# COMPARISON OF DECISION TREE ALGORITHM AND K-NEAREST NEIGHBOR (KNN) ALGORITHM PERFORMANCE IN DIABETES CASE STUDY

**[1]Silvano Pratama Jubilate Deo, [2]Y.B. Dwi Setianto**
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata
[2]setianto@unika.ac.id

## ABSTRACT

*Diabetes is a chronic metabolic disease characterized by elevated blood sugar levels, which can result in damage to the eyes and vital organs. Type 2 diabetes is a variant of diabetes that most often affects adults over 18 years old, the symptoms caused by this variant are not very noticeable and to identify it requires a long test process. The use of classification algorithms in predicting diabetes, can help minimize the risk in the early stages of the disease and help health practitioners in controlling the impact of diabetes. In this study, the authors compare the performance of Decision Tree and K-Nearest Neighbor algorithms in predicting diabetes, on the Pima Indian Diabetes dataset. Both algorithm models were trained with 3 dataset sharing ratios, which are 80:20, 70:30 and 65:35. In addition, the authors also implemented GridSearchCV hyperparameter tuning to find the best parameters of both models. Accuracy, precision, recall and F-1 score of the two models are used to determine which model has the best performance. The results show that the Decision Tree algorithm without hyperparameter tuning has the best performance at a ratio of 70:40, resulting in accuracy 83.33%. K=7 is the most optimal K value in the KNN algorithm, resulting in an accuracy of 77.65%. Hyperparameter tuning GridSearchCV can work optimally at a ratio of 80:20 and 65:35, in finding the best parameters in decision algorithms. But there is still overfitting in decision tree algorithms.*

**Keywords:** Diabetes, Decision Tree, K-Nearest Neighbor, GridSearchCV

## 1. INTRODUCTION

Diabetes is a chronic metabolic disease characterized by elevated blood sugar levels that result in damage to the eyes, and vital organs such as the heart, blood vessels, kidneys, and nerves. Diabetes has 2 variants, namely type 1 and type 2 diabetes. Type 2 diabetes is the most common diabetic disease that affects adults above the age of 18 years. The symptoms of type 2 diabetes are similar to the symptoms of type 1 diabetes, but often the symptoms are not very striking and to identify them requires a long test process.

Prediction of various diseases, especially in the discipline of computer science, has been carried out by many researchers, through data mining and machine learning approaches using classification algorithms. The use of classification algorithms in predicting diseases can minimize the risk in the early stages of the disease and help health practitioners in controlling the impact of diabetes.

In this project, the author implements the Decision Tree and K-Nearest Neighbor (KNN) classification algorithms in detecting diabetes early. This research uses the Pima Indians Diabetes dataset obtained through Kaggle, and the performance of these two algorithms is tested using accuracy, precision, recall, F-1 score.

## 2. LITERATUR STUDY

Sivanesan, R et al. [1] performed Diabetes Mellitus diagnosis using classification algorithms to divide data into certain classes with the aim of finding functions or models based on their characteristics. The Decision Tree algorithm was applied with a data division of 65% train data and 35% test data and evaluated with different metrics, the experimental results showed the effectiveness of the decision tree algorithm from various data sets with an accuracy of 84.11%.

Talha Mahboob Alam et al. [2] this study aims to anticipate diabetes in the early stages. The Pima Indian dataset was cleaned by removing zero values in the glucose, blood pressure, skin thickness, insulin, and BMI attributes, removed and replaced with their median values.

G Tripathi et al. [3] this study aims to create a machine learning model for early prediction of diabetes. In the data preprocessing stage, imbalanced data is handled using the oversampling method, which aims to avoid bias and facilitate model evaluation. Minmax normalization is used to equalize each feature in a certain range.

FA Khaleel et al [4] this research aimed to predict diabetes in patients using regression, naïve bayes and KNN algorithms. The Indian pima dataset is MinMax normalized with a scale range of 0 and 1, then divided into 70% training data and 30% testing data. The results obtained from this study, using the Logistic Regression (LR), Naive Bayes (NB), and K-Nearest Neighbors algorithms were 94%, 79%, and 69%, respectively.

Abedini, M. et al. [5] applied machine learning in predicting the risk of diabetes in the early stages of the disease to facilitate health workers in controlling diabetes. The PID dataset is divided into 30% test data and 70% training data. The results of this study show the highest accuracy obtained by the ensemble model algorithm with 83.08% accuracy, logistic regression with 80.71% accuracy, and decision tree with 65.84% accuracy.

Karyono, G. [6] this research applies data mining techniques to diagnose diabetes mellitus. K=9, 10, 11, 12, and 13 values are used in the KNN algorithm. Karyono G. mentioned, the higher the value of K used, can result in a decrease in the model because more neighbors are used. The limitation of this research is that it requires handling missing attribute values.

Hashi, E. K. et al. [7] suggested a system that can help health workers predict diseases. The Pima Indian Diabetes dataset was divided into 70% train data and 30% test data, the results of this study resulted in 90.43% accuracy on the Decision Tree and 79.96% accuracy with a value of K = 7 on KNN.

Kandhasamy, J. P. et al. [8] this study aims to compare the performance of J48 Decision Tree, KNN, and Random Forest classification algorithms to predict patients with diabetes mellitus, using the WEKA platform. Noisy in the Pima Indian dataset is removed by treating zero values as missing values, and replacing them with mean and median values.

P Sinha et al. [9] applied a classification approach in a chronic kidney disease prediction system. SVM and KNN algorithms were evaluated based on accuracy, precision, recall and F-measure metrics. The results showed that the KNN algorithm performed better in predicting chronic kidney disease.

B Pranto et al. [10] applied machine learning in medical research. Pima Indian and Kurmitoal Hospital datasets were removed and replaced zero values with mean, normalized minmax in the range of 0 and1. GridSearchCV was used to find the best parameters for the model. The parameters tuned in this study include the K value and euclidean distance metric in KNN, maximum depth and information gain entropy in decision tree, and number of trees in random forest algorithm.

## 3. RESEARCH METHODOLOGY

### 3.1.Dataset Collection

This research used the Pima Indian Diabetes (PID) dataset obtained through https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database. This dataset contains patient diagnosis information based on the number of pregnancies. glucose levels, blood pressure, skin thickness, insulin, BMI, diabetes pedigree and age.

### 3.2 Data Preprocessing

This research used the Pima Indian Diabetes (PID) dataset obtained through https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database. This data set contained information on patient diagnosis based on the number of pregnancies. glucose levels, blood pressure, skin thickness, insulin, BMI, diabetes pedigree and age. Data imbalance is an often encountered problem in classification and influences prediction results and accuracy [3]. Handling imbalances in the data is done by the method of oversampling, which equalizes the minority class by taking random samples, until the number is equal to the majority class. Outlier or noise is deviant data that has a value that is too high or too low in the data. Z-score method is used to calculated how far away a data point is, then remove or eliminate data based on threshold of 3 and -3. The final step of data pre-processing is data normalization, which aims to equalize each feature in the data with a certain scale range. MinMax normalization is used to scale each feature to 0 and 1[3][11].

### 3.3 Train Test Split Data

In this research, the datasets are divided into training data and testing data with 3 categories, namely, 80% training data and 20% testing data, 70% training data and 30% testing data [6],[8],[11], 65% training data and 35% testing data [1].

### *3.4 Decision Tree*

The following are the steps taken in implementing the Decision Tree algorithm in this research:

1. Pima Indian dataset that has entered the preprocessing process and division into X and Y, then train test split data with 3 categories predetermined.
2. Classification is performed with the Decision Tree algorithm based on 3 categories of train test split data that have been determined
3. Obtained classification results and calculated the accuracy, precision, recall and F-1 score values.

### *3.5 K-Nearest Neighbor*

The following are the steps taken in applying the K-Nearest Neighbor algorithm in this study:

1. Pima Indian dataset that has entered the preprocessing process and division into X and Y, then train test split data with 3 categories yang predetermined.
2. Determining the K value to be used in classification, in this study the K values used are 7, 9, 10, 11, 12 and 13 [7][8]. Conduct training and testing based on the train test split data category that has been determined.
3. Obtained classification results and calculated the accuracy, precision, recall and F-1 score values.

### *3.6 Model Evaluation*

The two algorithm are evaluated according to the accuracy, precision, recall and F-1 Score values. In this research, a comparison is made and find which algorithm has better results in predicting diabetic patients.

### *3.8 Hyperparameter Tuning with GridSearchCV*

GridSearchCV is a hyperparameter tuning method that work to find the optimal parameters of a particular algorithm or model. GridSearchCV comprises of several parameter like estimator, grid parameter, and scoring. The estimator is the model or algorithm used. Parameter grid is a dictionary that contains the parameters to be tried. Scoring is a method used to evaluate the performance of the model through the cross validation process on the training set. In this research, the author uses 5 cross validation. Maximum depth 1-40 and criterion entropy are detuning parameters in the Decision Tree algorithm. N Neighbors (7, 9, 10, 11, 12, 13) and euclidean metrics are parameters that are tuned in the KNN algorithm.

## 4. RESULTS AND DISCUSSION

In this research, 2 stages of testing were carried out, namely without hyperparameter tuning and using GridSearchCV hyperparameter tuning, with different datasets. The results of the first stage of research can be seen in Table 4.1 :

**Table 4.1 Pengujian tanpa hyperparameter tuning**

| 80% Data Training dan 20% Data Testing | | | | |
|---|---|---|---|---|
| Algoritma | Accuracy | Precision | Recall | F-1 Score |
| Decision Tree | 78.72 | 70.19 | 89.02 | 78.49 |
| KNN n=7 | 76.59 | 69.38 | 82.92 | 75.55 |
| KNN n=9 | 77.12 | 69.69 | 84.14 | 76.42 |
| KNN n=10 | 75.53 | 69.14 | 79.26 | 73.86 |
| KNN n=11 | 75.00 | 67.67 | 81.70 | 74.03 |
| KNN n=12 | 73.93 | 67.36 | 78.04 | 72.31 |
| KNN n=13 | 72.87 | 65.34 | 80.48 | 72.13 |
| 70% Data Training dan 30% Data Testing | | | | |
| Algoritma | Accuracy | Precision | Recall | F-1 Score |
| Decision Tree | 83.33 | 76.92 | 91.60 | 83.62 |
| KNN n=7 | 77.65 | 72.66 | 83.20 | 77.58 |
| KNN n=9 | 74.46 | 69.28 | 80.91 | 74.64 |
| KNN n=10 | 75.17 | 71.32 | 77.86 | 74.45 |
| KNN n=11 | 74.46 | 69.28 | 80.91 | 74.64 |
| KNN n=12 | 74.46 | 70.34 | 77.86 | 73.91 |
| KNN n=13 | 73.75 | 68.15 | 81.67 | 74.30 |
| 65% Data Training dan 35% Data Testing | | | | |
| Algoritma | Accuracy | Precision | Recall | F-1 Score |
| Decision Tree | 77.50 | 73.56 | 82.05 | 77.57 |
| KNN n=7 | 75.68 | 71.59 | 80.76 | 75.90 |
| KNN n=9 | 75.07 | 71.02 | 80.12 | 75.30 |
| KNN n=10 | 73.55 | 70.65 | 75.64 | 73.06 |
| KNN n=11 | 73.25 | 68.88 | 79.48 | 73.80 |
| KNN n=12 | 73.86 | 70.83 | 76.28 | 73.45 |
| KNN n=13 | 72.64 | 68.13 | 79.48 | 73.37 |

In table 4.1, with a ratio of 80% training data and 20% testing data, the best performance is obtained on the decision tree with an accuracy of 78.72%, precision 70.19%, recall 89.02% and F-1 Score 78.49%. While in KNN, the value of K = 9 is the most optimal K value among other K values, with an accuracy of 77.12%, precision 69.69%, recall 84.14% and F-1 score 76.42%.

With a ratio of 70% training data and 30% testing data, the best performance is obtained on the decision tree with 83.33% accuracy, 76.92% precision, 91.60% recall and 83.62% F-1 Score. While in KNN, the value of K = 7 is the most optimal K value among other K values, with the results of 77.65% accuracy, 72.66% precision, 83.20% recall and 77.58% F-1 score.

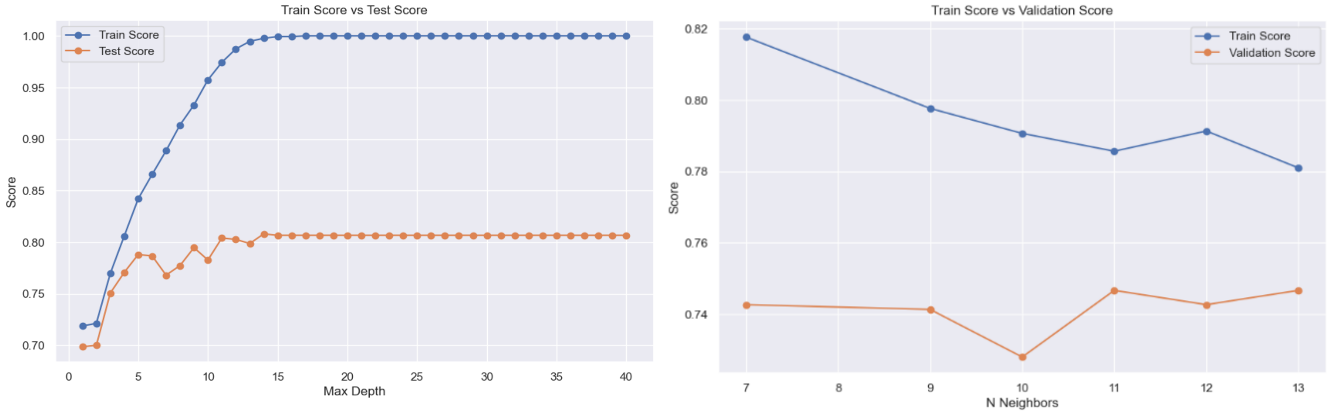
In the data division ratio of 65% training data and 35% testing data, the best performance is obtained on the decision tree with an accuracy of 77.50%, precision 73.56%, recall 82.05% and F-1 score 77.57%. While in KNN, the value of K = 7 is the most optimal K value among other K values, with accuracy results of 75.68% accuracy, 71.59% precision, 80.76% recall and F-1 score 75.90%. In the KNN algorithm there is a decrease in accuracy along with the higher value of K used,  this occurs in all three data divisions. This is in accordance with what was done by Karyono G [6] in his research.

The second testing in this research is to implement GridSearchCV hyperparameter tuning, to find the best parameters from both models. In Decision Tree algorithm, the detuning parameters are maximum depth (range 1-40) and criterion entropy. While in KNN algorithm, the parameters are N neighbors (7, 9, 10, 11, 12, 13) and Euclidean distance metric.

**Table 4.2 Result GridSearchCV ratio 80:20**

| Algoritma | Parameter | Train Score | Test Score | Accuracy | Precision | Recall | F-1 Score |
|---|---|---|---|---|---|---|---|
| Decision Tree | Max_depth : 14<br>Criterion : entropy | 99.76 | 80.80 | 82.97 | 74.50 | 92.68 | 82.60 |
| KNN | n_neighbors : 11<br>metric : Euclidean | 78.56 | 74.66 | 75.00 | 67.67 | 81.70 | 74.03 |

In table 4.2 the best parameters max_depth 14 and criterion entropy obtained on the Decision Tree resulted in an accuracy of 82.97%, precision 74.50%, recall 92.68% and F-1 Score 82.60%, these results are higher when compared to the first stage of testing. For KNN with the best parameters n_neighbors : 11 and metric: euclidean, resulted in an accuracy of 75.00%, precision 67.67%, recall 81.70% and F-1 Score 74.03%, this test result is the same as K = 11 in



**Gambar 1.** Visualisasi Train dan Test Score

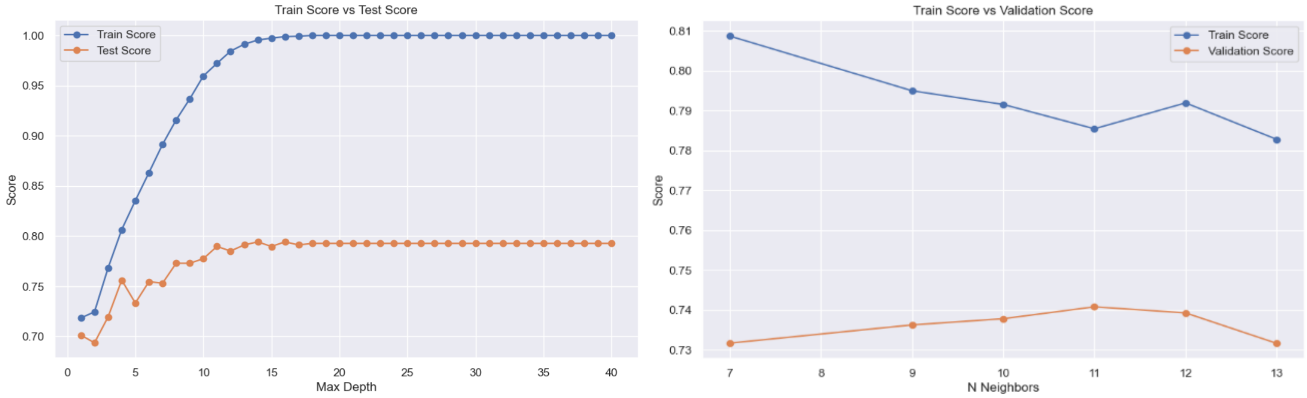
Figure 1, shows the visualization of train and test scores for both algorithms. Training score is an accuracy that measures how well the model fits the training data, while test score is an accuracy that measures how well the model performs on a separate data set or test set. In the

Decision Tree algorithm, there is an increase in the train score along with the higher maximum depth value used. This is an overfitting condition caused by the maximum depth used is too high, so that the model recognizes more patterns in the training data and results in errors in the test data. Whereas in KNN, there is a decrease in the train score value, along with the higher K value used.

**Table 4.3 Result GridSearchCV ratio 70:30**

| Algoritma | Parameter | Train Score | Test Score | Accuracy | Precision | Recall | F-1 Score |
|---|---|---|---|---|---|---|---|
| Decision Tree | Max_depth : 16<br>Criterion : entropy | 99.88 | 79.42 | 81.91 | 75.97 | 89.31 | 82.10 |
| KNN | n_neighbors : 11<br>metric : Euclidean | 78.54 | 74.08 | 74.46 | 69.28 | 80.91 | 76.64 |

In table 4.3 the best parameters max_depth 16 and criterion entropy obtained on the Decision Tree resulted in an accuracy of 81.91%, precision 75.97%, recall 89.31% and F-1 Score 82.10%. For KNN with the best parameters n_neighbors : 11 and metric: euclidean, resulting in accuracy 74.46%, precision 69.28%, recall 80.91% and F-1 Score 74.64%. the results of this test



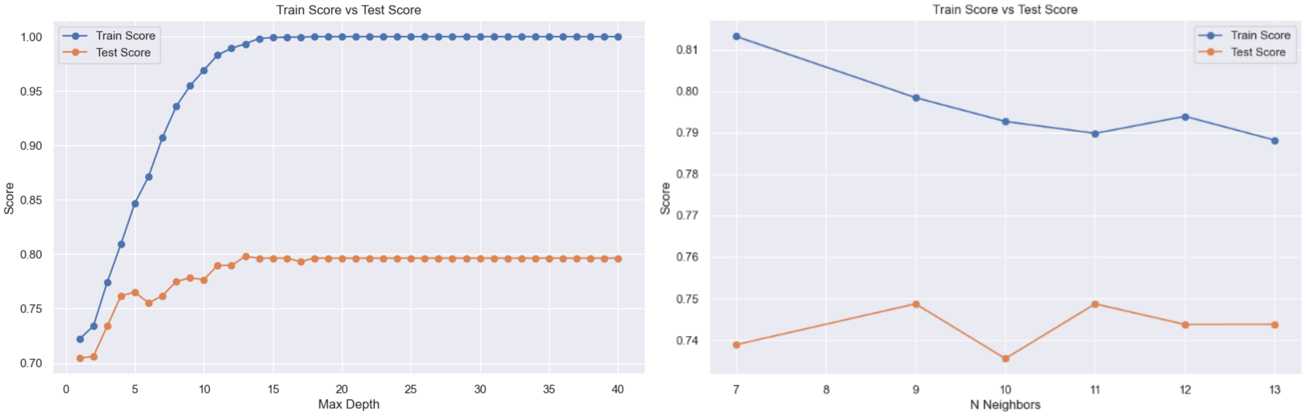
**Gambar 2.** Visualisasi Train dan Test Score

Figure 2, shows the visualization of train and test scores for both algorithms. Training score is an accuracy that measures how well the model fits the training data, while test score is an accuracy that measures how well the model performs on a separate data set or test set. In the Decision Tree algorithm, there is an increase in the train score along with the higher maximum depth value used. This is an overfitting condition caused by the maximum depth used is too high, so that the model recognizes more patterns in the training data and results in errors in the test data. Whereas in KNN, there is a decrease in the train score value, along with the higher K value used.

**Table 4.4 Result GridSearchCV ratio 65:35**

| Algoritma | Parameter | Train Score | Test Score | Accuracy | Precision | Recall | F-1 Score |
|---|---|---|---|---|---|---|---|
| Decision Tree | Max_depth : 13<br>Criterion : entropy | 98.34 | 79.80 | 79.02 | 74.57 | 84.61 | 82.10 |
| KNN | n_neighbors : 9<br>metric : Euclidean | 79.26 | 74.87 | 75.07 | 71.02 | 80.12 | 74.30 |

In table 4.4 the best parameters max_depth 13 and criterion entropy obtained on the Decision Tree resulted in an accuracy of 79.02%, precision 74.57%, recall 84.61% and F-1 score 82.10%, these results are higher when compared to the first stage of testing. For KNN with the best parameters n_neighbors : 9 and metric: euclidean, resulted in an accuracy of 75.07%, precision 71.02%, recall 80.12% and F-1 score 75.30%, these test results are the same as K = 9 in the first



**Gambar 3.** Visualisasi Train dan Test Score

Figure 3, shows the visualization of train and test scores for both algorithms. Training score is an accuracy that measures how well the model fits the training data, while test score is an accuracy that measures how well the model performs on a separate data set or test set. In the Decision Tree algorithm, there is an increase in the train score along with the higher maximum depth value used. This is an overfitting condition caused by the maximum depth used is too high, so that the model recognizes more patterns in the training data and results in errors in the test data. Whereas in KNN, there is a decrease in the train score value, along with the higher K value used.

## 5. CONCLUSION

Based on the results of the two tests with three dataset sharing ratios, several conclusions are drawn, namely the best performance is obtained in the Decision Tree algorithm without hyperparameter tuning at a ratio of 70% training data and 30% testing data, resulting in 83.33%

accuracy, 76.92% precision, 91.60% recall and 83.62% F-1 Score. K = 7 is the most optimal K value in the KNN algorithm, this can be seen from the results of accuracy 77.65%, precision 72.66%, recall 83.20% and F-1 score 77.58%. In addition, the higher the value of K used, it will result in a decrease in model accuracy.

Hyperparameter tuning with GridSearchCV can work well in finding the best parameters in both algorithms. The best performance is obtained in the Decision Tree algorithm with a division ratio of 80:20 and 65:35, which results in higher accuracy, precision, recall and F-1 score than the first stage of testing.

For future research, it is recommended to do further handling of preprocessing data to avoid overfitting, explore and add other parameters to the GridSearchCV tuning hyperparameter, so that the tuning results produced can be much more optimal.

## DAFTAR PUSTAKA

[1] Sivanesan, R., Devika, K., & Dhivya, R. (2017). A Review on Diabetes Mellitus diagnoses using classification on Pima Indian Diabetes Data Set. In *International Journal of Advance Research in Computer Science and Management Studies* (Vol. 5, Issue 1). https://www.academia.edu/35039029/A_Review_on_Diabetes_Mellitus_diagnoses_using_classification_on_Pima_Indian_Diabetes_Data_Set

[2] Mahboob Alam, T., Iqbal, M. A., Ali, Y., Wahab, A., Ijaz, S., Imtiaz Baig, T., Hussain, A., Malik, M. A., Raza, M. M., Ibrar, S., & Abbas, Z. (2019). A model for early prediction of diabetes. *Informatics in Medicine Unlocked*, *16*. https://doi.org/10.1016/j.imu.2019.100204

[3] G. Tripathi and R. Kumar, "Early Prediction of Diabetes Mellitus Using Machine Learning," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 1009-1014, doi: 10.1109/ICRITO48877.2020.9197832. https://doi.org/10.1016/j.imu.2019.100204

[4] Alaa Khaleel, F., & Al-Bakry, A. M. (2021). Diagnosis of diabetes using machine learning algorithms. *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2021.07.196

[5] Abedini, M., Bijari, A., & Banirostam, T. (2020). Classification of Pima Indian Diabetes Dataset using Ensemble of Decision Tree, Logistic Regression and Neural Network. *IJARCCE*, *9*(7), 1–4. https://doi.org/10.17148/ijarcce.2020.9701

[6] Karyono, G. (2016). ANALISIS TEKNIK DATA MINING “ALGORITMA C4.5 DAN K-NEAREST NEIGHBOR” UNTUK MENDIAGNOSA PENYAKIT DIABETES MELLITUS. In *Seminar Nasional Teknologi Informasi* (Vol. 12). http://ojs.palcomtech.ac.id/index.php/SNTIBD/article/view/396

[7] E. K. Hashi, M. S. U. Zaman and M. R. Hasan, "An expert clinical decision support system to predict disease using classification techniques," 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 2017, pp. 396-400, doi: 10.1109/ECACE.2017.7912937. https://doi.org/10.1109/ECACE.2017.7912937

[8] Kandhasamy, J. P., & Balamurali, S. (2015). Performance analysis of classifier models to predict diabetes mellitus. *Procedia Computer Science*, *47*(C), 45–51. https://doi.org/10.1016/j.procs.2015.03.182

[9] Sinha, P., & Sinha, P. (2015). Comparative study of chronic kidney disease prediction using KNN and SVM. *International Journal of Engineering Research and Technology*, *4*(12), 608-

12. https://www.ijert.org/research/comparative-study-of-chronic-kidney-disease-prediction-using-knn-and-svm-IJERTV4IS120622.pdf

[10] Pranto, B., Mehnaz, S. M., Mahid, E. B., Sadman, I. M., Rahman, A., & Momen, S. (2020). Evaluating machine learning methods for predicting diabetes among female patients in Bangladesh. *Information (Switzerland), 11*(8). https://doi.org/10.3390/INFO11080374