Performance Impact of Genetic Operators in a Hybrid GA-KNN Algorithm

Raghad Sehly[1], Mohammad Mezher[2]
Department of Computer Science
Fahad Bin Sultan University
Tabuk, KSA

Abstract—**Diabetes is a chronic disease caused by a deficiency of insulin that is prevalent around the world. Although doctors diagnose diabetes by testing glucose levels in the blood, they cannot determine whether a person is diabetic on this basis alone. Classification algorithms are an immensely helpful approach to accurately predicting diabetes. Merging two algorithms like the K-Nearest Neighbor (K-NN) Algorithm and the Genetic Algorithm (GA) can enhance prediction even more. Choosing an optimal ratio of crossover and mutation is one of the common obstacles faced by GA researchers. This paper proposes a model that combines K-NN and GA with Adaptive Parameter Control to help medical practitioners confirm their diagnosis of diabetes in patients. The UCI Pima Indian Diabetes Dataset is deployed on the Anaconda python platform. The mean accuracy of the proposed model is 0.84102, which is 1% better than the best result in the literature review.**

Keywords—Data mining; classification; K-NN; GA; Pima Indian Diabetes Dataset; UCI

I. Introduction

The world is facing many prevalent and chronic diseases. Diabetes is one of them. According to statistics for 2019 from the International Diabetes Federation (IDF), about 463 million adults (20-79 years old) were suffering from diabetes that year [1]. Many methods exist for diagnosing diabetes. Doctors diagnose diabetes by measuring glucose levels in the blood using tests such as the Fasting Plasma Glucose Test (FPG), the Postprandial Glucose Test, the Random Blood Glucose Test, the Oral Glucose Tolerance Test (OGTT), and the Glycated Hemoglobin Test [2]. One measurement of blood sugar is not enough to diagnose diabetes, particularly if the patient is diabetic and has no other symptoms. In differential diagnosis, doctors must examine the medical records of previous patients with the same conditions. Data were taken from patients and experts is the most important factor [3]. If these data are classified and predicted in a precise way, the global health expenditure can be reduced by up to 10% (760 billion USD) [1]. Data mining is a logical process for discovering and predicting models with huge data sets to find useful information. Data mining has three steps: exploration, pattern identification, and deployment [4].

The first step of data exploration involves cleaning and transforming data. The second step involves determining the important variables and the nature of the data, depending on the problem. After the data are explored, refined, and defined for particular variables, it is necessary to form a pattern of identification. Identifying and choosing the patterns can contribute to enhancing prediction. Deploying patterns is implemented for the desired outcome.

However, classifying the dataset is one of the most popular techniques in data mining. It employs a set of pre-classified examples for developing a model that can classify the records population in general. The classification has many different models, such as the K-Nearest Neighbor (KNN), the Decision Tree Induction, Neural Networks, the Bayesian Classification, Support Vector Machines (SVM), and Classification Based on Associations. With the help of classification algorithms, diabetes can be diagnosed more accurately [4].

A. K-NN Classifier

KNN is a technique used to classify objects depending on the closest training examples in the space of feature. The K-NN classifier is a kind of instance-based learning where the function is convergent in a local way, and the computation is postponed until the classification is finished. In K-NN, the training samples are generally described by n-dimensional numeric attributes. The training samples are saved in an N-dimensional space. K-NN [5, 6] begins searching for the "K" training samples that are closest to the test sample or unknown sample when a test sample is given. Closeness is defined in terms of Euclidean distance. Some distance measures are used to determine closeness. Occasionally, one minus correlation value is taken as a distance metric. The following three distance measures are used for continuous variables: Euclidean distance measure, Manhattan distance measure, and Minkowski distance measure, the latter of which has been used in this paper. The Hamming distance should be used in the instance of categorical variables. The distance measures between x and y points, (x_1, x_2… x_n) and y (y_1, y_2… y_n), are calculated in below distance functions:

$$Euclidean = \sum_{i=1}^{k} (x_i - y_i)^2$$

$$Manhattan = \sum_{i=1}^{k} |x_i - y_i|$$

$$Minkowski = \left(\sum_{i=1}^{k} (|x_i - y_i|)^q \right)^{1/q}$$

The pseudocode of K-NN:

```
K-NN Classifier (X, Y, x) :
 // where X: data, Y: decision labels
        // x: test sample
for i = 1 to m do # i: data point
k-distance = compute the
Sort the k distances.
distance d ($X_i$, x)
end for
return k-nearest reference point for {$Y_i$ where i ϵ $I_m$}
```

B. Genetic Algorithm (GA)

A programming method inspired by biological evolution; GA is used in problem-solving strategy to identify optimal solutions. Given that GA is a general algorithm, it works appropriately with any search space. GA is a useful tool for classifying the K-NN algorithm. Although the traditional K-NN algorithm employs Euclidian distance regularly, other measures can be used as well. GA can improve the performance of K-NN by using both Euclidian distance and cosine similarity to evaluate the optimal linear weights of features [7]. In search of an optimal reference point, GA is capable of optimizing additive measures to evaluate similarity using cosine measures. In [7], GA used the selection principles and evolution to find potential solutions to a given problem as shown in the pseudocode below:

```
1) Set a group of chromosomes (any applicable representation).
2) Generate an initial population out of (1).
3) Evaluate the generated population in (2).
4) Search Space until n generations.
    a) Select fittest chromosome.
    b) Mutate and recombine chromosomes.
    c) Evaluate the generated (b) chromosomes.
    d) Replace weak chromosomes with new chromosomes.
```

II. Literature Review

This section lists and reviews previous works that experiment with K-NN classification and GA algorithms. It also compares the algorithms based on accuracy performance.

Research paper [5] employed the Pima Indian Diabetes Dataset to measure the performance of combined K-NN and GA algorithms. The combination of K-NN and GA successfully improved the feature selection in the K-NN classifier. The accuracy of this model was 83.12%.

The authors of [8] conducted experiments to predict diabetes by using the Pima Indian Diabetes Dataset. For this evaluation, they chose to use the results of the 10-fold cross-validation. The accuracy of the K-NN algorithm was 71.84%.

Author in [9] evaluated the performance of the K-NN algorithm to classify the Pima Indian Diabetes Dataset. The authors used the data imputation, scaling, and normalization techniques to improve the classifier accuracy. A voting classifier was used to measure the performance of the Pima Indian Diabetes Dataset in predicting diabetes. The accuracy of the K-NN algorithm was 71.3%.

Author in [10] aimed to compare the performance of many classification algorithms to predict diabetes with the Pima Indian Diabetes Dataset. The authors compared many machine learning classifiers to classify patients with diabetes. The K-NN algorithm was one of these classifiers, and its accuracy was measured as 72.65% with using of 3-fold cross-validation.

The authors of [11] combined the GA algorithm with the K-NN algorithm to improve the feature selection of the Pima Indian Diabetes Dataset. The achieved accuracy of this model was 73.8%.

Table I shows the accuracy results of different research papers in which the best accuracies were for the hybrid models of GA and KNN.

K-NN and GA real-life applications:

K-NN algorithm: It is applied in many different ways in everyday life, including life, including:

Weather forecasting: The K-NN algorithm had been used to help with rainfall forecasts by using many weather factors, such as mean temperature, dew point temperature, humidity, sea pressure, and wind speed [12].

Economic forecasting: K-NN is a capable technique for economic prediction, especially concerning the financial distress of companies [13].

GA algorithm: It is used in many helpful practical applications, including:

Image Processing: GA is used to solve image segmentation, which is one of the most common problems in the image-processing area [14].

Bioinformatic: GA is a helpful model in interpreting huge data of bioinformatics more accurately and concisely [15].

TABLE I. Algorithms Comparison Results

Research Paper	**Algorithm**	**Accuracy**
[5]	K-NN + GA	83.12%
[8]	K-NN	71.84%
[9]	K-NN	71.32%
[10]	K-NN	72.65%
[11]	K-NN + GA	73.78%

III. Model and Implementation

The present researchers implemented the PIMA Dataset to calculate the accuracy of the combined K-NN and GA model. The Pima Indian Diabetes Dataset has eight attributes: Number of Times of Pregnancy (NTP), Plasma Glucose Concentration (PGC), Diastolic Blood Pressure (DBP), Triceps Skinfold Thickness (TSF), Body Mass Index BMI, Diabetes Pedigree Function (DPF), Age in years (AGE), and Binary Target (CLASS). This dataset has been divided into two sets: one for training the model (training set) and another for testing the model (testing set).

A. Data Preprocessing

Quality and quantity are the most effective factors in the classification model's accuracy and prediction. The medical databases are currently facing many obstacles such as noisy, inconsistent, and incomplete data, due to a large amount of data. These obstacles are leading to the low quality of the mining results. Thus, data quality should be upgraded with the assistance of suitable techniques to improve the results of data mining. The data preprocessing technique has a crucial role in improving the data quality and thereby increasing the accuracy of classification. Data preprocessing helps to detect anomalies in the data and remove the data that can lead to big payoffs for decision support. There are many methods of data preprocessing, including data integration, data cleaning, data reduction, data transformations, and data normalization [16].

B. Data Normalization

Data normalization is a process in data preprocessing intended to change the attribute value depending on a common scale or range to improve the performance of the machine learning algorithm. There are many techniques of data normalization, including min-max and z-score. The Python platform has a machine learning framework for data preprocessing such as sklearn. This framework has a large number of useful normalization techniques, including MinMaxScaler (MMS), MaxAbsScaler (MAS), StandardScaler (SS), RobustScaler (RS), and Normalizer (NM) [17]. The technique used in this paper is StandardScaler, which affected the model positively by increasing the accuracy percent.

C. Initial Population

Create a new population by repeating the next steps to completing the new population.

D. Fitness Function

The fitness f(x) of every chromosome x in the population is evaluated. Function fitness(X) is defined as in the equation.

$$Accuracy(x) = fitness(x)$$

E. Selection Operator

Select the chromosomes of the parents from a population depending on their fitness. GA has many different techniques that can be used for selecting the individuals who will be replaced by the next generation. The technique used in this paper is the Elitist selection, which worked effectively. Table II shows the comparisons of GA techniques and the usage of each technique.

TABLE II. Comparison of GA Techniques

GA Technique	Usage
Elitist selection	GA chooses the fittest individuals in every generation.
Fitness-proportionate selection	Chooses fitter members, but not certain, to be selected.
Roulette-wheel selection	Selects the individual depending on fitness level among competitors.
Scaling selection	Distinguishes between individuals with high fitness and those with small differences.
Tournament selection	Chooses the individuals by subgrouping them and then takes only one individual.

F.Crossover Operator

The parents are crossed over with a crossover probability (probability = 0.9) to produce a new offspring. If there is no crossover, the offspring will be an exact duplicate of the parents. In the GA algorithm, the crossover operator is one of the genetic operators often used in the GA lifecycle. After the individuals are selected, the next step is to produce the offspring. Crossover is a commonly used solution for this step, and among its many variant types the single-point crossover is the most used. As shown in Fig. 1, the single-point crossover solution involves selecting a place for locus to replace the remaining alleles between parents.

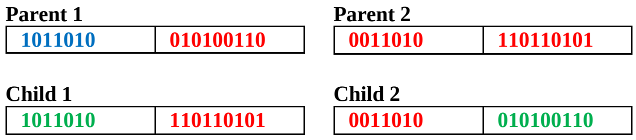


Fig. 1. Crossover Steps.

The children will receive just one section of the chromosome from their parents. The broken point of the chromosome is randomly selected by the crossover point. Because only one crossover point exists, this method is called single-point crossover. In some instances, only child 1 or child 2 is created, but in most cases, both offspring are created and located in the new population. However, the crossover does not occur always. In some cases, based on a probability set, the crossover does not occur, and the parents are copied into the new population directly. The range of the crossover occurrence probability is between 60% and 70%.

F. Mutation Operator

A new offspring will be mutated at every locus with the help of mutation probability (probability = 0.01). The second type of genetic operator used here and considered an exploitation operator is the mutation operator. When selection and mutation operators are used alone, a new individual will be mutated in some of its genes. Some genes are copied directly, and others will be mutated. To guarantee that the individuals are not completely the same, a mutation must be introduced. You loop over all the individuals' genes, and if that gene is chosen for mutation, you can replace it with a small amount, or you can replace it with a new value. The probability range of the mutation is usually less than 0.05. Fig. 2 shows the mutation process.

Before: 1101101001101110
After: 1101100001101110

Fig. 2. Mutation Steps.

The mutation is a simple operator in many ways. One simply must alter the selected alleles depending on what they feel is important and carry on. The mutation is necessary for ensuring the diversity of genes in the population [6].

G. *Model Lifecycle*

When the final condition is met, GA must stop, and return the best result in the current population [18]. There are two major types of parameter values setting, which are classified according to the behavior of the parameter values through the run. The first type is parameter tuning, a common technique based on experimenting with the diverse values of crossover and mutation. After selecting the value with the best results, the final run of the algorithm is carried out. There is no changing of the value during the course of this operation (fixed value).

The second type is parameter control based on the initial values for crossover and mutation, which are changed in some way during the run. Changes to the parameter values can be divided into three types:

1) Deterministic parameter control: It is used if the parameter value demands certain modifications using the same rule of the outcome. The parameter value should be tuned for producing the typical output with no results of the search process.

2) Adaptive parameter control: It is used if the specific type of feedback demanded from the search option helps in changing the parameter.

3) Self-Adaptive parameter control: With this type, the GA is capable of developing its parameters. The values of the parameters that are used are included in the individuals and persist during mutation and crossover.

These three types of parameter settings are commonly used by researchers in attempting to find optimal or near-optimal solutions with the best rates of the operators of crossover and mutation. This approach contributes to new strategies for controlling the rates of crossover and mutation. The proposed technique is classified as an adaptive parameter control [19].

Eight techniques of GA were used to compute the accuracy: Crossover and Mutation, Crossover only, Mutation only, crossover with adaptive parameter control (1), Crossover with adaptive parameter control (2), Crossover with adaptive parameter control (3), crossover with adaptive parameter control (4), and mutation with adaptive parameter control (5). Assume C: for Crossover, M: for Mutation, I: maximum of iterations, P: for population size, PF: for population fitness, MF: for Max Fitness,

$$C = (P * I)^2 \tag{1}$$

$$C = (P/I)^2 \tag{2}$$

$$C = \left(\frac{1}{2}\right)^{P*I} \tag{3}$$

$$C = \sum P - mean(PF) * MF - mean(PF) * \frac{1}{P-1} \tag{4}$$

$$M = \sum P - mean(PF) * MF - mean(PF) * \frac{1}{P-1} \tag{5}$$

The above four equations have increased the crossover to overpass mutation performance.

Fig. 3 depicts the steps used in the classified dataset. The first step is importing the Pima Indian Diabetes Dataset, after which the data must be preprocessed and normalized. Next, the five phases of GA are considered: initial population, fitness function, selection, crossover, and mutation, with the proposed four equations to improve the accuracy of the model by using adaptive parameter control because these operators play an important role in increasing the accuracy of GA. Finally, the K-NN is combined with the GA algorithm. The steps are repeated to produce a good result.

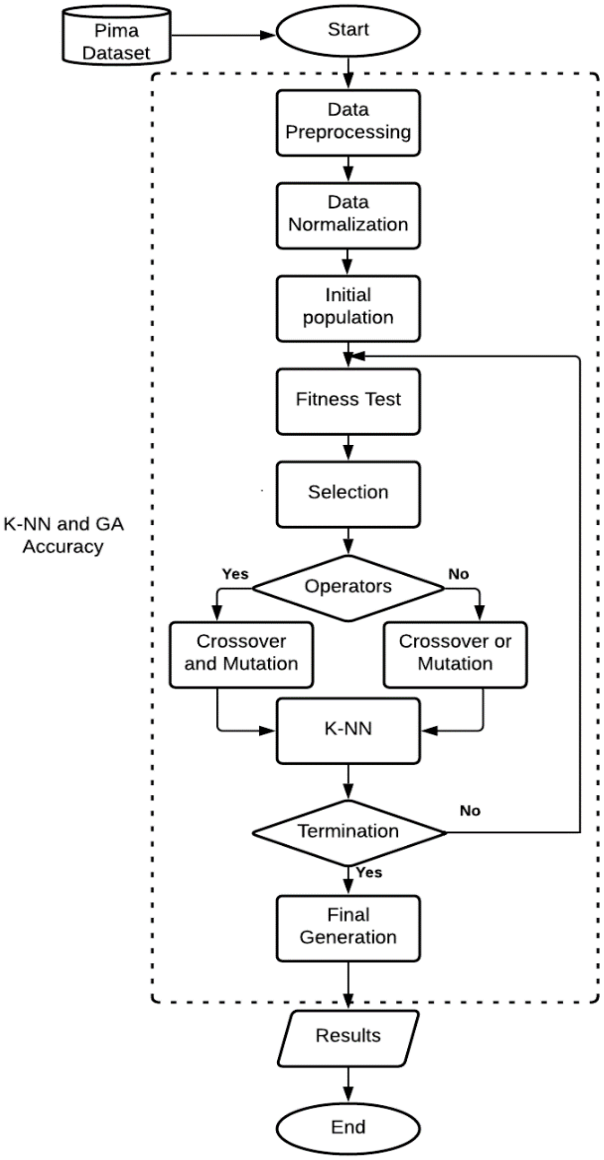


Fig. 3. The K-NN and GA Combination Model.

IV. Results and Discussions

This paper classifies the Pima Indian Diabetes Dataset into two groups: 0 or 1 (for negative '0', for positive '1') to compare the accuracy of combined K-NN and GA models using different genetic operators. Table III shows the Mean Accuracy of the eight techniques. Crossover operator with adaptive parameter control (4) achieved the best accuracy with 0.84, surpassing the performance of the mutation operator.

A. Scatter Plots Results

The mapping of scatter plots is similar to the mapping of line graphs in that they begin with mapping quantitative data points. The difference between the two plots is that the decision making on the scatter plots for the individual's point must not be connected directly with a line, but instead expresses a trend. This trend can be recognized directly through the point's distribution or the regression line [20].

Table IV shows the GA parameters and their values. The GA is configured to have a population size of 50 and was run for three generations. Crossover and mutation probability were 0.9 and 0.1, respectively.

Fig. 4 shows the scatter plot of the eight techniques in each group of generations.

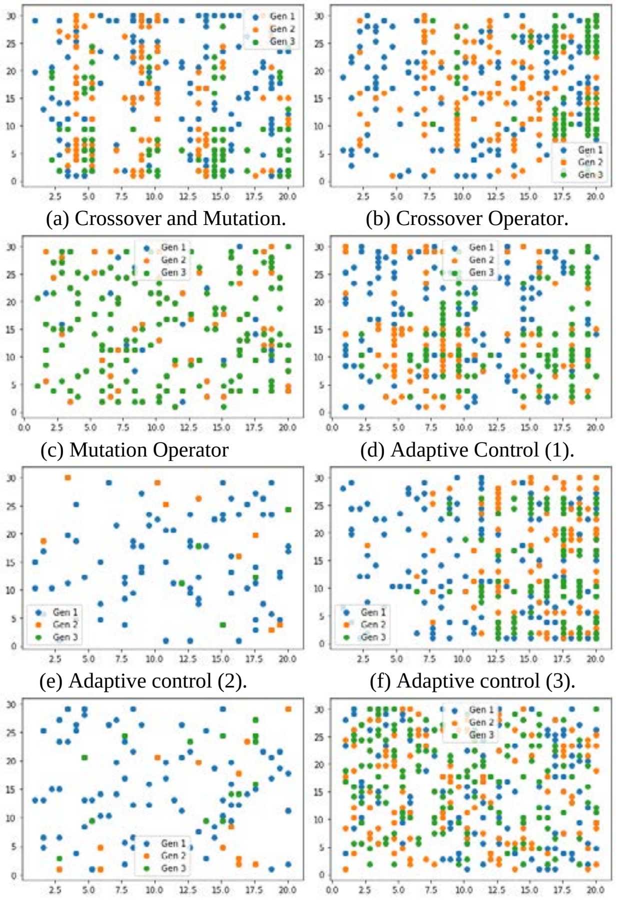
In experiment (a), a scatter plot shows that generation three has the best accuracy. In experiment (b), a scatter plot shows that generation two has the best accuracy. In experiment (c), a scatter plot shows that generation three has the best accuracy. In experiment (d), a scatter plot shows that generation two has the best accuracy. In experiment (e), a scatter plot shows that generation two has the best accuracy. In experiment (f), a scatter plot shows that generation three has the best accuracy. In experiment (g), a scatter plot shows that generation two has the best accuracy. Finally, in experiment (h), a scatter plot shows that generation two has the best accuracy.

TABLE III. Comparison of Mean Accuracy of the different Techniques

Technique	Accuracy
Crossover and Mutation operators	0.71687
Crossover operator	0.71428
Mutation operator	0.72726
Crossover with adaptive parameter control (1)	0.74804
Crossover with adaptive parameter control (2)	0.75064
Crossover with adaptive parameter control (3)	0.72726
Crossover with adaptive parameter control (4)	0.84102
Mutation with adaptive parameter control (5)	0.81538

TABLE IV. GA Parameters and Values

Value	Parameter
Population size	150
No. of generation group	3
Crossover probability	0.9
Mutation probability	0.01
Max iteration	300



(a) Crossover and Mutation. (b) Crossover Operator.

(c) Mutation Operator (d) Adaptive Control (1).

(e) Adaptive control (2). (f) Adaptive control (3).

(g) Crossover Adaptive Control (4). (h) Mutation Adaptive Control (5).

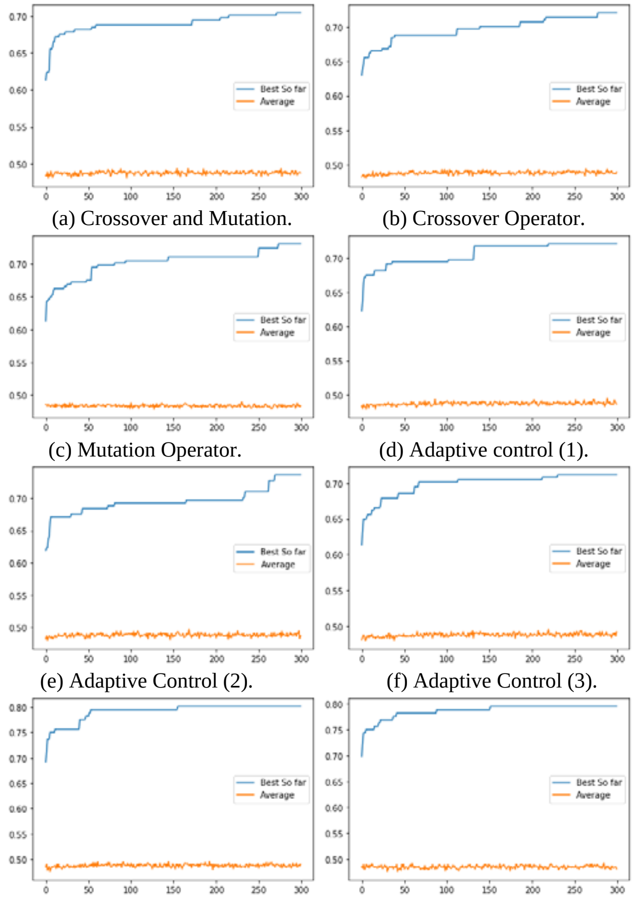
Fig. 4. Scatter Plots of the Eight Techniques.

B. Line Plots Results

Line plots are an excellent way of mapping quantitative variables. They can be either independent or dependent. If both variables are quantitative, the graph will have a slope that consists of the line segment. The latter can be visually interpreted relative to the slope of other lines or can be expressed as an accurate mathematical formula. [20] Fig. 5 shows the line plot of the eight techniques.

In experiment (a), the line plot of generation three achieved an accuracy level of 0.71. In experiment (b), the line plot of generation two achieved an accuracy level of 0.73. In experiment (c), the line plot of generation three achieved an accuracy level of 0.75. In experiment (d), the line plot of generation two achieved an accuracy level of 0.77. In experiment (e), the line plot of generation two achieved an accuracy level of 0.75. In experiment (f), the line plot of generation three achieved an accuracy level of 0.73. In experiment (g), the line plot of generation two achieved an accuracy level of 0.84. Finally, in experiment (h), the line plot of generation two achieved an accuracy level of 0.81.

Fig. 6 and 21, located in the appendix, show the results of the experiments according to the type of genetic operators used.



(a) Crossover and Mutation.

(b) Crossover Operator.

(c) Mutation Operator.

(d) Adaptive control (1).

(e) Adaptive Control (2).

(f) Adaptive Control (3).

(g) Crossover Adaptive Control (4).

(h) Mutation Adaptive Control (5).

Fig. 5. Line Plots of the Eight Techniques.

V. Conclusion

Diabetes comes in many forms. Among the general population, Type 2 Diabetes Mellitus (T2DM) is the most prevalent. The percentage of patients suffering from (T2DM) is approximately 90%. Thus, early and accurate diagnosis can help to reduce mortality rates. In this paper, a combined K-NN and GA model was conducted to improve the accuracy of the K-NN algorithm. The model was implemented by using the Anaconda Python platform on the Pima Indian Diabetes Dataset.

In terms of the accuracy of the proposed models of the two GA operators (crossover, mutation), the crossover with adaptive parameter control (4) is the best technique, achieving a mean accuracy of 84%, 1% better than the best result in the literature review. The combination of K-NN and GA is an effective choice for diagnosing diabetes accurately.

This paper aimed to enhance the performance of the K-NN algorithm. Many techniques have the potential to achieve this goal. This paper developed and used a hybrid GA-K-NN algorithm. Some suggestions for future work include:

- Modifying the mutation and crossover rates by using parameter tuning or parameter control (deterministic parameter control, adaptive parameter control, and self-adaptive parameter control) to achieve greater accuracy.
- Using different equations of parameter tuning or parameter control, which can modify the mutation and crossover rates to increase the accuracy percentage.
- Applying a hybrid GA-K-NN algorithm, which may achieve a good result if used in a real dataset to diagnose diabetes accurately.
- Merging another classification algorithm with GA, which might achieve an accurate diagnosis of diabetes.
- Combining the K-NN algorithm with other algorithms, which may enhance its performance.
- Applying GA with feature selection by using fewer input features, which may give rise to more accurate results.

References

[1] "International Diabetes Federation - Facts & figures", Idf.org, 2020. [Online]. Available: https://www.idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html. [Accessed: 06- May- 2020].

[2] M. Ngugi, J. Njagi, C. Kibiti, J. Ngeranwa and E. Njagi, "Diagnosis of Diabetes Mellitus", International Journal of Diabetes Research, 2012.

[3] M. Pradhan, G. Bamnote, V. Tribhuvan, K. Jadhav, V. Chabukswar and V. Dhobale, "A Genetic Programming Approach for Detection of Diabetes", International Journal Of Computational Engineering Research, vol. 2, no. 6, 2012.

[4] B. Ramageri, "Data Mining Techniques and Applications", Indian Journal of Computer Science and Engineering (IJCSE), 2010.

[5] R. Patil and S. Tamane, "Upgrading the Performance of KNN and Naïve Bayes in Diabetes Detection with Genetic Algorithm for Feature Selection", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJCSE), vol. 3, no. 1, 2018.

[6] A. Thengade and R. Dondal, "Genetic Algorithm – Survey Paper", International Journal of Computer Applications (IJCA), 2012.

[7] lmandoust and M. Bolandraftar, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events Theoretical Background", international Journal of Engineering Research and Applications (IJERA),, 2013.

[8] Y. Christobel and P. Sivaprakasam, "A New Classwise k Nearest Neighbor (CKNN) Method for the Classification of Diabetes Dataset", International Journal of Engineering and Advanced Technology (IJEAT), vol. 2, no. 3, 2013.

[9] Y. Christobel and P. Sivaprakasam, "Improving the performance of k-nearest neighbor algorithm for the classification of diabetes dataset with missing values", International Journal of Computer Engineering and Technology (IJCET), vol. 3, no. 3, 2012.

[10] J. Kandhasamy and S. Balamurali, "Performance Analysis of Classifier Models to Predict Diabetes Mellitus", Procedia Computer Science, vol. 47, pp. 45-51, 2015.

[11] N. Bidi and Z. Elberrichi, "Best Features Selection for Biomedical Data Classification Using Seven Spot Ladybird Optimization Algorithm", International Journal of Applied Metaheuristic Computing, vol. 9, no. 3, pp. 75-87, 2018. Available: 10.4018/ijamc.2018070104.

[12] D. Gupta and U. Ghose, "A comparative study of classification algorithms for forecasting rainfall," 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Noida, 2015, pp. 1-6, doi: 10.1109/ICRITO.2015.7359273.

[13] S. Imandoust and M. Bolandraftar, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background", International Journal of Engineering Research and Application, vol. 3, no. 5, 2013. [Accessed 23 November 2020].

[14] T. Alam, S. Qamar, A. Dixit and M. Benaida, "Genetic Algorithm: Reviews, Implementations, and Applications", International Journal of Engineering Pedagogy (IJEP), 2020. [Accessed 23 November 2020].

[15] [4]Z. Piserchia, "Applications of Genetic Algorithms in Bioinformatics", Master, University of California, 2018.

[16] M. Panwar, A. Acharyya, R. Shafik and D. Biswas, "K-Nearest Neighbor Based Methodology for Accurate Diagnosis of Diabetes Mellitus", in Sixth International Symposium on Embedded Computing and System Design (ISED)., 2016.

[17] L. Haldurai, T. Madhubala and R. Rajalakshmi, "A Study on Genetic Algorithm and its Applications", international journal of computer sciences and engineering (JCSE), vol. 4, no. 10, 2016. [Accessed 4 June 2020].

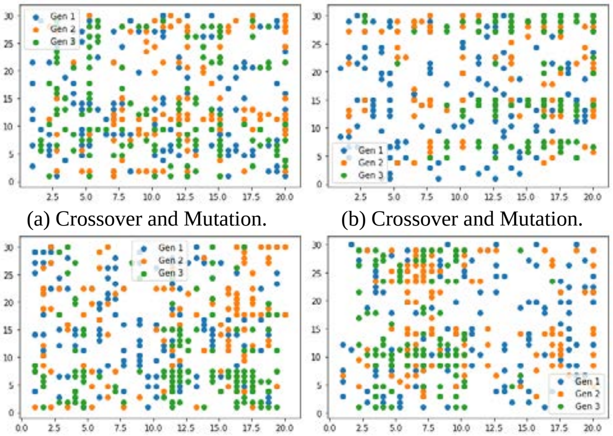
[18] K. Abd Halim, A. Jaya and A. Fadzil, "Data Pre-Processing Algorithm for Neural Network Binary Classification Model in Bank Tele-Marketing", International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 3, pp. 272-277, 2020.

[19] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri and V. Prasath, "Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach", Information, vol. 10, no. 12, p. 390, 2019. Available: 10.3390/info10120390.

[20] "Graphing - Line Graphs and Scatter Plots", Labwrite.ncsu.edu, 2020. [Online]. Available: https://labwrite.ncsu.edu/res/gh/gh-linegraph.html. [Accessed: 06- May- 2020].

APPENDIX

The scatter plot of the Crossover and Mutation technique:

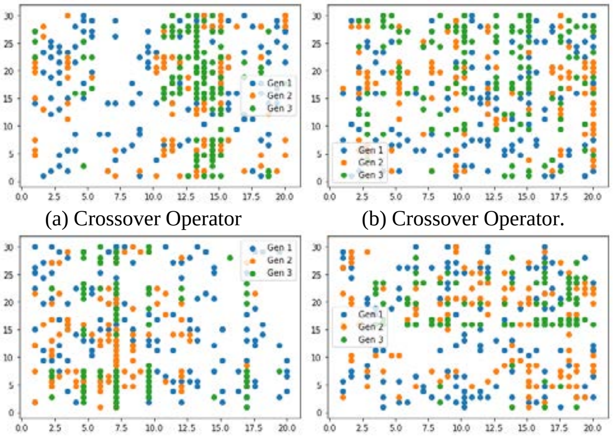


(a) Crossover and Mutation. (b) Crossover and Mutation.

(c) Crossover and Mutation. (d) Crossover and Mutation.

Fig. 6. Fitness Distribution Values of Crossover and Mutation.

The scatter plot of the Crossover only technique:

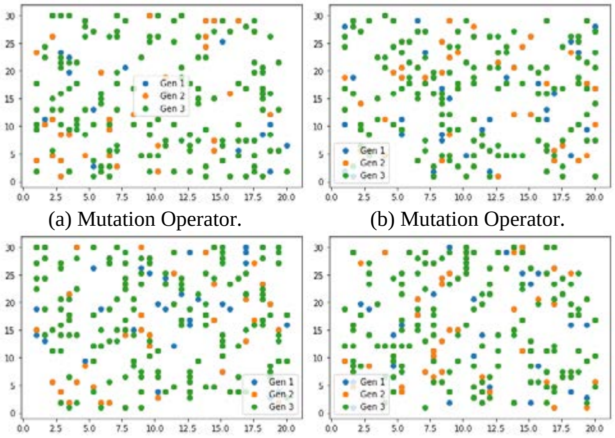


(a) Crossover Operator (b) Crossover Operator.

(c) Crossover Operator. (d) Crossover Operator.

Fig. 7. Fitness Distribution Values of Crossover Only.

The scatter plot of the Mutation only technique:

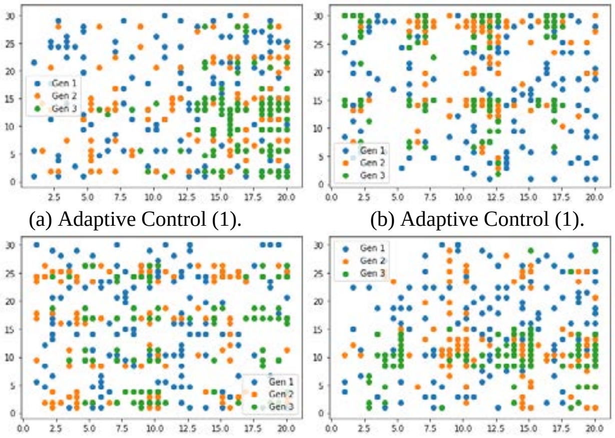


(a) Mutation Operator. (b) Mutation Operator.

(c) Mutation Operator (d) Mutation Operator.

Fig. 8. Fitness Distribution Values Mutation Only.

The scatter plot of the Proposed Crossover Equation (1):

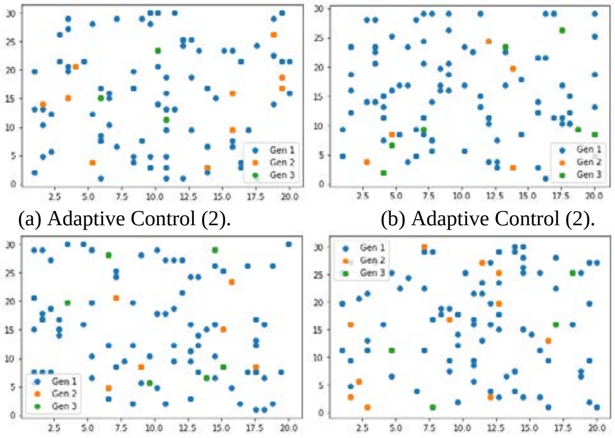


(a) Adaptive Control (1). (b) Adaptive Control (1).

(c) Adaptive Control (1). (d) Adaptive Control (1).

Fig. 9. Fitness Distribution Values of Crossover Equation (1).

The scatter plot of the Proposed Crossover Equation (2):

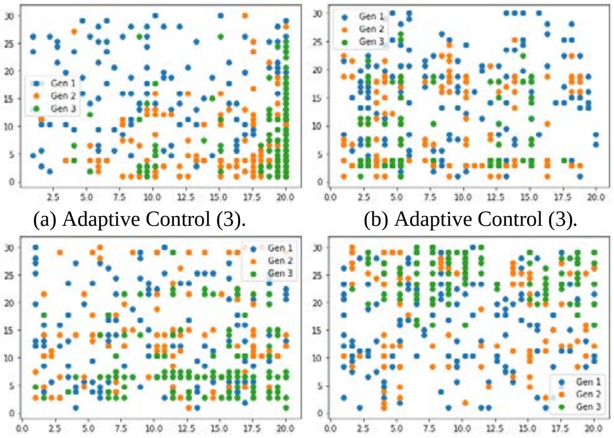


(a) Adaptive Control (2). (b) Adaptive Control (2).

(c) Adaptive Control (2). (d) Adaptive Control (2).

Fig. 10. Fitness Distribution Values of Crossover Equation (2).

The scatter plot of the Proposed Crossover Equation (3):

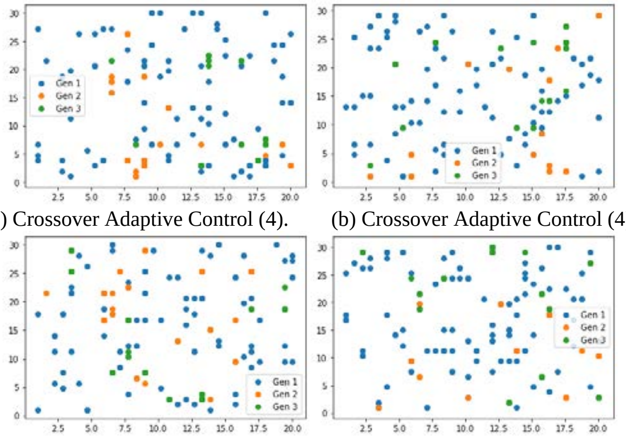


(a) Adaptive Control (3). (b) Adaptive Control (3).

(c) Adaptive Control (3). (d) Adaptive Control (3).

Fig. 11. Fitness Distribution Values of Crossover Equation (3).

The scatter plot of the Proposed Crossover Equation (4):

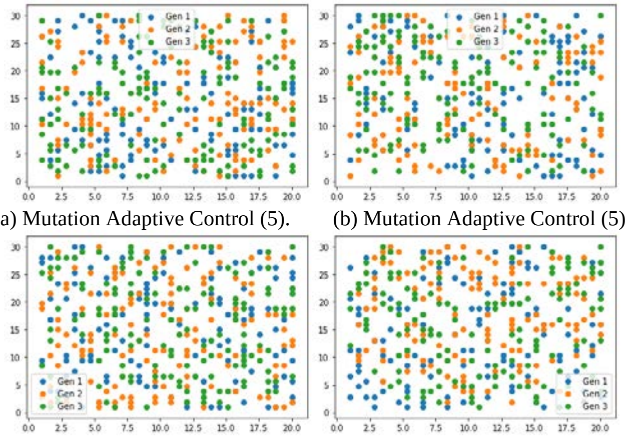


(a) Crossover Adaptive Control (4). (b) Crossover Adaptive Control (4).

(c) Crossover Adaptive Control (4). (d) Crossover Adaptive Control (4).

Fig. 12. Fitness Distribution Values of Crossover Equation (4).

The scatter plot of the Proposed Mutation Equation (5):

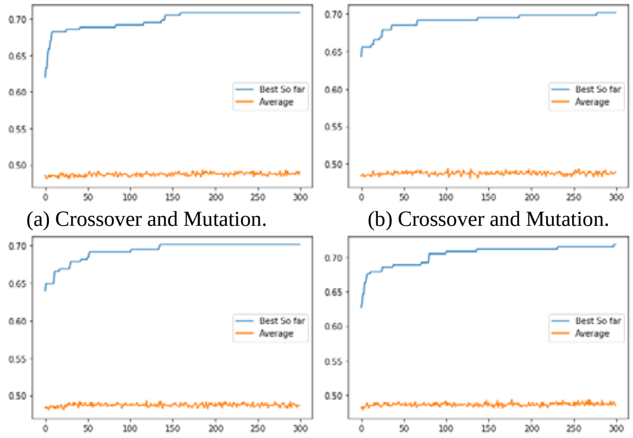


(a) Mutation Adaptive Control (5). (b) Mutation Adaptive Control (5).

(c) Mutation Adaptive Control (5). (d) Mutation Adaptive Control (5).

Fig. 13. Fitness Distribution Values of Mutation Equation (5).

The line plot of the Crossover and Mutation technique:

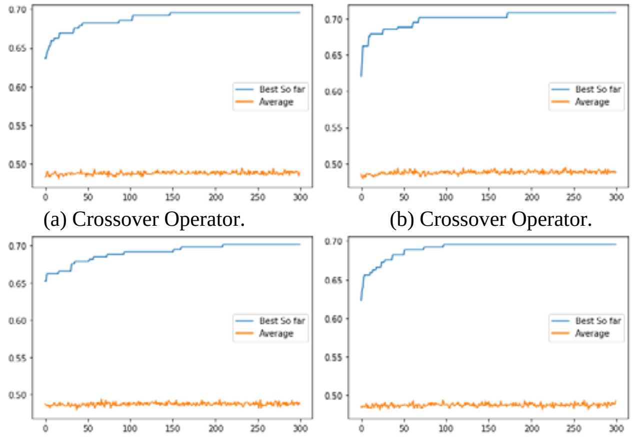


(a) Crossover and Mutation. (b) Crossover and Mutation.

(c) Crossover and Mutation. (d) Crossover and Mutation.

Fig. 14. Fitness Distribution Values of Crossover and Mutation.

The line plot of the Crossover only technique:

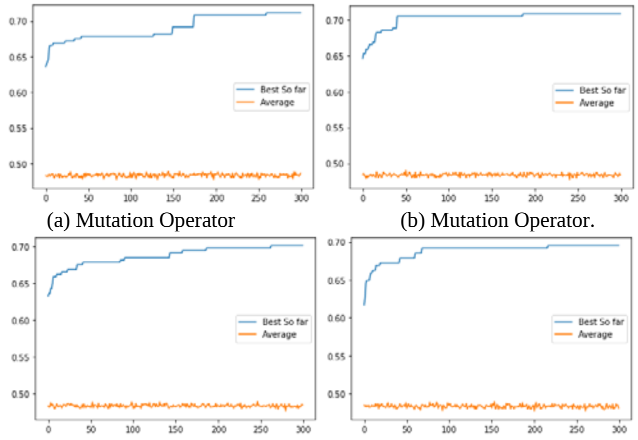


(a) Crossover Operator. (b) Crossover Operator.

(c) Crossover Operator (d) Crossover Operator.

Fig. 15. Fitness Distribution Values of Crossover Only.

The line plot of the Mutation only technique:

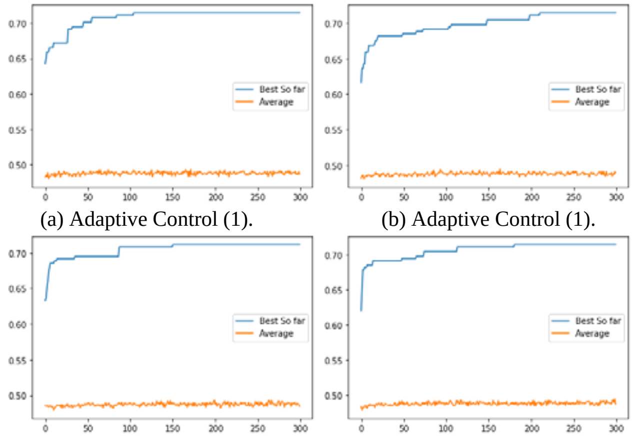


(a) Mutation Operator (b) Mutation Operator.

(c) Mutation Operator (d) Mutation Operator.

Fig. 16. Fitness Distribution Values of Mutation Only.

The line plot of the Proposed Crossover Equation (1):

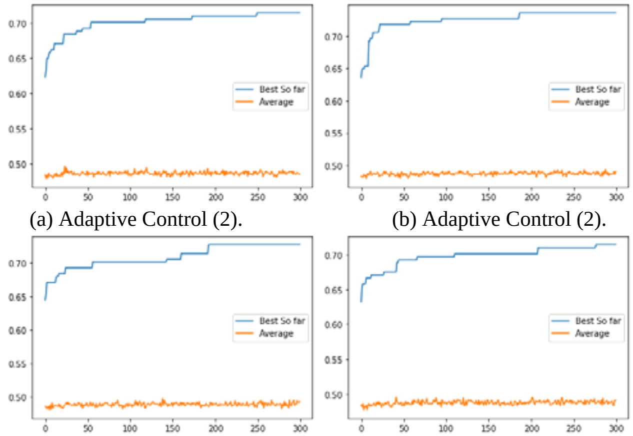


(a) Adaptive Control (1). (b) Adaptive Control (1).

(c) Adaptive Control (1). (d) Adaptive Control (1).

Fig. 17. Fitness Distribution Values Crossover Equation (1).

The line plot of the Proposed Crossover Equation (2):

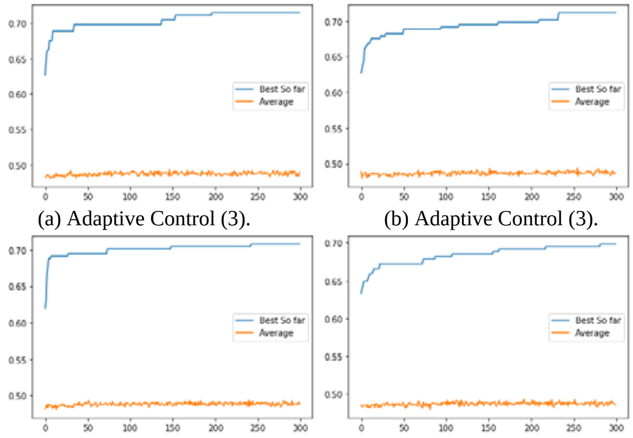


(a) Adaptive Control (2). (b) Adaptive Control (2).

(c) Adaptive Control (2). (d) Adaptive Control (2).

Fig. 18. Fitness Distribution Values Crossover Equation (2).

The line plot of the Proposed Crossover Equation (3):

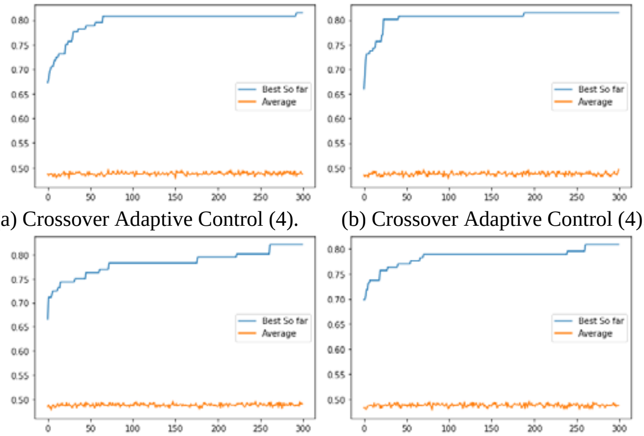


(a) Adaptive Control (3). (b) Adaptive Control (3).

(c) Adaptive Control (3). (d) Adaptive Control (3).

Fig. 19. Fitness Distribution Values Crossover Equation (3).

The line plot of the Proposed Crossover Equation (4):

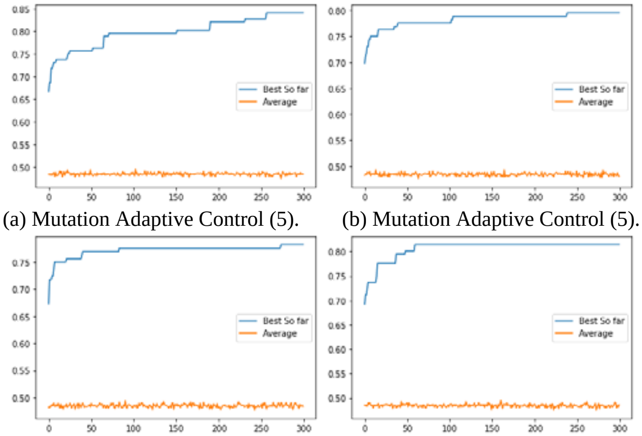


(a) Crossover Adaptive Control (4). (b) Crossover Adaptive Control (4).

(c) Crossover Adaptive Control (4). (d) Crossover Adaptive Control (4).

Fig. 20. Fitness Distribution Values Crossover Equation (4).

The line plot of the Proposed Mutation Equation (5):



(a) Mutation Adaptive Control (5). (b) Mutation Adaptive Control (5).

(a) Mutation Adaptive Control (5). (b) Mutation Adaptive Control (5).

Fig. 21. Fitness Distribution Values Mutation Equation (5).