Voting Scheme.pdf

By Angga Pradipta

WORD COUNT 4922

TIME SUBMITTED 20-NOV-2023 09:07AM

PAPER ID 104372279

Voting Scheme Nearest Neighbors by Difference Distance Metrics Measurement

Gede Angga Pradipta[1*], Made Liandana[2], Putu Desiana Wulaning Ayu[3], Dandy Pramana Hostiadi[4] , I Putu Sumardika Eka Putra[5]

[1,3,4] *Department of Magister Information Systems, Institut Teknologi dan Bisnis STIKOM Bali, Indonesia*
[2,5] *Departement of Informatics and Computer, Institut Teknologi dan Bisnis STIKOM Bali, Indonesia*
*corr_author: angga_pradipta@stikom-bali.ac.id

Abstract - K-Nearest Neighbor (KNN) is a widely used method for both classification and regression cases. This algorithm, known for its simplicity and effectiveness, relies primarily on the Euclidean formula for distance metrics. Therefore, this study aimed to develop a voting model where observations were made using different distance calculation formulas. The nearest neighbors algorithm was divided based on differences in distance measurements, with each resulting model contributing a vote to determine the final class. Consequently, three methods were proposed, namely k-nearest neighbors (KNN), Local Mean-based KNN, and Distance-Weighted neighbor (DWKNN), with an inclusion of a voting scheme. The robustness of these models was tested using umbilical cord data characterized by imbalance and small dataset size. The results showed that the proposed voting model for nearest neighbors consistently improved performance by an average of 1-2% across accuracy, precision, recall, and F1 score when compared to the conventional non-voting method.

Keywords: KNN, Euclidean, Manhattan, Minowski, Voting

I. INTRODUCTION

K-Nearest Neighbor (KNN) is a distance-based classification method whose operation includes identifying the nearest neighbors of the test data within the range of the training data. This proximity can be measured using a distance function, with Euclidean distance serving as the most prevalent choice. KNN has several advantages, including its simplicity, ease of explanation, and adaptability to irregular feature spaces. Over time, it has been subject to various modifications for performance improvement. However, the method has garnered attention in previous literature due to four problematic issues. Firstly, it is sensitive to the neighborhood size parameter k [1]–[3]. The performance can deteriorate when outliers are present, whether k is set to a smaller or larger value. A selection of small k parameter often results in suboptimal classification outcomes, particularly in discrete and noisy datasets. Conversely, the selection of a large k parameter can lead to compromised classification outcomes, due to the influence of outliers. Secondly, KNN is sensitive to the distance function used for selecting k nearest neighbors [4]–[6]. Thirdly, the method can be highly complex due to the search of nearest neighbor (NN) [7]–[9]. This aspect provides a significant challenge as KNN is required to calculate the distances of all samples in order to identify the k nearest neighbors for each given query (test data).

The development of the nearest neighbors method has been widely carried out, with a focus on addressing the three identified weaknesses. Several studies aimed at improving the method have been previously proposed, particularly in addressing the sensitivity issue. One approach used incorporates a local mean factor to reduce the sensitivity effect of the k value. Various methods, such as k-harmonic nearest neighbors (KHNN)[10], local mean-based KNN (LMKNN)[11], local mean-based pseudo-NN (LPMNN)[12], and multi-local means-based NN (MLNN)[13], have been developed to reduce the impact of outliers around the sample points. Some other methods, such as pseudo nearest neighbors (PNN) [14], weighted representation-based KNN (WRKNN), and weighted local mean representation-based KNN (WLMRKNN) [15], introduced weights for each neighborhood data point. These weighting methods are based on the premise that each nearest neighbor contributes differently to the classification outcome. The development of the dual distance-weight technique has led to the introduction of the distance-weighted k-nearest neighbor rule (DWKNN) [16]. The new method reduces the weight of each nearest neighbor, except the first closest and the k-th. Several alternative neighborhood methods have been successively applied to classification problems in order to address practical issues in KNN. For instance, the surrounding neighborhood-nearest centroid neighbor (NCN) was derived for finite sample-size situations, with extensions like KNCN [17] and

LMKNCN [18] exhibiting satisfactory performance. The second key concern is the selection of a suitable distance metric for evaluating the distance between query and training samples, a crucial factor in classification decisions. To enhance the classification performance of KNN, several local and global feature weighting-based distance metrics methods have been developed [19], [20]. However, these approaches often overlook the correlations between all training samples, signifying the importance of accurately defining a distance metric for KNN classification. Based on the fuzzy sets theory, fuzzy nearest neighbor classifiers that introduce fuzziness into KNN were proposed in [21]–[23]. Some evidence-theoretic KNN classifiers have also been explored [24], [25] from the perspective of Dempster–Shafer theory. Derrac et al. recently conducted a comprehensive review of the most relevant algorithms for fuzzy nearest neighbor classification [26]. In contrast to using all training samples in some of the algorithms, several prototype-based classifiers have emerged. These approaches, including the selection [27]–[29], generation [30], [31] and optimization [32]–[34] of prototype, leverage a few well-represented prototypes to achieve optimal classification performance, improving speed, storage, and accuracy. Based on the extensive literature on KNN development, the current study aimed to analyze the impact of different distance metric formulations, such as Euclidean, Minowski, and Manhattan distances. The proposed model adopted a voting scheme to these three distinct distance metrics, and was subsequently applied to the original KNN method, LMKNN, and DWKNN. The effect of the model was evaluated using umbilical cord data. The main contribution of this research is applying voting techniques to the three methods above using different distance metric methods. Voting is carried out with the same weight between one metric and another.

II. METHOD

A. Voting Scheme for Nearest Neighbors

The proposed method in this study utilized voting calculations to determine the final class for the test data. The final prediction result of this model was determined by the highest number of votes received. A total of 12 models were analyzed for their performance, based on the differences in distance metric measurements, specifically Euclidean, Manhattan, and Minowski. The classifier algorithm used was the KNN method, along with its state-of-the-art developments, namely LMKNN and DWKNN. Fig. 1 shows the proposed model scheme based on the voting method.

B. Local Mean Based k-Nearest Neighbors (LMKNN)

Local Mean K-Nearest Neighbor (LMKNN) Classification extended the K-Nearest Neighbor algorithm and was specifically designed to address the sensitivity of KNN to outliers, specifically when training sample sizes are small. The basic concept behind the development of LMKNN is presented as follows:

1. Identification of the KNN for each class corresponding to the query sample.

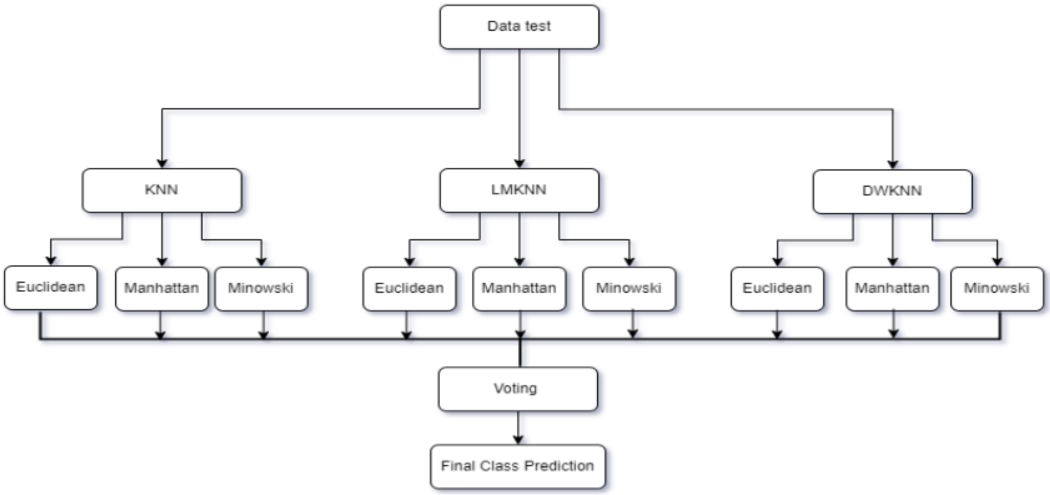

Fig. 1 Proposed voting scheme for Nearest Neighbors

2. Calculation of the local mean in the KNN for each class.
3. Calculation of the distances between the query sample and the local mean of each class.
4. Classification of the query sample into minimum local mean distance.

LMKNN was designed to mitigate the challenges of outliers and enhance KNN performance. By calculating the local mean for each class, LMKNN aimed to effectively capture the underlying data structure, particularly in cases of small training samples or in the presence of outliers.
In summary, it was a classification method that leveraged the local mean of nearest neighbors to classify query samples and effectively address outlier issues in KNN as in (1).

$$m_{wj}^{k} = \frac{i}{k}\sum_{i=1}^{k} y_{i.j}^{NN} \tag{1}$$

Where TS $=\{pi \in \mathbb{R}^d\}_{i=1}^{N}$ is the training samples from a d-dimensional feature space, with N as the total number of samples, and $ci \in \{c1, c2, \ldots, cm\}$ indicates the class label for pi, with a total of M classes. TR$=\{pij \in \mathbb{R}^d\}_{i=1}^{Ni}$ represents the subset of TS corresponding to class ci, with Ni training samples. LMKNN follows these steps to classify the query sample $x \in Rd$ into a class c:

Step 1. Identify the KNN TR set for each class ci for the query pattern x.

Where $TR_k^{NN}(x) = \{p_{ij}^{NN} \in \mathbb{R}^d\}_{j=1}^{k}$ is the set of KNN for x in class ci using the Euclidean distance metric (x, p_{ij}^{NN}) , where $k \leq Ni$ as in (2).

$$d\left(x, p_{ij}^{NN}\right) = \sqrt{(x, p_{ij}^{NN})^T (x, p_{ij}^{NN})} \tag{2}$$

Step 2. Calculate the local mean vector lm_{ik}^{NN} for class ci, using set $TR_k^{NN}(x)$ as in (3).

$$lm_{ik}^{NN} = \frac{1}{k}\sum_{j=1}^{K} p_{ij}^{NN} \tag{3}$$

Step 3. Assign x to class c when the mean distance between the local mean vector for c and the query sample falls within the minimum Euclidean space using (4).

$$c = argmind_{ci}(x, lm_{ik}^{NN}) \tag{4}$$

LMKNN is equivalent to the 1-NN classifier when $k = 1$. The significance of K differed between KNN and LMKNN. KNN selected the K nearest neighbors from the entire training sample set, while LMKNN utilized the local mean vectors of K nearest neighbors within each class. LMKNN aimed to identify the class with the locally closest region to the query sample, effectively mitigating the negative impact of outliers, particularly in small sample sizes.

C. Distance-Weighted k-Nearest Neighbors (DWKNN)

DWKNN, an extension of KNN, was designed in order to reduce the sensitivity to the neighborhood size parameter k and achieve good pattern classification performance. Let $\bar{T} = \{(x_i^{NN}, y_i^{NN})\}_{i=1}^{k}$ set of k -nearest neighbors to the query $\bar{x}$ sorted in ascending order of their distances $d\left(\bar{x}, x_i^{NN}\right)$ between $\bar{x}$ and $\bar{x}_i^{NN}$, and $\bar{W} = \{\bar{w}_1, \ldots\ldots, \bar{w}_k\}$ being the corresponding set of dual weights. DWKNN built upon WKNN by assigning different weights to the k-nearest neighbors based on their distances, with a higher weight given to the nearest neighbors. It also assigned dual weights to the i-th nearest neighbor x_i^{NN} of the query $\bar{x}$ a dual weight $\bar{w}_i$, which were determined by a dual distance-weighted function as in (5). The classification of the query$\bar{x}$ was determined through a majority weighted vote from the k nearest neighbors, as described by (6).

$$\bar{w}_i = \begin{cases} \frac{d\left(\bar{x}, x_k^{NN}\right) - d\left(\bar{x}, x_i^{NN}\right)}{d\left(\bar{x}, x_k^{NN}\right) - d\left(\bar{x}, x_1^{NN}\right)} \times \frac{d\left(\bar{x}, x_k^{NN}\right) + d\left(\bar{x}, x_1^{NN}\right)}{d\left(\bar{x}, x_k^{NN}\right) + d\left(\bar{x}, x_i^{NN}\right)} & , if\ d\left(\bar{x}, x_k^{NN}\right) \neq d\left(\bar{x}, x_1^{NN}\right) \\ 1 & , if\ d\left(\bar{x}, x_k^{NN}\right) \neq d\left(\bar{x}, x_1^{NN}\right) \end{cases} \tag{5}$$

$$\bar{y} = \underset{y}{arg\,max} \sum_{\left(x_i^{NN}, y_i^{NN}\right) \in \bar{T}} \bar{w}_i \times \delta\left(y = y_i^{NN}\right) \tag{6}$$

The dual weight calculation considered two components: the first part was similar to the weight in WKNN, while the second represented a newly determined weight, both based on the fundamental idea of the distance weighting scheme. The dual weights $\bar{w}_i$ is, as shown in Eq. (5), were generally smaller than the weights computed in WKNN, except for the first and k -th nearest neighbors. Consequently, the corresponding neighbors x_i^{NN} had a smaller influence on the query classification outcome. Dual weights decreased rapidly from 1 at the first nearest neighbor distance to 0 at the farthest k-th nearest neighbor distance.

III. RESULT AND DISCUSSION

A. Dataset Description

The umbilical cord, also known as the navel string, is a connecting tissue or channel establishing a link between the placenta and fetus. It serves as a lifeline, fulfilling several crucial roles, including maintaining the viability and growth of fetus, eliminating waste compounds, and transporting essential elements such as oxygen, nutrients, and antibodies. These factors collectively contribute to the optimal development of fetus in the womb.

The umbilical cord dataset comprised 19 distinct features and three classes, namely normal, hypercoiling, and hypocoiling. The dataset exhibited an imbalanced ratio (IR) of 6.3%, with a total of 63 data points, as shown in Table I. The testing phase was performed using a training data split of 70% and a testing data split of 30%.

B. Performance Metrics

The performance of the classification model was assessed using three metrics, including accuracy, precision, recall, and F-Measure. In machine learning classification tasks, these metrics were derived from the confusion matrix parameters, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These parameters served as the basis for computing other performance metrics such as Precision, Recall, and F1 scores. Accuracy measures the amount of correctly classified data points relative to the ground-truth label divided by the total data used for testing. Precision is the rate of accurate predictions among all samples predicted to belong to the minority class, and it indicates the number of accurate positive predictions. Also, recall reflects the proportion of minority class samples labeled as positive. Table II shows the formulas for measuring accuracy, precision, recall, and F-Measure.

TABLE I
DESCRIPTIVE INFORMATION OF THE DATASET

Class	Attribute	Majority Percentage	Minority Percentage	Positive Class	Negative Class
3	19	6.3%	93.7%	59	4

TABLE II
PERFORMANCE METRICS

No	Metrics	Expression
1	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
2	Precision	$\frac{TP}{TP+FP}$
3	Recall	$\frac{TP}{TP+FN}$
4	F-Measure	$F_\beta = \frac{1}{\beta \times \frac{1}{precision} + (1-\beta) \times \frac{1}{recall}}$

The F-measure parameter represented the *harmonic mean* of precision and recall. It was governed by the value of β, ranging from 0 to 1. A higher β value indicated that the testing model prioritized the results of precision and vice versa.

C. Performance Analysis

The testing phase included the assessment of KNN, LMKNN, and DWKNN, utilizing three different distance metrics, namely Euclidean, Minowski, and Manhattan. In the first testing phase, the KNN method was evaluated with varying values of k (k = 1, 3, 5, 7, 9). Table III shows the testing results for the KNN voting algorithm.

The results showed that the KNN voting model attained its highest accuracy of 98.37% when k = 3, surpassing the conventional KNN method for Minowski, Manhattan, and Euclidean distances. In terms of precision performance, the KNN voting method also demonstrated highest achievement of 98.49% at k = 1 and 3. In the case of recall, all four models achieved the highest value of 98.34%. Similarly, in the F1 score, all four models attained a maximum result of 98.33%. The second testing phase continued with the DWKNN voting model, while DWKNN (Euclidean), DWKNN (Manhattan), DWKNN (Minowski), and DWKNN (Voting all) models were subsequently analyzed for optimal performance. Table IV shows the results of the second testing phase.

TABLE III
RESULTS OF ACCURACY, PRECISION, RECALL, AND F1-MEASURE MEASUREMENTS FOR KNN VOTING MODEL

Metrics	Number of k	Euclidean (KNN)	Manhattan (KNN)	Minkowski KNN	Voting KNN
Accuracy	1	0.982301	0.982301	**0.983776**	0.982301
	3	0.982301	**0.983776**	0.980826	**0.983776**
	5	0.983776	0.980826	0.976401	0.982301
	7	0.976401	**0.977876**	0.973451	0.976401
	9	0.973451	0.970501	0.973451	**0.973451**
Precision	1	0.98261	0.98261	0.983993	**0.984449**
	3	0.982933	**0.984449**	0.981432	**0.984449**
	5	**0.984449**	0.981559	0.976914	0.983145
	7	0.976914	**0.97843**	0.973798	0.976914
	9	0.974121	0.97118	0.974029	**0.974121**
Recall	1	0.98212	0.98212	**0.983498**	0.98212
	3	0.981858	0.983367	0.98035	**0.983367**
	5	**0.983367**	0.980308	0.975741	0.981858
	7	**0.975741**	0.977207	0.972682	**0.975741**
	9	0.972556	0.969539	0.972598	**0.972556**
F1_Score	1	0.982275	0.982275	0.983623	**0.982275**
	3	0.982139	**0.983693**	0.980585	**0.983693**
	5	**0.983693**	0.980705	0.976061	0.982234
	7	0.976061	**0.977647**	0.972975	0.976061
	9	0.973157	0.970137	0.973128	**0.973157**

In the second testing phase, the DWKNN voting algorithm demonstrated performance relatively similar to the DWKNN algorithm with Manhattan distance. The highest accuracy was achieved at 98.52% for k = 2, a result matched by DWKNN Manhattan at k = 5 and 7, also achieving a 98.52% accuracy rate. In terms of precision performance, the highest values were nearly identical for DWKNN Manhattan and DWKNN voting, both reaching 98.57 at k = 3. The recall values reached their peak at 98.48% for k = 3, 5, and 7. For the F1 score, DWKNN (Euclidean), DWKNN (Manhattan), and DWKNN (Voting all) all attained the same peak value of 98.36%. The third testing phase proceeded with a comparison of the performance results for Voting LMKNN, LMKNN (Euclidean), LMKNN (Manhattan), and LMKNN (Minowski) methods. Table V presents the results of the third testing phase

In the third test, the DWKNN voting algorithm consistently outperformed the other three methods. It achieved the highest accuracy of 98.86% at k = 5, the highest precision of 98.70% at k = 5, the peak recall at 98.80% with k = 3, and the highest F1 score of 98.63% at k = 5. The tests showed that the proposed nearest neighbors voting algorithm could enhance performance. While the performance improvement might not be highly significant, the inclusion of voting enhanced the ability of the algorithm to identify a decision boundary, particularly in imbalanced data conditions. The average performance improvement in terms of accuracy, precision, recall, and F1 score fell within the range of 1-2%. The incorporation of voting, based on different distance measurements in the nearest neighbors algorithm, broadened the scope for methods to make final class decisions for test data. Fig. 2 until 5 show a plot graph of the performance of the nearest neighbors voting model. The figure shows a comparison of the performance of the KNN, LMKNN, and DWKNN methods and the proposed method with parameters of different k values.

.TABLE IV
RESULTS OF ACCURACY, PRECISION, RECALL, AND F1-MEASURE MEASUREMENTS FOR LMKNN VOTING MODEL

Metrics	Number of k	Eclidean (DWKNN)	Manhattan (DWKNN)	Minkowski (DWKNN)	Voting DWKNN
Accuracy	1	0.982301	0.982301	**0.983776**	0.982301
	3	0.985251	0.985251	0.982301	**0.985251**
	5	0.982301	**0.985251**	0.977876	0.982301
	7	0.980826	**0.985251**	0.977876	0.980826
	9	0.980826	**0.983776**	0.979351	0.980826
Precision	1	0.98261	0.98261	**0.983993**	0.98261
	3	0.985762	0.985762	0.982745	**0.985762**
	5	0.982933	**0.985762**	0.97864	0.982933
	7	0.981432	**0.985978**	0.97864	0.981432
	9	0.981852	**0.984413**	0.980337	0.981852
Recall	1	0.98212	0.98212	0.983498	0.98212
	3	0.984875	0.984875	0.981858	**0.984875**
	5	0.981858	**0.984875**	0.977333	0.981858
	7	0.98035	**0.984875**	0.977333	0.98035
	9	0.98035	**0.983325**	0.978842	0.98035
F1_Score	1	0.982275	0.982275	0.983623	0.982275
	3	0.982139	**0.983693**	0.980585	**0.983693**
	5	**0.983693**	0.980705	0.976061	0.982234
	7	0.976061	**0.977647**	0.972975	0.976061
	9	**0.973157**	0.970137	0.973128	**0.973157**

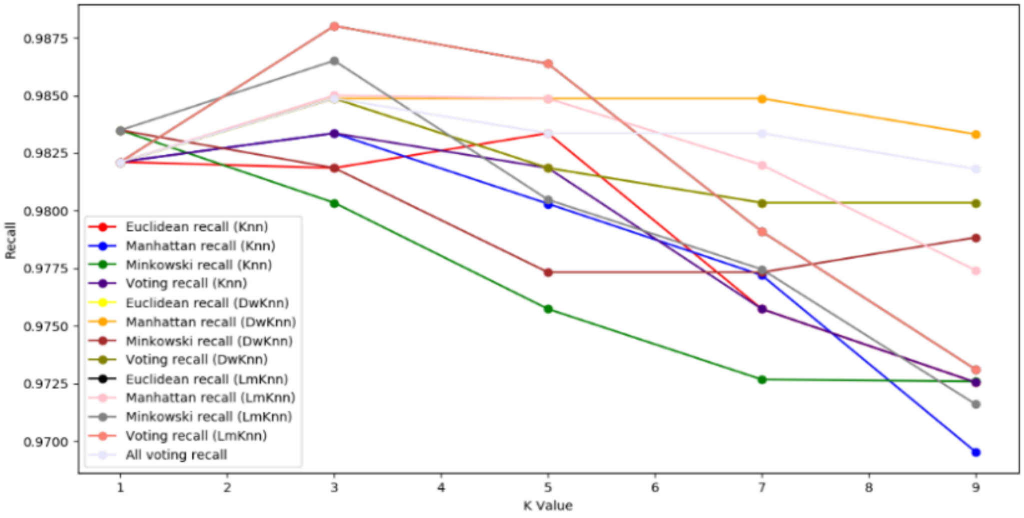

Fig. 2 Plot recall performance of proposed method in umbilical cord dataset

TABLE V
RESULTS OF ACCURACY, PRECISION, RECALL, AND F1-MEASURE MEASUREMENTS FOR DWKNN VOTING MODEL.

Metrics	Number of k	Eclidean (LMKNN)	Manhattan (LMKNN)	Minkowski (LMKNN)	Voting LMKNN
Accuracy	1	0.982301	0.982301	**0.983776**	0.982301
	3	**0.988201**	0.985251	0.986726	**0.988201**
	5	**0.986726**	0.985251	0.980826	**0.986726**
	7	**0.979351**	0.982301	0.977876	**0.979351**
	9	0.973451	**0.977876**	0.971976	0.973451
Precision	1	0.98261	0.98261	**0.983993**	0.98261
	3	**0.988221**	0.985516	0.986699	**0.988221**
	5	**0.987086**	0.985978	0.978038	**0.987086**
	7	0.979565	**0.982853**	0.978038	0.979565
	9	0.973729	**0.978429**	0.972177	0.973729
Recall	1	0.98212	0.98212	**0.983498**	0.98212
	3	**0.988022**	0.985006	0.986514	**0.988022**
	5	0.986383	0.984875	0.980481	**0.986383**
	7	0.979104	**0.981989**	0.977464	0.979104
	9	0.973117	**0.977422**	0.971609	0.973117
F1_Score	1	0.98212	0.98212	**0.983498**	0.98212
	3	0.988022	0.985006	**0.986514**	0.988022
	5	0.986383	0.984875	0.980481	**0.986383**
	7	0.979104	**0.981989**	0.977464	0.979104
	9	0.973117	**0.977422**	0.971609	0.973117

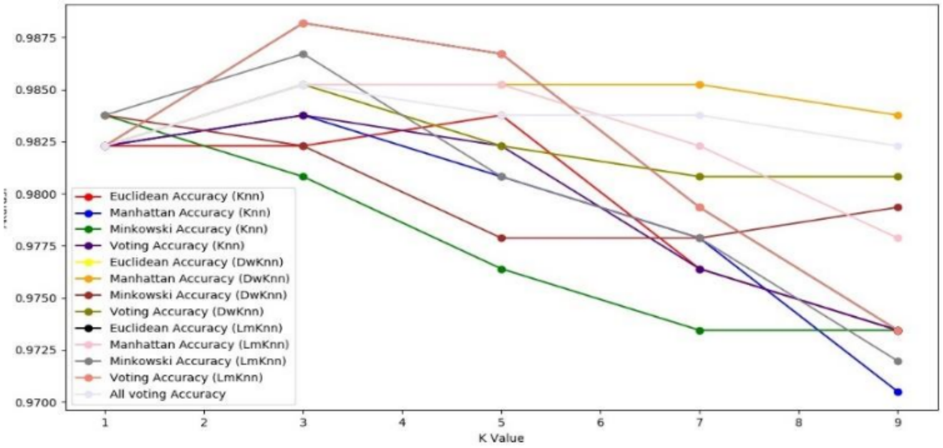

Fig. 3 Plot accuracy performance of proposed method in umbilical cord dataset

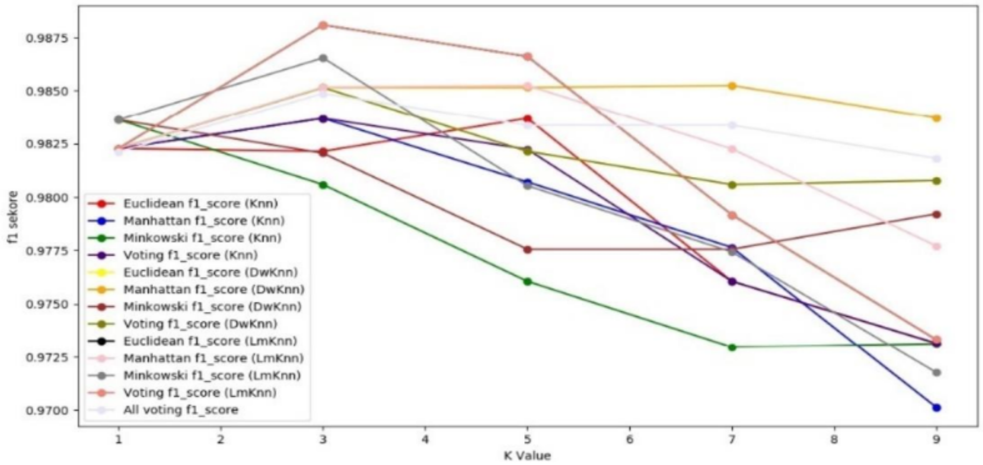

Fig. 4 Plot F1-score performance of proposed method in umbilical cord dataset

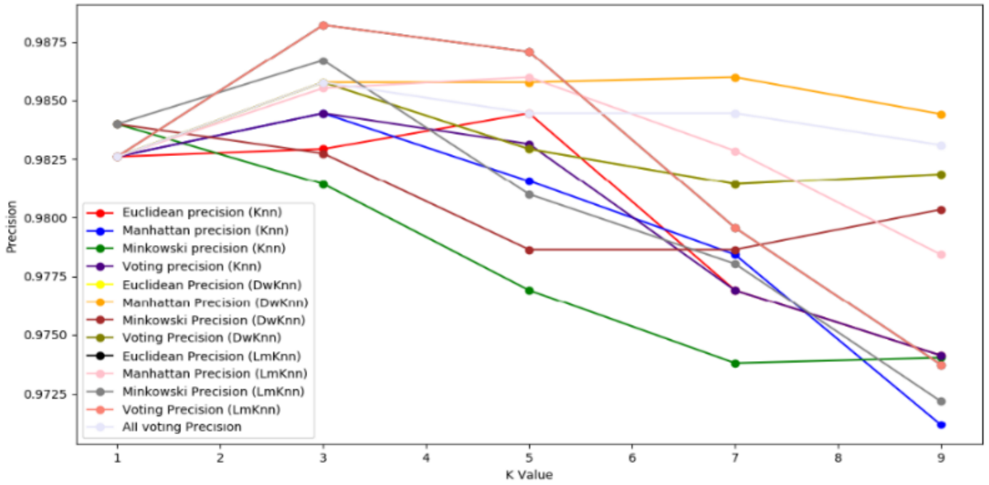

Fig. 5 Plot precision performance of proposed method in umbilical cord dataset

IV. CONCLUSION

In conclusion, this study developed a novel approach to enhance the KNN algorithm by incorporating distance-based metrics and implementing a voting scheme. The algorithms tested with the inclusion of voting based on distance metrics included KNN, LMKNN, and DWKNN. The evaluation was carried out on the umbilical cord dataset, characterized by its limited data volume and class imbalance. This was achieved using a 70:30 split for mechanism for training and testing data, with varying k-values for observations. The experimental results showed that the proposed method of nearest neighbors voting yielded an improvement of approximately 1% to 2% in terms of accuracy, precision, recall, and F1 score when compared to the non-voting approach. The integration of voting based on diverse distance measurements within the nearest neighbors algorithm provided a broader perspective for refining the process of making final class decisions from test data. For future developments, this study could consider addressing the relatively high computational time associated with the proposed method. The increased computational demands were a result of the various distance metric calculations and the inclusion of voting steps performed by the model for each test data point.

ACKNOWLEDGEMENT

The authors are grateful to the Directorate of Research, Technology, and Community Service (DPRM) Indonesia for funding this study through Program Funding (Fundamental Regular) for the 2023 Fiscal Year.

REFERENCES

[1] S. Suyanto, P. E. Yunanto, T. Wahyuningrum, and S. Khomsah, "A multi-voter multi-commission nearest neighbor classifier," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 6292–6302, Sep. 2022, doi: 10.1016/j.jksuci.2022.01.018.

[2] J. Gou, W. Qiu, Z. Yi, X. Shen, Y. Zhan, and W. Ou, "Locality constrained representation-based K-nearest neighbor classification," *Knowl Based Syst*, vol. 167, pp. 38–52, Mar. 2019, doi: 10.1016/j.knosys.2019.01.016.

[3] J. Gou, Y. Zhan, Y. Rao, X. Shen, X. Wang, and W. He, "Improved pseudo nearest neighbor classification," *Knowl Based Syst*, vol. 70, pp. 361–375, Nov. 2014, doi: 10.1016/j.knosys.2014.07.020.

[4] Y. Li and S. Ercisli, "Data-efficient crop pest recognition based on KNN distance entropy," *Sustainable Computing: Informatics and Systems*, vol. 38, Apr. 2023, doi: 10.1016/j.suscom.2023.100860.

[5] S. Klikovits, C. Ho Thanh, A. Cetinkaya, and P. Arcaini, "Trust your neighbours: Handling noise in multi-objective optimisation using kNN-averaging[Formula presented]," *Appl Soft Comput*, vol. 146, Oct. 2023, doi: 10.1016/j.asoc.2023.110631.

[6] G. Bhattacharya, K. Ghosh, and A. S. Chowdhury, "An affinity-based new local distance function and similarity measure for kNN algorithm," *Pattern Recognit Lett*, vol. 33, no. 3, pp. 356–363, Feb. 2012, doi: 10.1016/j.patrec.2011.10.021.

[7] L. Niu *et al.*, "Residual Vector Product Quantization for approximate nearest neighbor search," *Expert Syst Appl*, vol. 232, p. 120832, Dec. 2023, doi: 10.1016/j.eswa.2023.120832.

[8] Z. Zhu, Z. Wang, D. Li, and W. Du, "NearCount: Selecting critical instances based on the cited counts of nearest neighbors ☆," vol. 190, p. 105196, 2020, doi: 10.1016/j.knosys.

[9] J. Fan, Z. Pan, L. Wang, and Y. Wang, "Codebook-softened product quantization for high accuracy approximate nearest neighbor search," *Neurocomputing*, vol. 507, pp. 107–116, Oct. 2022, doi: 10.1016/j.neucom.2022.08.002.

[10] Z. Pan, Y. Wang, and W. Ku, "A new k-harmonic nearest neighbor classifier based on the multi-local means," *Expert Syst Appl*, vol. 67, pp. 115–125, Jan. 2017, doi: 10.1016/j.eswa.2016.09.031.

[11] Y. Mitani and Y. Hamamoto, "A local mean-based nonparametric classifier," *Pattern Recognit Lett*, vol. 27, no. 10, pp. 1151–1159, Jul. 2006, doi: 10.1016/j.patrec.2005.12.016.

[12] J. Gou, Y. Zhan, Y. Rao, X. Shen, X. Wang, and W. He, "Improved pseudo nearest neighbor classification," *Knowl Based Syst*, vol. 70, pp. 361–375, Nov. 2014, doi: 10.1016/j.knosys.2014.07.020.

[13] J. Gou, W. Qiu, Q. Mao, Y. Zhan, X. Shen, and Y. Rao, "A multi-local means based nearest neighbor classifier," in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, IEEE Computer Society, Jun. 2018, pp. 448–452. doi: 10.1109/ICTAI.2017.00075.

[14] Y. Zeng, Y. Yang, and L. Zhao, "Pseudo nearest neighbor rule for pattern classification," *Expert Syst Appl*, vol. 36, no. 2 PART 2, pp. 3587–3595, 2009, doi: 10.1016/j.eswa.2008.02.003.

[15] J. Gou, W. Qiu, Z. Yi, X. Shen, Y. Zhan, and W. Ou, "Locality constrained representation-based K-nearest neighbor classification," *Knowl Based Syst*, vol. 167, pp. 38–52, Mar. 2019, doi: 10.1016/j.knosys.2019.01.016.

[16] T. Xiong, J. Gou, L. Du, and Y. Zhang, "A New Distance-weighted k-nearest Neighbor Classifier A New Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance View project pattern recognition View project Jianping Gou A New Distance-weighted k-nearest Neighbor Classifier," 2011. [Online]. Available: http://www.joics.com

[17] J. S. Sanchez, F. Pla, and F. J. Ferri, "On the use of neighbourhood-based non-parametric classifiers 1," 1997.

[18] J. Gou, Z. Yi, L. Du, and T. Xiong, "A local mean-based k-nearest centroid neighbor classifier," *Computer Journal*, vol. 55, no. 9, pp. 1058–1071, Sep. 2012, doi: 10.1093/comjnl/bxr131.

[19] Y. Lin, J. Li, M. Lin, and J. Chen, "A new nearest neighbor classifier via fusing neighborhood information," *Neurocomputing*, vol. 143, pp. 164–169, Nov. 2014, doi: 10.1016/j.neucom.2014.06.009.

[20] L. Chen and G. Guo, "Nearest neighbor classification of categorical data by attributes weighting," *Expert Syst Appl*, vol. 42, no. 6, pp. 3142–3149, Apr. 2015, doi: 10.1016/j.eswa.2014.12.002.

[21] P. Gader Kotturu Satyanarayana, "Landmine Detection with Ground Penetrating Radar using Fuzzy K-Nearest Neighbors Hichem Frigui," 2004.

[22] H. Zhu and O. Basir, "An adaptive fuzzy evidential nearest neighbour formulation for classifying remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 8, pp. 1874–1889, Aug. 2005, doi: 10.1109/TGRS.2005.848706.

[23] Maozhen. Li, Yantai da xue., Institute of Electrical and Electronics Engineers., and IEEE Circuits and Systems Society., *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery : proceedings, 10-12 August 2010, Yantai, Shandong, China*. [IEEE], 2010.

[24] T. Denceux, "A k-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory," 1995.

[25] S. Surathong, C. Maisen, and P. Piyawongwisal, "Modified Fuzzy Dempster-Shafer Theory for Decision Fusion," in *2021 13th International Conference on Information Technology and Electrical Engineering, ICITEE 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 244–248. doi: 10.1109/ICITEE53064.2021.9611927.

[26] J. Derrac, S. García, and F. Herrera, "Fuzzy nearest neighbor algorithms: Taxonomy, experimental analysis and prospects," *Inf Sci (N Y)*, vol. 260, pp. 98–119, Mar. 2014, doi: 10.1016/j.ins.2013.10.038.

[27] S. R. Kheradpisheh, F. Behjati-Ardakani, and R. Ebrahimpour, "Combining classifiers using nearest decision prototypes," *Applied Soft Computing Journal*, vol. 13, no. 12, pp. 4570–4578, 2013, doi: 10.1016/j.asoc.2013.07.028.

[28] X. Zhang, H. Xiao, R. Gao, H. Zhang, and Y. Wang, "K-nearest neighbors rule combining prototype selection and local feature weighting for classification," *Knowl Based Syst*, vol. 243, May 2022, doi: 10.1016/j.knosys.2022.108451.

[29] E. Pękalska, R. P. W. Duin, and P. Paclík, "Prototype selection for dissimilarity-based classifiers," *Pattern Recognit*, vol. 39, no. 2, pp. 189–208, Feb. 2006, doi: 10.1016/j.patcog.2005.06.012.

[30] C. W. Yen, C. N. Young, and M. L. Nagurka, "A vector quantization method for nearest neighbor classifier design," *Pattern Recognit Lett*, vol. 25, no. 6, pp. 725–731, Apr. 2004, doi: 10.1016/j.patrec.2004.01.012.

[31] I. Triguero, J. Derrac, S. García, and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 1, pp. 86–100, Jan. 2012, doi: 10.1109/TSMCC.2010.2103939.

[32] F. Fernández and P. Isasi, "Local feature weighting in nearest prototype classification," *IEEE Trans Neural Netw*, vol. 19, no. 1, pp. 40–53, Jan. 2008, doi: 10.1109/TNN.2007.902955.

[33] S. Seo, M. Bode, and K. Obermayer, "Soft nearest prototype classification," *IEEE Trans Neural Netw*, vol. 14, no. 2, pp. 390–398, Mar. 2003, doi: 10.1109/TNN.2003.809407.

[34] P. Wohlhart, M. Kostinger, M. Donoser, P. M. Roth, and H. Bischof, "Optimizing 1-nearest prototype classifiers," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 460–467. doi: 10.1109/CVPR.2013.66.

Voting Scheme.pdf

ORIGINALITY REPORT

4%
SIMILARITY INDEX

PRIMARY SOURCES

1	www.researchgate.net Internet	99 words — 3%
2	Suyanto Suyanto, Prasti Eko Yunanto, Tenia Wahyuningrum, Siti Khomsah. "A multi-voter multi-commission nearest neighbor classifier", Journal of King Saud University - Computer and Information Sciences, 2022 Crossref	58 words — 2%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	< 2%
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	OFF