Modified K-Nearest Neighbor Optimization with Genetic Algorithm in Chronic Kidney Disease Classification

I W Supriana[1,*], Cokorda Pramartha[1,2], I G N A C Putra[1], M D A Raharja[1] and P P K Wiguna[3]

[1]Computer Science Departement, Udayana University, Indonesia
[2]Center for Interdisciplinary Research on the Humanities and Social Sciences, Udayana University, Indonesia
[3]Department of Agroecotechnology, Faculty of Agriculture, Udayana University, Indonesia
*wayan.supriana@unud.ac.id

Abstract. Chronic kidney disease is a global health problem that requires early diagnosis for effective management. According to a WHO survey, in Indonesia alone it is estimated that around 70,000 cases occur each year, while the percentage increase in cases will occur by 46% from 1955-2025. Early detection of kidney disease can provide early help to reduce the death rate. There are similarities between indications which make the diagnosis process difficult. This research proposes an innovative approach in optimizing chronic kidney disease classification models using the Modified K-Nearest Neighbor (MKNN) method with the application of a Genetic algorithm. MKNN has been proven to be effective in classification, however determining critical parameters such as the number of neighbors (k) can affect the model performance. In this research, Genetic algorithm was used to find the optimal k value of the MKNN parameter. This approach allows the model to automatically adapt to data characteristics, increasing classification accuracy and reducing overfitting. Genetic algorithm was used to optimize the k parameters, and its fitness function was based on the classification performance of the model. Testing was carried out using a chronic kidney disease dataset that includes 24 clinical features. The research results show that the Modified K-Nearest Neighbor algorithm with an accuracy of 93%, precision of 93.2% and recall of 93.2%. Based on the research results, the MKKN model optimized using a genetic algorithm provides significant results based on accuracy, precision and recall.

Keywords: Genetic Algorithms, Chronic Kidney, Modified K-Nearest Neighbor

1 Introduction

Chronic kidney disease is a pathological condition that affects kidney function gradually, often without clear symptoms, a person will experience a decrease in kidney function that leads to kidney damage that can lead to death [16]. In Indonesia there is an estimated 70,000 sufferers of chronic kidney failure according to the WHO survey,

K. Dharmawan and N. A. Sanjaya ER (eds.), *Proceedings of the First International Conference on Applied Mathematics, Statistics, and Computing (ICAMSAC 2023)*, Advances in Computer Science Research 110,
https://doi.org/10.2991/978-94-6463-413-6_20

there is an increase in the number of people with chronic kidney disease by 46% from 1955-2025 [3]. Chronic kidney disease can be prevented and overcomed by getting effective therapy and the results of the therapy will be better if the disease is known earlier [3]. Therefore, early detection and accurate classification are needed to initiate interventions and minimize the negative impacts. Development of efficient classification methods can help in early kidney disease detection and chronic kidney disease management [13]. Classification of chronic kidney disease involves analysis of various clinical factors and parameters. A good classification method is able to handle the complexity of clinical data and can provide accurate results in identifying chronic kidney disease. One classification method that is often used is K-Nearest Neighbor (KNN). KNN method can predict classes from data based on the closest progress. But KNN has weaknesses such as sensitivity to outlier and the inability to handle high features dimensions. Therefore, modifications to this algorithm need to be done to improve their performance [10]. The Modified K-Nearest Neighbor (MKNN) method as the basis of the classification model, has been modified to improve performance. MKNN is a modification of the KNN algorithm to improve performance and flexibility in handling various data types with the use of weight on the attributes in calculating the distance between the closest neighbors. With a different weight of each attribute, MKNN can increase the accuracy of classification by giving a higher value to more informative attributes[6].

This research was carried out innovatively in optimizing the classification model for chronic kidney disease using the Modified K-Nearest Neighbor (MKNN) method with the application of a genetic algorithm. MKNN has been proven to be effective in classification [7], however determining critical parameters such as the number of neighbors (k) can affect the model performance. This approach allows the model to automatically adapt to data characteristics, increasing classification accuracy and reducing overfitting. A genetic algorithm was used to optimize the biased k parameters, and the fitness function was based on the model's classification performance. Genetic algorithms work probabilistically to create a new populations through continuous iteration of the initial population until an optimal population was obtained [11]. Testing was carried out using a chronic kidney disease dataset that includes 24 clinical features [2]. By combining MKNN and genetic algorithms, this research were aims to create a classification model that is more accurate, efficient and reliable in detecting chronic kidney disease. The results of this research are expected to make a significant contribution to the development of classification methods that can be used in decision support systems to diagnose chronic kidney disease more precisely, enable earlier medical intervention, and improve the management of patients with this disease[20].

2 Materials

Classification methods can help further identification and finding indications that lead to people with kidney disease [4]. The method used was Modified K-Nearest Neighbor (MKNN). The MKNN method takes K nearest neighbors by modifying the determination of the class of classification results, namely giving each nearest neighbor a weight,

then voting on the weight of the neighbor. The neighbor with the highest vote was chosen as the class to be the classification result. The value of K is biased for each data, and optimized with Genetic Algorithms with the aim of getting the best classification results [8]. The research data was sourced from the UCI Machine Learning Repository with a total of 4000 datasets with two classes of data affected by kidney disease and not affected by kidney disease [5]. Important data preprocessing steps were carried out handling missing values, outliers, data normalization and dataset imbalance. The final stage of testing the classification results was done to determine the optimal K value of the Genetic algorithm, by performing hyperparameter tuning. The classification results of each K value were evaluated using a confusion matrix to see the performance of the classification model.

3 Research Method

The research was carried out through several stages that explain the process of the research, as shown in Figure 1. These stages include the data preprocessing stage to prepare research data, such as dealing with missing values, outliers and normalizing the data scale. The next stage divides the data into testing data using K-Fold Cross Validation. The next stage is optimizing the *k* value with a genetic algorithm and continuing with the classification stage. At the classification stage, two methods were used to compare model performance, namely Modified K-Nearest Neighbor (MKNN) and K-Nearest Neighbor (KNN). The classification results are then evaluated using the Confusion Matrix on each Fold section of the K-Fold Cross Validation[9].

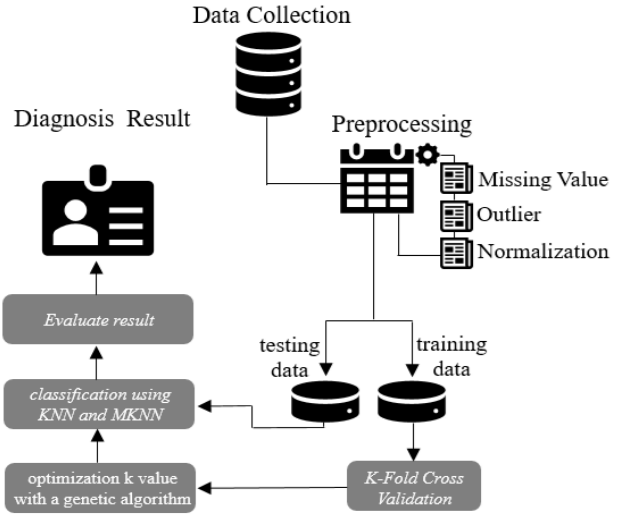

Figure 1. The Research Stages

The *k* value optimization stage with a genetic algorithm is based on Darwin's concept of evolutionary theory. Where the main concept is that individuals survive in a population based on their fitness or how strong the individual survives. This algorithm generally starts from: the initial stage of initializing the initial population, namely the process

of randomly generating as many chromosomes as the predetermined population size[11]. The second stage of individual evaluation is a process that calculates the fitness value of each chromosome using Equation 1.

$$f_i(X) = \frac{\sum_{a=i}^{u} validitas(a)}{u} \quad (1)$$

u is the amount of training data, validity(a) is the validity of the k value, i is the fitness function for k in population i.

The third stage of chromosome selection for crossover, this selection process was using roulette wheel selection by calculating the best fitness value based on the crossover probability [18]. The fourth stage was crossover, which aims to give birth to new chromosomes that inherit the characteristics of their parents during the reproductive process[12]. In this research, the crossover technique uses one-point crossover, two-point crossover, and uniform crossover as shown in Figure 2.

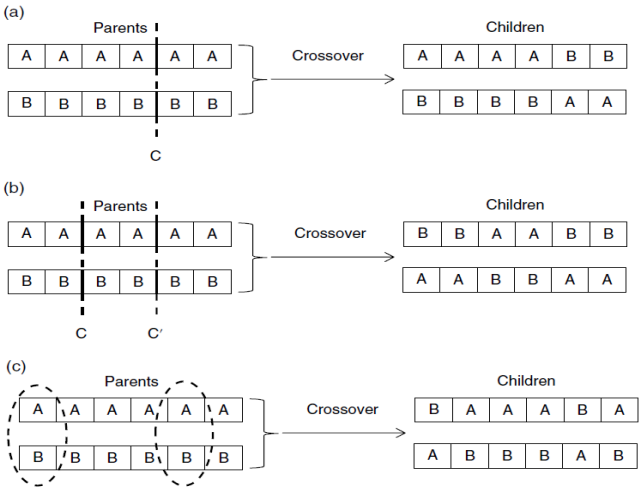

Figure 2. a) One-point crossover, b) two-point crossover and c) uniform crossover techniques

The fifth stage of mutation aims to introduce a random element into the evolutionary process, thereby preventing stagnation and helping to better explore the search space. Mutations help avoid rapid convergence to a local optimum by introducing new genetic variation into the population, as shown in Figure 3 [15].

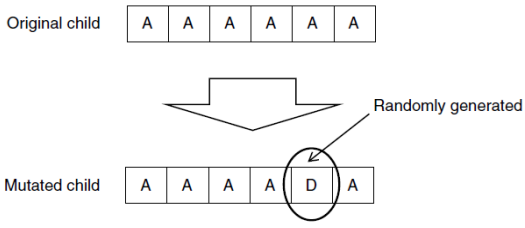

Figure 3. Mutation Technique

The final stage is the evaluation of the stopping criteria, namely: determining when the algorithm should be stopped based on criteria, such as reaching an adequate solution or reaching a specified maximum number of generations.
The Modified K-Nearest Neighbor algorithm classification is a modification of the K-Nearest Neighbor algorithm by adding a process for calculating validity values and voting weights to the dataset [19]. Determining class labels is not only based on the number of nearest neighbors but also based on weights. The largest class label weight will be selected as the classification class. The classification stages of the Modified K-Nearest Neighbor algorithm start from: calculating the Euclidean distance, namely a measure of the closeness between two data, calculated from the closeness between the data attributes, which calculated using Equation 2:

$$d(xi, yi) = \sqrt{\sum_{i=0}^{n}(xi - yi)^2} \quad (2)$$

x_i is testing data, y_i is training data and $d(x_i, y_i)$ is the Euclidean distance value.
The second stage determines the similarity value of the S function to calculate the similarity between point x and data i from the nearest neighbor, using Equation 3:

$$S(a, b) = \begin{cases} 1a = b \\ 0a \neq b \end{cases} \quad (3)$$

a is class a in the training data and b is a class other than a in the training data.
The third stage is determining the validity value, namely the process of determining the weight calculation, or weight voting, using Equation 4:

$$Validity(x) = \frac{1}{H}\sum_{i=0}^{n} S(label(x), label(Ni(x))) \quad (4)$$

H is the number of nearest points/number of nearest neighbors selected, label(x) is the class label of x and label($Ni(x)$) is the class label of the nearest point x.
The fourth stage is determining the voting weight, namely determining the highest class label from the nearest neighbors for the classification results, using Equation 5:

$$W(i) = Validity(i)\, x \frac{1}{de + a} \quad (5)$$

$W(i)$ is the weight of neighbor i, $Validity(i)$ is the validity of neighbor i, de is the Euclidean distance value of neighbor i and a is the smoothing parameter (0.5).
The classification testing technique uses a Confusion Matrix. This technique represents the performance of the MKNN algorithm in tabular form [17]. This table contains correctly classified and incorrectly classified test data. There are three criteria for assessing classification results using the Confusion Matrix, including Precision, Recall, Accuracy. The values are based on 4 categories, namely TP (True Positive), FP (False Positive), FN (False Negative) and TN (True Negative).

4 Result and Discussion

4.1 Optimal Parameter Selection

Parameter selection is carried out by determining the optimal k value using a genetic algorithm with evaluation based on crossover rate, mutation rate and number of populations or chromosomes in one generation. The optimal or best k value is determined by measuring the fitness value of each chromosome and the best fitness is optimal k.

The optimal *k* value is used to classify with the evaluation parameters accuracy, precision and recall.

4.2 Optimal Parameter Selection

Genetic algorithm parameters have an important role in determining classification optimization results, so that the appropriate parameters are expected to find the most optimal *k* value. How good these parameters are is based on the fitness value obtained. Based on the *k* test for fold 2, fold 5 and fold 10 with test parameters including crossover rate, mutation rate and population size, only one best *k* value will be the parameter for the number of nearest neighbors[14].

Testing on fold 2:

Fold 2 testing was carried out with the dependent parameters mutation rate = 0.02, population = 8 and change in crossover rate with a value of 0.3 - 0.9. Table 1 is the optimal *k* value resulting from each crossover rate. It can be seen that a crossover rate of 0.4 produces the best optimal *k* with a value of 3, fitness 0.936. In general, the range of *k* values produced is in the range 1-3 with changes in fitness values not being too significant.

Table 1. Fold 2 Crossover Rate Testing

Crossover rate	Optimal K value	Fitness value
0.3	1	0.920
0.4	3	0.936
0.5	2	0.926
0.6	1	0.936
0.7	3	0.920
0.8	2	0.926
0.9	1	0.936

Testing on fold 5:

Fold 5 testing was carried out with the dependent parameters mutation rate = 0.02, population = 8 and crossover rate change with a value of 0.3 - 0.9. Table 2 is the optimal *k* value resulting from each crossover rate. It can be seen that a crossover rate of 0.5 produces the best optimal *k* with a value of 3, fitness 0.955. In general, the resulting *k* value range is in the range 1-3 with insignificant changes in fitness values.

Table 2. Fold 5 Crossover Rate Testing

Crossover rate	Optimal K value	Fitness value
0.3	1	0.930
0.4	2	0.941
0.5	3	0.955
0.6	3	0.930
0.7	2	0.941

0.8	4	0.923
0.9	5	0.921

Testing on fold 10:
Fold 10 testing was carried out with the dependent parameters mutation rate = 0.02, population = 8 and crossover rate changes with a value of 0.3 - 0.9. Table 3 is the optimal *k* value resulting from each crossover rate. It can be seen that a crossover rate of 0.4 produces the best optimal *k* with a value of 3, fitness 0.962. In general, the range of *k* values produced is in the range 1-3 with changes in fitness values not being too significant.

Table 3. Fold 10 Crossover Rate Testing

Crossover rate	Optimal K value	Fitness value
0.3	1	0.928
0.4	3	0.962
0.5	8	0.920
0.6	2	0.945
0.7	5	0.945
0.8	3	0.938
0.9	1	0.962

Based on the fold 2, fold 5 and fold 10 tests, the optimal k value that has the best fitness is *k*=3, this *k* value will be used in the classification process for the MKNN and KNN models as a comparison of the classification results.

4.3 Optimization *k* Value Testing and Comparison of K-Nearest Neighbor Classification Results with Modified K-Nearest Neighbor

This test aims to determine the influence of the *k*=3 value obtained in the Genetic algorithm on the classification results [17]. The *k* value used is 1-5 for the MKNN and KNN models. Table 5 shows the evaluation results matrix in the form of accuracy, precision and recall. The optimal *k* test is to see the comparison of classification results between the Modified K-Nearest Neighbor and K-Nearest Neighbor methods.

Table 4. Evaluation of MKNN and KNN Model Classification Results

k value	K-Nearest Neighbor			Modified K-Nearest Neighbor		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
1	81,80%	82,30%	81,20%	92,40%	92,60%	91,80%
2	84,60%	84,60%	83,30%	91,80%	91,80%	90,80%
3	86,80%	88,00%	88,10%	93,00%	93,20%	93,20%
4	82,40%	82,10%	82,00%	92,80%	91,90%	93,00%
5	85,70%	86,60%	85,30%	92,80%	93,10%	92,70%
rata-rata	84,26%	84,72%	83,98%	92,56%	92,52%	92,30%

Table 5 shows the model evaluation results. The average accuracy of MKNN is 92.6%, while KNN is 84.3%. The accuracy of KNN tends to be lower compared to MKNN at all *k* values. The accuracy of MKNN increases consistently with different K values, indicating that the modified method provides overall performance improvement. The average precision of MKNN is 92.5%, while KNN is 84.7%. The precision of the KNN and MKNN models has a difference that is not too significant in the *k* value. In general, both methods have a relatively high level of precision, with MKNN being higher for each *k* value. The average recall for MKNN is 92.3% while KNN is 83. 9%. Modified KNN shows better consistency in terms of Recall compared to KNN, with a fairly stable increase. Overall, MKNN with various *k* values shows good performance compared to KNN. MKNN shows good performance with relatively high accuracy values at each k value tested, at *k*=3 the MKNN model provides the best performance with an accuracy of 93.0%, precision 93.2% and recall 93.2% compared to other *k* values.

5 Conclusion

Evaluation of the *k* parameters of the genetic algorithm process by determining the optimal *k* value, shows that the value *k*=3 produces the best fitness with the input parameters: mutation rate=0.02, population=8 and crossover rate change 0.0-0.9. The performance of the classification process after determining the optimal *k* value based on evaluation shows that *k*=3 provides the best value compared to other *k*. In the MKNN model, accuracy, precision and recall were obtained at 93%, 93.2%, 93.2%, while in the KNN model, accuracy, precision and recall were obtained at 86.6%, 88%, 88.1%. In general, the MKNN model has better performance than KNN. The *k* parameter in classification really determines the performance of the algorithm. Evaluation shows that optimizing the *k* value with a genetic algorithm has a significant effect on increasing the accuracy of the MKNN model. Comparison of the classification results of the two models shows that the MKNN model is significantly better when compared to the KNN classification model.

References

1. Abdelaziz, A., Salama, A. S., Riad, A. M., Mahmoud, A. N.:A Machine Learning Model for Predicting of Chronic Kidney Disease Based Internet of Things and Cloud Computing in Smart Cities. In: Hassanien A., Elhoseny M., Ahmed S., Singh A. (eds) Security in Smart Cities: Models, Applications, and Challenges. Lecture Notes in Intelligent Transportation and Infrastructure, Springer, pp.93-114 (2019)
2. Elhoseny, M., Shankar, K., Uthayakumar, J.: Intelligent Diagnostic Prediction and Classification System for Chronic Kidney Disease. Scientific Report 9 (9583), 1-14(2019)
3. Gamanarendra, I., W., Waspada, I.:Implementasi Data Mining Untuk Deteksi Penyakit Ginjal Kronis (PGK) Menggunakan K-Nearest Neighbor (KNN) Dengan Backward Elimination. Jurnal Teknologi Informasi dan Ilmu Komputer(JTIIK) 7(2), 417-426 (2020)
4. Gharibdousti, M. S., Azimi, K., Hathikal, S., Won, D. H.: Prediction of Chronic Kidney Disease using data mining techniques. Proceedings of the 2017 Industrial and Systems Engineering Conference, pp. 2135–2140(2017)

5. Hamedan, F., Orooji, A., Sanadgol, H., Sheikhtaheri, A.: Clinical decision support system to predict chronic kidney disease: A fuzzy expert system approach. International Journal of Medical Informatics 138(104134), June 2020
6. Jabbar, M. A., Deekshatulu, B. L., Chandra, P.: Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm. Procedia Technology 10, 85-94(2013)
7. Kunwar, V., Chandel, K., Sabitha, A. S.,Bansal, A.:Chronic Kidney Disease analysis using data mining classification techniques. 6th International Conference, Cloud System and Big Data Engineering (Confluence), Noida, India, pp. 300-305 (2016)
8. Levey, A. S., Jong, P. E.D., Coresh, J., Nahas, M. E., Astor, B. C., Matsushita, K., Gansevoort, R. T., Kasiske, B. L., Eckardt, K. U.:The definition, classification, and prognosis of chronic kidney disease: A KDIGO Controversies Conference report. Kidney Int 80(1), 17–28(2011).
9. Liu, Z., Pan, Q., Dezert, J.: A new belief-based K-nearest neighbor classification method. Pattern Recognition 46(3), 834–844(2013)
10. Parvin, H., Alizadeh, H., Minati, B.:A Modification on K-Nearest Neighbor Classifier. Global Journal of Computer Science Technology 10(14), 37–41(2010)
11. Prasetio, R. T., Rismayadi, A. A., Anshori, I. F.: Implementasi Algoritma Genetika pada k-nearest neighbours untuk Klasifikasi Kerusakan Tulang Belakang. Jurnal Informatika 5(2), 186-194(2018).
12. Putria, N.M.E.A., Kadyanana, I.G.A.G.A., Supriana, I.W., Pramartha, C.R.A., Karyawatia, A.A.I.N.E. and Gede, I.B., Identifikasi Topeng Bali Dengan Metode KNN (K Nearest Neighbor). Jurnal Elektronik Ilmu Komputer Udayana p-ISSN, 2301, p.5373.
13. Rady, E. H. A. Anwar, A. S.: Prediction of kidney disease stages using data mining algorithms. Informatics in Medicine Unlocked 15(100178), 2019.
14. Ravi, M. R., Adinugroho, S.: Implementasi Algoritme Modified K-Nearest Neighbor (MKNN) Untuk Mengidentifikasi Penyakit Gigi dan Mulut. JPTIIK: Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer 3(3), 2596-2602(2019).
15. Shah, S., Kusiak, A.:Cancer gene search with data-mining and genetic algorithms. Computers in Biology and Medicine 37(2), 251-261(2007)
16. Subas, A., Alickovic, E., Kevric, J.: Diagnosis of Chronic Kidney Disease by Using Random Forest. in Part of the IFMBE Proceedings book series 62, 589–594 (2017)
17. Supriana, I. W., Raharja, M. A., Bimantara, I. M. S., Bramantya, D.:Implementasi Dua Model Crossover Pada Algoritma Genetika untuk Optimasipenggunaan Ruang Perkuliahan. Jurnal Rekayasa Sistem Komputer 4(2), 167-177 (2021)
18. Supriana, I. W., Raharja, M. A., Gunawan, P. W.:Sistem Informasi Prediksi Penilaian Kredit Perbankan Menggunakan Algoritma K-Nearest Neighbor Classification. Jurnal Sains & Teknologi 8(1), 44-154 (2019)
19. Wahyudi, N., Wahyuningsih, S., Amijaya, F. D. T.: Optimasi Klasifikasi Batubara Berdasarkan Jenis Kalori dengan menggunakan Genetic Modified K-Nearest Neighbor (GMK-NN). Jurnal Exponensial 10(2), 103-112 (2019)
20. Zeng, Y., Yang, Y., Zhao, L.: Nonparametric classification based on local mean and class statistics. Expert Systems with Applications 36, 8443–8448 (2009)