Optimizing Machine Learning Classifiers for Enhanced Cardiovascular Disease Prediction

Sultan Munadi Alanazi
Department of Computer Science, Science College, Northern Border University, Arar, Saudi Arabia
sultan.aalanazi@nbu.edu.sa

Gamal Saad Mohamed Khamis
Department of Computer Science, Science College, Northern Border University, Arar, Saudi Arabia
gamal.khamees@nbu.edu.sa (corresponding author)

Received: 29 November 2023 | Revised: 24 December 2023 | Accepted: 29 December 2023

 DOI: https://doi.org/10.48084/etasr.6684

ABSTRACT

A key challenge in developing Machine Learning (ML) models for predicting or diagnosing Cardiovascular Disease (CVD), is selecting suitable algorithms and fine-tuning their parameters. In this study, we employed three ML techniques, namely Auto-WEKA, Decision Table/Naive Bayes (DTNB), and Multiobjective Evolutionary (MOE) fuzzy classifier to create diagnostic models using the Heart Disease Dataset from IEEE Dataport. Auto-WEKA generated a highly accurate model with a 100% success rate through optimal classifier selection and hyperparameter configuration. The DTNB classifier yielded a satisfactory 85.63% prediction accuracy concerning patients' risk levels. Further refinements, though, could help reduce possible misclassifications. Finally, the MOE fuzzy classifier achieved approximately 81.6% accuracy, indicating the potential for enhancing precision and recall values by adjusting classifier settings. Our findings underscore the promise of ML tools in CVD diagnosis and suggest further optimization of classifier parameters for superior performance.

Keywords-Machine Learning (ML); Auto-WEKA; Lazy IBK; DTNB; MOE fuzzy classifier; CVD

I. INTRODUCTION

As reported by the World Health Organization, Cardiovascular Disease (CVD) ranks as the primary death cause globally [1]. CVD occurs when the human heart struggles to function effectively and becomes incapable of supplying adequate blood to other bodily regions, therefore resulting in heart failure [2]. By facilitating early detection and professional prognosis, it is possible to reduce patient mortality rates. Several factors, such as medical history, age, gender, and lifestyle choices influence the prevalence of CVD. Adopting a healthy lifestyle can mitigate CVD risks by managing cholesterol levels and maintaining optimal blood pressure. Numerous studies have employed data mining techniques to identify methods which will reduce health risks and detect heart conditions quickly. The vast amount of data available in the healthcare sector allows these techniques to uncover previously unnoticed patterns and extract essential information that can support effective decision making. Data mining and machine-learning technologies are particularly valuable in healthcare due to their ability to analyze extensive volumes of data, ultimately yielding actionable insights [3-5]. However, diagnosing heart diseases quickly and accurately remains a significant challenge. Conventional diagnostic methods include blood tests, chest X-rays, and electrocardiograms (ECG). Recently, Machine Learning (ML) and artificial intelligence methods have emerged as critical contributors in the medical field. Thus, various machine-learning and deep learning models can now be harnessed for disease diagnosis while offering classifications or outcome predictions. Furthermore, multiple studies have investigated the use of different ML models to categorize and predict CVD patients effectively. This study aims to meticulously examine and delineate the optimization of machine-learning classifiers for improving heart disease prediction accuracy through the evaluation of Auto-WEKA, DTNB, and MOE Fuzzy System performance.

II. LITERATURE REVIEW

Regarding heart disease prediction, numerous studies have utilized various ML methods for diagnostic purposes. For example, a previous study employed the ordinary learning method [5], which achieved an accuracy of approximately 98% using clinical data from the UCI standard Cleveland dataset. Other approaches have utilized the long short-term memory with symbolic aggregate approximation for ECG-based categorization [6], resulting in a 98.4% accuracy rate. Additionally, ML algorithms have been applied to identify key features that improve prediction accuracy [7] using a Support Vector Machine (SVM), with the obtained results showing an impressive accuracy rate of 91%. Alternative methods include the majority voting ensemble model [8], which demonstrated a 90% accuracy rate based on low-cost medical tests. The extreme learning machine algorithm has also been utilized [9] to construct a diagnosis model, achieving an accuracy rate of approximately 80%. Moreover, the effective heart disease prediction system [10] displayed a 100% prediction rate engaging a neural network. Previous comparative studies [11] found that neural networks outperformed convolutional neural networks in terms of heart disease diagnosis in most instances. Improvements have been made by integrating naive Bayes classifiers with decision tables [12] and applying multiobjective evolutionary algorithms for fuzzy classification in different medical contexts [13]. In a more comprehensive study, various ML algorithms and deep learning techniques were implemented to analyze the UCI Heart Disease Dataset [14]. For instance, authors in [15] proposed a method that utilizes machine learning and enhanced auto categorical particle swarm optimization for early heart disease prediction, achieving 98% accuracy with logistic regression and SVMs on the Statlog and Cleveland datasets.

Selecting the appropriate algorithm and fine-tuning its parameters, constitutes a significant challenge in ML, often leading to suboptimal results due to improper decisions. The Auto-WEKA tool, integrated within the open-source WEKA package, addresses this issue by automating the process of algorithm selection and parameter adjustment for classification models. Auto-WEKA employs Bayesian optimization to map hyperparameters to algorithms, thereby enhancing their performance. Being compatible with any system that supports WEKA, Auto-WEKA operates similarly to other WEKA classifiers. It automatically identifies the optimal model and its parameters through Sequential Model-based Algorithm Configuration (SMAC) [16]. This method takes into consideration the noise in function evaluations and provides a reliable estimate of the best configuration performance [17].

The DTNB approach is a hybrid technique that leverages the strengths of both decision tables and naive Bayes classifiers. A decision table serves as a mechanism for constructing and utilizing a hybrid classifier. At each point during the search process, the DTNB algorithm assesses the benefit of dividing the dataset attributes into two disjoint subsets, one for the decision table and one for naive Bayes. A forward selection search technique is employed, in each step of which the selected attributes are modeled by naive Bayes and the remaining attributes are modeled by the decision table. It is important to note that all attributes are initially modeled by the decision table. In every step, the algorithm also considers the possibility of removing an attribute entirely from the model. The DTNB method strengths lie in its ability to capture complex relationships in the target data while maintaining simplicity. The decision table classifier is adept at handling complex interactions among attributes; however, it may suffer from overfitting when faced with a large number of irrelevant features. On the other hand, Naive Bayes classifiers perform well in high-dimensional spaces due to their attribute independence assumption. Nevertheless, they may be less effective when there are strong dependencies between attributes [12]. By combining the decision table and Naive Bayes classifiers, the DTNB method offers the following advantages over each individual method [12]:

- Improved accuracy. The hybrid DTNB method frequently produces better classification results by exploiting both complex attribute interactions and independent feature probabilities.
- Reduced overfitting. By considering both decision rules and feature independence, the model prevents sole reliance on complex interactions that may result in overfitting.
- Scalability. The DTNB method handles high-dimensional data efficiently because it exploits the advantages of both methods.

The construction of the fuzzy rule-based classifier is accomplished by employing the ENORA Multiobjective Evolutionary (MOE) algorithm within the multiobjective evolutionary fuzzy classifier. This process entails the optimization of two distinct objectives. The primary objective can be customized to either maximize accuracy, maximize the area under the ROC curve, or minimize the root mean squared error. Conversely, the secondary objective focuses on minimizing the number of fuzzy rules associated with the classifier. Ultimately, the non-dominated solutions in the final population exhibiting optimal fitness for the primary objective are selected as output [13]. This extensive body of research serves as a foundation for our investigation, through which we attempt to optimize the ML classifiers Auto-WEKA, DTNB, and evolutionary fuzzy systems to realize improved CVD prediction performance by providing a comprehensive performance assessment of these systems.

III. DATASET AND METHODOLOGY

A. Dataset

In this study, we used the Heart Disease Dataset [15], which was obtained from IEEE Dataport. This dataset contains 12 input features, as shown in Table I.

B. Method

We present a method to classify cardiovascular patients by leveraging the dataset and employing the Auto-WEKA, DTNB, and MOE fuzzy classifiers. The proposed methodology comprises the following steps.

1. Data preprocessing. First, we acquire the CVD dataset and perform necessary preprocessing steps to ensure data consistency and quality. These procedures include error removal, handling missing values, encoding categorical variables, and data normalization or standardization.
2. Feature selection. We identify the most significant features for predicting CVDs in the dataset using various methods, e.g. correlation-based analysis, chi-squared tests, and mutual information, to determine the most impactful features for classification.
3. The Auto-WEKA classifier is trained on the training set. Auto-WEKA is an automated ML tool designed for classification tasks that iteratively selects suitable algorithms and hyperparameters.

4. The DTNB classifier is trained using the training set. This classifier combines the decision table and Naive Bayes algorithms to maintain sufficient accuracy while minimizing computational costs.

5. The MOE fuzzy classifier is trained on the training set to optimize accuracy, interpretability, and simplicity concurrently using genetic algorithms while incorporating fuzzy logic techniques in managing imprecision and uncertainty.

C. Model Evaluation

In order to evaluate the performance of the trained classifiers, various performance metrics are utilized on the test set. These key evaluation metrics comprise accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic (AUROC) curve. To better understand these metrics, we will delve into their definitions, calculations, and potential relationships. TP, TN, FP, and FN denote True Positives, True Negatives, False Positives, and False Negatives, respectively.

- Accuracy measures the proportion of correct predictions out of the total number of predictions made. It is calculated as (TP+TN)/(Total instances). However, when the data are imbalanced with uneven class distribution, accuracy may not be a reliable metric.
- Precision focuses on the ratio of TP to total predicted positives (TP / (TP + FP)). It essentially gauges how well a model can identify true positive instances while avoiding false positive instances.
- Also known as sensitivity or True Positive Rate (TPR), recall assesses the capability of a model to correctly identify positive cases from all actual positive cases. It is calculated as: TP / (TP + FN).
- F1-score is the harmonic mean of precision and recall that allows for a balance between both metrics making suitable to evaluate class imbalance scenarios where low prevalence rates are observed. It is defined as 2 × ((Precision * Recall) / (Precision + Recall)).
- The AUROC curve summarizes the performance of a classifier across all possible threshold levels by plotting the TPR against the False Positive Rate (FPR). Higher area under this curve, typically ranging from 0 to 1, indicates better classification performance.

Based on our analysis and the proposed methodology, we classified cardiovascular patients using Auto-WEKA, DTNB, and the MOE fuzzy classifiers. To ensure the highest accuracy and effective model selection, we implemented cross-validation methods or adjusted the dataset splits based on the obtained accuracy values. This approach allows for a rigorous evaluation and optimization of classifier performance, while mitigating concerns of overfitting related to specific training sets. By meticulously assessing the performance of each classifier—according to accuracy, precision, recall, F1-score, and AUROC curve—on different dataset splits or cross-validation folds, we are able to draw more reliable conclusions about the performance of each classifier. This process aids in selecting the most appropriate method for CVD classification.

TABLE I. HEART DISEASE DATASET FEATURES

Feature	Feature discretion	Value range
age	Years	[29, 77]
sex	Sex	0 = female 1 = male
cp	Chest pain	0 = typical angina 1 = atypical angina 2 = nonangina pain 3 = asymptomatic
trestbps	Resting blood pressure (mmHg) on hospital admission	[94, 200]
chol	Serum cholesterol (mg/dl)	[126, 564]
fbs	Fasting blood sugar > 120 mg/dl	0 = false 1 = true
restecg	Resting electrocardiographic results	0 = normal 1 = ST-T wave abnormality 2 = left ventricular hypertrophy
thalach	Maximum heart rate	[71, 202]
exang	Exercise-induced angina	0 = no 1 = yes
oldpeak	ST depression induced by exercise relative to rest	[0, 6.2]
slope	Slope of the peak exercise ST segment	0 = up sloping 1 = flat 2 = down sloping
Target	Class	0: no 1: yes

IV. EXPERIMENTS

A. Experiment 1

Auto-WEKA was utilized to optimize machine-learning algorithms and their respective hyperparameters. By leveraging the automated search process offered by WEKA's extensive toolkit, this experiment simplified the process of model selection and configuration, leading to highly accurate predictions and increased efficiency in handling complex data analysis tasks. In pursuit of unmatched performance, Experiment 1 rigorously assessed various algorithmic combinations. This approach allowed researchers to uncover insightful patterns and make well-informed decisions across numerous domains. The classifier was applied to the option values depicted in Figure 1. Detailed explanations of these options are provided in Table II. Auto-WEKA was directly implemented on the Heart Disease Dataset via the Auto-WEKA panel. This involved launching the tool on the dataset, which was used as the training set, in accordance with the option values outlined in Figure 1.

The results are presented in Table III. As can be observed, the Auto-WEKA tool selected lazy.IBk [18] as the best classifier. This model delivers an impressive accuracy of 100%, and a kappa statistic of 1, signifying flawless classifier performance [19]. Table VI provides a detailed breakdown of accuracy results by class, while Table VII exhibits the confusion matrix for the corresponding model. The remarkable performance of this model can be attributed to two primary factors: the selection of an appropriate classifier and the correct setting of classifier hyperparameters, as evidenced by the values in Table II and Figure 1. Figure 2 depicts the ROC curve for the Auto-WEKA model, illustrating the model's performance in terms of the true positive rate versus the false positive rate.

TABLE II. AUTO-WEKA OPTIONS

seed	Seed for the random number generator
memLimit	Memory limit for runs (MB)
parallel runs	Number of runs to perform in parallel experiment
numDecimalPlaces	Number of decimal places in the output of numbers in the model
batchSize	Preferred number of instances to process if a batch prediction is performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size.
timeLimit	Time limit for tuning (minutes)
debug	If true, the classifier may output additional info to the console
best configs	How many best configurations should be returned as output
doNotCheckCapabilities	If set, classifier capabilities are not checked before the classifier is built (use with caution to reduce runtime)
metric	Target metric to optimize

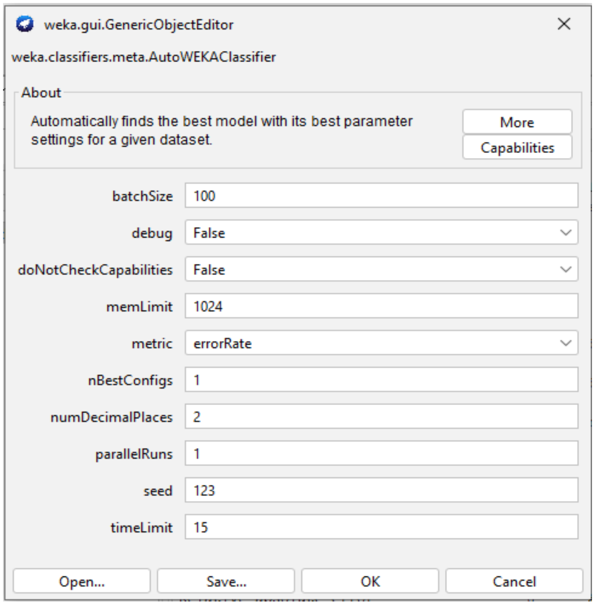

Fig. 1. Auto-WEKA settings.

TABLE III. AUTO-WEKA CLASSIFIER OUTPUT

Best classifier	**WEKA.classifier.lazy.IBK**
Correctly classified instances	1190 (100%)
Incorrectly classified instances	0 (0%)
Kappa Statistic	1
Mean absolute error	0.03
Root mean squared error	0.06
Relative absolute error	0%
Root relative squared error	0%
Total number of instance	1190

TABLE IV. AUTO-WEKA ACCURACY BY CLASS

TPR	**FPR**	**Precision**	**Recall**	**F1-score**	**MCC**	**ROC Area**	**PRC Area**	**Class**
1	0.0	1.00	1.00	1.00	1.00	1.00	1.00	no
1	0.0	1.00	1.00	1.00	1.00	1.00	1.00	yes

TABLE V. AUTO WEKA CONFUSION MATRIX

Actual	**Predicted**		
	Output	**Yes**	**No**
	Yes	560	0
	No	0	629

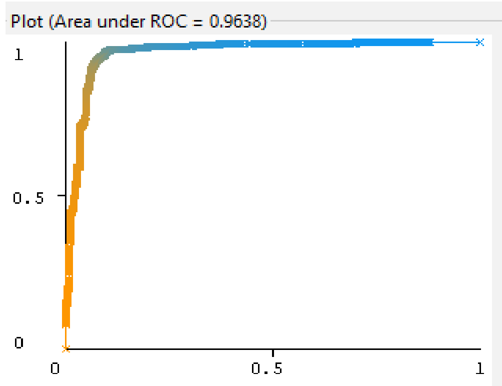

Fig. 2. ROC curve for the Auto-WEKA model.

B. Experiment 2: DTNB Algorithm

In this experiment, we employed the DTNB algorithm from the WEKA software, following this scheme: weka.classifiers.rules.DTNB-X 1. The dataset was preprocessed to include only pertinent features such as age, chestpain type, resting bps, fasting blood sugar, resting ecg, and class. For feature selection, we used the cross-validation method (leave one out). In this context, we conducted a 10-fold stratified cross-validation to assess the classification model performance, therefore ensuring a constant distribution of classes for each fold. TABLE VI and Figure 3 illustrate the DTNB classifier options and settings, respectively.

TABLE VI. DTNB CLASSIFIER OPTIONS

numDecimalPlaces	Number of decimal places used for the output of numbers in the model.
batchSize	Preferred number of instances to process if batch prediction is performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size.
debug	If true, the classifier may output additional information to the console.
doNotCheckCapabilities	If set, classifier capabilities are not checked before the classifier is built (use with caution to reduce runtime).
evaluationMeasure	Measure used to evaluate the performance of attribute combinations used in the decision table.
search	Search method used to find good attribute combinations for the decision table.
displayRules	Sets whether rules are to be printed.
useIBk	Sets whether Simple instance-based learner that uses the class of the nearest k training instances(IBk)should be used rather than the majority class.
crossVal	Sets the number of folds for cross validation (1 = leave one out).
metric	Target metric to optimize

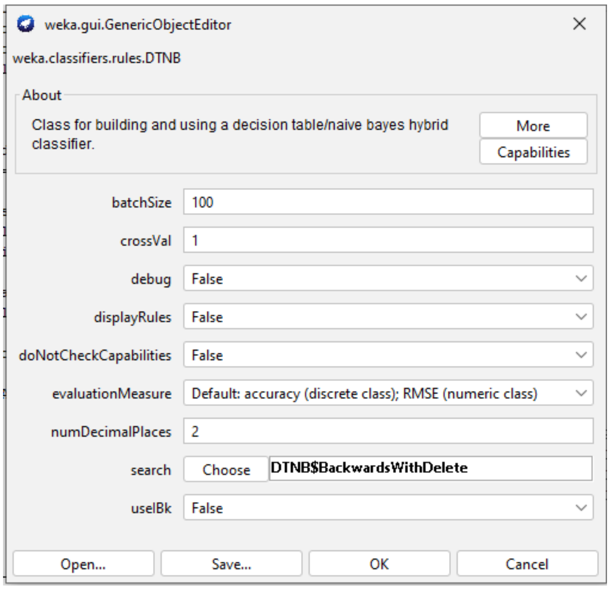

Fig. 3. DTNB settings.

We constructed the DTNB model in 0.24 s. It comprised 108 rules and covered all potential nonmatching instances by the majority class. Table VII presents the accuracy measurements during the cross-validation analysis, along with other significant evaluation metrics. Additionally, Table VIII provides detailed accuracy for each class and Table IX depicts the confusion matrix for the corresponding model. The area under the ROC curve of the DTNB model was found to be 0.9153, which highlights its efficacy in distinguishing between classes.

TABLE VII. DTBN CLASSIFIER OUTPUT

Correctly classified instances	1019 (85.63%)
Incorrectly classified instances	171 (14.36%)
Kappa statistic	0.7115
Mean absolute error	0.1849
Root mean squared error	0.1849
Relative absolute error	37.092%
Root relative squared error	68.511%
Total number of instance	1190

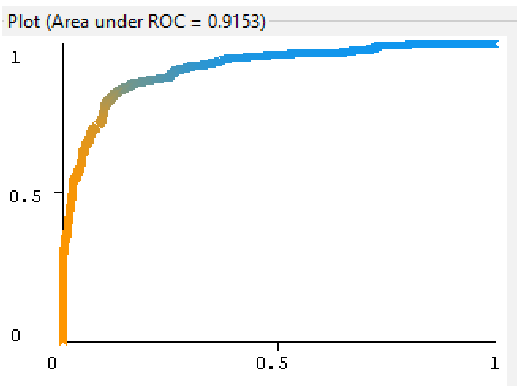

Fig. 4. ROC curve for the DTNB classifier.

TABLE VIII. TDBN ACCURACY BY CLASS

TPR	FPR	Precision	Recall	F1 Score	MCC	ROC Area	PRC Area	Class
0.841	0.130	0.852	0.841	0.847	0.712	0.915	0.914	no
0.870	0.159	0.860	0.870	0.865	0.712	0.915	0.916	yes

TABLE IX. TDNB CLASSIFIER CONFUSION MATRIX

Actual	**Predicted**		
	Output	**Yes**	**No**
	Yes	472	89
	No	82	547

C. Experiment3: MultiObjective Evolutionary Fuzzy Classifier

In this experiment, we employed the MOE algorithm. The dataset was preprocessed to include only pertinent features such as age, chestpain type, resting bps, fasting blood sugar, resting ecg, and class. For feature selection, we used 10-fold stratified cross-validation to assess the performance of the classification model, thereby ensuring a constant distribution of classes for each fold. TABLE X and Figure 5 illustrate the MOE classifier options and settings, respectively.

TABLE X. MOE FUZZY CLASSIFIER OPTIONS

seed	Set the random seed.
maxRules	Set the value for the maximum number of rules (default: 10 + number of class labels).
minV	Set the value by which the domain of the variable is divided to obtain the minimum variance.
populationSize	Set the number of individuals in the population.
generations	Set the number of generations to evolve the population.
reportFrequency	Set the frequency to print the status of the evolutionary search.
numDecimalPlaces	Number of decimal places used in the output of numbers in the model.
batchSize	Preferred number of instances to process if batch prediction is being performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size.
logFile	Set the name for the log file.
maxSimilarity	Set the maximum similarity value for the fuzzy sets.
algorithm	Set the algorithm.
debug	If true, the classifier may output additional information to the console.
doNotCheckCapabilities	If set, classifier capabilities are not checked before the classifier is built (use with caution to reduce runtime).
evaluationMeasure	Set the evaluation criteria.
maxV	Set the value by which the domain of the variable is divided to obtain the maximum variance.
maxLabels	Set the value of maximum number of labels.

TABLE XI. OUTPUT OF MULTIOBJECTIVE EVOLUTIONARY FUZZY CLASSIFIER.

Correctly classified instances	972 (81.6807 %)
Incorrectly classified instances	218 (18.3193 %)
Kappa statistic	0.6307
Mean absolute error	0.2374
Root mean squared error	0.428
Relative absolute error	36.7584
Root relative squared error	85.742
Total number of instance	1190

Table XI presents the accuracy measurements during the cross-validation analysis along with other significant evaluation metrics. Additionally, Table XII provides detailed accuracy for each class, and Table XIII presents the confusion matrix for the corresponding model. The area under the ROC curve of the MOE model was found to be 0.8139.

TABLE XII. DETAILED ACCURACY RESULTS FOR THE MULTI-OBJECTIVE EVOLUTIONARY FUZZY CLASSIFIER

TPR	FPR	Precision	Recall	F1 Score	MCC	ROC Area	PRC Area	Class
0.763	0.135	0.834	0.763	0.797	0.633	0.814	0.7480	no
0.865	0.237	0.804	0.865	0.833	0.633	0.814	0.766	yes

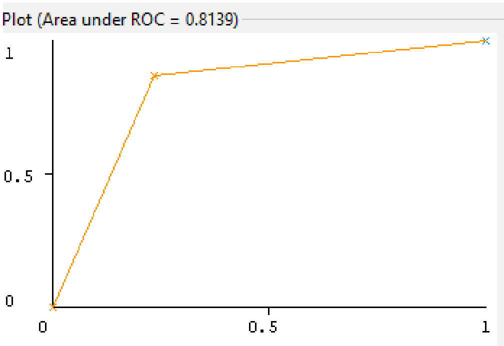

Fig. 5. ROC curve for MOE fuzzy classifier.

TABLE XIII. MOE FUZZY CLASSIFIER CONFUSION MATRIX

Actual	Predicted		
	Output	Yes	No
	Yes	428	133
	No	85	455

V. CONCLUSION

In this study, we utilized three machine learning tools: Auto-WEKA, DTNB, and the MOE fuzzy classifier. These tools were used to construct diagnostic models for heart disease, employing the Heart Disease Dataset from IEEE Dataport. The Auto-WEKA tool automatically identifies the most suitable model and optimizes parameter settings for classification or regression tasks. In addition, we also investigated the hybrid DTNB and MOE fuzzy classifiers to achieve optimal performance.

Our findings suggest that the Auto-WEKA tool was effectively utilized to construct a highly accurate model, with lazy.IBk emerging as the best classifier. This model achieved an impressive accuracy rate of 100%. This success can be primarily attributed to the efficient selection of classifiers and the appropriate configuration of hyperparameters. Our study also revealed that the DTNB classifier developed a model with an accuracy of approximately 85.63% in predicting patients' risk levels based on their medical records. This demonstrates satisfactory performance in making reliable predictions. However, further enhancements may be necessary to reduce misclassification instances.

Finally, the MOE fuzzy classifier attained an overall accuracy of approximately 81.6%. Although this classifier made satisfactory distinctions among patient classes, there is still potential for improvement in the precision and recall results. Future studies could explore this potential by adjusting various classifier settings, such as the population size and the number of generations. Table XIV offers a comparison between the outcomes derived from default settings and those achieved after fine-tuning the classifier's hyperparameters. It becomes clear that the precise configuration of these settings leads to an enhancement of the overall results.

To contextualize our findings with other studies that have adopted machine learning methods for heart disease prediction, our work demonstrates varying degrees of success in comparison with alternative approaches. These include ordinary learning methods [5], long short-term memory with symbolic aggregate approximation [6], support vector machines [7], majority voting ensemble models [8], extreme learning machine algorithms [9], effective heart disease prediction systems [10], and neural networks [14]. Notable accomplishments from these studies encompass accuracy rates ranging between 80% and 100%. This underscores the effectiveness of machine learning tools in aiding heart disease diagnosis using standard datasets.

In conclusion, our study's findings underscore the effectiveness of machine learning tools in facilitating heart disease diagnosis through standard datasets. However, optimizing performance may require fine-tuning critical classifier parameters to yield more precise results, thereby opening avenues for future research. This study adds to the burgeoning body of research on the application of machine learning techniques in medical diagnosis and introduces innovative approaches to optimize heart disease diagnostic models

TABLE XIV. RESULT COMPARISON FOR DEFAULT AND ADJUSTED PARAMETER SETTINGS

Classifier / Metrics	MOE Fuzzy		TDBN		AUTO-WEKA	
	Default setting Accuracy	Adjusted setting Accuracy	Default setting Accuracy	Adjusted setting Accuracy	Default setting Accuracy	Adjusted setting Accuracy
Correctly classified instances	76.2%	81.6%	79.5%	85.6%	90.5%	100%
Incorrectly classified instances	3.73%	18.3%	20.4%	14.3%	9.41%	0%
Kappa statistic	0.52	0.63	0.58	0.71	0.81	1
Mean absolute error	0.23	0.23	0.24	0.18	0.09	0.03
Root mean squared error	0.48	0.428	0.40	0.18	0.30	0.06

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2024-1060-01".

REFERENCES

[1] M. Armbrust, A. D. Joseph, R. H. Katz, and D. A. Patterson, "Above the clouds : A Berkeley view of cloud computing," *Science*, vol. 53, no. 4, pp. 50-58, 2010, https://doi.org/10.1145/1721654.1721672.

[2] A. L. Bui, T. B. Horwich, and G. C. Fonarow, "Epidemiology and risk profile of heart failure," *Nature Reviews Cardiology*, vol. 8, no. 1, pp. 30-41, Jan. 2011, https://doi.org/10.1038/NRCARDIO.2010.165.

[3] B. Trstenjak, D. Donko, and Z. Avdagic, "Adaptable Web Prediction Framework for Disease Prediction Based on the Hybrid Case Based Reasoning Model," *Engineering, Technology & Applied Science Research*, vol. 6, no. 6, pp. 1212–1216, Dec. 2016, https://doi.org/ 10.48084/etasr.753.

[4] R. Ramesh and S. Sathiamoorthy, "A Deep Learning Grading Classification of Diabetic Retinopathy on Retinal Fundus Images with Bio-inspired Optimization," *Engineering, Technology & Applied Science Research*, vol. 13, no. 4, pp. 11248–11252, Aug. 2023, https://doi.org/ 10.48084/etasr.6033.

[5] G. Someshwaran and V. Sarada, "A Research Review on Fetal Heart Disease Detection Techniques," in *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, Feb. 2022, pp. 1674–1681, https://doi.org/10.1109/ICICCS53718. 2022.9788226.

[6] M. Liu and Y. Kim, "Classification of Heart Diseases Based On ECG Signals Using Long Short-Term Memory," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Honolulu, HI, USA, Jul. 2018, pp. 2707–2710, https://doi.org/10.1109/EMBC.2018.8512761.

[7] M. N. R. Chowdhury, E. Ahmed, Md. A. D. Siddik, and A. U. Zaman, "Heart Disease Prognosis Using Machine Learning Classification Techniques," in *2021 6th International Conference for Convergence in Technology (I2CT)*, Maharashtra, India, Apr. 2021, https://doi.org/ 10.1109/I2CT51068.2021.9418181.

[8] R. Atallah and A. Al-Mousa, "Heart Disease Detection Using Machine Learning Majority Voting Ensemble Method," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, Amman, Jordan, Jul. 2019, https://doi.org/10.1109/ICTCS.2019.8923053.

[9] S. Ismaeel, A. Miri, and D. Chourishi, "Using the Extreme Learning Machine (ELM) technique for heart disease diagnosis," in *2015 IEEE Canada International Humanitarian Technology Conference (IHTC2015)*, Ottawa, ON, Canada, Feb. 2015, https://doi.org/10.1109/ IHTC.2015.7238043.

[10] P. Singh, S. Singh and G. S. Pandi-Jain. journal of, and undefined 2018, "Effective heart disease prediction system using data mining techniques," *International Journal of Nanomedicine*, vol. 13, no. T-NANO 2014 Abstracts, pp. 121-124, 2018, https://doi.org/ 10.2147/IJN.S124998.

[11] C.-H. Lin, P.-K. Yang, Y.-C. Lin, and P.-K. Fu, "On Machine Learning Models for Heart Disease Diagnosis," in *2020 IEEE 2nd Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS)*, Tainan, Taiwan, May 2020, pp. 158–161, https://doi.org/ 10.1109/ECBIOS50299.2020.9203614.

[12] M. A. Hall and E. Frank, "Combining Naive Bayes and Decision Tables," in *Proceedings of Twenty-First International Florida Artificial Intelligence Research Society Conference*, Coconut Grove, FL, USA, May 2008, pp. 318–319.

[13] F. Jiménez, G. Sánchez, and J. M. Juárez, "Multi-objective evolutionary algorithms for fuzzy classification in survival prediction," *Artificial Intelligence in* Medicine, vol. 60, no. 3, pp. 197-219, 2014, https://doi.org/10.1016/J.ARTMED.2013.12.006.

[14] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, "Prediction of heart disease using a combination of machine learning and deep learning," *Computational Intelligence and Neuroscience*, vol. 2021, 2021, Art. no. 8387680, https://doi.org/ 10.1155/2021/8387680.

[15] A. K. Dubey, A. K. Sinhal, and R. Sharma, "An Improved Auto Categorical PSO with ML for Heart Disease Prediction," *Engineering, Technology & Applied Science Research*, vol. 12, no. 3, pp. 8567–8573, Jun. 2022, https://doi.org/10.48084/etasr.4854.

[16] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA," *Journal of Machine Learning Research*, vol. 18, pp. 1-5, Mar. 2017, https://doi.org/10.1007/978-3-030-05318-5_4.

[17] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques,"* 4th ed. Burlington, NJ, USA: Morgan Kaufmann, 2016.

[18] R. E. Schapire, "Explaining AdaBoost," in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, B. Schölkopf, Z. Luo, and V. Vovk, Eds. Berlin, Heidelberg, Germany: Springer, 2013, pp. 37–52, https://doi.org/10.1007/978-3-642-41136-6_5.

[19] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: the kappa statistic," *Family Medicine*, vol. 37, no. 5, pp. 360–363, Dec. 2005.