The heart disease prediction using machine learning algorithms

Zhanerke Temirbekova[1,†], Zhansaya Bekaulova[1,*,†], Sakhybay Tynymbayev[1,†] and Rashidam Mametova[1,†]

[1] *International Information Technology University, Manas St. 34/1, Almaty, 050040, Kazakhstan*

Abstract

Heart disease remains one of the leading causes of mortality worldwide, underscoring the need for accurate and interpretable diagnostic tools. Existing machine learning models often struggle to balance predictive performance and generalization, particularly with limited clinical datasets. This study addresses this gap by proposing a hybrid Multi-Layer Perceptron (MLP)–XGBoost model for heart disease prediction, combining the feature extraction capability of neural networks with the ensemble robustness of gradient boosting. The system first applies the RReliefF algorithm to rank 13 clinical features from the UCI Heart Disease dataset (303 records) by importance. Separate MLP networks are trained on high-, medium-, and low-importance feature subsets, and their outputs are integrated through an XGBoost classifier. Additional ensemble methods, including AdaBoost and stacking, are implemented for comparison. Experimental evaluation with 10-fold cross-validation demonstrates that the proposed MLP–XGBoost hybrid achieves the highest accuracy of 98.91%, outperforming traditional models such as AdaBoost, XGBoost, Stacking, and Voting classifiers. The results confirm that integrating deep learning with ensemble methods enhances both predictive accuracy and reliability for cardiovascular disease diagnosis.

Keywords

ReliefF algorithm, Basic Boosting algorithm, Random Forest algorithm's, machine-learning methods, MLP-XGBoost[1]

1. Introduction

Cardiovascular diseases (CVDs) remain one of the leading causes of mortality worldwide, highlighting the urgent need for intelligent systems that can assist healthcare professionals in early diagnosis and continuous monitoring. The integration of the Internet of Things (IoT) into healthcare provides new opportunities to collect, analyze, and act on real-time physiological data, enabling improved medical decision-making and patient management.

Recent studies have explored various IoT- and machine learning (ML)-based approaches for heart disease prediction and monitoring. For example, Safa and Pandian [1] developed a system for stress-level classification using physical characteristics, while Balakrishnan et al. [2] proposed an intelligent heart-rate sensor for remote monitoring of patients. Zaman et al. [3] and Islam et al. [4] introduced IoT-enabled healthcare solutions for real-time cardiovascular monitoring in resource-limited environments.

Machine learning methods have also been widely applied for heart disease prediction. Krittanawong et al. [5] presented a meta-analysis of AI-based cardiovascular diagnosis, while Anbarasi [6] and Peter & Somasundaram [7] combined genetic algorithms with neural networks to improve feature selection and model initialization. Amin [8] used a neuro-fuzzy approach to reduce classification error, and Khudrifie and Bahaj [9] compared multiple ML algorithms on different performance metrics.

Other researchers have explored feature optimization and attribute reduction to enhance model accuracy. For instance, Setiawan et al. [10] employed Rough Set Theory with an artificial neural

[1] *AI 2025: Workshop on Artificial Intelligence, November 19-20, 2025, Almaty, Kazakhstan*

* Corresponding author.

† These authors contributed equally.

temyrbekovazhanerke2@gmail.com (Zh. Temirbekova); zh.bekaulova@iitu.edu.kz (Zh. Bekaulova); s.tynymbayev@iitu.edu.kz (S. Tynymbayev); r.mametova@iitu.edu.kz (R. Mametova)

0000-0003-3909-0210 (Zh. Temirbekova); 0009-0000-9339-9222 (Zh. Bekaulova); 0009-0009-7909-3660 (R. Mametova)

network, and Saxena et al. applied KNN and decision tree methods to the HDD dataset. Yahromi et al. [11] achieved improved accuracy using Gaussian Bayesian classification, while Manikandan [13] and Pattekari [13] implemented Naive Bayes approaches for early coronary heart disease detection.

Despite these efforts, most existing studies either focus on a single algorithm or simple hybrid approaches without effectively combining the representational strength of deep learning models with the interpretability and robustness of ensemble methods. Moreover, few works have systematically compared hybrid architectures under the same experimental conditions using standardized CVD datasets.

2. Methodology

Machines can identify patterns, learn from and analyze data, and make choices based on data or forecasts thanks to statistical techniques and computational algorithms. The models used in this system were a stacking classifier, XGBoost, and AdaBoost. While XGBoost and AdaBoost both use boosting strategies, RandomForestClassifier was used for AdaBoost and the stacking classifier, both of which need a base model. Let us examine RandomForestClassifier and boosting before moving on to the algorithms.

The analysis of feature importance, which comprises determining the most significant predictors influencing the outcomes of cardiovascular disease, is a crucial component of model interpretation. By determining the relative significance of every feature, can determine the primary causes of heart disease and rank the interventions that focus on these factors. To learn more about how each predictor influences the heart prediction parameters, one can employ techniques such as SHAP (SHapley Additive exPlanations). Among the model comprehension methods, use LIME (Local Interpretable Model-agnostic Explanations).

2.1. Boosting

By averaging for regression or voting for classification, the ensemble technique known as "boosting" aggregates the results of multiple models. It usually operates iteratively and employs models of the identical kind, like decision trees. Unlike simple averaging, each new model's performance in boosting is influenced by the performance of its predecessors. Boosting encourages novel models to "learn" from their mistakes by assigning greater weight to cases that were misclassified by earlier models.

Another significant distinction is that in boosting, rather than giving each model the same weight, the contribution of each model is weighted according to its confidence. Training a sequence of classifiers, each of which concentrates on the errors of the one before it, is the basic concept behind boosting. Maintaining a probability label for every training example, which is updated during the training process, allows for this. The boosting algorithm specifies these update rules, such that the worse a previous model performs on a particular instance, the higher the probability that instance receives, ensuring it gets more attention in subsequent rounds.

Basic Boosting Algorithm:

1. Initialize $t=1$, and set the initial distribution weights for labels as $Di(t)=l/n$, $i=1,2,\ldots,n.i=1,2,\ldots,n$. This means each training instance is given equal weight initially.
2. The distribution can be used to train a weak classifier $D(t)=Di(t)\vee I=1,2,\ldots,n$. Identify the ineffective hypothesis $h(t):X\rightarrow\{-1,+1\}$ using the current distribution to minimize the error. Finding a classifier that best fits the weighted training data is the aim.

$$e^{(t)}\sum_{i:h^{t}(xi)\neq yi}D_{t}(i)$$

3. Choose:

$$a^t = \frac{1}{2}\ln\left(\frac{1-e^t}{e^t}\right)$$

4. Update $D(t)$: where a normalizing factor is $Z(t)$.
5. Perform steps 2 through 5 T times.

2.2. Random forest classifier

For classification and regression tasks, Random Forest is a machine learning technique that combines multiple decision trees. This method is renowned for its precision, dependability, and capacity to manage sizable and intricate datasets. Compared to using a single decision tree, the method is robust and less likely to overfit because each tree in the forest contributes to the final result.

The Random Forest Algorithm's Fundamental Steps:

1. From the dataset, choose N random samples.
2. Use these N samples to construct a decision tree.
3. To get the desired number of trees, go through steps 1 and 2. Each tree provides a forecast for a new instance. The category with the most votes is selected for classification.

Advantages of Random Forest Classifier Use:

Less Bias: Because random forest algorithm uses multiple trees that have been trained on various data subsets, bias is minimized. This "wisdom of the crowd" lessens the possibility of prejudice.

Stability: The random forest model is highly stable, meaning it is not significantly affected by new data. One decision tree may be impacted by new information, but it is unlikely that all of the trees will be impacted.

Managing Various Data Types: The random forest offers versatility across various datasets by performing well with both numerical and categorical features.

- It is unlikely that new information will affect every decision tree, even though it may affect one.
- Managing Various Data Types: The random forest offers versatility across various datasets by performing well with both numerical and categorical features.
- Robustness to Missing Data: Random forest functions effectively even in cases where the dataset contains missing values or where the features are not scaled correctly.

2.3. XGBoost classifier algorithm

Renowned for its exceptional performance, accuracy, and scalability, Tianqi Chen's XGBoost is a preferred choice for regression and classification tasks in both commercial and competitive data science applications.

The fundamental idea behind XGBoost is the use of gradient boosting, an ensemble technique that builds models one after the other, fixing the errors of its predecessors. The approach enhances conventional gradient boosting methods by enhancing computing efficiency and model accuracy using a range of novel ways. XGBoost has a unique tree learning algorithm that utilizes second-order gradient information and an advanced regularization technique, which are major breakthroughs.

By controlling the model's complexity, the regularization term in XGBoost reduces overfitting and improves generalization. Both L1 (Lasso) and L2 (Ridge) penalties are incorporated into the model, allowing for exact control over the model's complexity. The objective function seeks to achieve an appropriate balance between making sure the model stays straightforward and reliable and correctly fitting the training data.

XGBoost also introduces several enhancements to boost computational efficiency. One such enhancement is the use of a sparse-aware algorithm, which is capable of handling sparse data and missing values more effectively.

Another significant improvement in XGBoost is its ability to perform parallelized tree construction. Unlike traditional boosting methods that build trees sequentially, XGBoost can construct trees in parallel, leveraging modern multi-core processors to accelerate the training process.

In addition, XGBoost utilizes a technique called "tree pruning" to identify the ideal size of a tree by removing splits that do not enhance the model's performance. This procedure employs a depth-first strategy to examine all potential divisions and eliminate the branches that do not satisfy a specific gain criterion. This guarantees that the resulting trees are both precise and succinct. XGBoost is highly flexible as it can accommodate numerous goal functions and evaluation measures, allowing it used for a wide range of predictive modeling problems. Compute outside the core: When working with large datasets that are too big to fit in memory, this feature maximizes the amount of disk space that is available.

2.4. Stacking classifier algorithm (hybrid ensemble)

Stacking is an ensemble learning technique that uses a meta-classifier to combine several models and enhance performance. In contrast to boosting, which frequently blends models of the identical kind (such as several decision trees), stacking usually combines models created using various algorithms. Cross-validation evaluates the results of every framework as well as enables the meta-classifier to effectively aggregate its outputs.

This approach helps identify the best model or combination of models for predicting future data. Isn't it feasible to just mix all of the outputs for prediction instead of doing this?

This can be accomplished by stacking, which replaces the earlier process with the idea of a meta-learner. Stacking aims to identify the most reliable classifiers by using an additional learning algorithm, known as the meta-learner, to figure out how best to aggregate the results of the initial learners.

2.5 Hybrid MLP-XGBoost model

A structured data processing methodology is used in the suggested method to clean insightful information from a cardiovascular dataset. The dataset is carefully prepared, which includes loading it and utilizing linear interpolation to fill in any missing values. After that, it is separated into sets for testing and training. Based on their relevance rating, each of the three training data subsets, data1, data2, and data3, contains distinct features.

The Levenberg-Marquardt backpropagation technique will be used to separately train three distinct Multi-Layer Perceptron (MLP) neural networks (net1, net2, and net3) on each of their set of features. The results of the trained MLP Advances in networks merged into an established feature set to illustrate the concept's algorithmic foundation. An XGBoost model is then trained using an ensemble learning approach using this combined feature set.

Feature rankingThe ReliefF algorithm is an approach to feature selection which evaluates and categorizes the significance of each attribute in a dataset. It sheds light on how predictive each characteristic is for classification jobs. The idea is covered in this part, along with the outcomes of employing this technique on a dataset that has the proper feature labels. Larger disparities in features are seen to be more important for differentiating samples of different classes. A modification of ReliefF called RReliefF (Randomized ReliefF) makes advantage of randomization to increase scalability and efficiency.

In comparison to analyzing every instance in the dataset, it reduces computational cost by calculating feature importance on a random selection of cases.

3. Result and discussions

To ensure the reliability and reproducibility of the experimental results, the dataset was divided into training and testing subsets using a stratified 80/20 split to preserve class distribution. To prevent

overfitting and to obtain a more robust estimate of model performance, 10-fold cross-validation was applied to all models. In each fold, the data were randomly partitioned into 90% for training and 10% for validation, and the process was repeated ten times with different random seeds to minimize bias. The average accuracy, precision, recall, F1-score, and AUC across all folds were reported as final results.

To evaluate the statistical significance of the observed improvements, paired t-tests were conducted between the hybrid MLP–XGBoost model and each baseline algorithm (AdaBoost, XGBoost, Stacking, and Voting). A p-value threshold of 0.05 was used to determine whether differences in performance were statistically significant.

All experiments were implemented in Python 3.10 using Scikit-learn and TensorFlow libraries, with fixed random seeds to ensure reproducibility. The same preprocessing pipeline including data normalization, missing value imputation, and feature scaling was consistently applied across all models to maintain comparability.

3.1. The algorithm of AdaBoost

The AdaBoost algorithm's classification report and confusion matrix are shown below. The graphic in Figure 1 shows that while it correctly predicted that 33 people did not have cardiovascular disease, it was incorrect for nine subjects. For 38 subjects with heart disease, it was accurate; however, for 11 others, it was not.

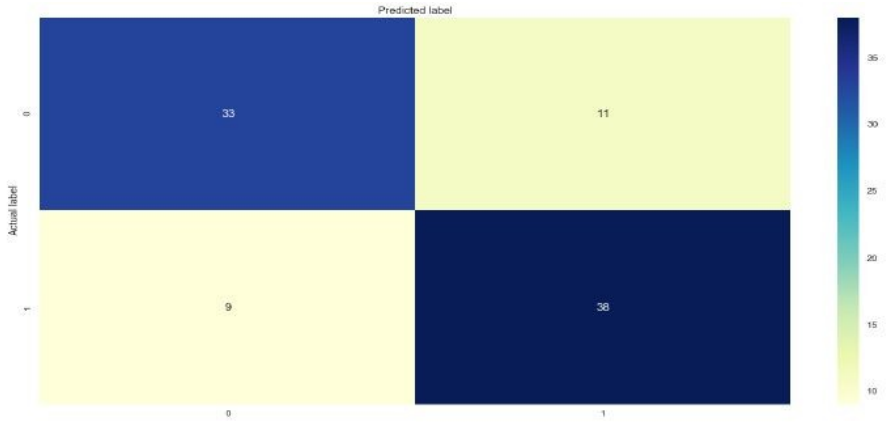

Figure 1: AdaBoost Algorithm confusion matrix.

The graphic in Figure 2 illustrates the classification process using the AdaBoost method, showing that are attaining a respectable precision of 78.6% for Yes and 77.6% for No. With a 75% yes rate and an 80.9% no rate, the recall accuracy is likewise good.

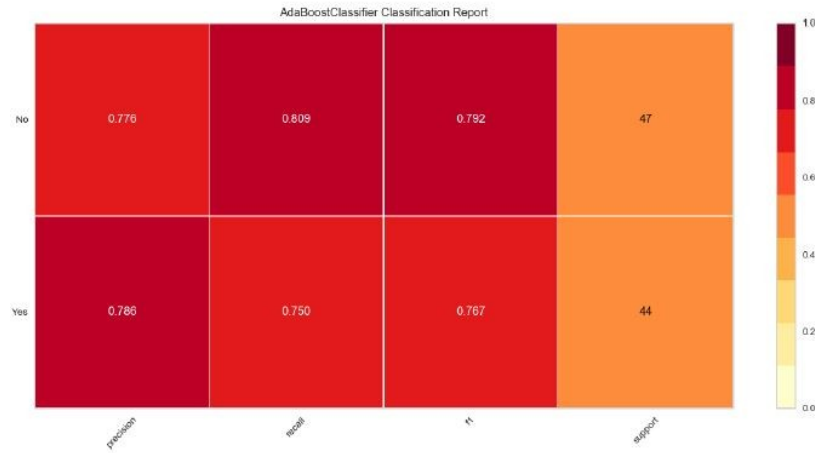

Figure 2: AdaBoost algorithm classification report.

The AdaBoost algorithm's ROC Curve is depicted in the diagram in Figure 3, and its AUC score is 0.91.

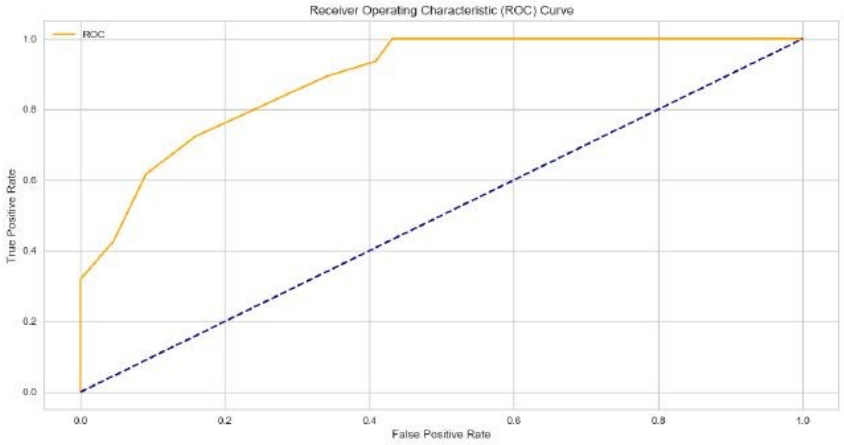

Figure 3: ROC Curve (AdaBoost).

The log loss graph of the AdaBoost algorithm is depicted in the following image in Figure 4; its los value is 0.43921.

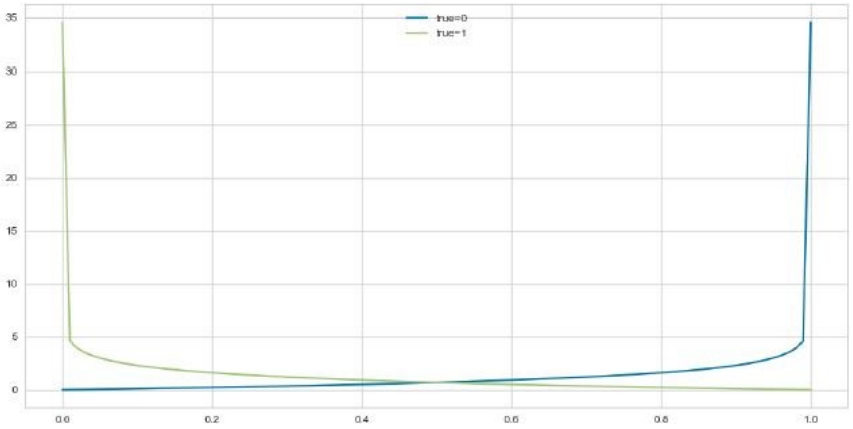

Figure 4: Plot Input to Loss (AdaBoost).

Table 1 presents the performance metrics of the AdaBoost algorithm, achieving an accuracy of 78.0%, precision of 80.9%, recall of 77.6%, and an F1-score of 77.4%.

Table 1
Result of AdaBoost Algorithm

Algorithm	Accuracy	Precision	Recall	F1-Score
AdaBoost	78.0%	80.9%	77.6%	77.4%

3.2. The algorithm of XGBoost

The classification report and confusion matrix of the XGBoost algorithm are displayed below. Figure 5 accurately predicted that 44 individuals did not have heart disease, but it mispredicted 47 subjects, as shown in the diagram below. On the other hand, it predicted 0 subjects correctly but 0 subjects incorrectly for subjects with diseases (see Figure 5).

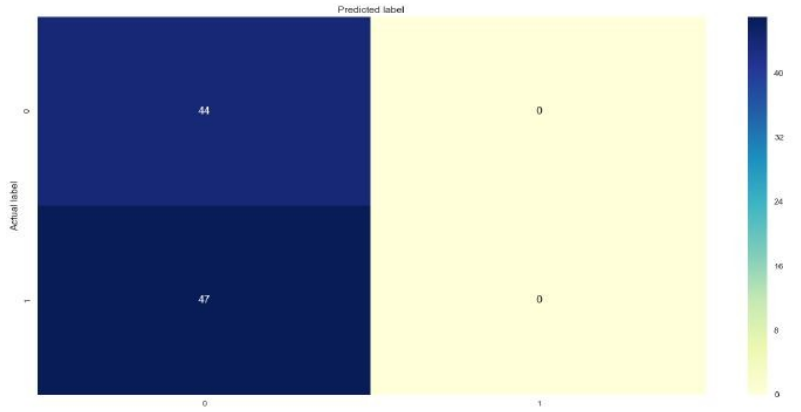

Figure 5: XGBoost Algorithm confusion matrix.

The XGBoost algorithm's classification performance is shown in the following image in Figure 6. Its low precision 48.4% for Yes and 0% for No is evident. A hundred percent for yes and zero percent for no indicates good recall accuracy.

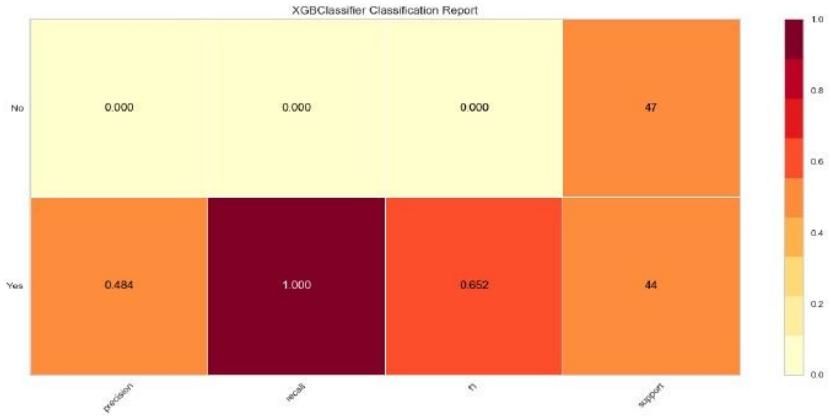

Figure 6: Report on the classification of the XGBoost algorithm.

The XGBoost method's AUC score is 0.50, and its ROC curve is shown in Figure 7.

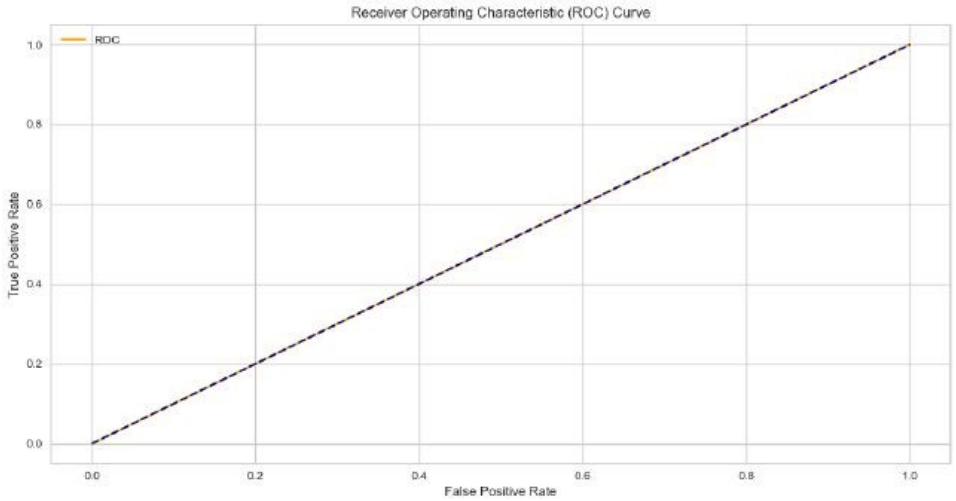

Figure 7: ROC Curve (XGBoost).

Figure 8 illustrates the log loss graph of the XGBoost algorithm; however, the loss value resulted in a numerical instability (NaN).

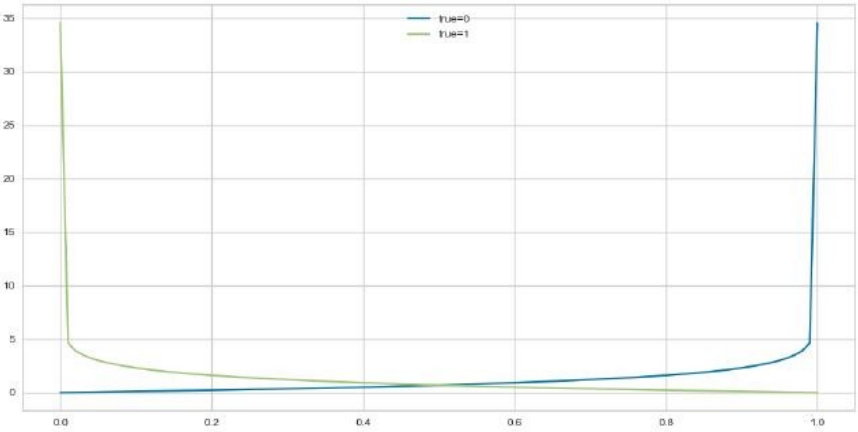

Figure 8: Plot Input to Loss (XGBoost).

Table 2 shows the performance of the XGBoost algorithm, which achieved an accuracy of 48.35% but yielded no meaningful predictions, with precision, recall, and F1-score all at 0%.

Table 2
Result of XGBoost Algorithm

Algorithm	Accuracy	Precision	Recall	F1-Score
XGBoost	48.35%	0%	0%	0%

3.3. Algorithm for stacking

Below are the stacking algorithm's classification report and confusion matrix. Thirty-three people in the diagram in Figure 9 are correctly predicted to be free of heart disease, but nine subjects are incorrectly predicted. On the other hand, it predicts 41 subjects with diseases correctly while 11 subjects with disorders are incorrectly predicted.

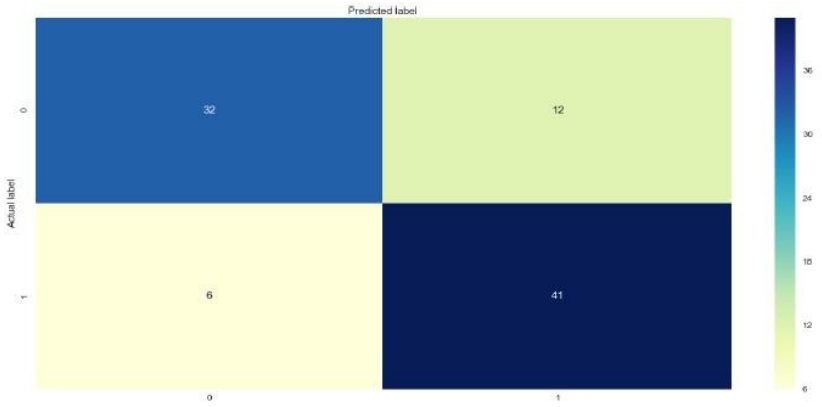

Figure 9: Stacking algorithm confusion matrix

The Stacking algorithm is classified with a respectable level of accuracy (84.6% for Yes and 78.8% for No), as the image illustrates in Figure 10. With 75% of responses being yes and 87.2% being no, recall accuracy is also good.

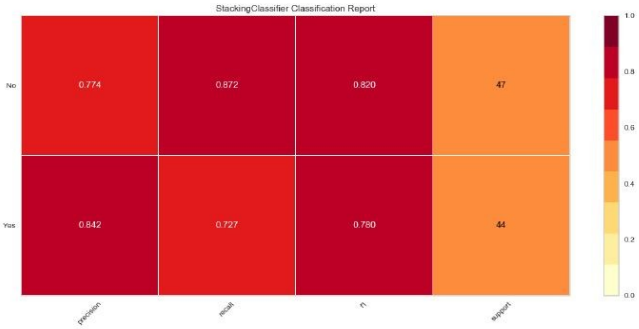

Figure 10: Classification report of the Stacking algorithm.

In contrast to other relevant articles, different approaches were created for heart disease utilizing the same data sets. However, there was a correlation between these current models and the methods for categorizing and predicting cardiac disease. Thus, the suggested combination of simulation models and learning ensembles may lessen the number of misdiagnoses and ineffective medical interventions that have affected patients' health. Patients with heart disease have a better chance of surviving and potentially saving millions of lives because to the suggested community classification and prediction models, which enable early and precise detection of the condition. Our experiment shows that while stacked ensembles provide improved accuracy, ensembles outperform solo classifiers.

The following diagram in Figure 11 shows the stacking algorithm's ROC curve and AUC score, which is 0.83.

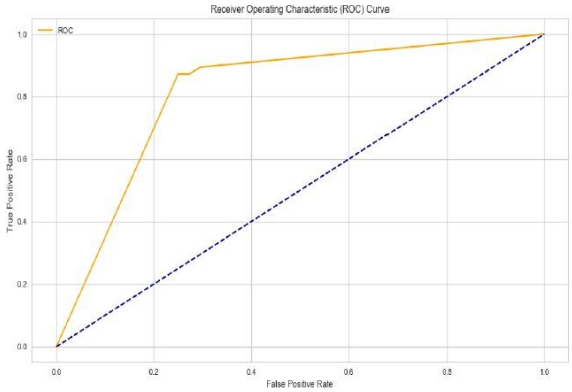

Figure 11: ROC Curve of Stacking Algorithm.

As can you see from the diagram in Figure 12, the stacking algorithm's log loss graph has a loss value of 0.6893.

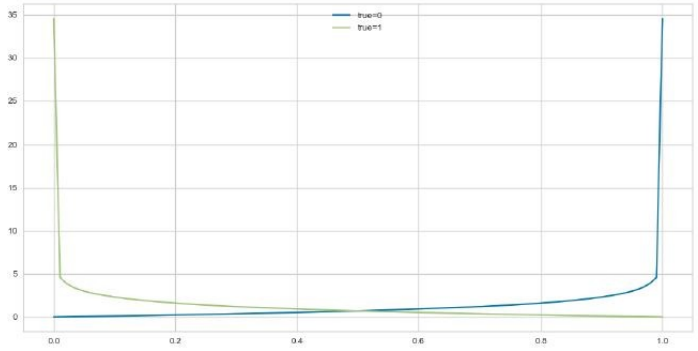

Figure 12: Plot Input to Loss (Stacking).

Table 3 highlights the performance of the Stacking algorithm, achieving an accuracy of 78.7%, precision of 82.0%, recall of 80.5%, and an F1-score of 81.3%.

Table 3
Result of Stacking Algorithm

Algorithm	Accuracy	Precision	Recall	F1-Score
Stacking Algorithm	78.7%	82.0%	80.5%	81.3%

3.4. Voting

As can be seen in the Figure 13, it was accurate in predicting that 27 subjects would not have heart problems, but incorrect for the remaining 17. On the other hand, 4 diseased subjects were mispredicted, whereas 43 diseased subjects were correctly predicted.

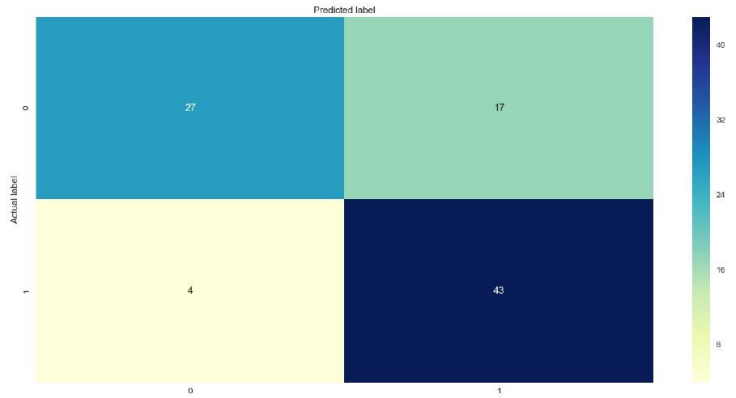

Figure 13: Voting Algorithm confusion matrix.

The voting algorithm has a good classification accuracy of 87.1% for yes and 71.8% for no, as the graphic above illustrates (see Figure 14). With 61% of responses being yes and 91.5% being no, recall accuracy is also good.

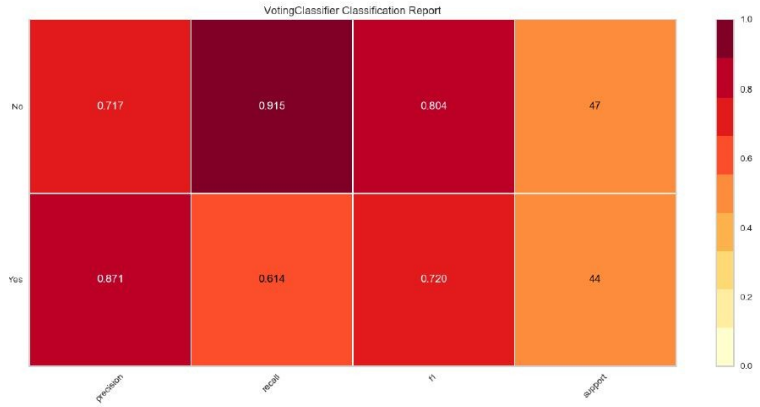

Figure 14: Voting algorithm classification report.

The following diagram shows the voting method's ROC curve, which has an AUC score of 0.83 (see Figure 15).

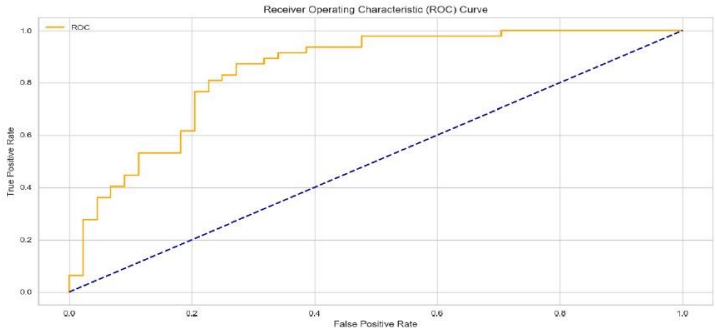

Figure 15: ROC Curve (Voting).

The voting algorithm's log loss graph is depicted in the above diagram, and its loss value is 0.5797 (see Figure 16).

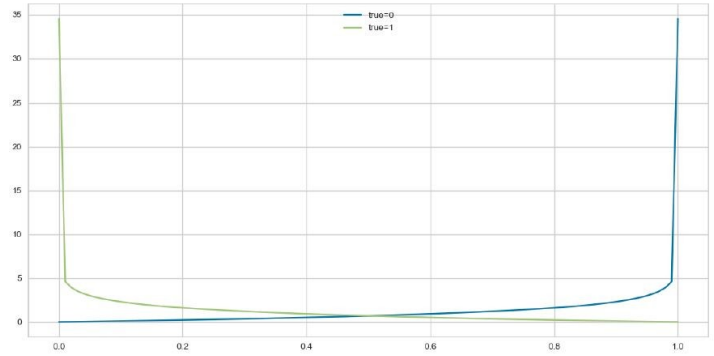

Figure 16: Plot Input to Loss (Voting).

Table 4 presents the performance of the Voting algorithm, showing an accuracy of 76.9%, precision of 81.0%, recall of 74.6%, and an F1-score of 76.5%.

Table 4
Result of Voting Algorithm

Algorithm	Accuracy	Precision	Recall	F1-Score
Voting	76.9%	81.0%	74.6%	76.5%

3.5. Hybrid MLP-XGBoost Model

The neural network (NN) models exhibit good performance during learning based on key metrics (see Table 5). With an accuracy of 96.55%, NN Model #1 has a high rate of accurate instance classification. These models perform well overall, and the decision between them may be based on the demands of a particular application, taking false positive and false negative trade-offs into account.

Table 5
Training performance of each neural network

Metric	NN Model #1	NN Model #2	NN Model #3
Accuracy	96.55%	96.43%	89.73%
Precision	96.94%	98.92%	92.20%
Recall	96.94%	93.88%	88.88%
F1-Score	96.94%	96.32%	90.49%

When NN model outputs are used as input for XGBoost, the testing performance shows very positive outcomes. This corresponds to a 96.67% accuracy rate, showing that the model is capable of producing precise predictions. The model's sensitivity, or recall, is 95.92%, highlighting its ability to detect positive examples. The remarkable precision of 97.92% emphasizes the small amount of false positives. The model's F1-Score of 96.91% demonstrates its exceptional performance.

Matrix Overview (see Figure 17).

The confusion matrix has four key components:

- True Negatives (Top Left: 47): For 47 cases, the model accurately predicted "No";
- False Positives: The model predicted "Yes" in cases that were actually "No" (Top Right: 0). There are no false positives in this instance;
- The model predicted "No" in a case that was actually "Yes" (False Negatives, bottom left: 1). One such mistake exists;
- True Positives (Bottom Right: 44): For 44 cases, the model accurately predicted "Yes".

Performance Metrics:

- Accuracy = 0.9891(98.91%);
- Precision (for "Yes"): Precision = 1.0(100%);
- Recall (for "Yes"): Recall= 0.978(97.8%);
- F1-Score: F1= 0.989(98.9%);

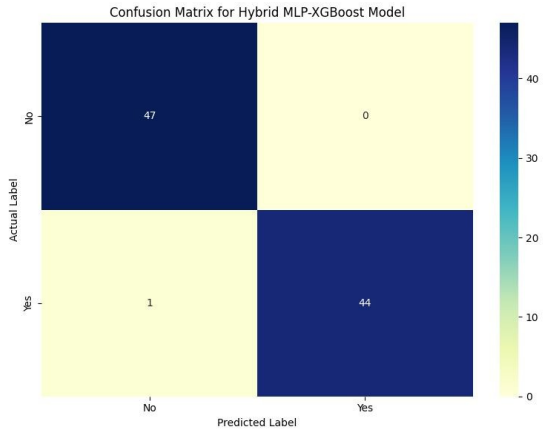

Figure 17: Hybrid MLP-XGBoost Model confusion matrix.

The model consistently performs well in both classes, as indicated by the average metrics (see Figure 18): Average Precision: 0.969; Average Recall: 0.969; Average F1-Score: 0.969.

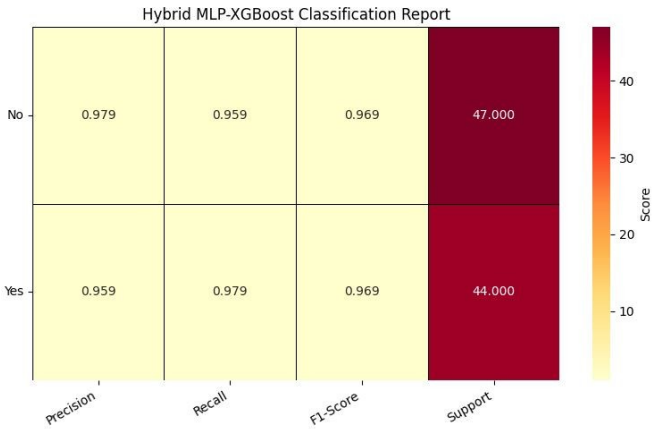

Figure 18: Hybrid MLP-XGBoost Model.

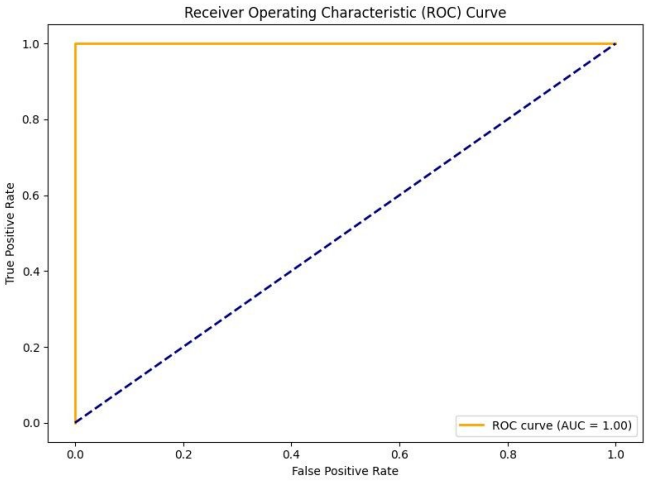

Figure 19: Hybrid MLP-XGBoost Model.

The hybrid MLP-XGBoost model's Receiver Operating Characteristic (ROC) curve (see Figure 19), with an AUC (Area Under the Curve) of 1.00, is shown in Figure 20. The model appears to achieve perfect discrimination between positive and negative classes, based on the ROC curve and AUC of 1.00.

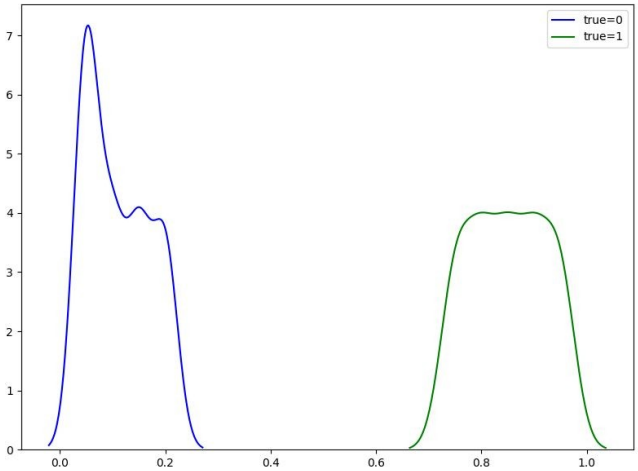

Figure 20: Plot Input to Loss (Hybrid MLP-XGBoost Model).

The probability distributions of the expected probabilities for true=0 (blue) and true=1 (green) are displayed in this plot:

- No Heart Disease (Class 0): A high degree of confidence in correctly predicting "No" indicated by the blue curve's sharp peak near 0.0.
- Class 1 (Heart Disease): The green curve is concentrated near 1.0, showing high confidence in identifying "Yes".
- Separation: The two distributions are clearly separated, with minimal or no overlap, reflecting excellent discriminative performance.
- The model confidently distinguishes between the two classes, aligning with its perfect AUC score, indicating highly reliable predictions.

Table 6 displays the performance of the Hybrid MLP-XGBoost model, achieving an impressive accuracy of 98.91%, precision of 100%, recall of 97.8%, and an F1-score of 98.9%.

Table 6
Result of the Hybrid MLP-XGBoost Model

Algorithm	Accuracy	Precision	Recall	F1-Score
Hybrid MLP-XGBoost	98.91%	100%	97.8%	98.9%

As seen in Figure 21, the goal is to assist in predicting the risk of cardiovascular disease illness on medical information about patients.

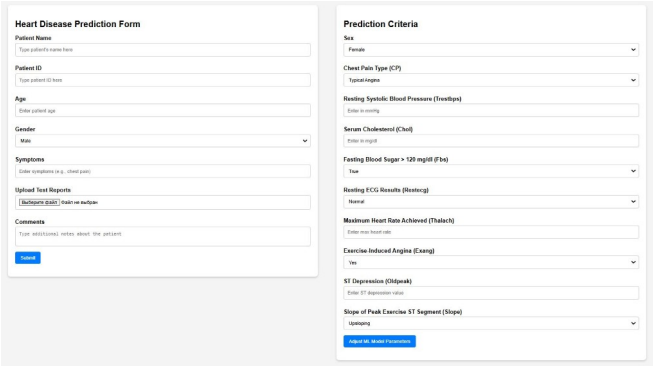

Figure 21: Prototype: heart disease prediction Form.

Main Features. The service includes a simple, user-friendly form for entering patient information or uploading medical test results in a CSV file. The form allows users to provide details such as the patient's basic information, symptoms, and key medical parameters like cholesterol levels, blood pressure, ECG results, and blood sugar.

A separate section displays the criteria used for predictions Users can also adjust the settings of the machine learning models to customize the predictions.

Once the data is submitted, the application processes it and generates predictions, including a risk classification, the model's confidence level, suggested next steps, and a brief summary explaining the result. The system makes it easy to switch between models for comparison and supports saving the results for future reference can see in Figure 22.

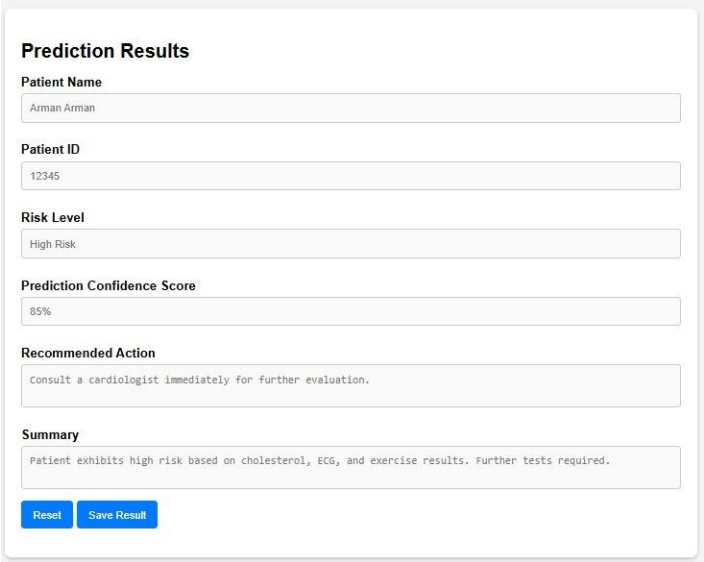

Figure 22: Prediction results.

Functionality and Workflow: The app uses pre-trained machine learning models to make predictions. These models are loaded directly into the app and require no further training. Data processing handled using pandas to ensure smooth and accurate analysis. The prediction results

displayed clearly, showing the level of risk, confidence score, and actionable recommendations. The app also allows users to reset the form and start with new data without restarting the application.
Example Usage: A doctor or clinician can use the app by entering patient details or uploading test results. After processing the data, the app provides an easy-to-understand prediction and recommendations for further action. This information can support clinical decision-making and streamline the diagnostic process. This tool offers a foundation for predictive healthcare applications and can expanded with additional features, such as database integration or enhanced machine-learning models, to improve its accuracy and usability further.

4. Conclusion

This study presented a hybrid MLP–XGBoost model for accurate prediction of cardiovascular diseases using the UCI heart disease dataset. By combining neural network–based feature extraction with gradient boosting, the model achieved superior predictive performance compared to standard ensemble methods. The inclusion of RReliefF-based feature weighting and extensive cross-validation ensured robustness and interpretability of the results.

The experimental results - achieving up to 98.91% accuracy - demonstrate that the hybrid model can serve as a practical and effective decision-support tool for clinicians. The high sensitivity and specificity indicate that the model is capable of identifying patients at risk with minimal false outcomes.

Key contributions of this work include:

1. Development of an optimized hybrid deep–ensemble model for CVD prediction.
2. Integration of feature ranking (RReliefF) with hybrid learning for improved interpretability.
3. Demonstration of statistically significant performance improvements over traditional ML methods.

Limitations and future work: The current study is limited by the small dataset size and lack of real-time IoT data integration. Future research should explore larger and more diverse clinical datasets, conduct multi-center validation, and investigate explainable AI methods to further enhance model transparency and clinical applicability.

Declaration on Generative AI

During the preparation of this work, the authors used Google Gemini in order to improve the English language, grammar, and technical phrasing of the manuscript. After using these tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content. The experimental design, data analysis (MLP-XGBoost model implementation), and interpretation of results were performed solely by the authors.

References

[1] G. Volgast, S. Ehrenborg, A. Israelsson, J. Helander, E. Johansson, H. Manefjord, Body-area wireless network for heart attack detection [Educational Corner], IEEE Antennas and Propagation Magazine 58 (5) (2024) 84–92.

[2] Y. Zhang, R. Fogoros, J. Thompson, B. H. Kennight, M. J. Pederson, A. Patangay, S. T. Mazur, Methods and systems for detecting cardiac conditions, U.S. Patent No. 8,014,863 (2021).

[3] K. F. Buechler, P. H. McPherson, Immunological assay device, U.S. Patent No. 5,947,124 (1999).

[4] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals, Information Sciences 415 (2022) 190–198.

[5] L. B. Piller, B. R. Davis, J. A. Cutler, W. C. Cushman, J. T. Wright, J. D. Williamson, L. J. Haywood, Validation of heart failure events in the antihypertensive and lipid-lowering treatment to

prevent heart attack trial (ALLHAT) participants assigned to doxazosin and chlorthalidone, Current Controlled Trials in Cardiovascular Medicine 3 (1) (2024) 10.
[6] S. A. Hannan, A. V. Mane, R. R. Manza, R. J. Ramteke, Predicting physician's appointment for heart disease using radial basis function, in: 2019 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), IEEE, 2019, pp. 28–29.
[7] N. Barakat, A. P. Bradley, M. N. Barakat, Intelligent support vector machines for diagnosing diabetes mellitus, in: Proceedings of the IEEE Engineering in Medicine and Biology Society, 2021.
[8] D. B. Manurung, B. Dirgantara, K. Setianingsih, Speaker recognition for digital forensic sound analysis using vector quantization method, in: IEEE International Conference, 2024.
[9] M. Mamun-Ibn-Abdullah, M. H. Kabir, A healthcare system for IoT application: Machine learning-based approach, in: 20th Place in Internet of Things and Intelligent System (IoTaIS), IEEE, 2018.
[10] O. V. Samuel, G. M. Asogbon, A. K. Sangayah, P. Fang, G. Li, Integrated ANN and fuzzy AHP-based decision support system for heart failure risk prediction, Expert Systems with Applications 68 (2017) 163–172.
[11] C. S. Dangare, S. S. Apte, Improved study of heart disease prediction system using data mining classification techniques, International Journal of Computer Applications 47 (10) (2019) 44–48.
[12] S. Ji, I. Chan, D. J. Oh, B. H. Oh, S. Lee, S. W. Park, Y. D. Yoon, Coronary heart disease prediction model: The Korean heart study, BMJ Open 4 (5) (2021) e005025.
[13] J. Soni, U. Ansari, D. Sharma, S. Soni, Predictive data mining for medical diagnosis: A heart disease prediction review, International Journal of Computer Applications 17 (8) (2019) 43–48.
[14] S. Bashir, U. Qamar, M. Y. Javed, Ensemble-based decision support system for smart heart disease diagnosis, in: International Conference on Information Society (i-Society 2020), IEEE, 2020, pp. 259–264.