Optimization of Hyperparameter K in K-Nearest Neighbor Using Particle Swarm Optimization

Muhammad Rizki[1*], Arief Hermawan[2], Donny Avianto[3]

[1,2,3]*Universitas Teknologi Yogyakarta, Indonesia*

*corr_author: 6220211002.rizki@student.uty.ac.id

Abstract - This study aims to enhance the performance of the K-Nearest Neighbors (KNN) algorithm by optimizing the hyperparameter K using the Particle Swarm Optimization (PSO) algorithm. In contrast to prior research, which typically focuses on a single dataset, this study seeks to demonstrate that PSO can effectively optimize KNN hyperparameters across diverse datasets. Three datasets from different domains are utilized: Iris, Wine, and Breast Cancer, each featuring distinct classification types and classes. Furthermore, this research endeavors to establish that PSO can operate optimally with both Manhattan and Euclidean distance metrics. Prior to optimization, experiments with default K values (3, 5, and 7) were conducted to observe KNN behavior on each dataset. Initial results reveal stable accuracy in the iris dataset, while the wine and breast cancer datasets exhibit a decrease in accuracy at K=3, attributed to attribute complexity. The hyperparameter K optimization process with PSO yields a significant increase in accuracy, particularly in the wine dataset, where accuracy improves by 6.28% with the Manhattan matrix. The enhanced accuracy in the optimized KNN algorithm demonstrates the effectiveness of PSO in overcoming KNN constraints. Although the accuracy increase for the iris dataset is not as pronounced, this research provides insight that optimizing the hyperparameter K can yield positive results, even for datasets with initially good performance. A recommendation for future research is to conduct similar experiments with different algorithms, such as Support Vector Machine or Random Forest, to further evaluate PSO's ability to optimize the iris, wine, and breast cancer datasets.

Keywords: Particle Swarm Optimization, K-Nearest Neighbor, Euclidean, Manhattan, breast cancer

I. INTRODUCTION

K Nearest Neighbors (KNN) is a supervised learning-based classification algorithm [1]–[3], KNN has been used in a wide range of research fields [4]–[7], Some research fields that apply KNN as a classification algorithm include the health sector [6], e-commerce [8], Detecting Vehicle [9]. Although KNN has been widely used in various fields, it has some limitations, including the difficulty of determining the appropriate hyperparameter K. This difficulty arises because an incorrect value of K can lead to overfitting or underfitting problems. If the value of K is too small, the KNN model tends to be too sensitive to small fluctuations in the data, resulting in a less reliable model for new data. On the other hand, if the value of K is too large, the model may become too general and lose the ability to capture significant patterns or variations in the data [10]-[11]. Some earlier studies have attempted to address this issue, such as the research conducted by Maincer in 2022. This study compared the methods of Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) for diagnosing seven classes of sensor errors in a manipulator robot. Simulation results on the Selective Compliance Assembly Robot Arm (SCARA) model showed that SVM with Particle Swarm Optimization (PSO) achieved an accuracy of 96.95%, while KNN achieved a correlation of up to 94.62%. This confirms that SVM with PSO is more accurate in diagnosing errors in manipulator robots [12], Further research conducted by Bui in 2019 developed a new artificial intelligence technique, PSO-KNN (Particle Swarm Optimization-K-Nearest Neighbors), to estimate ground vibrations induced by explosions (PPV). The study optimized the hyperparameters of the K-Nearest Neighbors algorithm using Particle Swarm Optimization and compared it with benchmark models such as random forest, support vector regression, and empirical techniques. The research successfully improved the efficiency of the PSO-KNN model, particularly PSO-KNN-T, identified as a powerful tool for mitigating undesirable impacts caused by PPV in surface mining [13], And there are still many other studies. Although these studies utilize PSO to improve KNN, unfortunately, previous research attempting to optimize KNN hyperparameters only validated on a single type of dataset. In contrast to prior research, this study validates the optimized hyperparameter results of KNN by employing three distinct datasets, namely iris, breast cancer, and wine datasets. The principal objective is to substantiate that Particle Swarm Optimization (PSO) can effectively mitigate the intricacies associated with hyperparameter tuning for the KNN algorithm across diverse datasets. This is evidenced through variations in data volume, encompassing both balanced and imbalanced distributions, as well as across different numbers of classes, including binary and multiclass scenarios. Moreover, it is also important to note that this research also employing two different distance measurement matrices: Euclidean and Manhattan Distance. Particle Swarm Optimization (PSO) is utilized as the optimization algorithm in this research. This choice is made because PSO is considered effective for optimizing K-Nearest Neighbors hyperparameters due to its ability to explore parameter space with flexibility, achieve local and global optima, and provide parallelism that enhances convergence speed without requiring gradient information. Additionally, its simple implementation makes it easily adoptable [14]- [15].

II. METHOD

This study was conducted following the research flow depicted in Fig. 1. The diagram shown in Fig. 1 illustrates that this research is divided into three stages, and these stages are explained in the following points.

A. Literature Study

This stage is carried out to ensure and verify that this research is distinct and crucial to undertake. The literature review phase involves collecting various references limited to the last 5 years (2018 – 2023) from reputable sources such as IEEE or Sciencedirect.

B. Data Collection and Visualization

This research utilizes the wine, breast cancer, and iris datasets, each with its description and visualization as follows:

1) Dataset Wine: The wine dataset consists of 13 attributes and 1 class. The description of the wine dataset can be found in Table I. Table I shows the description of the wine dataset, which comprises 177 rows and 14 columns. One of these columns is the class attribute, which has 3 classes, each marked with an integer. Additionally, the other columns or attributes in the dataset are of float data type. The distribution of data for each class can be observed in Fig. 2.

Fig. 2 illustrates the class distribution, with each class having 59 rows for Class 1, 71 rows for Class 2, and 48 rows for Class 3. Furthermore, the dataset exhibits a very good correlation, indicating its direct usability. This is confirmed by the correlation matrix presented in Fig. 3.

Fig. 1 Research flow

TABLE I
DESCRIPTION DATASET WINE

Instance	Descriptions
Total Row	177
Total Column	14
Total Class	3
Attr : Class	Class | Int
Attr : Alcohol	Float
Attr : Malic Acid	Float
Attr : Ash	Float
Attr : Alcalinity of ash	Float
Attr : Magnesium	Float
Attr : Total phenols	Float
Attr : Flavanoids	Float
Attr : Nonflavanoid phenols	Float
Attr : Proanthocyanins	Float
Attr : Color intensity	Float
Attr : Hue	Float
Attr : OD280/OD315 of diluted wines	Float
Attr : Proline	Float

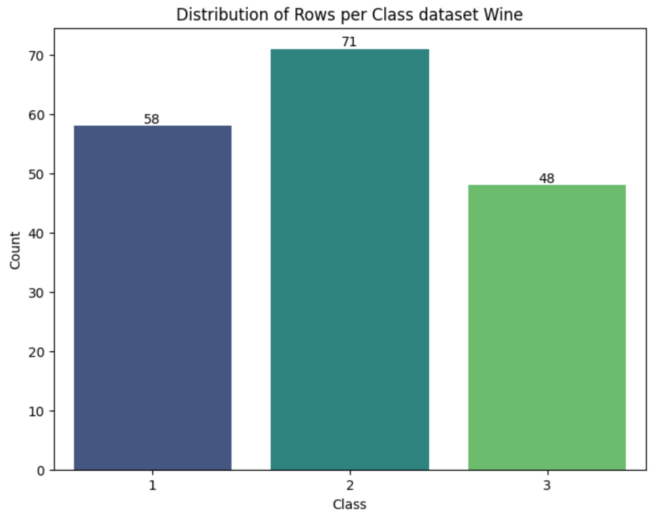

Fig. 2 Distribution of rows per class dataset wine

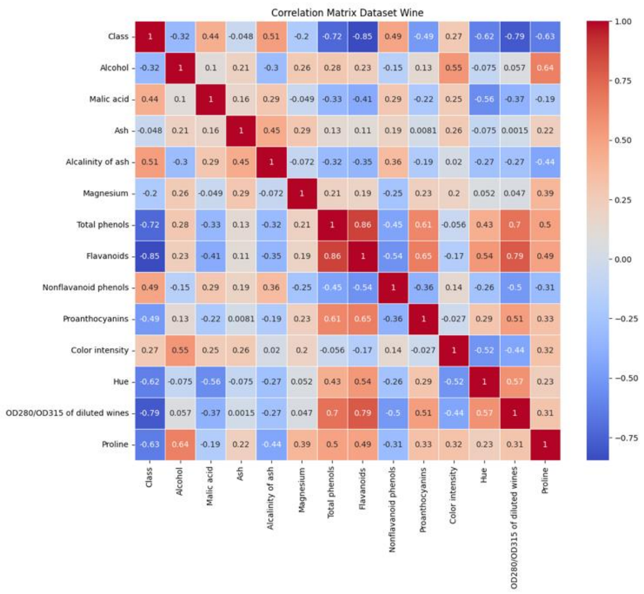

Fig. 3 Correlations matrix of dataset wine

The correlation matrix shown in Fig. 3 indicates that the correlation between variables or attributes is very good. This is demonstrated by the absence of attributes with correlation values exceeding 0.8, indicating that all attributes can be used for classification.

2) Dataset Breast Cancer: The breast cancer dataset is used for the purpose of breast cancer classification. The description of the breast cancer dataset can be found in Table II. The Table II indicates that several columns have data types of range and string. Of course, this needs to be preprocessed to convert the string or range into a numerical value that can be calculated by KNN later. This research preprocesses the breast cancer data by converting nominal data into numerical data. The results of the preprocessing can be seen in the following Table III.

TABLE II
DESCRIPTION DATASET BREAST CANCER

Instance	Descriptions
Total Row	286
Total Column	10
Total Class	2
Attr : Class	Class | Str
Attr : age	Range | Int
Attr : menopause	Str
Attr : tumor-size	Range | Int
Attr : inv-nodes	Range | Int
Attr : node-caps	Boolean
Attr : deg-malig	Int
Attr : breast	Boolean
Attr : breast-quad	Str
Attr : irradiat	Boolean

TABLE III
DESCRIPTION DATASET BREAST CANCER AFTER PREPROCESSING

Instance	Descriptions
Attr : Class	Class | Str
Attr : age	Int
Attr : menopause	Int
Attr : tumor-size	Int
Attr : inv-nodes	Int
Attr : node-caps	Boolean
Attr : deg-malig	Int
Attr : breast	Boolean
Attr : breast-quad	Int
Attr : irradiat	Boolean

After preprocessing, the breast cancer dataset is ready to be used for further analysis. The dataset has 2 classes, namely the no-recurrence-events and recurrence-events classes, the following class distribution of the dataset can be seen in Fig. 4.

The distribution of the class counts shown in Fig. 4 indicates that the breast cancer dataset is dominated by the 'no-recurrence-events' class, which has around 200 rows, while the remaining rows belong to the 'recurrence-events' class. The correlation between parameters/variables can be observed in Fig. 5.

Fig. 5 illustrates the correlation between variables in the breast cancer dataset. The matrix shown in Fig. 5 confirms that the breast cancer dataset used in this research has excellent correlation because there are no attributes or variables with correlation values exceeding 0.4. Therefore, all variables can be directly used without the need for elimination.

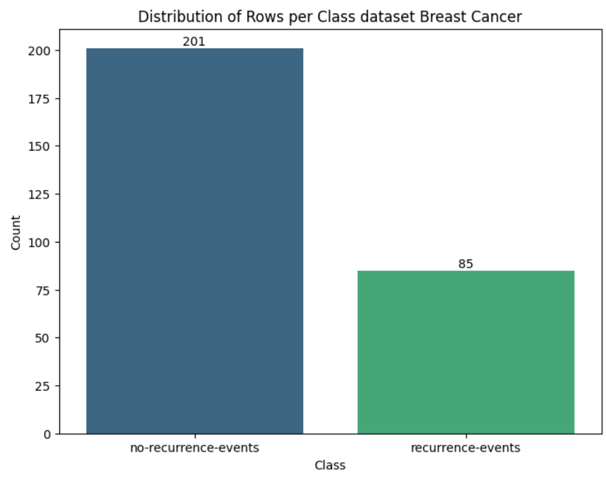

Fig. 4 Distribution row per class dataset breast cancer

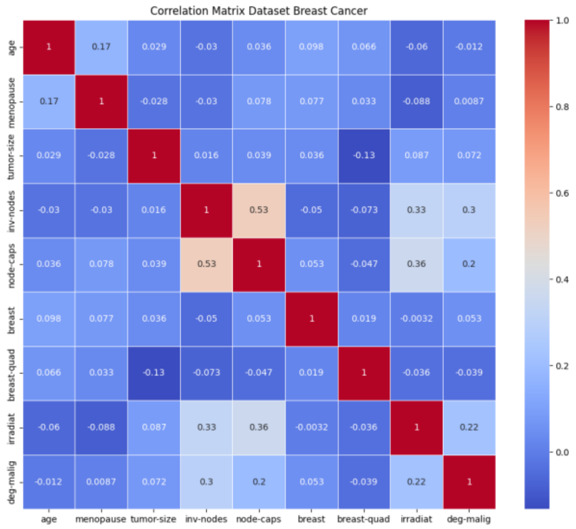

Fig. 5 Correlations matrix of dataset breast cancer

3) Dataset Iris: The iris dataset is a public dataset used to classify the types of iris flowers, containing 3 classes. The class distribution is depicted in Fig. 6.

Each class has an equal distribution of 50 rows of data. Further description of the iris dataset is provided in Table IV. Table IV shows that the iris dataset has 5 columns, with 4 of them being attributes that determine the classification results. These four attributes are already in float data format, making them directly usable. The distribution of correlations between variables in the dataset can be seen in Fig. 7.

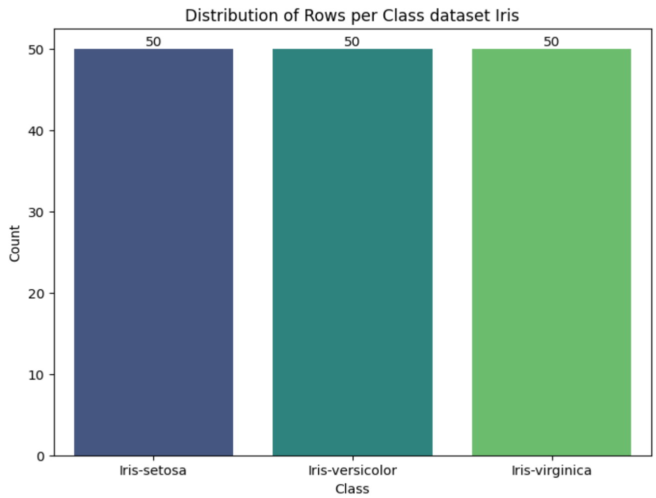

Fig. 6 Distribution of rows per class dataset iris

TABLE IV
DESCRIPTION DATASET IRIS

Instance	Descriptions
Total Row	150
Total Column	5
Total Class	3
Attr : Class	String
Attr : sepal length	Float
Attr : sepal width	Float
Attr : petal length	Float
Attr : petal width	float

There is a variable with a very high correlation level, namely the 'petal length' variable with the 'petal width' variable, with a value of 0.96. However, despite having a very high correlation, this variable is not removed because the dataset is relatively small, consisting of only 150 rows.

C. Combination PSO KNN

This research optimizes KNN using PSO, as illustrated in Fig. 8.

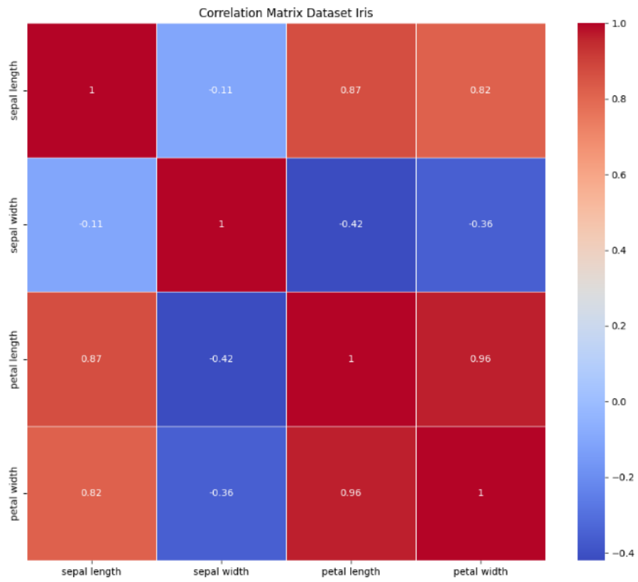

Fig. 7 Correlations matrix of dataset iris

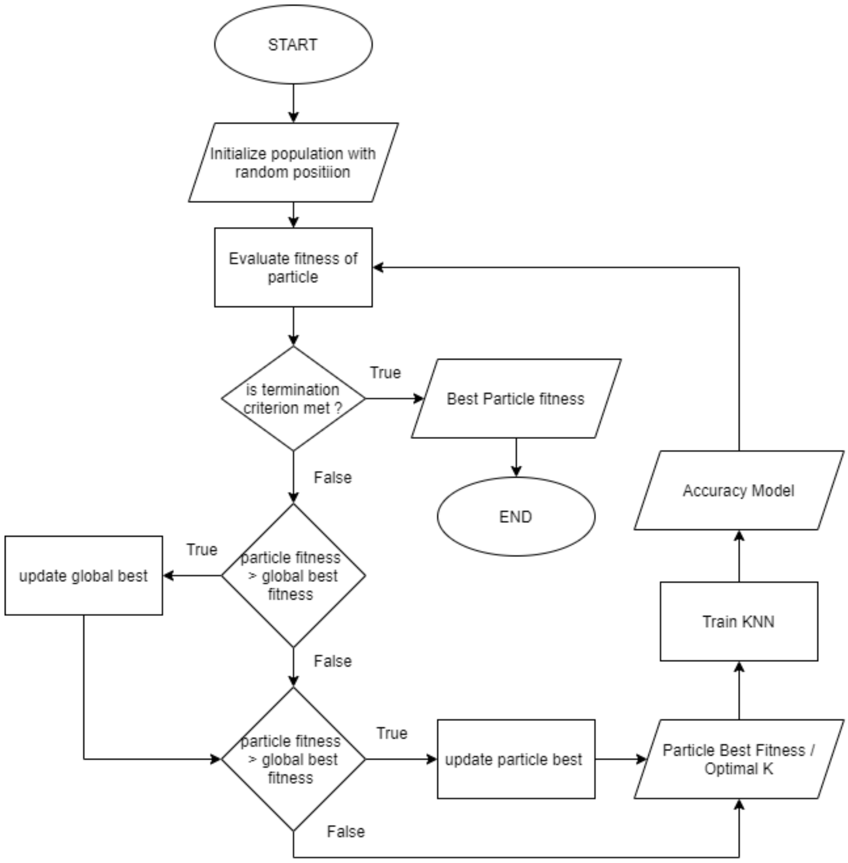

Fig. 8 Flowchart PSO-KNN

The flowchart shown in Fig. 8 indicates that PSO begins its operation by determining a random population, which is then evaluated. After the evaluation, the next step is to check whether the previously evaluated particles meet the termination criteria. If this condition is met, the evaluated particle becomes the best particle and represents the K value in the KNN algorithm. If not, the next step is to proceed with comparisons. Comparisons are carried out in two stages. The first stage involves checking whether the particle fitness has the potential or a better fitness value than the global fitness value. If so, the global fitness value is updated. PSO updates the global fitness by applying the formula shown in (1) [16]–[20].

$$V_i\ (t+1) = w\ .V_i(t) + C_1\ .r\ .\left(X_i^{Pb} - X_i(t)\right) + C_2\ .r\ .\left(X_i^{Gb} - X_i(t)\right) \quad (1)$$

w : Inertia Weight
$V_i(t)$: velocity of each particle
C_1 : best particle weight coefficient constant
C_2 : best global weight coefficient constant
X_i^{Pb} : best particle in a group
X_i^{Gb} : best particle in all groups
r : random decimal from 0 to 1

After updating the global fitness, the next step carried out by PSO is to examine the updated result by comparing it with the particle fitness. If the particle fitness has a better value, PSO will update the best particle by applying the equation shown in (2).

$$X_i\ (t+1) = X_i(t) + V_i\ (\ t+1) \quad (2)$$

$X_i(t)$: position of each particle
$V_i\ (\ t+1)$: updated velocity value

After updating the best particle, the result of this update is temporarily considered as the best particle or the optimized K value. Therefore, the next step is to train the dataset using the KNN algorithm with the optimized hyperparameter K. K-NN is a classification method based on finding the nearest neighbors of unclassified samples and making predictions according to the classes with high similarity values. KNN assigns unclassified samples to a class based on their proximity to data points that have been classified before [21]-[22]. This study utilizes Euclidean and Manhattan to calculate the distance between classes, employing the following (3) and (4) [23]–[28].

$$j(v1, v2) = \sqrt{\sum_{k=1}^{n}(v_1(k) - v_2(k))^2} \tag{3}$$

$v_1(k)$: vector class 1
$v_2(k)$: vector class 2

$$d_{ij} = |X - Y|_1 = \sum_{i=1}^{n}|X_i - Y_1| \tag{4}$$

The next step is to perform evaluation and continue iterations until the best particle is found [29]–[31].

III. RESULT AND DISCUSSION

This study divides the results and discussion section into several parts, including the following:

A. *The Accuracy of the Dataset for K values of 3, 5, and 7*

Before optimizing the K hyperparameter using PSO, here are the dataset optimization results for K values of 3, 5, and 7, presented in Fig. 9.

The use of K values of 3, 5, and 7 in this study was determined randomly, it aims to make a comparison between the KNN classification results before and after the K value is optimized by PSO. Among the three datasets trained with K values of 3, 5, and 7, the iris dataset achieved very stable accuracy, specifically 95.24% for each K value. However, both the Wine and Breast Cancer datasets exhibited a similar pattern. When these two datasets were trained with K = 3, the accuracy decreased, although the decrease was not considered significant. For the Wine dataset, it decreased from 69.30% to 68.80%, and for the Breast Cancer dataset, it decreased from 70.50% to 69.00%. This is attributed to the fact that the Wine and Breast Cancer datasets have more attributes compared to the iris dataset, influencing the accuracy results.

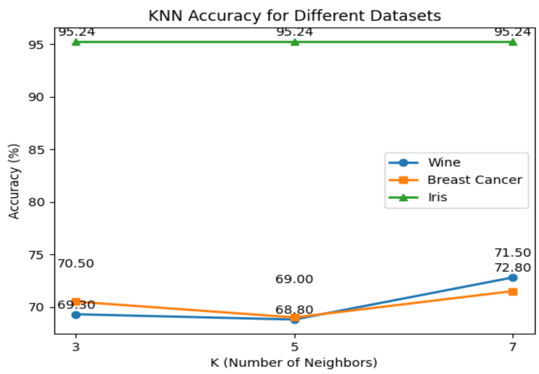

Fig. 9 KNN accuracy score for unoptimized K value

B. *Dataset Accuracy for K Values Optimized by PSO*

This study optimized the K hyperparameter using PSO on three different datasets: Iris, Wine, and Breast Cancer. In this research, the PSO search space was constrained by a lower bound of 3 and an upper bound of 50. This implies that the range of K values to be optimized ranged from 3 to 50. Additionally, 10 particles were used with a maximum search iteration of 100. Below is the discussion for each experiment conducted.

1) Optimization of the K Hyperparameter on the Iris Dataset: This section presents the results of the optimization of the KNN hyperparameter (K parameter) conducted on the Iris dataset. The results can be observed in Fig. 10.

Fig. 10 shows that PSO successfully improved the accuracy of KNN. When using the default K values (i.e., 3, 5, and 7), the achieved accuracy ranged only around 95.24%. In contrast, PSO increased the accuracy to 97.37%, indicating an improvement of 2.13% for both Euclidean and Manhattan distance matrices. Although both experienced a 2.13% increase, in this case, the Manhattan distance matrix achieved this accuracy at K = 3, while the Euclidean distance matrix achieved it at K = 7.

2) Optimization of the K Hyperparameter on the Wine Dataset: In contrast to the optimization results obtained by PSO on the Iris dataset, the optimization results for the K parameter on the Manhattan and Euclidean distance matrices in the wine dataset show the same outcomes. The optimization results for both distance matrices can be observed in Fig. 11.

The optimization results indicate that the most optimal K parameter for the Wine dataset is 9 for each matrix. However, Manhattan distance yields better accuracy compared to the accuracy produced by the Euclidean distance matrix. Manhattan distance achieves an accuracy of 77.78%, while Euclidean distance has an accuracy of 73.33%.

3) Optimization of the K Hyperparameter on the Breast Cancer Dataset: The optimized K hyperparameter in the KNN algorithm, tested on the breast cancer dataset, shows better results compared to the experiment using default KNN values. The optimization results for the K hyperparameter on the breast cancer dataset are presented in Fig. 12.

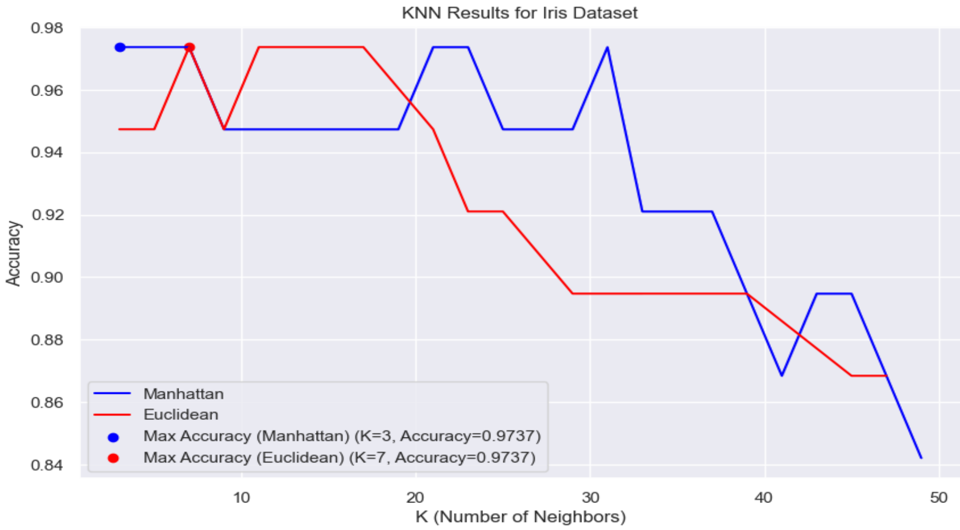

Fig. 10 KNN results for iris dataset

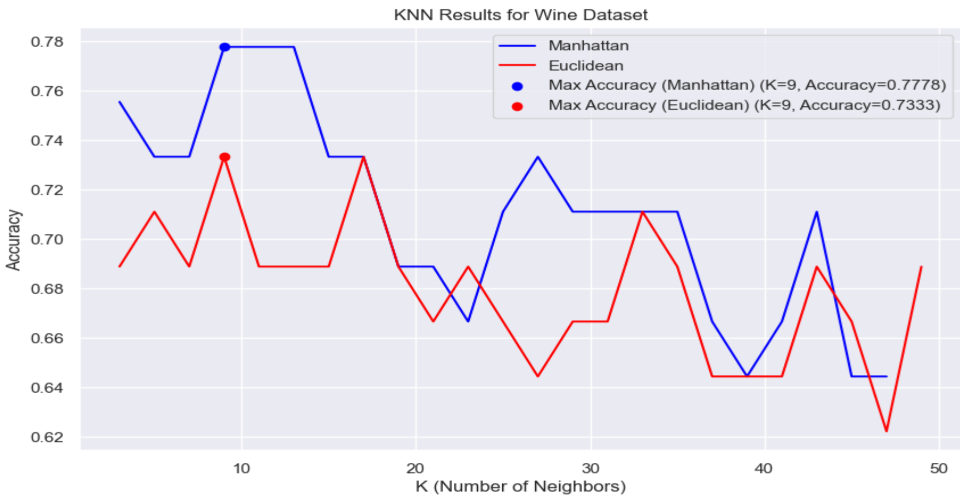

Fig. 11 KNN results for wine dataset

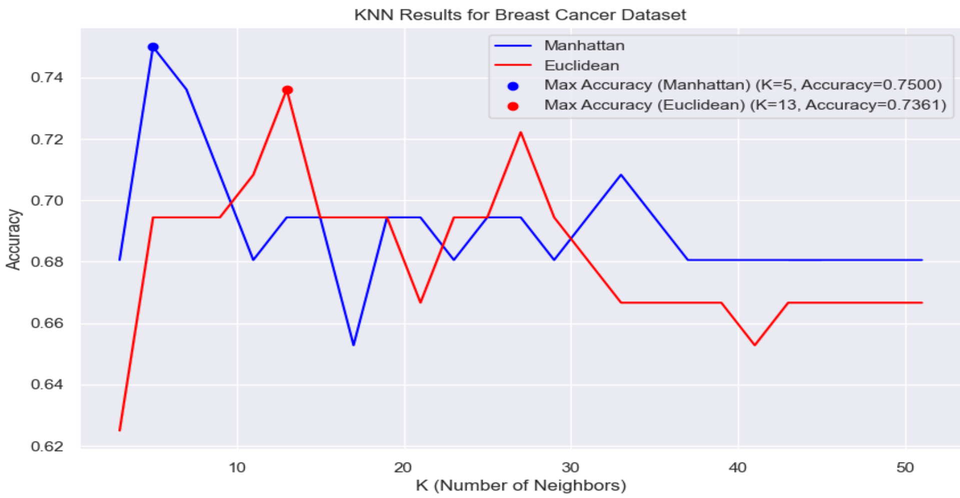

Fig. 12 KNN results for breast cancer dataset

Based on Fig. 12, Manhattan still outperforms Euclidean, as evidenced by the accuracy results. Manhattan achieves an accuracy of 75.00%, while Euclidean achieves 73.61%, indicating a superiority of 1.39% for Manhattan.

IV. CONCLUSION

Implementing Particle Swarm Optimization (PSO) to optimize the hyperparameter K, this study aims to enhance the performance of the K-Nearest Neighbors (KNN) algorithm. Optimization is conducted using Manhattan and Euclidean distance matrices on three different datasets: iris, breast cancer, and wine. Prior to optimization, experiments were conducted with default K values (3, 5, and 7) to observe the behavior of KNN on each dataset. Initial results indicate that the iris dataset has stable accuracy, while the wine and breast cancer datasets show a decrease in accuracy at K=3, attributed to attribute complexity. The optimization process of the hyperparameter K with PSO results in a significant increase in accuracy, especially in the wine dataset, where accuracy improves by 6.28% with the Manhattan matrix. The improved accuracy in the optimized KNN algorithm demonstrates the effectiveness of PSO in overcoming KNN constraints. Although the increase in accuracy for the iris dataset is not as significant, this research provides insight that optimizing the hyperparameter K can yield positive results, even for datasets with initially good performance. A suggestion for future work is to perform similar experiments with different algorithms such as Support Vector Machine or Random Forest to further test the ability of PSO to optimize the iris, wine, and breast cancer datasets.

REFERENCES

[1] T. Denœux, O. Kanjanatarakul, and S. Sriboonchitta, "A new evidential K-nearest neighbor rule based on contextual discounting with partially supervised learning," *International Journal of Approximate Reasoning*, vol. 113, pp. 287–302, 2019, doi: 10.1016/j.ijar.2019.07.009.

[2] J. Jasmir, S. Nurmaini, and B. Tutuko, "Fine-grained algorithm for improving knn computational performance on clinical trials text classification," *Big Data and Cognitive Computing*, vol. 5, no. 4, 2021, doi: 10.3390/bdcc5040060.

[3] D. C. G. Pedronette, Y. Weng, A. Baldassin, and C. Hou, "Semi-supervised and active learning through Manifold Reciprocal kNN Graph for image retrieval," *Neurocomputing*, vol. 340, pp. 19–31, 2019, doi: 10.1016/j.neucom.2019.02.016.

[4] S. Zhang and J. Li, "KNN Classification With One-Step Computation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 2711–2723, 2023, doi: 10.1109/TKDE.2021.3119140.

[5] J. Hu, H. Peng, J. Wang, and W. Yu, "kNN-P: A kNN classifier optimized by P systems," *Theoretical Computer Science*, vol. 817, pp. 55–65, 2020, doi: 10.1016/j.tcs.2020.01.001.

[6] W. Xing and Y. Bei, "Medical Health Big Data Classification Based on KNN Classification Algorithm," *IEEE Access*, vol. 8, pp. 28808–28819, 2020, doi: 10.1109/ACCESS.2019.2955754.

[7] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Scientific Reports*, vol. 12, no. 1, pp. 1–11, 2022, doi: 10.1038/s41598-022-10358-x.

[8] B. Octaviana, "Combination of K-Means Clustering and K-Nearest Neighbor on Ecommerce Customer Spending Rate Prediction," *Infokum*, vol. 9, no. 2, pp. 496–503, 2021.

[9] F. A. I. Achyunda Putra, F. Utaminingrum, and W. F. Mahmudy, "HOG Feature Extraction and KNN Classification for Detecting Vehicle in The Highway," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 3, p. 231, 2020, doi: 10.22146/ijccs.54050.

[10] S. Ambesange, R. Nadagoudar, R. Uppin, V. Patil, S. Patil, and S. Patil, "Liver Diseases Prediction using KNN with Hyper Parameter Tuning Techniques," *Proceedings of B-HTC 2020 - 1st IEEE Bangalore Humanitarian Technology Conference*, pp. 1–6, 2020, doi: 10.1109/B-HTC50970.2020.9297949.

[11] R. Ghawi and J. Pfeffer, "Efficient Hyperparameter Tuning with Grid Search for Text Categorization using kNN Approach with BM25 Similarity," *Open Computer Science*, vol. 9, no. 1, pp. 160–180, 2019, doi: 10.1515/comp-2019-0011.

[12] D. Maincer, Y. Benmahamed, M. Mansour, M. Alharthi, and S. S. M. Ghonein, "Fault Diagnosis in Robot Manipulators Using SVM and KNN," *Intelligent Automation and Soft Computing*, vol. 35, no. 2, pp. 1957–1969, 2023, doi: 10.32604/iasc.2023.029210.

[13] X. N. Bui, P. Jaroonpattanapong, H. Nguyen, Q. H. Tran, and N. Q. Long, "A Novel Hybrid Model for Predicting Blast-Induced Ground Vibration Based on k-Nearest Neighbors and Particle Swarm Optimization," *Scientific Reports*, vol. 9, no. 1, pp. 1–14, 2019, doi: 10.1038/s41598-019-50262-5.

[14] M. A. M. Shaheen, H. M. Hasanien, and A. Alkuhayli, "A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 621–630, 2021, doi: 10.1016/j.asej.2020.07.011.

[15] K. Song, F. Yan, T. Ding, L. Gao, and S. Lu, "A steel

property optimization model based on the XGBoost algorithm and improved PSO," *Computational Materials Science*, vol. 174, no. December 2019, p. 109472, 2020, doi: 10.1016/j.commatsci.2019.109472.

[16] H. Liu, X. W. Zhang, and L. P. Tu, "A modified particle swarm optimization using adaptive strategy," *Expert Systems with Applications*, vol. 152, p. 113353, 2020, doi: 10.1016/j.eswa.2020.113353.

[17] X. W. Zhang, H. Liu, and L. P. Tu, "A modified particle swarm optimization for multimodal multi-objective optimization," *Engineering Applications of Artificial Intelligence*, vol. 95, no. October 2019, p. 103905, 2020, doi: 10.1016/j.engappai.2020.103905.

[18] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. Hone, and X. Liu, "A Dynamic Neighborhood-Based Switching Particle Swarm Optimization Algorithm," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9290–9301, 2022, doi: 10.1109/TCYB.2020.3029748.

[19] W. Liu, Z. Wang, N. Zeng, Y. Yuan, F. E. Alsaadi, and X. Liu, "A novel randomised particle swarm optimizer," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 2, pp. 529–540, 2021, doi: 10.1007/s13042-020-01186-4.

[20] X. fang Song, Y. Zhang, D. wei Gong, and X. yan Sun, "Feature selection using bare-bones particle swarm optimization with mutual information," *Pattern Recognition*, vol. 112, p. 107804, 2021, doi: 10.1016/j.patcog.2020.107804.

[21] A. Shokrzade, M. Ramezani, F. Akhlaghian Tab, and M. Abdulla Mohammad, "A novel extreme learning machine based kNN classification method for dealing with big data," *Expert Systems with Applications*, vol. 183, no. December 2019, p. 115293, 2021, doi: 10.1016/j.eswa.2021.115293.

[22] D. Zhao *et al.*, "k-means clustering and kNN classification based on negative databases," *Applied Soft Computing*, vol. 110, p. 107732, 2021, doi: 10.1016/j.asoc.2021.107732.

[23] N. Hidayati and A. Hermawan, "K-Nearest Neighbor (K-NN) algorithm with Euclidean and Manhattan in classification of student graduation," *Journal of Engineering and Applied Technology*, vol. 2, no. 2, pp. 86–91, 2021, doi: 10.21831/jeatech.v2i2.42777.

[24] A. Alim Murtopo, B. Priyatna, and R. Mayasari, "Signature Verification Using The K-Nearest Neighbor (KNN) Algorithm and Using the Harris Corner Detector Feature Extraction Method," *Buana Information Technology and Computer Sciences (BIT and CS)*, vol. 3, no. 2, pp. 35–40, 2022, doi: 10.36805/bit-cs.v3i2.2763.

[25] X. Gao and G. Li, "A KNN Model Based on Manhattan Distance to Identify the SNARE Proteins," *IEEE Access*, vol. 8, pp. 112922–112931, 2020, doi: 10.1109/ACCESS.2020.3003086.

[26] T. Wira Harjanti, S. Madenda, J. Harlan, and E. T. P. Lussiana, "Study and research on the identification of the leaves of indonesian herbal medicines using manhattan distance and neural network algorithms," *2020 5th International Conference on Informatics and Computing, ICIC 2020*, 2020, doi: 10.1109/ICIC50835.2020.9288564.

[27] Z. Yu, K. Wang, S. Xie, Y. Zhong, and Z. Lv, "Prototypical Network Based onManhattan Distance," *CMES - Computer Modeling in Engineering and Sciences*, vol. 130, no. 3, 2022, doi: 10.32604/cmes.2022.019612.

[28] Y. Zeng, H. Ren, T. Yang, S. Xiao, and N. Xiong, "A Novel Similarity Measure of Single-Valued Neutrosophic Sets Based on Modified Manhattan Distance and Its Applications," *Electronics (Switzerland)*, vol. 11, no. 6, 2022, doi: 10.3390/electronics11060941.

[29] H. R. R. Zaman and F. S. Gharehchopogh, *An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems*, vol. 38, no. 0123456789. Springer London, 2022. doi: 10.1007/s00366-021-01431-6.

[30] M. Jahandideh-Tehrani, O. Bozorg-Haddad, and H. A. Loáiciga, "Application of particle swarm optimization to water management: an introduction and overview," *Environmental Monitoring and Assessment*, vol. 192, no. 5, 2020, doi: 10.1007/s10661-020-8228-z.

[31] W. Li, X. Meng, Y. Huang, and Z. H. Fu, "Multipopulation cooperative particle swarm optimization with a mixed mutation strategy," *Information Sciences*, vol. 529, pp. 179–196, 2020, doi: 10.1016/j.ins.2020.02.034.