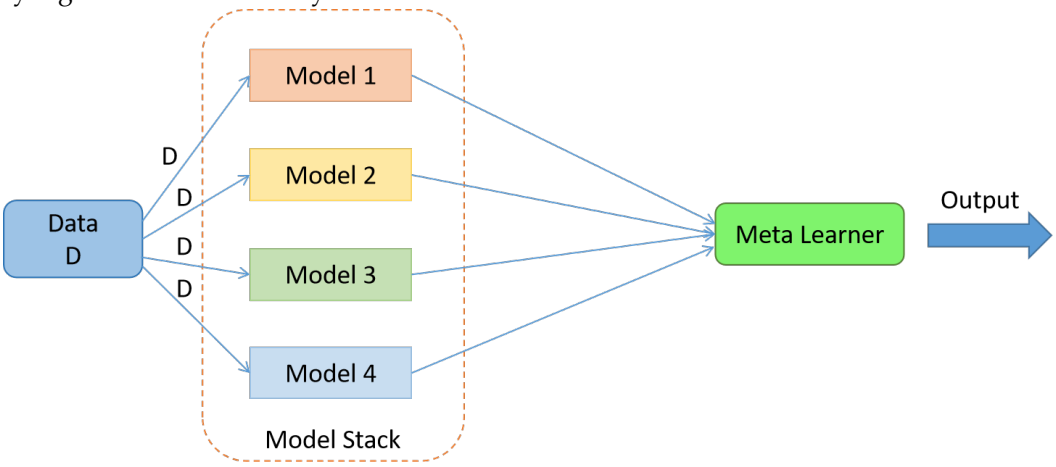

Figure 5. Meta Learner Model Stack

2.5.3. Recall

Sensitivity is additionally known as recall, and recalls counts the proportion of true positive results the the hypothesis was accurately identified. In the scenario of virus detection, recall examines a model's capacity for recognizing real cases of the condition. Strong remember is crucial to make sure that just as many sick people may receive an appropriate diagnosis.

2.5.4. F1 Score

The F1 Rating, essentially calculates the harmonic mean of both recall and precision, maintains a balance amongst the two distinct the units of measurement. Whether precision and recall have equal importance or if it is an uneven class shipment, it is quite helpful. A solid F1 score highlights both the sensitivity and the accuracy of the model, thus serving as an excellent gauge to determine the total efficacy of the classification system.

2.5.5. Cohen's Kappa

The Kappa coefficient of Cohen evaluates the degree of concurrence amongst the actual with expected labels without incorporating accidental concurrence. Since it takes to account both likelihood agreements and preciseness, this indicator is extremely successful if applied on datasets that are not balanced. Higher numbers coincide with stronger concordance between the expected or real labels.

2.5.6. Hamming Loss

The hamming Loss evaluates the amount of wrong predictions the model generates.

This serves as a highly helpful fraction for estimating the fraction of wrongly anticipated labels on multi-label classifying issues. When screening viral infections, a lower Hamming in Loss signals superior efficiency of the classifier.

2.6. Matthews Correlation Coefficient (MCC)

The MCC, an objective and fair gauge for grouping quality, factors into consideration the genuine positives as well as fraudulent negatives. For tasks such as binary classification including data that is not balanced, it can be particularly useful since it provides greater detail than precision. Greater Dams values suggest significantly superior performance for the model.

2.6.1. Jackyard Index

The James Index, referred to as the Convergence over Integration, is used for comparing actual successes with predicted occurrences. This statistic is helpful to measure the efficacy of binary classifiers, especially when dealing with data sets with imbalances. Greater James Index value result in more variation among the planned

and observed positive events. Various combination methods along with evaluation criteria had been carefully chosen in order to obtain a comprehensive and accurate verdict on the method's effectiveness for identifying Hepatitis C infection. With the goal to achieve the most effective outcomes, all combination approaches were modified through hyper parameter optimizing procedures, providing significant information into the effectiveness in algorithmic learning in medical evaluation.

3. Results and Discussion

In the present study, researchers integrated various ensemble learning methods including Random Forest Training, Boosting, and Carrying with parameter optimised procedures like Array Search as well as Random Search to assess the results on the sorting challenge. The simulations were assessed using important evaluation requirements which include the score given by F1, MCC, which stands Jacquard's Index, Cohen's Kappa, preciseness, precision, and remembering. A breakdown of the findings is shown in Table 1, which also provides a comparison of effectiveness of every technique and identifies both its advantages and disadvantages.

Table 1. Evaluation Metrics Results Computed Various Ensemble Techniques

Classifier	Accuracy	Precision	Recall	F1 Score	Cohen's Kappa	Hamming Loss	MCC	Jaccard Index
Random Forest (Grid)	0.898305	0.735047	0.525	0.5989	0.566575	0.101695	0.591994	0.486562
Random Forest (Random)	0.889831	0.634906	0.52298	0.564553	0.543995	0.110169	0.561591	0.451359
Boosting (Grid)	0.915254	0.933333	0.658333	0.700511	0.62963	0.084746	0.671423	0.591667
Boosting (Random)	0.923729	0.838571	0.641667	0.701737	0.692975	0.076271	0.709044	0.568571
Bagging (Grid)	0.889831	0.623625	0.52298	0.551202	0.57915	0.110169	0.585116	0.439977
Bagging (Random)	0.881356	0.446667	0.42298	0.428088	0.521852	0.118644	0.534086	0.350707
XGBoost (Grid)	0.915254	0.560784	0.5	0.515745	0.684154	0.084746	0.688993	0.432451
XGBoost (Random)	0.923729	0.567451	0.533333	0.543119	0.71551	0.076271	0.719783	0.469356
Stacking (Grid)	0.915254	0.667393	0.59798	0.620286	0.692227	0.084746	0.694478	0.500728
Stacking (Random)	0.915254	0.775238	0.575	0.646425	0.658367	0.084746	0.672897	0.521905

3.1. Accuracy

The Boosting models outperform the others in terms of accuracy; Boosting (Random Search) has the highest score of 0.9237, closely followed by Boosting (Grid Search) at 0.9153. The Random Forest models have a somewhat reduced accuracy of 0.8983 and 0.8898 for Random Forest (Grid Search) and Random Forest (Random Search), respectively. With an accuracy of 0.8898, the Bagging (Grid Search) method also matches the Random Forest (Random Search) method, indicating that, when applied to the Random Forest and Bagging models, grid and random search strategies offer little difference in performance.

3.2. Precision

With a score of 0.9333, Boosting (Grid Search) has a major advantage when it comes to precision, indicating its great ability to detect positive occurrences correctly with fewer false positives. Random Search, or Boosting, comes in second with an accuracy score of 0.8386, still beating the Random Forest models. Random

Forest (Grid Search), with a score of 0.7350 as opposed to 0.6349, demonstrates greater precision than Random Forest (an arbitrary Search). Nevertheless, bagging (Grid Search) exhibits the lowest performance in this group, with an accuracy rating of 0.6236, suggesting a greater probability of incorrectly identifying negative cases as positive.

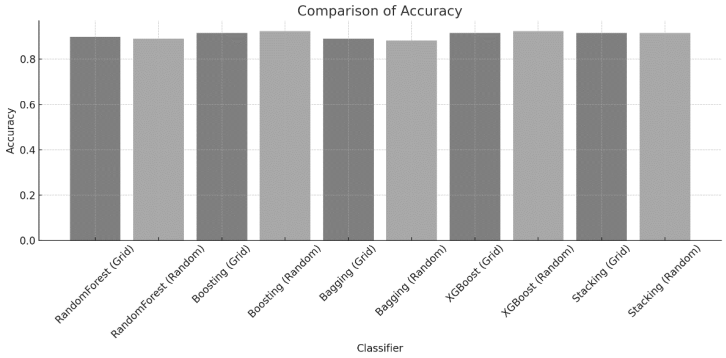

Figure 6. Accuracy

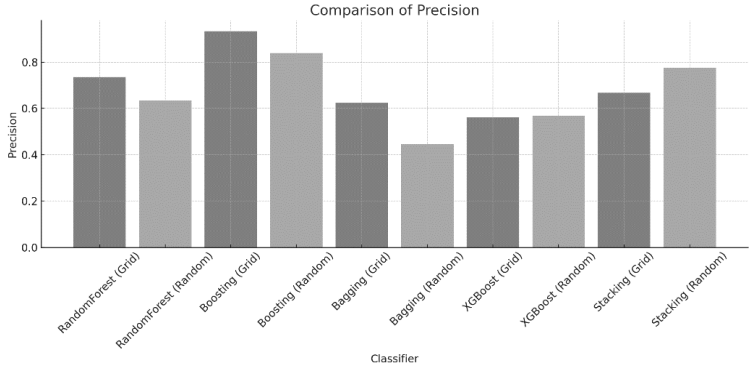

Figure 7. Precision

3.3. Recall

Random Search hypertuning generally produces superior recall outcomes across all models, according to the recall metric, which gauges the model's accuracy in identifying positive cases. Outperforming RandomForest (Random Search) at 0.5230, Boosting (Random Search) achieves 0.6417 in recall. However, the Boosting grid search variation performs marginally better at 0.6583, while the grid search variants from RandomForest and Bagging both reach similar recall levels around 0.5230, suggesting that the latter two models have greater difficulty detecting all positive cases.

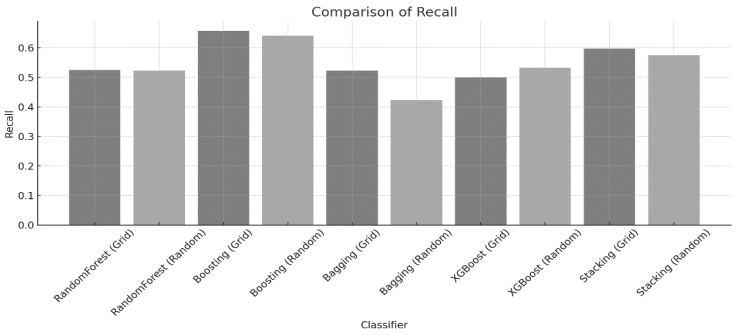

Figure 8. Recall

3.4. F1 Score

Boosting (Random Search) scores 0.7017, the highest F1 score, which strikes a balance between recall and precision. With a score of 0.7005, Grid Search Boosting comes in second. With F1 scores of 0.5989 and 0.5646,

respectively, RandomForest (Grid Search) and RandomForest (Random Search) exhibit a worse trade-off between precision and recall when compared to the Boosting models. With its lowest F1 score of 0.5512, Bagging (Grid Search) exhibits the greatest challenge in striking a satisfactory balance between recall and precision.

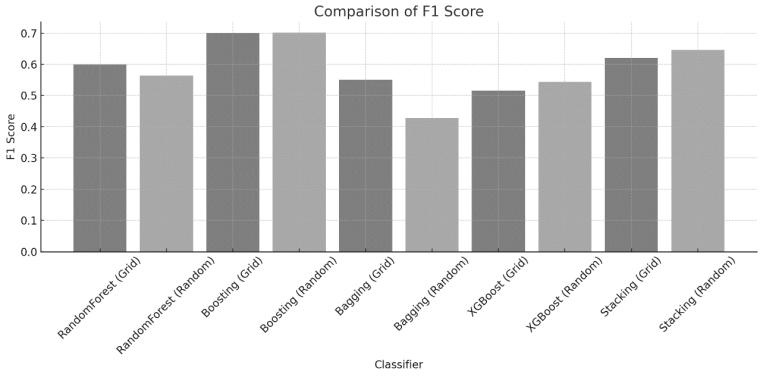

Figure 9. F1 Score

3.5. Cohen's Kappa

With a score of 0.6930, the Cohen's Kappa metric which gauges the degree of agreement between expected and actual labels further emphasizes the advantages of Boosting (Random Search), demonstrating the highest degree of prediction consistency. While RandomForest (Grid Search) performs marginally better than its random search variation, scoring 0.5666 compared to 0.5440, Boosting (Grid Search) also performs well, scoring 0.6296. With a Kappa score of 0.5791, Bagging (Grid Search) exhibits performance comparable to the RandomForest models, indicating reasonable consistency in predictions.

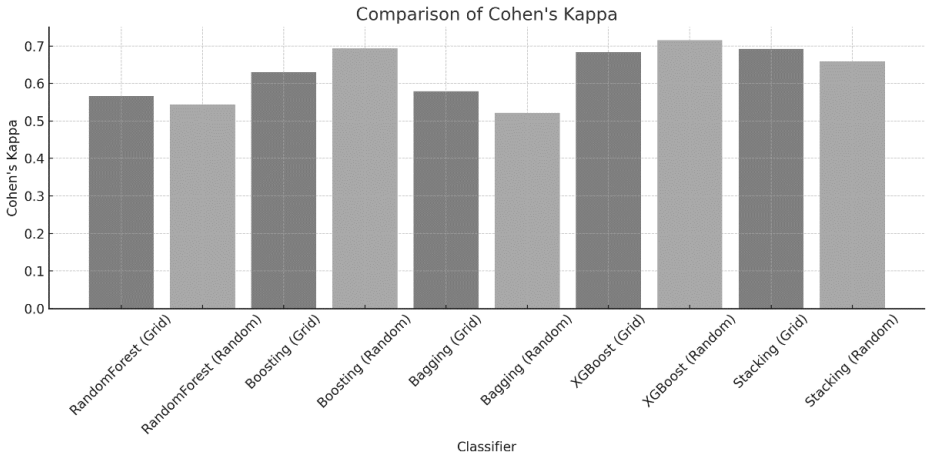

Figure 10. Comparison of Cohen's Kappa

3.6. Hamming Loss

When considering Hamming Loss, which gauges the percentage of inaccurate forecasts, the Boosting models perform exceptionally well. With a Hamming Loss of 0.0763, Boosting (Random Search) produces the lowest number of classification errors. With a 0.0847 loss, Boosting (Grid Search) comes in close second. Compared to RandomForest (Random Search) and Bagging (Grid Search), which both have a Hamming Loss of 0.1102, RandomForest (Grid Search) has a slightly better score of 0.1017, indicating that these models tend to make more mistakes in their classifications.

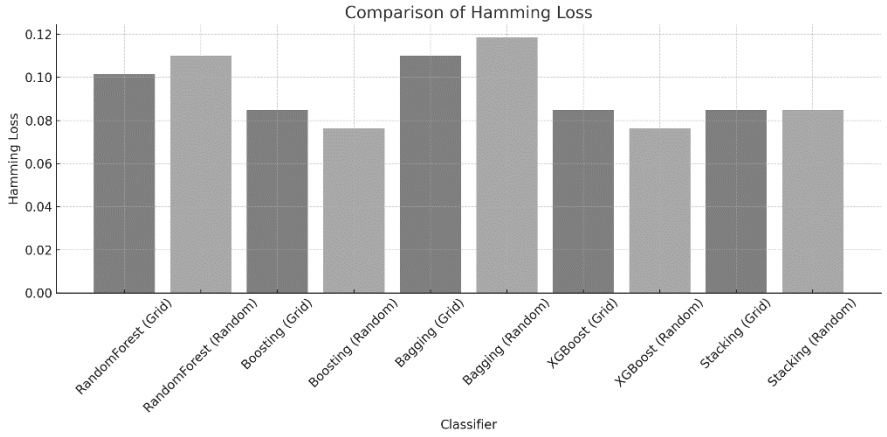

Figure 11. Hamming Loss

3.7. Matthews Correlation Coefficient (MCC)

Boosting (Random Search) has the highest score of 0.7090 on the Matthews Correlation Coefficient (MCC), which evaluates the quality of binary classifications. Boosting (Grid Search) comes in second place with a score of 0.6714. Grid search for RandomForest performs somewhat better than random search with scores of 0.5616 and 0.5920, respectively. With a score of 0.5851, Bagging (Grid Search) performs worse overall in terms of predictive quality.

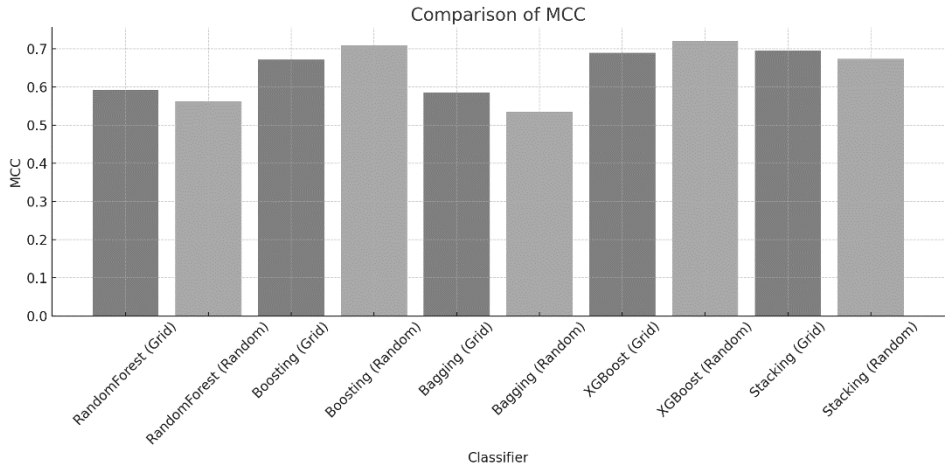

Figure 12. MCC

3.8. Jaccard Index

Lastly, the Jaccard Index, which analyses the closeness between predicted and real labels, demonstrates that Boosting models perform best. With a score of 0.5917, Boosting (Grid Search) comes in top, followed by Boosting (Random Search) with a score of 0.5686. With a score of 0.4866 as opposed to 0.4514, RandomForest (Grid Search) exhibits a superior Jaccard Index than RandomForest (Random Search). Once more, bagging (also known as grid search) performs the worst in this domain, as evidenced by its inferior overall predictive performance as measured by the Jaccard Index of 0.4400. In summary, boosting strategies typically beat the other models across a wide range of measures, including accuracy, F1 score, Cohen's Kappa, and MCC. This is especially true when combined with random search hypertuning. Although it is marginally superior to its random search variant in certain measures, Boosting (Grid Search) likewise demonstrates outstanding performance. While RandomForest models, especially those that have been adjusted via grid search, perform reasonably well across measures, they fall short of Boosting models in a few areas. Boosting and RandomForest

approaches outperform Bagging (Grid Search) in general, suggesting that it might not be as appropriate for this dataset.

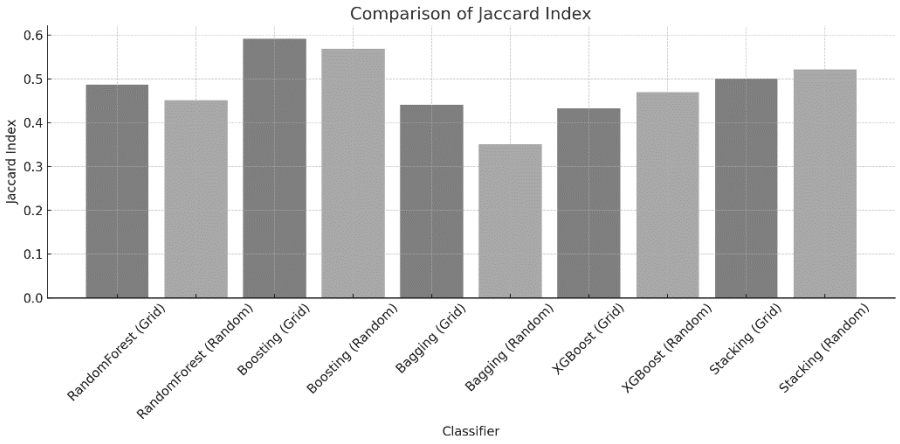

Figure 13. Jaccard Index

4. Conclusions

Finally, a number of important conclusions can be drawn from the comparison of different ensemble approaches with hyperparameter tuning using Grid Search and Random Search. Boosting is the most efficient method for this classification assignment because it consistently beat other models across most assessment metrics, including as accuracy, F1 score, Cohen's Kappa, and MCC, especially when paired with Random Search. The RandomForest models exhibited commendable performance as well; Grid Search produced more consistent outcomes than Random Search, particularly in terms of precision and F1 score. But compared to Boosting and RandomForest, Bagging performed worse on most criteria, suggesting it might not be as appropriate for the dataset. These results imply that, in order to enhance predictive performance, en masse learning techniques specifically, boosting with Random Search optimization should be taken into consideration for application in comparable classification problems. In order to improve practical applications, future study might concentrate on fine-tuning and experimenting with additional ensemble approaches, investigating a wider range of datasets, and investigating the influence of computational efficiency and interpretability of the model.

Reference

1. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
2. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189-1232.
3. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.
4. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.
5. Rokach, L. (2010). Ensemble-based classifiers. Artificial Intelligence Review, 33(1-2), 1-39.
6. Dietterich, T. G. (2000). Ensemble methods in machine learning. Multiple Classifier Systems, 1857, 1-15.
7. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer Science & Business Media.
8. Wainberg, M., Merico, D., Delong, A., & Frey, B. J. (2016). Deep learning in biomedicine. Nature Biotechnology, 35(4), 319-327.
9. Ejaz, F., Tanveer, F., Shoukat, F., Fatima, N., & Ahmad, A. (2024). Effectiveness of routine physical therapy with or without home-based intensive bimanual training on clinical outcomes in cerebral palsy children: a randomised controlled trial. Physiotherapy Quarterly, 32(1), 78-83.
10. Polat, K., & Güneş, S. (2007). Hepatitis disease diagnosis using a new hybrid system based on feature selection (FS) and artificial immune recognition system (AIRS) with fuzzy resource allocation mechanism. International Journal of Medical Informatics, 76(9), 651-662.
11. Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging Artificial Intelligence Applications in Computer Engineering, 160(1), 3-24.
12. Quinlan, J. R. (1993). C4.5: Programs for machine learning. Elsevier.
13. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139.
14. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.
15. Han, J., Kamber, M., & Pei, J. (2011). Data mining: Concepts and techniques. Elsevier.
16. Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123-140.
17. Schapire, R. E. (1990). The strength of weak learnability. Machine Learning, 5(2), 197-227.
18. Vapnik, V. N. (1998). Statistical learning theory. Wiley.
19. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30.
20. Ho, T. K. (1995). Random decision forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, 1, 278-282.
21. Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., & Steinberg, D. (2008). Top 10 algorithms in data mining. Knowledge and Information Systems, 14(1), 1-37.
22. Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R News, 2(3), 18-22.
23. Zhang, H., & Singer, B. (2010). Recursive partitioning and applications. Springer Science & Business Media.
24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
25. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097-1105.
26. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., & Zhao, S. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1-2), 1-210.
27. Akella, V., & Zhang, Y. (2010). Hepatitis C virus and liver injury: Pathological and molecular mechanisms. Cell Bioscience, 10(1), 1-12.
28. El-Serag, H. B. (2012). Epidemiology of viral hepatitis and hepatocellular carcinoma. Gastroenterology, 142(6), 1264-1273.

29. Tang, Y., & Liu, T. (2017). Ensemble learning techniques for combating concept drift in Hepatitis C diagnosis. Journal of Healthcare Engineering, 2017, 1-9.
30. Lichman, M. (2013). UCI Machine Learning Repository [https://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
31. Zhang, Y., & Yang, Q. (2020). A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering, 34(1), 1-20.
32. Sajjad, R., Khan, M. F., Nawaz, A., Ali, M. T., & Adil, M. (2022). Systematic analysis of ovarian cancer empowered with machine and deep learning: a taxonomy and future challenges. Journal of Computing & Biomedical Informatics, 3(02), 64-87.
33. Shah, A. M., Aljubayri, M., Khan, M. F., Alqahtani, J., Sulaiman, A., & Shaikh, A. (2023). ILSM: Incorporated Lightweight Security Model for Improving QOS in WSN. Computer Systems Science & Engineering, 46(2).