SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 21, No. 3, October 2024, 375-390

UDC: 004.85::616.1 **DOI: https://doi.org/10.2298/SJEE2403375B**

Enhancing Heart Disease Prediction Accuracy by Comparing Classification Models Employing Varied Feature Selection Techniques

Lorena Balliu[1], Blerina Zanaj[2], Gledis Basha[1], Elma Zanaj[1], Elinda Kajo Meçe[1]

Abstract: ML (Machine Learning) is frequently used in health systems to alert physicians in real time. This helps to take preventive measures, such as predicting a future heart attack. This study presents ML combined with various forms of feature selection to identify heart disease. It includes the analysis of different algorithms such as Decision Tree, Logistic Regression, Support Vector Machine, Random Forest and hybrid models. This results in SVM and RM performing better after applying feature selection for individual ML models. Meanwhile, hybrid cases provide good results if the ensemble is done using a Voting Classifier. Our approach in this paper is based on our study of existing literature and methodologies. We can conclude that, for the used dataset, the Voting Classifier appears to be the most accurate and precise model out of all individual and hybrid classifiers that use feature selection techniques.

Keywords: Machine Learning, Cardio Vascular System, Decision tree, Logistic regression, Support vector machine, Random Forest, Hybrid models, Feature selection.

1 Introduction

CVD (Cardio Vascular Diseases) have gained considerable focus in the last decade as they can be considered one of the major contributors to global mortality [1]. Modern lifestyle, genetic predisposition, global air pollution, poor diet, lack of physical activity, smoking, and stress can be considered the main factors that have led to an increased number of CVD cases.

The use of ML in CVD research and healthcare has evolved significantly. We can mention some of its uses:

– Risk prediction (part of prevention). ML is used to develop predictive models. It is related to identifying cardiovascular diseases based on different factors.

[1]Faculty of Technology of Information, Polytechnic University of Tirana, Albania; E-mails: lorena.balliu@fti.edu.al; ezanaj@fti.edu.al; gledis.basha@fti.edu.al; ekajo@fti.edu.al
[2]Faculty of Economics and Agribusiness, Agriculture University of Tirana, Albania; E-mail: bzanaj@ubt.edu.al

– Medical imaging-based. Analysis of medical imaging data retrieved from CT (Computed Tomography) scans, ECG (Electrocardiogram), and MRI (Magnetic Resonance Imaging).
– Customized treatments. Analyzing the data to predict the correct medication dosages or recommend specific therapies.
– Remote monitoring. This is becoming possible by integrating ML algorithms into wearable devices that track real-time parameter changes.
– Drug development. Assisting in the creation of new and innovative medications.

The latest technological advances, the availability of large amounts of data related to CVS (Cardio Vascular System) health issues and the need for precise and individualized solutions have made ML more applicable and effective. Hence, developments in ML algorithms have grown rapidly. We can mention some of the most used ML algorithms in CVD fields, which we also use during this paper: DT (Decision Trees), RF (Random Forests), LR (Logistic Regression), and SVM (Support Vector Machine).

Nowadays, hybrid MLs are also widely used. These aim to combine the predictions from different models into a single prediction, leveraging the strengths and mitigating the weaknesses of single modes. This process can be done through different ensemble methods, which we have also used during this work: Voting Classifier, Bagging, Boost, and Stacking.

Feature selection could also be applied to any individual or hybrid MLs for better results. These techniques are important in MLs for several reasons, such as model improvement, computational efficiency, data quality enhancement, and high-dimensional data management.

In this paper, we will evaluate the performance of several ML algorithms, including hybrid ones, by applying suitable feature selection methods. We use RFE (Recursive Feature Elimination), Feature Selection, Select Percentile, Lasso Regression and GA (Genetic Algorithms). We aim to identify the most effective approach for maximizing precision and accuracy on the given CVD dataset.

The rest of the paper is organized as follows: Section 2 provides a literature review background. Section 3 examines the suggested system and methodology used, followed by Section 4, which presents the actual design and implementation of the proposed system. In Section 5, we discuss the paper's results and draw a final conclusion in Section 6.

2 Background

CVD has inspired many studies that include ML applications and their comparisons. In this section, we will summarize the most relevant papers that use ML techniques to cope with CVD or other medical issues.

The study [2] uses PPG (PhotoPlethysmoGraphy) to identify cardiovascular abnormalities. The research uses PPG signals and feature selection-based classifiers to identify cardiorespiratory disorders. Seven classifiers were used, and from the obtained results, the NB classifier achieved the highest accuracy in both cases with and without FS. The results highlight the potential of PPG as a reliable diagnostic tool.

The article [3] uses various ML techniques such as SVR (Support Vector Regression), Adaptive Multivariate Regression Splines, M5Tree model, Neural Networks, ANFIS (Adaptive Neuro-Fuzzy Inference System), Nearest Neighbor/Naive Bayes classifiers, and statistical approaches to provide seventeen CVD risk factors and to classify CVDs. A real-world dataset is studied to evaluate hybrid models against existing ML models. The study shows that of all the other methods tested, ANFIS has the highest accuracy prediction, 96.56%, which is better than other statistical and ML methods. Some of the other factors include the age of the patient and amounts of cholesterol and glucose. The study suggests that such a relationship exists among the complex and non-linear CVS factors.

Research [4] proposes an approach that uses DL (Deep Learning) techniques to identify important features and enhance prediction accuracy. It presents a new hybrid DL intelligent system that has been developed. The model, combining MDenseNet201 (Modified DenseNet201) for feature extraction and IDRSNet (Improved Deep Residual Shrinkage Network) for prediction, achieves an accuracy of 99.12% on the UCI dataset.

Paper [5] examines how filtering-based feature selection techniques affect classification in ML. It investigates correlation-based methods (Pearson, Spearman, and Kendall) and statistical methods (mutual information, chi-squared score, ANOVA, ROC-AUC (Receiver Operating Characteristic curve and the Area under the Curve)) on classifiers such as KNN, SVM, DT, and GNB (Gaussian Naïve Bayes). All the analysis is done over a fetal heart rate dataset. Experiments showed that using statistical feature selection techniques improved the performance of GNB and KNN by 3%, while results with DT and SVM improved by 4% when correlation-based techniques were used. Overall, the statistical methods ANOVA and ROC-AUC improved accuracy by 92%. From the study, we can deduce that feature selection techniques can improve the accuracy of different classification methods.

The research [6] investigates the use of the Activity Index for monitoring physical activity. The study compared Activity Index values across different activity groups and hand usage scenarios. The Activity Index was then tested with SVM, KNN, and RF Algorithms as a classification feature. The RF algorithm achieved the highest accuracy (97% F1 score) for classifying hand usage.

Even in previous studies, we have focused on ML applications in the medical field [7]. We used DL models to diagnose illnesses from X-ray chest images,

examining the CNN+VGG19 (Convolutional Neural Network + Visual Geometry Group) Deep Learning architecture before and after modifications. The updated model showed signs of overfitting, suggesting the need for regularization techniques or avoiding additional extraction and classification layers to improve performance.

The studies mentioned in this paper emphasize the value of early detection and the importance of using ML to increase diagnosis accuracy. ML and hybrid ML methodologies have shown huge potential in diagnosing CVD diseases. Most cases have used feature selection techniques for fine-tuning data inputs and choosing the most influencing factors. Further, hybrid ML approaches using ML models with optimization strategies have shown increased effectivity and accuracy predictions. Prediction can be improved by combining ML, hybrid models, and sophisticated feature selection.

3 Methodology

The work presented in this paper focuses on FS methods. We compare different individual ML algorithms with and without a FS technique and hybrid methods combined with hybrid FS techniques.

3.1 Dataset

The data set we use is the Cleveland Clinic Heart Disease Dataset, obtained from Kaggle, which consists of 303 records. The estimated parameters to predict the probability of heart disease will be included in the FS process before being used in the selected algorithms. Understanding the input is important for better analysis. The dataset contains features like age, sex, CP (Chest Pain), treetops (Resting Blood Pressure), cholesterol, classification (the presence or not of the disease), etc. The dataset target values show patients' presence (value 1) or absence (value 0) of heart disease.

3.2 ML models used in the analysis

The selected ML models are the ones that are mostly used in the field of CVD prediction, Fig. 1.

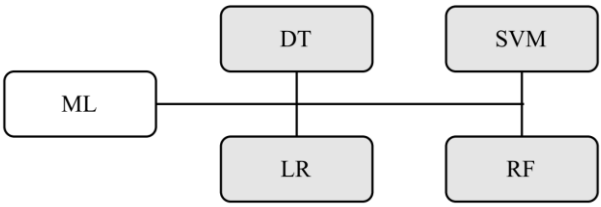

Fig. 1 – *ML models used in this paper.*

1. DT [8, 9] use a recursive partitioning of the input, taking decisions at each iteration step. A decision tree includes a root node (starting point), branches (arrows connecting nodes), interior nodes (nodes with children), and leaf nodes (nodes without children, representing possible target values). The branching factor denotes the number of children per node. The final step of this method is prediction. Decision Trees' algorithm is simple to understand and interpret, and it handles both classification and regression.
2. LR [9]. Logistic regression is a fundamental ML algorithm, and even though the name has regression in it, it is a classification algorithm rather than a regression one. Logistic regression performs best on large datasets where target values occur equally and is unsuitable for datasets with high multicollinearity among independent variables. It is a simple, efficient, and easy-to-use algorithm, but it can be sensitive to outliners and limited to binary classification. The algorithm in [9] is implemented through scikit-learn (Python).
3. SVM [9]. An SVM ML algorithm can be used for classification, outlier detection and regression. It can maximize the margin between different classes and minimize classification errors. SVMs perform well when there is a clear separation between classes but are unsuitable for large datasets. Implementations of SVM are done through the scikit-learn (Python) library.
4. RF [9]. Another ML algorithm widely used both for classification and regression is Random Forest. It is very powerful and well-known for producing reliable results. RF creates multiple decision trees from subsets of data and combines their outputs, reducing overfitting and variance, thus improving accuracy. They are stable and less affected by noise. RF can handle large datasets but is computationally expensive and can have performance problems with very high-dimensional sparse data. Implementations of Random Forest are done in the scikit-learn (Python) library.

3.3 Ensemble methods

Combining different ML models to form an ensemble can often improve performance compared to individual models. We will use some of the most known ensemble methods to create hybrid ML models. Fig. 2 shows a summary of the ones used in our analysis.

The following is a summary of ensemble techniques used in this paper:

1. Voting classifier [10]. Voting classifiers are ensemble learning techniques that combine the predictions of multiple models to improve overall performance. By leveraging the strengths of different classifiers, a voting classifier can achieve better generalization.

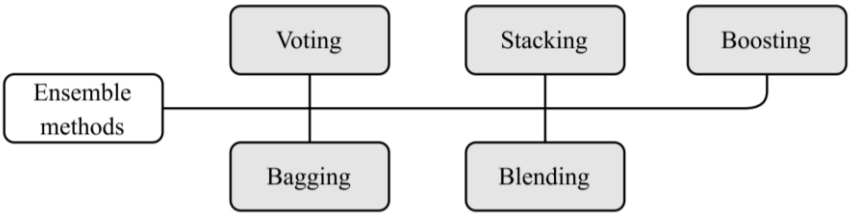

Fig. 2 – *Ensemble methods used in this study.*

2. Bagging (Bootstrap Aggregating) [11]. It builds multiple models on different subsets of the training data and then averages the predictions. Aggregating predictions from randomly generated training sets improves the classification performance of ML models.
3. Boosting [11]. This method trains models sequentially, meaning each step corrects the errors of its predecessor. Each subsequent model focuses on the misclassified instances of the previous models. The main idea behind boosting is to iteratively apply the base learning algorithm to modified versions of the input data.
4. Stacking (Meta-Ensemble) [11]. It is similar to a Voting classifier but trains a meta-model that learns how to combine the predictions of multiple base models best by training on their outputs. A voting classifier simply aggregates the predictions through majority voting or averaging but without involving additional learning.
5. Blending [12]. This method is similar to stacking but typically simpler. It operates by training multiple models on the entire dataset and then blending their predictions using a weighted average or a simple model. It involves trying out several data splits, resulting in different train and test sets for the base models, and then using only the test sets to train the metamodel; this technique is known as blending.

3.4 FS (Feature Selection) Techniques

The types of FS used in this paper are summarized in Fig. 3.

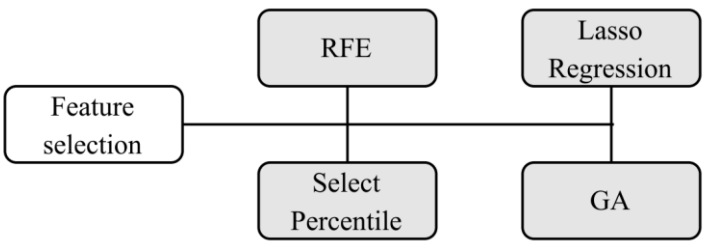

Fig. 3 – *Feature selection methods used in this paper.*

1. RFE (Recursive Feature Elimination) [13] is a feature selection technique commonly used to identify the most relevant features in a dataset. It works recursively, meaning it removes less important features after each iteration. This reduces the overfitting phenomenon and improves model description.
2. The Select Percentile [14] feature selection method is used in ML to select the top features based on their scores relative to the percentile of the highest-scoring features. Leveraging `Select Percentile` for feature selection can effectively improve the efficiency and performance of ML models.
3. Lasso Regression [15] (Least Absolute Shrinkage and Selection Operator) is a linear regression technique used to improve the prediction accuracy and interpretability of the model. It adds a penalty term to the OLS (Ordinary Least Squares) objective function, encouraging sparsity in the coefficient vector by driving some coefficients to zero. This technique is beneficial for data with many features and prevents overfitting.
4. GA (Genetic Algorithms) [16] are used for optimization, hyperparameter tuning, FS, or model architecture search. GAs are evolutionary algorithms inspired by natural selection and genetics, where solutions evolve over generations to optimize an objective function.

Python offers several libraries, such as DEAP (Distributed Evolutionary Algorithms in Python), PyGAD (Python Genetic Algorithm Library), or custom implementations to integrate GAs with different ML algorithms.

4 Design and Implementation

This study tested two different approaches and flows. The first one is Flow 1, which gives the flow related to the performance evaluation of individual MLs with and without FS techniques. The provided algorithm outlines a comprehensive process for building, optimizing, and evaluating ML models.

On the other hand, Flow 2 presents the flow of hybrid FS methods performed over hybrid ML models. Analysis and metrics visualization are all done in Python and related libraries.

4.1 Flow 1 – Individual MLs with and without FS

The initial dataset with different features is used as the input for Flow 1. After importing the necessary libraries for data visualization and including individual ML algorithms, the dataset is prepared and transformed. This process converts categorical features to numerical ones by encoding and splitting the data into training and testing sets.

The individual classifiers for DT, LR, RF and SVM algorithms are initialized, trained and evaluated by computing the selected performance metrics without feature selection, and the results are visualized with confusion matrices.

Subsequently, each feature selection technique (RFE, Select Percentile, Lasso Regression, and GA) is applied to each ML algorithm, followed by hyperparameter tuning and model evaluation.

Flow 1: *Prediction flow for DT, LR, RF, SVM*

Step 1. *Input the dataset.*
Step 2. *Prepare and transform the data. Split the dataset into training and testing sets for evaluation. Identification of features and targets.*
Step 3. *Execute steps 4 - 9 for each ML algorithm.*
Step 4. *Initializing individual classifiers (Decision Tree, Logistic Regression, SVM, and Random Forest).*
Step 5. *Training.*
Step 6. *Evaluation function. Compute and return various performance metrics without feature selection.*
Step 7. *Execute steps 8-9 for each of the feature selection techniques used*
Step 8. *Select the feature selection technique.*
Step 9. *Hyperparameter tuning and model evaluation*
Step 10. *Compare the metrics reached after each round of testing.*

Finally, the metrics obtained after each round of testing (without and with FS) are compared to assess performance. **Table 1** summarizes some metrics obtained for each algorithm with and without FS techniques.

Table 1
Metrics of ML algorithms with and without Feature Selection.

	No Feature selection		RFE		Select Percentile		Lasso Regression		GA	
	Precision	Accuracy	Pr.	Acc.	Pr.	Acc.	Pr.	Acc.	Pr.	Acc.
DT	0.79	0.79	0.74	0.74	0.75	0.73	0.74	0.72	0.79	0.79
LR	0.84	0.84	0.83	0.84	0.73	0.72	0.8	0.8	0.8	0.8
SVM	0.85	0.85	0.81	0.81	0.75	0.74	0.8	0.8	0.87	0.87
RF	0.82	0.82	0.79	0.79	0.78	0.77	0.77	0.77	0.87	0.87

4.2 Flow 2 – Ensembled ML with hybrid FS

Testing involves data preparation, selecting key features, and training different hybrid models. Therefore, multiple feature selection techniques are combined to leverage their individual strengths. To take the best of each feature selection technique, they are combined and applied to hybrid Machine Learning

models. These ML hybrid models are built using different ensemble methods to create a stronger workflow that improves both feature selection and model performance. Each ML method has its strengths and hybrid approaches may capture different advantages.

The second flow represents our work based on the study and synthesis of existing literature related to this area. After studying various methods, we customized our approach to fit the tests of interest for this algorithm.

The Flow 2 algorithm combines multiple FS techniques and uses a GA to optimize the FS and preprocessing steps. It builds a final pipeline that integrates the selected features and trains different classifiers, evaluating the model with various metrics and visualizing the results with a confusion matrix. It compares different ensemble methods using four FS techniques combined under the same flow. **Table 3** summarizes some metrics obtained for each hybrid case combined with all FS techniques.

Flow 2: *Prediction flow for hybrid cases*

Step 1. *Input the dataset.*

Step 2. *Prepare and transform the data. Split the dataset into training and testing sets for evaluation. Identification of features and targets.*

Step 3. *Initialize classifiers: Decision Tree, Logistic Regression, SVM, and Random Forest.*

Step 4. *Initialize feature selection techniques: RFE, Select Percentile, Lasso (SelectFromModel).*

Step 5. *Use genetic algorithm for feature selection:*

– *Initialize TPOT Classifier (Tree-based Pipeline Optimization Tool) with desired parameters.*

– *Fit TPOT Classifier on scaled training data.*

– *Extract the best pipeline from the fitted TPOT Classifier.*

– *Extract preprocessing steps from the best pipeline.*

– *Transform training and test data using the preprocessing pipeline.*

Step 6. *Combine feature selection techniques (RFE, Select Percentile, and Lasso selectors).*

Step 7. *Build and train the final pipeline based on the combination of step 6 and the prechosen ensemble method over transformed training data.*

Step 8: *Evaluate the hybrid model on the testing data using various performance metrics (accuracy, precision, recall F1, plot confusion matrix).*

Step 9. *Compare the evaluated metrics*

The algorithm begins with the dataset as input. The data is prepared and transformed by splitting it into training and testing sets and identifying features and targets. So, 80% of the data is allocated to train the algorithm, while 20% is used to test its performance. It continues with initializing classifiers, including DT, LR, SVM, and RF. In the next step, feature selection techniques such as RFE, Select Percentile, and Lasso (SelectFromModel) are initialized.

Then, a GA is used for feature selection by initializing the TPOTClassifier with desired parameters, fitting it on scaled training data, extracting the best pipeline, and transforming both training and test data using the preprocessing steps from the pipeline. Then, a combination of FS techniques is done.

The final pipeline is built and trained using the combined FS techniques and the prechosen ensemble method on the transformed training data. The hybrid model is evaluated on the testing data using various performance metrics, such as accuracy, precision, recall, and F1 score, and includes plotting a confusion matrix. Finally, the last step compares the evaluated metrics.

5 Results

Our experiments cover four ML models tested with and without FS. In addition, we use hybrid ML methods created by combining four MLs simultaneously. Hybrid algorithms are created using different ensemble techniques. Several feature selection techniques are combined with hybrid selection methods to refine selection efficiency further.

We rely on different performance metrics to evaluate and compare the performance of these ML and hybrid models. The performance metrics derived from these experiments are used to compare the models and identify the most efficient solutions. Performance metrics or evaluation metrics obtained from the execution of our tests are summarized in **Tables 1** and **3**.

5.1 Evaluation process

Evaluating the performance of an ML or hybrid model is one of the crucial steps of the model development process or workflow. Various performance metrics or evaluation metrics are used to evaluate the performance or quality of the model. All ML models aim to generalize well to unseen/new data, and performance metrics help determine how well the model generalizes to the new data set.

Accuracy

The accuracy metric is defined as the number of correct predictions over the total number of predictions as in Equation (1):

$$Accuracy = Nr.\ of\ correct\ predictions/Total\ Nr.\ of\ predictions. \quad (1)$$

Precision

The precision metric defines the percentage of positive predictions that were correct (Equation 2):

$$Precision = TP/(TP+FP). \quad (2)$$

The Confusion Matrix

A confusion matrix is a tabular representation of the prediction results of each binary classifier. It is represented by a matrix, where the columns are forecast values, and the rows specify actual values. Actual and forecast rows give two possible classes: Yes or No. Therefore, if we predict the presence of a disease in a patient, the prediction column with Yes means the patient has the disease, and for No, the patient does not have the disease.

Recall

It resembles the Precision metric; however, recall is intended to account for the percentage of actual positives 3). It can be counted as a True Positive or a true prediction over the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (True Positive and False Negative). The formula for calculating recall (3):

$$Recall = TP/(TP+FN). \quad (3)$$

To minimize false negatives, the *Recall* should be closer to 100%, and if we want to minimize false positives, then precision should be closer to 100%. So, if we maximize precision, we will minimize FP error, and if we maximize recall, we will minimize FN error.

F1-score

The F1-score is the harmonic mean of precision and recall, providing a single measure that balances both false positives and false negatives (4):

$$F1 = 2((precision \cdot recall) / (precision + recall)). \quad (4)$$

The F1 score is particularly useful when the class distribution is imbalanced.

5.2 Results from individual ML algorithms

This session will discuss the DT, LR, SVM, and RM test results. All the models are trained without and with FS techniques, including RFE, Select Percentile, Lasso Regression, and GA.

Table 1 shows the accuracy and precision of ML models without and with selection techniques. Each row relates to an ML model, and each column heading combines FS and model evaluation metrics (precision and accuracy).

The DT model applied to the selected dataset achieves the highest precision and accuracy without FS. If FS is used, then RFE, Select Percentile, and Lasso Regression slightly decrease precision and accuracy compared to the base (no FS). GA provides performance metrics that are the same as the baseline.

The results show that similar to the decision tree model, the logistic regression (LR) model achieves the highest precision and accuracy without feature selection (FS). When employing FS methods like Recursive Feature Elimination (RFE) and Select Percentile, there is a noticeable decrease in both precision and accuracy compared to the baseline (with no FS). Among these methods, Select Percentile leads to the lowest precision and accuracy. In contrast, Lasso Regression and Genetic Algorithms (GA) were observed to be close to the baseline values, albeit slightly lower than when no FS was applied. The SVM model reaches the highest precision and accuracy with FS. When using FS methods like RFE, Lasso Regression, and Select Percentile, precision and accuracy decrease compared to the baseline (no FS). Select Percentile results are the lowest. GA significantly improves the performance of the SVM model, achieving the highest precision and accuracy among all methods considered.

The last row of the table presents the performance metrics of an RF model. When using FS methods like RFE, Select Percentile, and Lasso Regression for model tuning, precision and accuracy decrease compared to the baseline. GA significantly improves the performance of the RF model.

The confusion matrix significantly represents how well the TP and TN are predicted. **Table 2** displays the confusion matrices of the best cases reached by each ML.

Figs. 4 and 5 display the cases that reached the best recall values and F1 scores for each of the ML algorithms. For both recall and F1-score, all the algorithms reached the best performance under GA except LR, which reached the best in the case without FS.

Table 2
Confusion matrixes of individual ML algorithms.

Method		Predicted Negative	Predicted Positive
DT without FS	Actual Negative	19	9
	Actual Positive	9	24
LR without FS	Actual Negative	22	6
	Actual Positive	4	29
SVM With GA	Actual Negative	25	4
	Actual Positive	4	28
RF with GA	Actual Negative	24	5
	Actual Positive	3	29

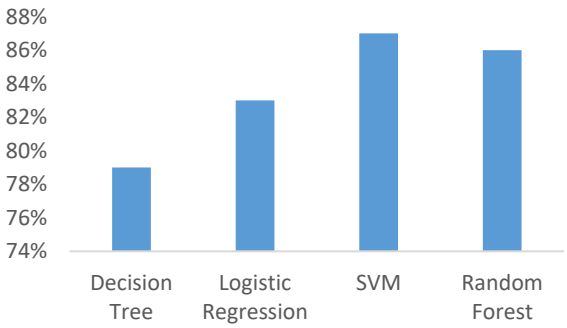

Fig. 4 – *Best recall (in %) for each individual ML algorithm.*

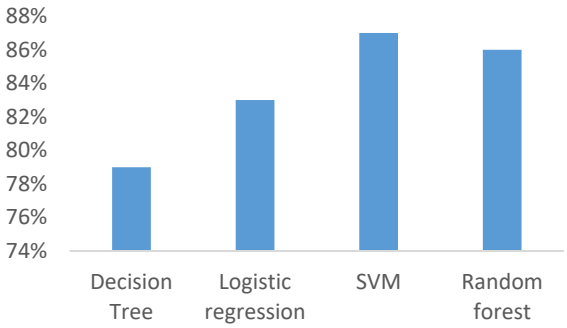

Fig. 5 – *Best F1-score (in %) for each individual ML algorithm.*

5.3 Results from hybrid approaches

We have applied four feature selection techniques to various hybrid Machine Learning methods (developed through different ensemble techniques). After applying TPOS, GA is used to optimize the ML pipeline. Then, this data is used as input and combined with three other FS techniques. Consequently, ensemble techniques then combine all four MLs (DT, LR, SVM, and RF), which are applied, and at the end, pipeline construction and training are done. **Table 3** shows the performance evaluation reached by the process explained above.

The metrics show that our models perform well, with high accuracy, precision, recall, and F1 score, indicating a good balance between precision and recall. The ROC AUC score further confirms the model's strong discriminatory power.

The Voting Classifier shows the highest accuracy and precision, making it the most reliable in terms of overall correctness and minimizing false positives. The F1 score is also the highest, indicating a good balance between precision and recall.

Table 3
Metrics of hybrid algorithms with combined Feature Selection.

	Accuracy	Precision	Recall	F1	ROC AUC
Voting Classifier	0.89	0.90	0.88	0.89	0.89
Bagging	0.85	0.87	0.84	0.86	0.85
Boosting	0.80	0.86	0.75	0.80	0.81
Stacking	0.87	0.85	0.91	0.88	0.87

Bagging has solid but lower scores across all metrics compared to the Voting Classifier. It performs well but is not as strong as the Voting Classifier in any metric.

Boosting has the lowest accuracy and recall, indicating it struggles with correctly identifying True Positives. Its precision is relatively high, meaning it is often correct when predicting a positive but misses more actual positives.

Stacking excels at recall, indicating it identifies true positives very well. Its slightly lower precision suggests it may have more false positives than the Voting Classifier. The F1 score and ROC AUC are strong, indicating balanced performance and good discrimination ability. **Table 4** displays the confusion matrices for different hybrid approaches.

Table 4
Confusion Matrix for Voting, Bagging, Boosting, Stacking

Method		Predicted Negative	Predicted Positive
Voting	Actual Negative	26	3
	Actual Positive	4	28
Bagging	Actual Negative	25	4
	Actual Positive	5	27
Boosting	Actual Negative	25	4
	Actual Positive	8	24
Stacking	Actual Negative	24	5
	Actual Positive	3	29

6 Conclusion

This paper aims to identify the ML (Machine Learning) /hybrid algorithm that performs better with FS (Feature Selection) techniques based on the used dataset. This paper's contribution and tests relate to determining whether an

individual or hybrid model performs best when combined with FS (Feature Selection) techniques.

Hybrid models (Voting, Bagging, Boosting, and Stacking) perform better than individual ML algorithms (Decision Trees, Logistic Regression, Support Vector Machine, and Random Forest) across most metrics.

Based on the used dataset, the results showed that for individual ML (Machine Learning) algorithms, DT (Decision Tree) generally achieves the best precision and accuracy without FS (Feature Selection). LR (Logistic Regression) consistently performs well with different FS (Feature Selection) techniques, with slight variation in precision and accuracy. SVM (Support Vector Machine) achieves precision and accuracy of 0.87 with GA (Genetic Algorithms), the same as the RF (Random Forests) method. Models like SVM (Support Vector Machine) and RF (Random Forest) benefit notably from FS.

SVM is the best-performing algorithm for the used dataset, which further improves if used together with GA (Genetic Algorithms) feature selection. These results show that even though FS (Feature Selection) can help make simpler models, a method should be carefully considered based on the specific dataset and model. In the case of hybrid models, the Voting Classifier stands out due to its high accuracy and precision, balanced recall, high F1 score and ROC AUC. It provides a strong overall performance, making it the most reliable and well-rounded model for this dataset.

Let's compare all the individual and hybrid models. Overall, based on the provided data, the model created with Voting Classifier seems to be the best-performing model among both the individual classifiers and the hybrid classifiers, offering the highest accuracy and precision. The Voting Classifier is a leading performer overall- it shows the highest accuracy of 0.89 and maintains a strong balance between F1 score, precision and recall. Bagging also has a good performance but is slightly lower in recall. Boosting excels in precision but has challenges with recall, scoring 0.75. Stacking demonstrates excellent recall at 0.91 and maintains a solid balance among other metrics.

Feature selection plays an important role in improving model outcomes. In addition, ensemble techniques have been shown to enhance performance.

It must be emphasized that the specific characteristics and requirements of the task are of high importance. We can mention accuracy, speed, and scalability. Based on this, additional analyses may be necessary to confirm the findings. This study has limitations due to the small dataset; therefore, more varied data could give more accurate and reliable results. Another concern to consider is the cost and complexity of hybrid algorithms. It is necessary to provide more training and develop more advanced coding to optimize algorithms, reduce runtime, and address the increased processing power needed for training. We aim to address these drawbacks in the future.

Future research will require a broader collection of datasets, more ensemble models, and other FS (Feature Selection) techniques.

7 References

[1] S. Dattani, V. Samborska, H. Ritchie, M. Roser: Cardiovascular Diseases, Our World in Data, Available at: https://ourworldindata.org/cardiovascular-diseases

[2] A. S. Al Fahoum, A. O. Abu Al-Haija, H. A. Alshraideh: Identification of Coronary Artery Diseases Using Photoplethysmography Signals and Practical Feature Selection Process, Bioengineering, Vol. 10, No. 2, February 2023, p. 249.

[3] O. Taylan, A. S. Alkabaa, H. S. Alqabbaa, E. Pamukçu, V. Leiva: Early Prediction in Classification of Cardiovascular Diseases with Machine Learning, Neuro-Fuzzy and Statistical Methods, Biology, Vol. 12, No. 1, January 2023, p. 117.

[4] M. Mandava, S. Reddy Vinta: MDensNet201-IDRSRNet: Efficient Cardiovascular Disease Prediction System Using Hybrid Deep Learning, Biomedical Signal Processing and Control, Vol. 93, July 2024, p. 106147.

[5] I. J. Jebadurai, G. J. Leelipushpam Paulraj, J. Jebadurai, S. Silas: Experimental Analysis of Filtering-Based Feature Selection Techniques for Fetal Health Classification, Serbian Journal of Electrical Engineering, Vol. 19, No. 2, June 2022, pp. 207 – 224.

[6] J. Medarević, M. Novičić, M. Marković: Feasibility Test of Activity Index Summary Metric in Human Hand Activity Recognition, Serbian Journal of Electrical Engineering, Vol. 19, No. 2, June 2022, pp. 225 – 238.

[7] E. Zanaj, L. Balliu, G. Basha, E. Gjata, E. K. Meçe: Studying the Behavior of a Modified Deep Learning Model for Disease Detection Through X-ray Chest Images, International Journal of Advanced Computer Science and Applications, Vol. 15, No. 5, May 2024, pp. 843 – 850.

[8] A. Amelia: Decision Tree: Part 2/2 - Towards Data Science, Medium, December 2021, Available at: https://towardsdatascience.com/decision-tree-part-2-34b31b1dc328

[9] D. Papakyriakou, I. S. Barbounakis: Data Mining Methods: A Review, International Journal of Computer Applications, Vol. 183, No. 48, January 2022, pp. 5 – 19.

[10] Ensembles: Gradient Boosting, Random Forests, Bagging, Voting, Stacking, User Guide, Available at: https://scikit-learn.org/stable/modules/ensemble.html#voting-classifier

[11] I. D. Mienye, Y. Sun: A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects, IEEE Access, Vol. 10, September 2022, pp. 99129 – 99149.

[12] A. Chatzimparmpas, R. M. Martins, K. Kucher, A. Kerren: Empirical Study: Visual Analytics for Comparing Stacking to Blending Ensemble Learning, Proceedings of the 23rd International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, May 2021, pp. 1 – 8.

[13] RFE, API, Available at:

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

[14] SelectPercentile, API, Available at:

[15] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectPercentile.html

[16] T. Hastie, R. Tibshirani, J. Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Springer, New York, 2009.

A. Kapoor: Hands-On Artificial Intelligence for IoT, Packt Publishing, January 2019.