✔Media
Statistika

p-ISSN 1979 – 3693 e-ISSN 2477 – 0647
MEDIA STATISTIKA 17(2) 2024: 162-172
http://ejournal.undip.ac.id/index.php/media_statistika

PARAMETER INDEPENDENT FUZZY WEIGHTED k-NEAREST NEIGHBOR

Mayawi, Subanar
Department of Mathematics, Universitas Gadjah Mada, Yogyakarta, Indonesia

e-mail: mayawikarim27@mail.ugm.ac.id

DOI: 10.14710/medstat.17.2.162-172

Article Info:
Received: 20 December 2023
Accepted: 25 December 2024
Available Online: 8 October 2025

Keywords:
Classification; Parameter Independent Fuzzy Weighted k-Nearest Neighbor; k-Nearest Neighbor; Weighted k-Nearest Neighbor; Fuzzy k-Nearest Neighbor; Success History based Parameter Adaptive Differential Evolution

Abstract: Parameter Independent Fuzzy Weighted k-Nearest Neighbor (PIFWkNN) as a classification technique developed by combining Success History based Parameter Adaptive Differential Evolution (SHADE) with Fuzzy k-Nearest Neighbor (FkNN), where this PIFWkNN does not state the optimization of weights and k values as two separate problems, but they're combined into one and solved simultaneously by the SHADE algorithm. The steps for implementing the PIFWkNN method are explained, followed by its application to 10 different datasets, and then the accuracy is calculated. To see the consistency of the goodness of the classification of this method, the accuracy results are compared with the accuracy of the k-Nearest Neighbor (kNN), FkNN, and Weighted k-Nearest Neighbor (WkNN). The results show that the average accuracy of PIFWkNN, kNN, FkNN, and WkNN is 75.76%, 68.52%, 71.40% and 66.22% so PIFWkNN is higher than the three methods. Using the Wilcoxon Sign Rank (WSR) test also concluded that with a 95% confidence shows that every hypothesis had significant differences. Furthermore, it descriptively shows that the average rank of PIFWkNN is higher than the other. Thus, the PIFWkNN has higher accuracy than the kNN, FkNN, and WkNN.

1. INTRODUCTION

Statistical learning theory was first introduced in the late 1960s by Vladimir Vapnik. According to Gareth et al. (2013), statistical learning refers to a set of tools for understanding data, consisting of supervised and unsupervised methods. Several steps in the statistical learning process are: 1) data collection, 2) data preparation, treatment of missing or outlier data, 3) data analysis, and 4) training algorithms on training data and test data. Supervised learning algorithms rely heavily on the input and output of a given data set. The supervised learning approach has input and output, which can then be made into a mathematical relationship model so that it is able to make predictions and classifications based on pre-existing data. The unsupervised learning approach does not use training data or training data to make predictions or classifications. Based on the mathematical model, this algorithm does not have a target variable. One of the goals of this algorithm is to group objects that are almost the same in a certain area. k-Nearest Neighbor (kNN) is a method for making decisions using a supervised learning approach with the results of the newly classified input data based on the closest distance in the value data. kNN itself is one of the nonparametric classification algorithms (methods) where the parametric model is based on an independent distribution or has a defined distribution but with undefined distribution parameters. kNN is very dependent on the value of k in classifying, the distance function used to identify neighbors, and the importance of these neighbors, classes, or features.

So far, most of the research has concentrated on only one or three of these factors to improve the accuracy of kNN. For practical implementation, several techniques such as cross-validation and probability modeling are used to select the best k. In certain cases, the use of kNN becomes ineffective in classifying and predicting, so that then one of them was developed again, namely the Fuzzy k-Nearest Neighbor (FkNN) classification method. Fuzzy k-Nearest Neighbor (FkNN) is a combination of classification algorithms between Fuzzy Logic and k-Nearest Neighbor. Fuzzy k-Nearest Neighbor has two advantages when compared to the k-Nearest Neighbor algorithm, namely, it can consider if there is an ambiguous nature of the neighbor, and the object will have a membership degree value in each class, so that it will give more strength/trust for an object to be in a class (Józwik, 1983).

Based on previous research, it is a fact that FkNN performance can be improved by using the optimal choice of parameters. As research has been conducted by Siringoringo & Perangin-angin (2017), which presents Particle Swarm Optimization (PSO) to determine the best k and m parameter values. From the results obtained, the performance of FkNN with parameter optimization using PSO is better. Likewise, with the research conducted by Suguna et al. (2010) related to improving the classification performance of the k-Nearest Neighbor method using an algorithm, the results show that the application of the GA method to kNN is able to improve the classification performance. In line with this, previous research conducted by AlSukker et al. (2010) related to the optimization of kNN metric weighting using Differential Evolution also gave the result that kNN can provide accurate performance.

Differential Evolution (DE) is a stochastic search method, which is mainly designed for numerical optimization problems. DE has proven to be competitive with other optimizations that are more complex algorithms and have also been applied to many practical problems (Tanabe & Fukunaga, 2013). Success History-based Adaptive Differential Evolution (SHADE) as a method developed by Tanabe & Fukunaga (2013), who proposed SHADE, based on an adaptive DE algorithm using historical memory of parameter set successfully guides gene-value formation of new control parameters. SHADE was shown to outperform the previous DE algorithm, one of which is JADE (Joining Adaptive DE). JADE is a well-known and effective DE variant using the mechanism adaptation of control parameters. It uses a novel mutation strategy called current-top best/1 and external archives to store previously generated individuals. Based on the above background, in this study, the author will discuss the combination of the algorithm from SHADE with Fuzzy k-Nearest Neighbor, hereinafter referred to as Parameter Independent Fuzzy Weighted k-Nearest Neighbor (PIFWkNN), then measure the accuracy of the method and compare it with the k-Nearest method. Neighbor, Fuzzy k-Nearest Neighbor, and one of the other development methods of the kNN method, namely the Weighted k-Nearest Neighbor (WkNN) method. Consideration of choosing the WkNN method as one of the comparison methods in this study, because apart from having never been used as a comparison with the PIFWkNN method, the WkNN method also shows good classification performance (Kozak et al., 2007).

2. LITERATURE REVIEW

Based on research conducted by Angreni et al. (2018), the results obtained by testing the k values, which are different, namely 1, 8, and 15, resulted in a different level of accuracy for each type of damage. Based on trials conducted on 100 images with 50 images of crocodile skin cracks and cracks, each obtained a different accuracy value. For the value of k=1, the accuracy for cracking = 98% and crocodile skin cracking = 84%. For the value of k=8, the accuracy for cracking = 96% and crocodile skin cracking = 8%. For the value of k=15, the accuracy for cracking = 0% and crocodile skin cracking = 98%.

Furthermore, based on research conducted by Ahmad et al. (2018), which discusses the application of Fuzzy and k-Nearest Neighbor methods for predicting student graduation, with the conclusion that prediction accuracy with this algorithm of 77.35% with true positives totaling 1138 data and true negatives totaling 163 data. Class precision for predictions passing on time is 79.86% and for late predictions is 63.42%, has a class recall for true correct at 92.37% and true late at 36.22%. In this study, the k test was carried out 10 times to estimate the accuracy of the estimate.

The research conducted by Sianturi et al. (2018) diagnosed and classified Schizophrenia psychiatric disease using the FkNN algorithm. The classification process consists of three processes, namely the fuzzy initialization process, the k-nearest neighbor algorithm process, and the k-nearest neighbor fuzzy algorithm process. To test the system, the k-value is tested, and the k-fold is tested. Based on the test results on the value of k, the highest accuracy value was obtained at 38.33% at the value of k=5. The results of testing the effect of k-Fold obtained the highest average accuracy value of 34.17% at k-Fold = 10.

Research conducted by AlSukker et al. (2010) is a study that uses the Differential Evolution (DE) optimization technique to improve kNN performance by optimizing the metric weights of features, neighbors, and class. Several data sets were used to evaluate the performance of the proposed DE-based metrics and compare them with several kNN variants from the literature. The experimental results show that in many cases, combining DE in the kNN classification can provide more accurate performance.

Furthermore, the research conducted by Prasetio et al. (2020) proposes a methodology based on a data mining paradigm that integrates heuristic searches inspired by genetic algorithms. In this study, the k-Nearest Neighbor Genetic Algorithm is used for feature selection and parameter optimization, and k-Nearest Neighbor is used as a classification algorithm. The proposed method was tested on 5 medical datasets, with experimental results showing that the proposed method was able to achieve good performance. The results of the comparison with other classification methods show that there is a significant increase in the p-value of the t-test = 0.0011.

One of the research projects related to parameter optimization in Fuzzy k-Nearest Neighbor was conducted by Chen et al. (2011), which is a study that proposes a nonparametric classification method for predicting bankruptcy using an adaptive method, Fuzzy k-Nearest Neighbor with k nearest neighbors and fuzzy strength. Parameter m is adaptively determined by Particle Swarm Optimization (PSO). Time Varying Acceleration Coefficients (TVAC) and Time Varying Inertia Weight (TVIW) are used to efficiently control the local and global search capabilities of the PSO. Furthermore, continuous and binary PSO was applied to a parallel multicore platform, which was later named PTVPSO-FkNN, giving the result that this method is more efficient to be applied to predicting bankruptcy.

Siringoringo & Perangin-angin (2017) conducted a study about the application of MPSO in determining the parameters of FkNN, further giving the result that the model offered in this study provides a better classification performance value than the FkNN model alone. Comparison of the superiority of the research model with other classification models, such as IBK and Decision Tree. This research model has a better level of performance. The research conducted by Ubaiddillah et al. (2018) is a study that combines the Fuzzy k-Nearest Neighbor method with GAs gives the result that the program can produce a fairly optimal accuracy that reaches 98%, with parameters on GAs, namely population 40, generation 15, CR 0.5, and MR 0.8.

Based on previous research, the researcher was then interested in conducting research related to combining one of the Differential Evolution methods, namely Success-History Based Adaptive Differential Evolution (SHADE), with Fuzzy k-Nearest Neighbor (FkNN), hereinafter referred to as the Parameter Independent Fuzzy Weighted k-Nearest Neighbor (PIFWkNN), in classifying. As for the PIFWkNN method, previously Biswas et al. (2018) have conducted research related to PIFWkNN, namely a study that proposed the Parameter Independent Fuzzy Weighted k-Nearest Neighbor (PIFWkNN) classifier. PIFWkNN formulates the problem of selecting an appropriate k value and a set of class-dependent optimal weights for features as the sole objective of the continuous non-convex optimization problem. In this study, a Differential Evolution (DE) variant is also used, which is called Success-History based Adaptive Differential Evolution (SHADE). From these experiments, it can be concluded that the proposed PIFWkNN can not only work independently (without the need for additional parameter tuning) but can also show significant advantages over several other popular classification parameters. As for this research, it will be explained in relation to PIFWkNN, which will then be compared with the kNN, WkNN, and FkNN classification methods.

3. MATERIAL AND METHOD

3.1. Data Sources

The dataset used in this study came from the UCI (University of California, Irvine) Machine Learning Repository. These data have varying sizes and numbers of variables. The data used in this case are presented in Table 1.

Table 1. Datasets Description

No	Dataset	Number of Points	Number of Attributes	Number of Classes
1	Iris	150	5	3
2	Breast Cancer Wisconsin	699	10	2
3	E. coli	100	8	8
4	Glass	214	10	7
5	Hayes	132	5	3
6	Haberman	306	4	2
7	Diabetes	100	9	2
8	Wine	178	13	3
9	Lung Cancer	32	57	3
10	Vehicle	100	18	3

3.2. Notation

Suppose $P = x_1, x_2, \dots, x_n$ as training data and $Y = y_1, y_2, \dots, y_m$ as testing data, $P, Y \subseteq X$ where X is a real set of dimensions d. Furthermore, a vector with dimension d is defined, namely a $\in \mathbb{R}^d$ as $\left[a^{(1)}, a^{(2)}, a^{(3)}, \dots, a^{(d)}\right]$. Each $x \in X$ is a class consisting of c classes which are expressed by the set $L = l_1, l_2, \dots, l_c$.. Then, for the training data $x_i \in P$, it is known that the class $c_L(x_i)$ and testing data $y_j \in Y$ with $\hat{c}_L(y_j)$ is the predicted class with $c_L(x_i), \hat{c}_L(y_j) \in L$. Suppose the k-neighborhood of a point a, a set S is defined as $\Omega_k(S, a)$ and the number of members of a set S is denoted by $|S|$. Next, suppose that the distance between two points a and b is denoted as (a, b). A classifier $h(.)$ can be further defined as $h_\gamma\,(P, y_j)$ where is a parameter set, so $\hat{c}_L(y_j) = h_\gamma(P, y_j)$.

3.3. kNN, FkNN, WkNN, and Weighted Euclidean Distance

Given a dataset P and testing data $y_j \in Y$. The kNN classification is symbolized by $\hat{c}_L(y_j)$ a class label is defined by (Biswas et al., 2018),

$$\hat{c}_L(y_j) = l_r \tag{1}$$

$$r' = \arg\max |\{x_i : x_i \in \Omega_k(P, y_j) \wedge c_L(x_i) = l_r\}| \tag{2}$$

Thus, the classification of k-NN is determined as a $y_j \in Y$ with a class label in which most of the training data set k neighbors of y_j are located.

Next is to determine the value of U using the formula proposed by Keller et al. (1985) as follows,

$$u_{ir} = \begin{cases} 0.49\left(\frac{n_r}{v}\right) & \text{if } C_L(x_i) \neq l_r \\ 0.51 + 0.49\left(\frac{n_r}{v}\right) & \text{others} \end{cases} \tag{3}$$

where n_r is the number of training data labeled $l_r, v \geq 1, v$ integer members. Next, for a testing data point $y_j \in Y$, FkNN returns a set $M\,(y_j\,) = \{\mu_1(y_j\,), \mu_2(y_j\,), \dots, \mu_c(y_j)\}$ where $\mu_r(y_j)$ is the probability that y_j is in class r_{th}, with,

$$\mu_r(y_j) = \frac{\sum_{x_i \in \Omega_k(P, y_j)} u_{ir}\, \delta(y_j, x_i)^{\frac{2}{1-m}}}{\sum_{x_i \in \Omega_k(P, y_j)} \delta\,(y_j, x_i)^{\frac{2}{1-m}}} \tag{4}$$

where $m > 1$. After calculating $M\,(y_j\,)$, the next class label prediction will be determined with,

$$\hat{c}_L(y_j) = l_{r'}, \quad \text{where } r' = \underset{r=1,2,..,c}{\arg\max}\, \mu_r(y_j) \tag{5}$$

The distance used is the weighted Euclidean distance defined by,

$$\delta_W(x_i, y_j) = \left(\sum_{f=1}^{d} w_{C_L(x_i)} \left(x_i^{(f)} - y_j^{(f)}\right)^2\right)^{\frac{1}{2}} \tag{6}$$

where $W = [w_{rf}]_{c \times d}$ is the weight set of size $c \times d$. Classification using FkNN does not eliminate the basic nature of kNN in overcoming the shortcomings of the kNN method as described previously. However, the FkNN method will provide probabilities for each training data point for each class.

The steps in applying the k-Nearest Neighbor classification method, in general, are calculating the Euclidean distance, sorting by the value of the Euclidean distance, determining the closest k classification record, and the output target being the majority class.

The Fuzzy k-Nearest Neighbor (kNN) algorithm is one of the algorithms used for decision-making. Fuzzy K-Nearest Neighbor is a classification technique that combines the Fuzzy technique with K-Nearest Neighbor. The resulting value will be converted into several binary values, and then the weighting will be searched using the fuzzy technique. So that after finding the best neighbor point, a weight calculation will be carried out to find the final answer. As for this study, the kNN algorithm used was proposed by Keller (1985).

According to Hechenbichler & Schliep (2004), the classification steps using WkNN broadly consist of five steps. The first step is to determine data sampling, then look for the value of k+1, followed by distance normalization, and then weighting at a distance that has been normalized and looking for the largest weight to determine the classification class. As for this study, the kNN algorithm used was proposed by Hechenbichler & Schliep (2004).

3.4 Optimization, DE, and SHADE

Optimization is a process to achieve the best or optimal result. Suppose a function $g : S \rightarrow \mathbb{R}$, where S is the set defined as,

$$S = \{s: s \in \mathbb{R}^D, b_1^{(1)} \leq s^{(1)} \leq b_2^{(1)}, \dots, b_1^{(D)} \leq s^{(D)} \leq b_2^{(D)}\} \tag{7}$$

with $b_1 = \left[b_1^{(1)}, b_1^{(2)}, \dots, b_1^{(D)}\right], b_2 = \left[b_2^{(1)}, b_2^{(2)}, \dots, b_2^{(D)}\right]$, and $b_1 , b_2 \in \mathbb{R}^D$. An optimization problem, the solution of ŝ can be formulated in the form of a minimization problem (without reducing its generality) as,

$$\hat{s} = \arg\min_{s \in S} g(s) \tag{8}$$

In this case, $g(.)$ is defined as an objective function that represents the optimization problem and needs to be minimized.

In the PIFWkNN method, DE is used to find $\hat{s}$ starting with a random population $\Theta \subseteq S$ consisting of Np candidate solutions, and iterations are carried out until completion, until the condition limit is reached. Evolution is carried out by applying three operations sequentially, namely mutation (using the vector difference of the two candidate solutions), crossover (exponential or binomial), and population selection. There are many variations of mutation and crossover operations, which are designed to solve different optimization problems. Using DE, though, frees the user from setting k values and finding optimal feature weights; it still requires adjustment of F and Cr to effectively improve the accuracy of FkNN. This problem can be solved by using an improved DE variant called SHADE.

SHADE, which uses a self-adjustment technique to intelligently calculate optimal values of F and Cr. SHADE maintains historical memory with H entries for both DE control parameters, i.e., Cr and F. First, $M_{Cr,i}$ and $M_{F,i}$ $(i = 1, \dots, H)$ which is the average value S_{Cr} and S_F are stored generation-wise, respectively, initialized to 0.5 (Tanabe & Fukunaga, 2013).

In each generation, the control parameters CR_i and $F_i$are used for each individual x_i resulting from the first selection process of an index r_i randomly from $[1, H\ ]$ and will be determined with (Tanabe & Fukunaga, 2013),

$$\hat{s} = \arg\min_{s \in S} g(s) \tag{8}$$

$$CR_i = randn_i(M_{CR,r_i}, 0.1) \tag{9}$$

$$F_i = randn_i(M_{F,r_i}, 0.1) \tag{10}$$

If the resulting values for CR_i and F_i if it is outside [0, 1], it will be regenerated again. The CR_i and F_i values used by successful individuals are recorded in S_{CR} and S_F, and at the end of the generation, the memory contents are updated with (Tanabe & Fukunaga, 2013),

$$M_{CR,k,G+1} = \begin{cases} mean_{W\,A}(S_{CR}) & \text{if } S_{CR} \neq \emptyset \\ M_{CR,k,G} & \text{others} \end{cases} \tag{11}$$

$$M_{F,k,G+1} = \begin{cases} mean_{W\,L}(S_F) & \text{if } S_F \neq \emptyset \\ M_{F,k,G} & \text{others} \end{cases} \tag{12}$$

with index $k(1 \leq k \leq H)$ specifying the memory position to update. If $k > H,$ then the value of $k = 1$. If each generation G fails to produce a better experimental vector than the previous one (the parent), or in other words $S_{CR} = S_F = \emptyset$ then the memory is not updated. The weighted mean $mean_{W\,A}S_{CR}$ is calculated by (Tanabe & Fukunaga, 2013),

$$mean_{W\,A}(S_{CR}) = \sum_{k=1}^{|S_{CR}|} w_k.S_{CR,k} \tag{13}$$

with

$$w_k = \frac{\Delta f_k}{\sum_{k=1}^{|S_{CR}|} \Delta f_k} \tag{14}$$

$$\Delta f_k = \left|f\left(u_{k,G}\right) - f\left(x_{k,G}\right)\right| \tag{15}$$

Furthermore, the average Lehmer weight (weighted Lehmer mean) is calculated using the following formula,

$$mean_{W\,L}(S_F) = \frac{\sum_{k=1}^{|S_{CR}|} w_k\,.S_{F,k}^2}{\sum_{k=1}^{|S_{CR}|} w_k\,.S_{F,k}} \tag{16}$$

In SHADE, each individual x_i corresponds to p_i, whose set is based on the equation resulting from the following formula (Tanabe & Fukunaga, 2013),

$$p_i = rand[p_{min}, 0.2] \tag{17}$$

where p_min is the set arranged so that when the best p individuals are selected, at least 2 individuals are selected, $p_{min} = 2/N$ with a maximum value of 0.2.

3.5 Parameter Independent Fuzzy Weighted k-Nearest Neighbor (PIFWkNN)

In this method, it is necessary to design an objective function that minimizes it will optimize the performance of PIFWkNN. The choice of this function can be a misclassification that does not exist in the testing data. Such errors can be calculated by classifying each. $x_i \in P$ by PIFWkNN, whereas $P \backslash \{x_i\}$ is used as the training set. Since our goal is to optimize the parameters for PIFWkNN, the objective function must have a domain associated with the classifier parameter space. Let's express the domain of the objective function as $Z \in \mathbb{R}^D$. Thus, Biswas et al. (2017) formally define the objective function $e : Z \rightarrow \mathbb{R}^D$as shown below:

$$e(z) = \left(1 - \frac{\sum_{x_i \in P} I(h_z(P \backslash \{x_i\}, x_i), C_L(x_i))}{|P|}\right) \tag{18}$$

where $h_z(.)$ is the FkNN classifier using δ_W distance, and $I(a, b)$ is an indicator function that returns 1 if a and b are equal, otherwise the distance is 0. In addition, $z \in Z$ is a candidate solution of $e(.)$from the optimal choice of parameters k and W that can be found. Next, the last thing to do is to code z in such a way that not only W and k can be easily computed from it, but can also be correctly expanded by SHADE.

According to Biswas et al. (2017), the proposed PIFWkNN classifier starts by initializing a random population (sampled from a uniform D-dimensional distribution) Z

contains N_p number of candidate solutions (with $0 \le z^{(f)} \le 1$, and $f = 1,2,\ldots,D$), which are fixed respectively, and after extracting the corresponding W and k, which are evaluated by calculating the value of the function $e(.)$. For repair and extraction of W can be done by normalizing the set of weights for each class ($[z^{(rd-d+1)}, z^{(rd-d+2)}, \ldots, z^{(rd)}]$ for r^{th} class) in the range 0 to 1. As for k, the first $z^{(D)}$ Is bounded between 0 and 1, which is a very small positive real number (for $z^{(D)} \le 0$ the set $z^{(D)} = \varepsilon$ and for $z^{(D)} \le 1$ the set $z^{(D)} = 1$, otherwise the value of $z^{(D)}$ can be preserved). The modification $z^{(D)}$ is multiplied by the maximum permissible value of k and rounded to the next integer as in the following equation.

$$k = [z^{(D)} \times \sqrt{n}] \tag{19}$$

```
PIFWkNN Algorithm
Input: Data training $P$, data testing $y_j \in Y$.
Output: Label of classes $y_j, \hat{c}_L(y_j)$.
begin
    Random population initialization $Z = [z_{tj}]_{N_p \times D}$ with real value
     between 0 and 1.
    Expand Z by SHADE to minimize the objective function e(.) in
    Equation (18). Application checks required for each
    experimental solution and extract W and k in Equation (10) to be
    evaluated using the function e(.).
    $\hat{z} = \underset{z \in Z}{\operatorname{argmin}}\, e(z)$.
    Construct optimal $W$ and $k$ using $\hat{z}$.
    Calculate $U$ using optimal $k$, optimal $W$, by using the distance in
    Equation (6) and $P$.
    Find $M(y_j)$ using Equation (4).
    Find $\hat{c}_L(y_j)$ using Equation (5).
end
```

Figure 1. PIFWkNN Algorithm (Biswas et al., 2017)

4. RESULTS AND DISCUSSION

The application of PIFWkNN, kNN, WkNN, and FkNN methods in analyzing this case study used the Python programming language provided on Google Colab Research, which can be accessed through https://colab.research.google.com/. Furthermore, the three algorithms, such as kNN, WkNN, and FkNN, used as a comparison with the PIFWkNN method will be explained as follows.

The accuracy of Parameter Independent Fuzzy Weighted k-Nearest Neighbor (PIFWkNN), k-Nearest Neighbor (kNN), Fuzzy k-Nearest Neighbor (FkNN) classification method, and Weighted k-Nearest Neighbor (WkNN) on 10 datasets is provided in Table 2. From Table 2, it can be seen that the PIFWkNN method applied to 10 datasets to be tested, there are 7 datasets that have a higher accuracy value than the kNN, FkNN, and WkNN methods. For a comparison of the accuracy of each method, it can be see in Figure 2. According to Figure 2, it can be seen that the accuracy of the PIFWkNN method for 10 datasets has a higher rank of 1 than the other methods. Furthermore, Table 3 shows the results of the average accuracy and average ranking based on the accuracy results for the 10 datasets of each method that have been previously presented in Table 3.

Table 2. Method Accuracy

No	Dataset	kNN (%)	FkNN (%)	WkNN (%)	PIFWkNN (%)
1	Iris	96.00	90.00	93.33	94.67
2	Wisconsin	95.00	95.00	73.47	96.00
3	Ecoli	75.00	80.00	80.00	84.00
4	Glass	68.00	70.00	76.64	72.00
5	Hayes	48.15	57.00	28.79	71.10
6	Haberman	60.00	70.00	66.00	74.00
7	Diabetes	70.00	65.00	60.00	69.00
8	Wine	89.00	90.00	91.00	94.97
9	Lung Cancer	30.00	45.00	40.00	49.99
10	Vehicle	56.00	50.00	52.00	53.00

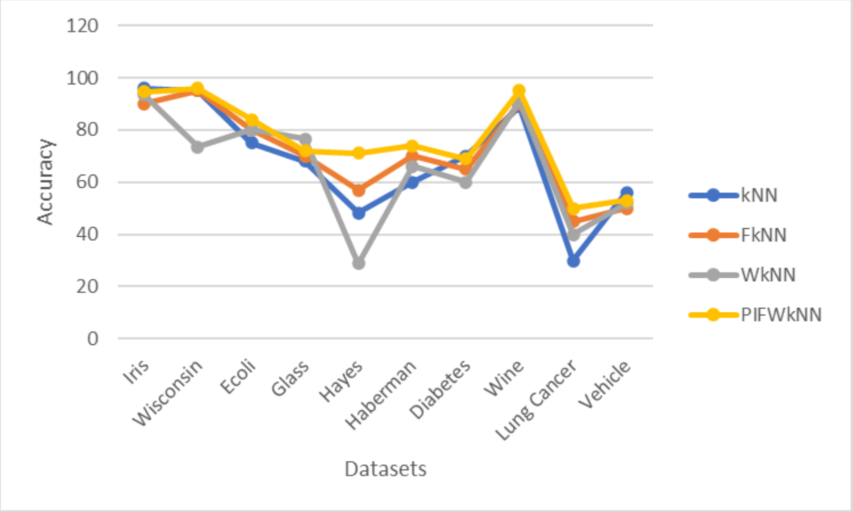

Figure 2. Method Accuracy Comparison

Table 3. Average Accuracy and Rank

Methods	Average of Accuracy (%)	Average of Rank
PIFWkNN	75,76	1,50
kNN	68,52	2,80
FkNN	71,40	2,60
WkNN	66,22	2,70

Based on the results shown in Table 3, it can be seen that the average accuracy of the PIFWkNN method that has been applied to 10 datasets is higher than the other 3 comparison methods. Table 3 also shows that the rank of the PIFWkNN method is higher than the other three comparison methods. Furthermore, to ensure that PIFWkNN is a classification method that is consistently better than the three comparison methods, an analysis will be carried out using the Wilcoxon Sign Rank (WSR) test by comparing the accuracy results of the PIFWkNN method with the 10 datasets used against the PIFWkNN method. kNN, FkNN, and WkNN. This WSR test aims to see if there is a difference in the average accuracy with each of the methods being compared in this study. From the results of the analysis using WSR, the following p-values are presented for each method in Table 4.

Table 4. p-Value WSR Analysis Results

Methods	kNN	FkNN	WkNN
PIFWkNN	0.008004	0.01032	0.02734

From the table above, it can be seen that the p-value of the analysis using the WSR test for PIFWkNN and kNN is smaller than = 0.05. Likewise, the p-value of the WSR test analysis between PIFWkNN and FkNN and the p-value of the test analysis of the WSR for PIFWkNN and WkNN, each of which is smaller than = 0.05. Thus, it can be concluded that for PIFWkNN with kNN, PIFWkNN with FkNN, and PIFWkNN with WkNN, there is a significant difference in average accuracy.

5. CONCLUSION

Based on the results of the analysis that has been carried out, several conclusions can be drawn. The average level of accuracy in the 10 datasets selected based on the application of each method of k-Nearest Neighbor, Weighted k-Nearest Neighbor, and Fuzzy k-Nearest Neighbor, respectively, was 68.52%, 66.22% and 71.40%. The average level of accuracy in 10 datasets selected based on the application of the Parameter Independent Fuzzy k-Nearest Neighbor method sequentially is 75% which is higher than the three methods used as comparison, namely kNN, WkNN, and FkNN. This is also reinforced by the higher average ranking value of PIFWkNN than the three methods and the results of the analysis using WSR, where there is a significant difference between the average value of PIFWkNN accuracy and each of the three comparison methods.

REFERENCES

Ahmad, I. & Saputra, H. S. D. H. (2018). Using Fuzzy K-Nearest Neighbor for Predicting University Students Graduation in Teknokrat. *Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)*, *1*(1), 47-52.

AlSukker, A., Khushaba, R., & Al-Ani, A. (2010, March). Optimizing the k-NN Metric Weights Using Differential Evolution. In *2010 International Conference on Multimedia Computing and Information Technology (MCIT)* (pp. 89-92). IEEE.

Angreni, I. A., Adisasmita, S. A., Ramli, M. I., & Hamid, S. (2018). Pengaruh Nilai K Pada Metode K-Nearest Neighbor (KNN) Terhadap Tingkat Akurasi Identifikasi Kerusakan Jalan. *Rekayasa Sipil Mercu Buana*, *7*(2), 63-70.

Biswas, N., Chakraborty, S., Mullick, S. S., & Das, S. (2018). A Parameter Independent Fuzzy Weighted K-Nearest Neighbor Classifier. *Pattern Recognition Letters*, *101*, 80-87.

Chen, H. L., Liu, D. Y., Yang, B., Liu, J., Wang, G., & Wang, S. J. (2011). An Adaptive Fuzzy K-Nearest Neighbor Method Based on Parallel Particle Swarm Optimization for Bankruptcy Prediction. 2010 *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 249-264. Springer, Berlin, Heidelberg.

Gareth, J., Daniela, W., Trevor, H., & Robert, T. (2013). *An Introduction to Statistical Learning: With Applications in R*. New York: Springer.

Hechenbichler, K. & Schliep, K. (2004). Weighted k-Nearest-Neighbor Techniques and Ordinal Classification. *Sonderforschungsbereich* 386, Paper 399.

Jóźwik, A. (1983). A Learning Scheme for a Fuzzy k-NN Rule. *Pattern Recognition Letters*, *1*(5-6), 287-289.

Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A Fuzzy k-Nearest Neighbor Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, *4*, 580-585.

Kozak, K., Kozak, M., & Stapor, K. (2007). Weighted k-Nearest-Neighbor Techniques for High Throughput Screening Data. *International Journal of Chemical and Molecular Engineering*, *1*(12), 161-166.

Prasetio, R. T. (2020). Seleksi Fitur dan Optimasi Parameter k-NN Berbasis Algoritma Genetika pada Dataset Medis. *Jurnal Responsif: Riset Sains & Informatika*, *2*(2), 213-221.

Sianturi, T. O. (2018). *Penerapan Algoritme Fuzzy K-Nearest Neighbor (FK-NN) pada Pengklasifikasian Penyakit Kejiwaan Skizofrenia* (Doctoral dissertation, Universitas Brawijaya).

Siringoringo, R., & Perangin-angin, R. (2017). Hibridisasi Metode Fuzzy K-Nearest Neighbor dengan Metode Modified Particle Swarm Optimization pada Pengklasifikasian Penyakit Tanaman Kedelai. *Sinkron: Jurnal dan Penelitian Teknik Informatika*, *2*(2), 6-12.

Suguna, N. & Thanushkodi, K. (2010). An Improved k-Nearest Neighbor Classification Using Genetic Algorithm. *International Journal of Computer Science Issues*, *7*(2), 18-21.

Tanabe, R. & Fukunaga, A. (2013, June). Success-History Based Parameter Adaptation for Differential Evolution. *2013 IEEE Congress on Evolutionary Computation*, 71-78.

Ubaiddillah, M., Ratnawati, D. E., & Dewi, C. (2017). Deteksi Dini Penyakit Gagal Ginjal menggunakan Gabungan Genetic Algorithm dan Fuzzy K-Nearest Neighbor (GAFKNN). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *2*(2), 577-583