IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 4, August 2025, pp. 2797~2804
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i4.pp2797-2804

Integrating random forest and genetic algorithms for improved kidney disease prediction

Bommanahalli Venkatagiriyappa Raghavendra[1], Anandkumar Ramappa Annigeri[1], Jogipalya Shivananjappa Srikantamurthy[1], Gururaj Raghavendrarao Sattigeri[2]
[1]Department of Mechanical Engineering, JSS Academy of Technical Education, Visvesvaraya Technological University, Bengaluru, India
[2]Department of Mechanical Engineering, Vishwanathrao Deshpande Institute of Technology, Haliyal, India

Article Info

Article history:

Received Nov 21, 2024
Revised Jun 16, 2025
Accepted Jul 10, 2025

Keywords:

Chronic kidney disease
Genetic algorithm
Machine learning
Optimization
Predictive model
Random forest

ABSTRACT

This work offers a novel method for predicting chronic kidney disease (CKD) by combining random forest (RF) classification with genetic algorithm (GA) to optimize important parameters. The dataset comprises 1,659 patients with 51 clinical parameters. The suggested method emphasizes the optimization of random state values, test size, and essential hyperparameters, such as the number of trees in the forest, the least number of samples needed at a leaf node, and the smallest number of samples necessary to split an internal node. The optimization process is conducted in two stages: the first stage optimizes the random state and test size, while the second stage focuses on hyperparameters. Through extensive simulations over 50 runs, the study demonstrates that the optimized model achieves an accuracy ranging from 0.9451 to 0.9738. The results indicate a maximum increase in accuracy of 2.09%, showcasing the effectiveness of the GA-RF integrated approach in enhancing model performance. This work provides valuable insights into the impact of parameter optimization on machine learning (ML) models, particularly in medical diagnostics, and offers a robust framework for developing highly accurate predictive models.

Corresponding Author:

Bommanahalli Venkatagiriyappa Raghavendra
Department of Mechanical Engineering, JSS Academy of Technical Education
Visvesvaraya Technological University
Bengaluru, India
Email: bvrjss@gmail.com

1. INTRODUCTION

The training and testing method for classifying biological information in machine learning (ML) is becoming increasingly significant. The researcher focuses carefully on choosing the right methods of classification and developing the predictive model. Few studies have examined classification and predictive model development. Literature studies are predominantly theoretical; there exists no practical model for sample selection in the training and testing process. The research focuses on type of classification, data training and testing is found the grey area. In order to build and fit a good model in ML, this article provides an in-depth analysis of the sizes of both the training and testing datasets. This work examines the effect of random state (shuffling the data set), test size (data set for training and testing), and hyper-parameters on performance in random forest (RF) ML. Accuracy is a metric that compares the performance metrics of the test and the predicted data from the model.

The genetic algorithm (GA) is integrated into RF classification in two stages. The purpose of the GA is to optimize the parameters during the process of model building. The simulation results (accuracy)

show that the random state, test ratio, and hyper-parameters are the criteria that have a direct impact on the model's correctness. To create a reliable system, a GA should be implemented into the RF classification to fine-tune the correctness of the results.

The RF algorithm is a powerful and versatile ML method, suitable for classification and regression. The features of the RF algorithm, such as accuracy, robustness in handling missing values, scalability, nonparametric support to handling complex interactions, robustness to noisy data and fit the effective model for prediction in diagnosing kidney disease. Many randomized algorithms in sklearn use the random state to select the random seed to feed to the pseudo-random number generator. The most popular integers are 0 and 42. When using an integer for the random state (changing the order of training samples yields varied results), the function will produce the same results across different executions. The results only change if the integer value is altered, and increasing the number of runs will likely decrease the variance. The reason for variance in performance is a sample that is too small (and/or many features/classes which is too high), which causes the models to over-fit. To decrease the variance by increasing the number of runs.

Chronic kidney disease (CKD) is a long-term health condition that typically lasts a lifetime and arises due to kidney cancer or diminished kidney function. The progression of this chronic illness can be stopped or slowed down to the point where the patient's life can be sustained only through dialysis or surgery [1]. Earlier detection of CKD is difficult among patients, due to no symptoms and varying rates of kidney disease progression. Timely and precise prediction of kidney disease is essential for effective disease management [2]. The third objective of the UN's sustainable development goals (SDG) focuses on good health and well-being, highlighting the growing challenges posed by non-communicable diseases. One of the SDG targets for 2030 is to reduce premature deaths from non-communicable diseases by one-third [3]. Kidney injury is irreversible and can advance to end-stage renal disease (ESRD), eventually requiring renal replacement therapy (RRT) due to the loss of remaining kidney function [4]–[6]. According to Zhao *et al*. [7], treating CKD and renal failure is expensive and often ineffective. Early and accurate diagnosis, along with timely treatment, is essential for effective management of CKD. This study aims to design and validate a predictive model for identifying CKD. In previous research, Pal [4] employed three ML algorithms—logistic regression (LR), decision tree (DT), and support vector machine (SVM)—to construct a predictive model. Similarly, Khalid *et al.* [8] introduced a hybrid approach that integrated Gaussian naïve Bayes (for gradient boosting) and a DT as the base learner, with a RF model serving as the meta-classifier. Debal and Sitote [3] investigated CKD using predictive models such as RF, SVM, and DT. In another study, Saif *et al.* [9] proposed three different models aimed at predicting CKD 6 to 12 months before clinical symptoms appear, employing sophisticated approaches like convolutional neural networks (CNNs), long short-term memory (LSTM) models, and deep ensemble learning techniques. Rahman *et al.* [10] focused on enhancing classification performance by applying feature selection methods such as recursive feature elimination (RFE) and the Boruta algorithm, along with multiple performance metrics, to identify optimal classifiers, striking a balance between high accuracy and low computational cost.

Additionally, Lei *et al*. [2] conducted a comprehensive meta-analysis to evaluate how accurately ML techniques can diagnose the progression of kidney disease. Dritsas and Trigka [11] proposed a ML-based strategy for assessing CKD risk, leveraging a range of models including probabilistic, tree-based, and ensemble approaches such as SVM, LR, stochastic gradient descent (SGD), artificial neural networks (ANN), and k-nearest neighbors (k-NN). Lei *et al*. [2] performed a systematic meta-analysis to assess the diagnostic accuracy of ML algorithms for kidney disease progression. Dritsas and Trigka [11] developed an ML methodology to predict CKD risk, utilizing probabilistic, tree-based, and ensemble learning models, including SVM, LR, SGD, ANN, and k-NN. Lim *et al*. [12] reviewed CKD and noted that Cox regression modeling was the most commonly used method among the few studies examined. Aoki *et al*. [13] explored the application of ML techniques, including RF survival models, to study CKD in the U.S., focusing on laboratory-derived risk factors as predictors of estimated glomerular filtration rate (eGFR). Binsawad [14] analyzed the correlation between kidney function and electrocardiogram (ECG) readings using an optimized RF model, demonstrating superior performance in terms of classification accuracy (CA), false positive rate (FPR), and true positive rate (TPR) when compared to other methods. Hema *et al.* [15], utilizing both standard and real-time datasets, assessed the effectiveness of various ML algorithms—such as k-NN, RF, DT, gradient boosting, and extreme gradient boosting (XGBoost)—in forecasting CKD. Takkavatakarn *et al*. [16] focused on stage-4 CKD, employing four different models—LASSO regression, RF, XGBoost, and ANN—to predict the progression to end-stage kidney disease (ESKD). Sanmarchi *et al.* [17] provided a comprehensive review of ML methodologies used in CKD research, outlining both the potential advantages and limitations of these techniques in diagnosis, prognosis, and disease management. Zhu *et al*. [18] developed a pipeline to process longitudinal electronic health records (EHRs) and applied recurrent neural networks (RNNs) to forecast the progression of CKD from stages II/III to IV/V. Additionally, Ghosh and

Khandoker [19] evaluated ML-based prediction models for CKD and highlighted the interpretability of Shapley additive explanations (SHAP) and local interpretable model-agnostic explanations (LIME) frameworks, which offer valuable clinical insights for healthcare professionals. Alturki *et al*. [20] studied CKD prediction using RF with accuracy of 92.85% with synthetic minority oversampling technique (SMOTE). Rahman *et al*. [10] used ML algorithm for CKD prediction with 99.75% accuracy. Rajeashwari and Arunesh [21] used deep convolutional neural network (DCNN) and modified extreme random forest (MERF) approaches were used to predict CKD with 98.5% accuracy.

From the literature review, it is learned that several studies are available on CKD prediction through machine-learning approaches. In the study of CKD, various parameters like dataset size, quality of dataset, and the timing of data collection play a crucial role. In the present study, a focus on CKD prediction using ML models for the big dataset is considered by a two-stage hybrid model of RF and GA. By this approach, the accuracy of the hybrid model improves.

2. METHOD

Diagnosing kidney disease is a crucial aspect of healthcare, which focuses on improving health through the prevention, diagnosis, treatment, and management of diseases and impairments. The early diagnosis helps society to undergo early treatment and hence avoid further damage to the health. Data about kidney disease is collected from open source at https://www.kaggle.com/datasets/rabieelkharoua/chronic-kidney-disease-dataset-analysis. The data of 1,659 persons tested for 51 various parameters before diagnosing the CKD.

A predictive model is generated using 1,659 persons with 51 tested parameters. The RF classifier is utilized for categorizing the data to fit the model and to predict. The effectiveness of the model is determined through the accuracy, which compares or deviation of the data considered and data generated through the model. Higher accuracy implies the model is effective in prediction. The effectiveness of the model in terms of accuracy in the RF classification depends upon various parameters viz, random state, test size, and hyperparameters. Key hyperparameters in RF classification include the number of trees (n_estimators), which indicates how many DTs the model will generate during training, selecting the best one through majority voting. Other important parameters are the minimum number of samples required at a leaf node (min_samples_leaf), which ensures that a split point at any depth leaves at least the specified minimum number of training samples in both branches, and the minimum number of samples needed to split an internal node.

In this research, a GA is employed to enhance the model's accuracy in two stages. Stage 1, during the classification of the data, and stage 2, in tuning the hyperparameters. In stage 1, the precision of the model in terms of accuracy is analyzed by shuffling the data sets along with the test dataset size. The optimization of these two parameters focuses on accuracy through the use of a GA. Figure 1 shows architecture of the process.

Stage 1: The lower and upper bound of the random state and test size is set as [13]

Random_state = [0, 100]

Test_size = [0.1, 0.5]

Stage 1&2 :

GA parameters set to

Maximum number of iterations :30

Population size :30

Mutation probability = 0.1

Elit ratio = 0.01

Crossover probability = 0.85

Parents portion = 0.3

Crossover type = Uniform

GA is used to optimize hyperparameters for RF classifications employing evolutionary strategies to search for the best hyperparameter set. GA are based on natural selection principles and employ techniques like selection, crossover, and mutation to evolve solutions to optimization problems. The optimized random state and test size, determined for model accuracy, are subsequently used as input parameters in the second level of the GA to refine the hyperparameters of the RF classifier. This additional step enhances the hyperparameters to achieve better accuracy for model fitting and prediction [22]–[24].

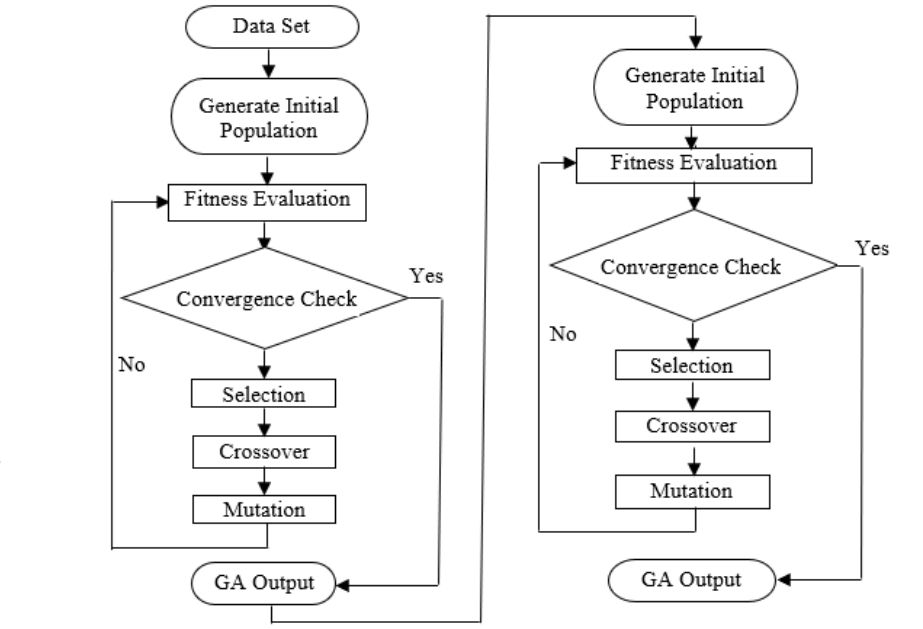
This study focuses on optimizing the random state, test size, and three key hyperparameters: the n_estimators, the min_samples_leaf, and the minimum number of samples required for a split.

The simulation is terminated if the conditions (lower/and upper bound) are not satisfied

Hyper parameters lower and upper bound set to : [25]

Number of trees (n_estimators) = [1, 100]

Minimum samples leaf (min_samples_leaf) = [1, 10]
Minimum samples to split() = [2, 10]



Stage 1: Random forest classification using GA to optimize random state and test size

Stage 2: Hyper parameter tuning using GA

Figure 1. Architecture of GA integrated with RF

3. RESULTS AND DISCUSSION

The predictive model is generated and simulations were performed with RF classification as follows the accuracy obtained in these simulations is tabulated:

– Test size 10 to 50 and Random State 0 to 100
– Integrating GA with RF in two levels,
  Stage 1: To optimize test size and random state,
  Stage 2: Input of optimized test size and random state to optimize hyperparameters such as N estimators, minimum leaf, and minimum split. The objective of the GA is to improve the accuracy of the classification by selecting appropriate effecting parameters.

The simulations were conducted for test size 10 to 50 and random state 0 to 100. The accuracy obtained is tabulated in Table 1. The simulation result in Table 1 reveals that a maximum accuracy of 0.9518 is obtained for test size 10 for random stages 10 & 100. Integrating GA with RF in two levels, stage 1: to optimize test size and random state, stage 2: input of optimized test size and random state to optimize hyperparameters such as N Estimators, minimum leaf, and minimum spit. A total of 50 simulation runs were conducted, and the findings are summarized in Table 2.

Table 1. Simulation result of test size and random state on accuracy

Test size	Random state										
	0	10	20	30	40	50	60	70	80	90	100
	Accuracy										
10	0.8916	0.9518	0.9337	0.9036	0.9337	0.9157	0.9337	0.9036	0.9036	0.9398	0.9518
15	0.8956	0.9116	0.9438	0.9157	0.9277	0.9197	0.9277	0.8916	0.9116	0.9197	0.9157
20	0.9006	0.9127	0.9367	0.9157	0.9187	0.9096	0.9277	0.9066	0.9187	0.9066	0.9157
25	0.9012	0.8988	0.9181	0.9205	0.9205	0.9181	0.9205	0.9060	0.9157	0.9060	0.9084
30	0.9076	0.8956	0.9197	0.9237	0.9197	0.9137	0.9217	0.9157	0.9177	0.9116	0.9096
35	0.9105	0.9002	0.9157	0.9243	0.9243	0.9191	0.9208	0.9157	0.9243	0.9105	0.9053
40	0.9142	0.9066	0.9142	0.9217	0.9292	0.9187	0.9187	0.9172	0.9187	0.9157	0.9066
45	0.9170	0.9090	0.9116	0.9183	0.9210	0.9183	0.9157	0.9264	0.9237	0.9157	0.9116
50	0.9157	0.9096	0.9072	0.9229	0.9193	0.9205	0.9157	0.9277	0.9205	0.9120	0.9157

Table 2. Simulation result of GA integrated RF in two levels

No. of Run	Stage 1: Output of GA to optimize random state and test size			Stage 2: Output of GA to optimize hyper-parameters			
	Random state	Test size		Random state	Test size		Random state
1	97	0.416	0.9507	26	2	8	0.9522
2	34	0.103	0.9593	8	2	5	0.9650
3	44	0.207	0.9623	13	1	9	0.9652
4	86	0.100	0.9581	31	1	3	0.9640
5	14	0.110	0.9529	85	2	3	0.9528
6	44	0.180	0.9658	7	8	7	0.9623
7	77	0.110	0.9581	14	1	3	0.9685
8	14	0.220	0.9539	22	1	6	0.9593
9	3	0.200	0.9451	4	7	9	0.9542
10	20	0.160	0.9513	9	2	5	0.9550
11	77	0.128	0.9624	26	2	10	0.9671
12	37	0.130	0.9509	11	2	5	0.9554
13	97	0.350	0.9504	30	2	2	0.9504
14	9	0.130	0.9577	17	1	5	0.9624
15	12	0.380	0.9504	17	6	5	0.9504
16	14	0.220	0.9526	26	1	2	0.9582
17	44	0.140	0.9612	27	7	4	0.9655
18	44	0.150	0.9592	47	1	2	0.9632
19	14	0.220	0.9539	4	5	5	0.9566
20	44	0.100	0.9651	8	4	10	0.9651
21	21	0.120	0.9598	12	5	6	0.9648
22	34	0.110	0.9558	12	1	7	0.9613
23	89	0.110	0.9572	10	5	10	0.9626
24	32	0.100	0.9538	11	5	9	0.9595
25	21	0.144	0.9585	21	2	8	0.9627
26	77	0.130	0.9630	13	4	7	0.9722
27	97	0.360	0.9502	31	4	7	0.9518
28	14	0.220	0.9570	56	2	6	0.9569
29	86	0.100	0.9540	3	7	6	0.9655
30	9	0.110	0.9529	6	2	2	0.9738
31	14	0.190	0.9527	17	1	5	0.9558
32	44	0.200	0.9585	33	1	10	0.9614
33	9	0.120	0.9543	28	4	10	0.9593
34	44	0.600	0.9590	5	10	4	0.9664
35	34	0.110	0.9545	18	1	4	0.9600
36	14	0.220	0.9530	26	1	3	0.9558
37	44	0.130	0.9628	13	1	3	0.9720
38	14	0.220	0.9562	7	7	6	0.9616
39	21	0.210	0.9590	3	2	6	0.9692
40	14	0.200	0.9527	61	1	4	0.9556
41	3	0.210	0.9484	14	4	4	0.9512
42	44	0.180	0.9623	25	1	4	0.9617
43	44	0.110	0.9568	33	1	10	0.9675
44	9	0.130	0.9526	5	6	6	0.9668
45	21	0.130	0.9628	8	4	7	0.9674
46	44	0.140	0.9657	21	1	4	0.9699
47	77	0.130	0.9593	2	10	4	0.9683
48	86	0.130	0.9471	5	4	10	0.9519
49	21	0.130	0.9631	87	4	5	0.9631
50	9	0.15	0.9547	4	10	8	0.967

The result of GA in stage 1 and stage 2 indicates that the accuracy varies from 0.9451 to 0.9738 for optimized random state and test size. In stage 2, the accuracy is decreased in 4 runs, with no change with 1 run out of 50, and in the remaining simulation run the accuracy increased to a maximum of 0.0209 (2.09%). 45 runs out of 50 indicated that integrating GA with RF in stages 1 and 2 improved the accuracy. This further leads to fitting the predictive model more effectively.

The GA-generated simulation of stage 1 (50^{th} run in Table 2) shows in Figure 2 that the accuracy obtained is 0.9547 with random stage 9 and test size 0.15. Figure 2 indicates that no improvement in the result was found after 23 iterations. Similarly, Figure 3 reveals stage 2- GA simulation to optimize hyperparameters. The accuracy obtained in this run is 0.967. Three parameters such as the number of tree (4), minimum leaves (10), and minimum leaf to spit (8) are optimized for an accuracy is 0.967 in the 50^{th} run as shown in Table 2. This shows that GA improves the accuracy from stage 1 to stage 2. The detailed DT is depicted in Figure 4 for pictorial representation. The simulation is carried out using Python v3.3 with 8 GB RAM, with GA function and ML algorithm in Windows 10 operating system.

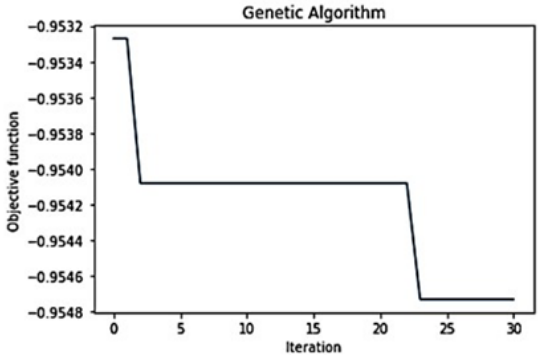


Figure 2. State 1- GA simulation to optimize random state and test size, accuracy is the objective function

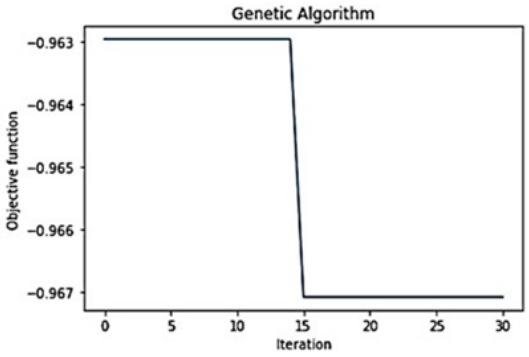


Figure 3. Stage 2- GA simulation to optimize hyperparameters

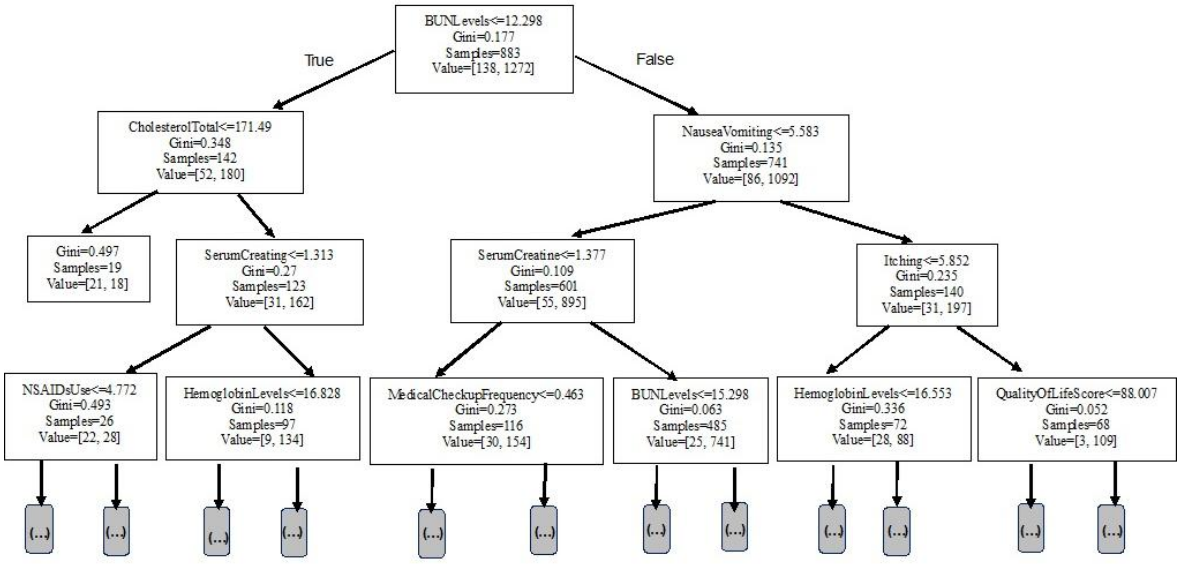


Figure 4. Pictorial representation of the DT

4. CONCLUSION

This study presents a novel approach to optimizing the performance of an RF classifier by integrating a GA for the prediction of CKD using a dataset of 1659 patients with 51 parameters. By systematically optimizing the random state, test size, and hyperparameters in a two-stage process, the method effectively enhances the accuracy of the RF model. The optimization process, conducted over 50 simulation runs, demonstrates a significant improvement in model accuracy, ranging from 0.9451 to 0.9738, with a maximum increase of 2.09%. These findings highlight the critical role of optimizing both model parameters and hyperparameters to enhance the predictive capabilities of ML models, especially within the realm of medical diagnostics. The combination of GA with RF not only enhances model performance but also establishes a strong and reliable framework for improving prediction accuracy in clinical settings. Future studies may consider applying this optimization strategy to different diseases and datasets to further assess its effectiveness and versatility.

ACKNOWLEDGEMENTS

Authors thank the organization for encouraging us to conduct research in multi-disciplinary areas.

FUNDING INFORMATION

This research was carried out independently, with no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Bommanahalli Venkatagiriyappa Raghavendra	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓		
Anandkumar Ramappa Annigeri	✓					✓	✓	✓	✓		✓	✓		
Jogipalya Shivananjappa Srikantamurthy		✓		✓	✓		✓	✓		✓			✓	
Gururaj Raghavendrarao Sattigeri		✓		✓	✓		✓			✓			✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest related to this study

DATA AVAILABILITY

The data that support this study are openly available in Kaggle dataset link at https://www.kaggle.com/datasets/rabieelkharoua/chronic-kidney-disease-dataset-analysis.

REFERENCES

[1] M. A. Islam, M. Z. H. Majumder, and M. A. Hussein, "Chronic kidney disease prediction based on machine learning algorithms," *Journal of Pathology Informatics*, vol. 14, 2023, doi: 10.1016/j.jpi.2023.100189.

[2] N. Lei *et al.*, "Machine learning algorithms' accuracy in predicting kidney disease progression: a systematic review and meta-analysis," *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, Dec. 2022, doi: 10.1186/s12911-022-01951-1.

[3] D. A. Debal and T. M. Sitote, "Chronic kidney disease prediction using machine learning techniques," *Journal of Big Data*, vol. 9, no. 1, Nov. 2022, doi: 10.1186/s40537-022-00657-5.

[4] S. Pal, "Chronic kidney disease prediction using machine learning techniques," *Biomedical Materials & Devices*, vol. 1, no. 1, pp. 534–540, Mar. 2023, doi: 10.1007/s44174-022-00027-y.

[5] S. J. Gilbert and D. E. Weiner, *National kidney foundation's primer on kidney diseases*, China: Elsevier, 2017.

[6] L. J. Lo *et al.*, "Dialysis-requiring acute renal failure increases the risk of progressive chronic kidney disease," *Kidney International*, vol. 76, no. 8, pp. 893–899, Oct. 2009, doi: 10.1038/ki.2009.289.

[7] J. Zhao *et al.*, "An early prediction model for chronic kidney disease," *Scientific Reports*, vol. 12, no. 1, Feb. 2022, doi: 10.1038/s41598-022-06665-y.

[8] H. Khalid, A. Khan, M. Z. Khan, G. Mehmood, and M. S. Qureshi, "Machine learning hybrid model for the prediction of chronic kidney disease," *Computational Intelligence and Neuroscience*, vol. 2023, no. 1, Jan. 2023, doi: 10.1155/2023/9266889.

[9] D. Saif, A. M. Sarhan, and N. M. Elshennawy, "Deep-kidney: an effective deep learning framework for chronic kidney disease prediction," *Health Information Science and Systems*, vol. 12, no. 1, Dec. 2023, doi: 10.1007/s13755-023-00261-8.

[10] M. M. Rahman, M. Al-Amin, and J. Hossain, "Machine learning models for chronic kidney disease diagnosis and prediction," *Biomedical Signal Processing and Control*, vol. 87, Jan. 2024, doi: 10.1016/j.bspc.2023.105368.

[11] E. Dritsas and M. Trigka, "Machine learning techniques for chronic kidney disease risk prediction," *Big Data and Cognitive Computing*, vol. 6, no. 3, Sep. 2022, doi: 10.3390/bdcc6030098.

[12] D. K. E. Lim *et al.*, "Prediction models used in the progression of chronic kidney disease: a scoping review," *PLOS ONE*, vol. 17, no. 7, Jul. 2022, doi: 10.1371/journal.pone.0271619.

[13] J. Aoki *et al.*, "CKD progression prediction in a diverse US population: a machine-learning model," *Kidney Medicine*, vol. 5, no. 9, Sep. 2023, doi: 10.1016/j.xkme.2023.100692.

[14] M. Binsawad, "Enhancing kidney disease prediction with optimized forest and ECG signals data," *Heliyon*, vol. 10, no. 10, May 2024, doi: 10.1016/j.heliyon.2024.e30792.

[15] K. Hema, K. Meena, and R. Pandian, "Analyze the impact of feature selection techniques in the early prediction of CKD," *International Journal of Cognitive Computing in Engineering*, vol. 5, pp. 66–77, 2024, doi: 10.1016/j.ijcce.2023.12.002.

[16] K. Takkavatakarn, W. Oh, E. Cheng, G. N. Nadkarni, and L. Chan, "Machine learning models to predict end-stage kidney disease in chronic kidney disease stage 4," *BMC Nephrology*, vol. 24, no. 1, Dec. 2023, doi: 10.1186/s12882-023-03424-7.

[17] F. Sanmarchi, C. Fanconi, D. Golinelli, D. Gori, T. Hernandez-Boussard, and A. Capodici, "Predict, diagnose, and treat chronic kidney disease with machine learning: a systematic literature review," *Journal of Nephrology*, vol. 36, no. 4, pp. 1101–1117, Feb. 2023, doi: 10.1007/s40620-023-01573-4.

[18] Y. Zhu, D. Bi, M. Saunders, and Y. Ji, "Prediction of chronic kidney disease progression using recurrent neural network and electronic health records," *Scientific Reports*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-49271-2.
[19] S. K. Ghosh and A. H. Khandoker, "Investigation on explainable machine learning models to predict chronic kidney diseases," *Scientific Reports*, vol. 14, no. 1, Feb. 2024, doi: 10.1038/s41598-024-54375-4.
[20] N. Alturki *et al.*, "Improving prediction of chronic kidney disease using KNN imputed SMOTE features and TrioNet model," *Computer Modeling in Engineering & Sciences*, vol. 139, no. 3, pp. 3513–3534, 2024, doi: 10.32604/cmes.2023.045868.
[21] S. Rajeashwari and K. Arunesh, "Chronic disease prediction with deep convolution based modified extreme-random forest classifier," *Biomedical Signal Processing and Control*, vol. 87, Jan. 2024, doi: 10.1016/j.bspc.2023.105425.
[22] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
[23] J. H. Holland, *Adaptation in natural and artificial systems*. United States: MIT Press, Cambridge, MA, 1992, doi: 10.7551/mitpress/1090.001.0001.
[24] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," *UC Berkeley Statistics Department*, pp. 1-12, 2004.
[25] E. Akkur and A. C. Öztürk, "Chronic kidney disease prediction with stacked ensemble-based model," *Alpha Journal of Engineering and Applied Sciences*, vol. 1, no. 1, pp. 50–61, 2023.

BIOGRAPHIES OF AUTHORS

Dr. Bommanahalli Venkatagiriyappa Raghavendra holds a bachelor's degree in mechanical engineering and a Ph.D. in mechanical engineering sciences from Visvesvaraya Technological University, Belagavi, India. Currently, he serves as an associate professor in the Department of Mechanical Engineering at JSS Academy of Technical Education, Bengaluru. His research interests include scheduling, the study of machining parameters, optimization, and metaheuristic algorithms. He has published more than 22 papers in peer-reviewed journals and is a member of the Institution of Engineers. He can be contacted at email: bvrjss@gmail.com.

Dr. Anandkumar Ramappa Annigeri holds a Ph.D. in mechanical engineering from IIT Madras, India, and has a strong academic foundation in the field. He is currently serving as a Professor in the Department of Mechanical Engineering at JSS Academy of Technical Education, Bengaluru. His research focuses on optimization and smart structure, and he has contributed significantly to the field through the publication of over 25 papers in peer-reviewed journals. He can be contacted at email: anandannigeri@gmail.com.

Dr. Jogipalya Shivananjappa Srikantamurthy holds a Bachelor's degree in mechanical engineering and a Ph.D. in the same field. Currently, he serves as an assistant professor in the Department of Mechanical Engineering at JSS Academy of Technical Education, Bengaluru. His research interests include optimization and finite element analysis. He is also a member of the Institution of Engineers. He can be contacted at email: jssrikantamurthy@jssateb.ac.in.

Gururaj Raghavendrarao Sattigeri is graduated in mechanical engineering, presently working as associate professor in the Department of Mechanical Engineering, KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, Karnataka. His research interest is jet impingement. He is a member in Institute of Engineers. He can be contacted at email: grs@klsvdit.edu.in.