**Communications in Mathematics and Applications**
Vol. 15, No. 4, pp. 1341–1351, 2024
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v15i4.3278

http://www.rgnpublications.com

***Special Issue*:**
**Frontiers in Applied and Computational Mathematics**
*Editors*: M. Vishu Kumar, S. Lakshmi Narayana, B. N. Hanumagowda, U. Vijaya Chandra Kumar

Research Article

# Enhancing Classification Accuracy With *K*-Nearest Neighbors: Optimizing Distance Metrics and Handling Unlabeled Data

C. Pavithra* and M. Saradha

*Department of Mathematics, REVA University, Bangalore, Karnataka, India*
***Corresponding author:** pavithracshekar5@gmail.com

**Received:** January 4, 2024 **Accepted:** June 27, 2024

**Abstract.** The K-Nearest Neighbors (K-NN) is a supervised machine learning algorithm, specifically within the realm of classification and regression tasks. It is a simple yet powerful method used for making predictions based on similarities between data points. The fundamental idea behind K-NN is to classify or predict a new data point's label or classify by looking at the labels of its nearest neighbors in the training dataset. This research introduces an adaptive K-NN classification approach that leverages local data characteristics to dynamically adjust neighborhood size and distance metrics. This adaptive K-NN classification approach is thoroughly examined through comparisons with $k = 3$, $k = 5$ and $k = 7$. By adjusting neighborhood size and distance metrics, the study yields nuanced performance insights. Calculated outlier-to-class distances offer valuable adaptability indications. The experimental results showcase its potential to enhance classification accuracy and adaptability in diverse data scenarios. This method contributes to the advancement of K-NN based classification techniques and provides a promising direction for improving the efficiency of data classification tasks in real-world applications.

**Keywords.** K-nearest neighbor, Non-classified data, Adaptive K-NN, Outliers

**Mathematics Subject Classification (2020).** 62H30, 68T05, 62C20

## 1. Introduction

The *K-Nearest Neighbors* (K-NN) classification technique, known for its simplicity and effectiveness in a wide range of scenarios, functions as a non-parametric method. When classifying a given data point $t$, the algorithm identifies the $k$-closest neighbors, forming a neighborhood around it. Typically, the classification of $t$ is determined by a majority voting process among the data points in this neighborhood. Optionally, the process can incorporate distance-based weighting. However, a crucial factor in K-NN is the choice of the parameter $k$, as the success of the classification heavily depends on it. Essentially, the K-NN method is inherently shaped by the selected $k$-value. While various methods exist for choosing the best $k$-value, a straightforward approach involves running the algorithm multiple times with different $k$-values and selecting the configuration that demonstrates the highest performance (Gou *et al.* [3], and Lopez-Bernal *et al.* [6]).

Recognizing the vulnerability of K-NN's performance to the chosen $k$-value, Wang *et al.* [8] introduced an inventive approach to alleviate this reliance. Instead of exclusively depending on a single set of K-Nearest Neighbors, the proposal involves considering multiple sets of nearest neighbors. This innovative framework centers around contextual probability, aiming to consolidate support from distinct nearest neighbor sets associated with different classes. The outcome is a more resilient support value that accurately reflects the true class of the data point, $t$. It is important to note, however, that the initial version of this technique comes with a computational overhead, necessitating $O(n \wedge 2)$ operations for classifying a new instance. Despite this computational cost, the approach significantly diminishes the need for the $k$-value and produces classification performance that closely approximates what's achieved with the optimal $k$ (Cai *et al.* [1]).

The *K-Nearest Neighbors* (K-NN) algorithm encounters a notable challenge in terms of the increased computational effort required for classifying new data instances. This higher computational demand is largely attributed to the fact that the bulk of calculations occurs during the classification stage, rather than during the initial processing of training examples. Despite its demonstrated effectiveness in applications like text categorization, where it performs exceptionally well on widely-recognized benchmarks such as the Reuters corpus of newswire stories, K-NN's 'lazy learning' nature poses a limitation (Pan *et al.* [7], and Wang *et al.* [8]). This nature implies that it does not involve pre-modeling or extensive preparation of the data prior to making predictions. Consequently, K-NN is best suited for situations where dynamic classification is needed on relatively smaller datasets. It is worth noting that techniques have been developed to mitigate the computational load during the querying phase. These methods include strategies like indexing the training examples to speed up the search process. However, these techniques go beyond the specific scope of the current paper's focus (Gou *et al.* [3]). To address these challenges, the paper introduces a novel K-NN-based classification approach called the 'K-NN Model'. This innovative technique centers on constructing a model using the available data and subsequently employing this model to classify new, unseen data points. The model is built by selecting representative data points from the training dataset and organizing them into distinct set. This approach aims to enhance the efficiency and performance of K-NN in scenarios where the conventional method faces limitations (Wang *et al.* [8], and Cai *et al.* [1]).

The paper introduces a modeling and classification algorithm and explains its foundational concept. The concepts of modeling and classification are illustrated through a practical example,

complemented by graphical aids, detailed information about experimental results is provided, followed by an in-depth discussion of these findings. The paper goes on to discuss prevailing challenges and offers insights into potential directions for future research. This study is presented as a natural extension of prior research that focused on *Data Reduction* (DR). Data Reduction offers a distinct advantage both the original and reduced datasets can be efficiently represented using hyper relations (Zhang *et al.* [10]). These hyper relations, when collected, create a comprehensive Boolean algebra seamlessly. This means that for any grouping of hyper tuples, their unique *least upper bound* (lub) can be derived through the reduction process. The experimental results confirm that data reduction demonstrates a relatively high reduction rate while maintaining classification accuracy. However, It is important to note that the basic procedure of constructing DR's model can be somewhat time-consuming due to the considerable time spent exploring potential merges (Chen and Kartini [2], and Zhang *et al.* [10]).

The *K-Nearest Neighbors* (K-NN) classifier presents a challenge due to its need to store the entire training set, which becomes particularly demanding when dealing with large datasets. To address this issue, researchers have worked on reducing redundancy within the training set, aiming to alleviate this challenge. The author introduces a computationally viable local search technique known as *Condensed Nearest Neighbor* (CNN). This approach aims to reduce the number of stored patterns by retaining only a subset of the training set for classification purposes. The method is built upon the notion that certain patterns within the training set are very similar and do not provide additional information, thus justifying their exclusion. Furthermore, the *Reduced Nearest Neighbor* (RNN) rule is proposed as an extension of CNN. RNN refines the subset stored by CNN by promptly removing elements that are unlikely to lead to errors. The paper also explores the use of voting schemes with multiple learners to enhance classification accuracy. Additionally, a different approach is presented where three compact groups of examples are identified. When used as sub-classifiers for K-NN, each group tends to make errors in distinct segments of the instance space. A simple voting mechanism corrects many of these individual sub-classifier errors (Lamba and Kumar [5]). The outcomes of these methodologies are thoroughly evaluated using publicly available datasets. Departing from both the *Data Reduction* (DR) methodology and other condensed nearest neighbor techniques, the proposed K-NN model-based approach introduces a fresh perspective. It involves constructing a model by identifying a set of representative instances within the training data, enhanced with additional informative attributes based on similarity principles. These representatives serve as virtual regions within the data space and hold the potential to significantly influence subsequent classification tasks. In the following sections, the paper delves into the uniqueness of the proposed K-NN model-based approach in comparison to existing DR and condensed nearest neighbor methods. The paper explains how this novel approach aims to build a model through the identification of representative instances and how these representative instances, akin to delineated regions within the data space, play a crucial role in subsequent classification endeavors (Keller *et al.* [4], and Lamba and Kumar [5]).

## 2. Preliminaries

The introductory context to the *K-Nearest Neighbors* (KNN) model serves as a foundation for comprehending the innovative approach this research aims to present. While the conventional K-NN classification method has demonstrated its effectiveness in various scenarios. It is not without its challenges, especially when applied to sizable training datasets. The typical

K-NN technique demands the retention of the entire training dataset, leading to potentially high computational expenses during classification, particularly in cases where efficiency and real-time performance are of utmost importance. In response to these challenges and with the intention of discovering new ways to enhance the efficiency and performance of K-NN, this study proposes a novel K-NN model-based approach. Unlike the conventional K-NN, which directly uses the original training instances for classification, the K-NN model approach introduces a fundamental change in how the training data is employed. This approach revolves around constructing a specialized model that captures the essence of the training data while concurrently reducing redundancy and improving classification efficiency. The fundamental concept underpinning the K-NN model approach centers on the identification of a subset of representative instances from the original training dataset (Keller *et al.* [4], and Xiao *et al.* [9]). These representatives encapsulate the essential characteristics of the original data while potentially integrating additional information based on a similarity-based principle. These chosen representatives can be conceptualized as distinct regions within the data space and act as stand-ins for the entire training dataset. This substitution enables a more streamlined and efficient classification process. This approach differs from conventional data reduction techniques and other condensed nearest neighbor methods, which often involve the selection of a subset of instances using diverse criteria. In contrast, the K-NN model approach aims to strike a balance between efficiency and accuracy. It provides a novel perspective on how K-NN classification can be improved and adjusted to tackle the challenges posed by large datasets. By focusing on representative instances that efficiently convey the underlying information of the data, the K-NN model approach introduces an innovative methodology to enhance the performance of K-NN classification (Cai *et al.* [1], and Lamba and Kumar [5]).

The Euclidean distance metric plays a crucial role in the functioning of the *K-Nearest Neighbors* (K-NN) classification algorithm. It serves as a fundamental tool for measuring the distance between two points within a multidimensional space. Specifically, the Euclidean distance represents the straight-line distance between these two points. In the context of K-NN, this metric is widely employed to quantify the similarity or dissimilarity between different data points. By calculating the Euclidean distance between data points, the K-NN algorithm gains a way to determine how 'close' or 'far' these points are from each other within the multidimensional space. This information is crucial for the algorithm to identify the nearest neighbors of a given data point. These nearest neighbors contribute to the decision-making process in K-NN classification, as they influence the class assignment of the data point in question. In essence, the Euclidean distance metric enables the algorithm to gauge the spatial relationships between data points, aiding in making informed classification choices based on the proximity of neighboring points (Xiao *et al.* [9]).

The formula for calculating the Euclidean distance between two data points $A = (x_1, y_1, z_1, \ldots, n_1)$ and $B = (x_2, y_2, z_2, \ldots, n_2)$ in an $n$-dimensional space is as follows:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{in} - x_{jn})^2}$$

or

$$d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}.$$

In this formula $x, y, z$ and $n$ represent the individual dimensions of the data points. The Euclidean distance calculation involves computing the squared differences between corresponding dimensions, summing them up, and then taking the square root of the result to obtain the distance (Keller *et al.* [4]).

Within the realm of K-NN classification (Figure 1):

- For every data point requiring classification, the Euclidean distance is computed between that specific point and all other data points within the training dataset.
- The resulting distances are organized in ascending order to discern the K-Nearest Neighbors, where $K$ is a value defined by the user.
- The class labels of these K-Nearest Neighbors are analyzed, and the predominant class among them establishes the classification label for the particular data point under consideration.

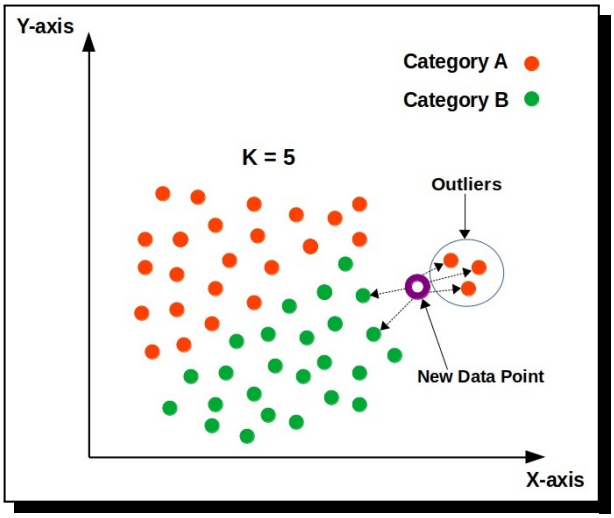


**Figure 1.** K-NN algorithm for classification

Euclidean distance metric is commonly used and easily understood, like data normalization or standardization, become necessary to prevent dimensions with larger ranges from exerting undue influence on the distance computation. Furthermore, considering alternative distance metrics is also valuable. These could include measures like Manhattan distance, cosine similarity, or Mahalanobis distance. The choice of metric should align with the data's characteristics and the specific demands of the classification task. In the grand scheme, the Euclidean distance metric plays a central role in the K-NN Algorithm. It empowers the algorithm to quantify the closeness between data points and to make informed classification decisions based on the labels of neighboring points. However, It is important to recognize that its suitability can vary, and practitioners should be open to considering alternatives when necessary (Keller *et al.* [4], and Xiao *et al.* [9]).

## 3. Research Methodology

The primary goal of this research paper is to accomplish two main tasks: identifying unlabeled data and subsequently classifying it using a combination of matrix distance and an adaptive

K-NN (K-Nearest Neighbors) technique. To achieve this objective, synthetic data is generated specifically for the purpose of identification and classification. The paper outlines a three-step approach to carry out this process:

- *Extracting Synthetic Features*: The first step involves extracting synthetic features from AI tools, particularly `Getrel.ai`. These synthetic features are essentially attributes or characteristics derived from the AI tool's output, which serve as the basis for subsequent analysis.
- *Dimensionality Reduction*: Following feature extraction, the dimensionality of these extracted features is reduced. This step aims to simplify the data while preserving its essential information. The reduction process involves considering the data points that are closest to the outliers. Outliers are data points that significantly differ from the majority of the data and might contain valuable information or anomalies.
- *Adaptive K-NN Classification*: The final step employs an adaptive K-NN algorithm for classification purposes. K-NN is a machine learning technique that classifies a data point based on the classes of its K-Nearest Neighbors. In this case, the K-NN algorithm is modified to be adaptive, likely implying that it adjusts its behavior based on the characteristics of the data. This modified algorithm is used to classify the reduced-dimensional data and, in this specific context, to identify instances of malware.

In essence, this paper presents a systematic approach that involves generating synthetic data, extracting relevant features, reducing dimensionality, and then classifying data using an adaptive K-NN algorithm. The ultimate objective is to identify and classify data instances, particularly focusing on identifying instances of malware.

### 3.1 Dataset Selection

We gather a variety of datasets to ensure a thorough assessment of the proposed strategies. This is crucial to showcase how these strategies can be applied effectively across a range of data types and levels of complexity.

### 3.2 Enhancing K-NN Classification

(a) *Distance Metrics*: We examine the influence of diverse distance metrics, including Euclidean, Manhattan, and Cosine, on the classification accuracy of K-NN. The rationale behind exploring these distinct distance metrics is to underscore how the selection of a metric can impact the algorithm's capacity to precisely gauge the similarity between data points.

(b) *Outliers*: Outliers refer to data points that deviate significantly from the majority of the dataset. These points can exhibit unusually high or low values and might not conform to the overall data pattern or trend. Outliers can emerge due to measurement inaccuracies, inherent variability, or even exceptional occurrences. They possess the capacity to skew statistical assessments and demand thoughtful attention during data analysis. Properly identifying and addressing outliers is crucial to guarantee the precision and relevance of insights drawn from the data.

## 4. Experiment

In this experiment and analysis, we aimed to investigate the impact of outliers on the nearest data points and their distances within a clustered dataset. Initially, we generated random data points for three distinct classes - Class 1, Class 2, and Class 3 - while also introducing three strategically placed outliers - Outlier 1, Outlier 2, and Outlier 3 - located between these classes. To quantify the effect of these outliers, we calculated the Euclidean distance between each outlier and all data points, including outliers themselves. Then, for each outlier, we sorted the calculated distances in ascending order and selected the nearest data points based on a varying number of closest points, such as 3, 5, 7, and so forth. This allowed us to explore how the number of nearest data points influenced the outcomes. We presented the results through visualizations, showcasing the dataset's scatter plot with distinct markers for each class and the outliers.

## 5. Implementation

We have identified 3 unlabeled data points (Figure 2). Now let us classify the unlabeled data by calculating the Euclidean distances numerically for each outlier to the nearest data points using the given data and outliers.

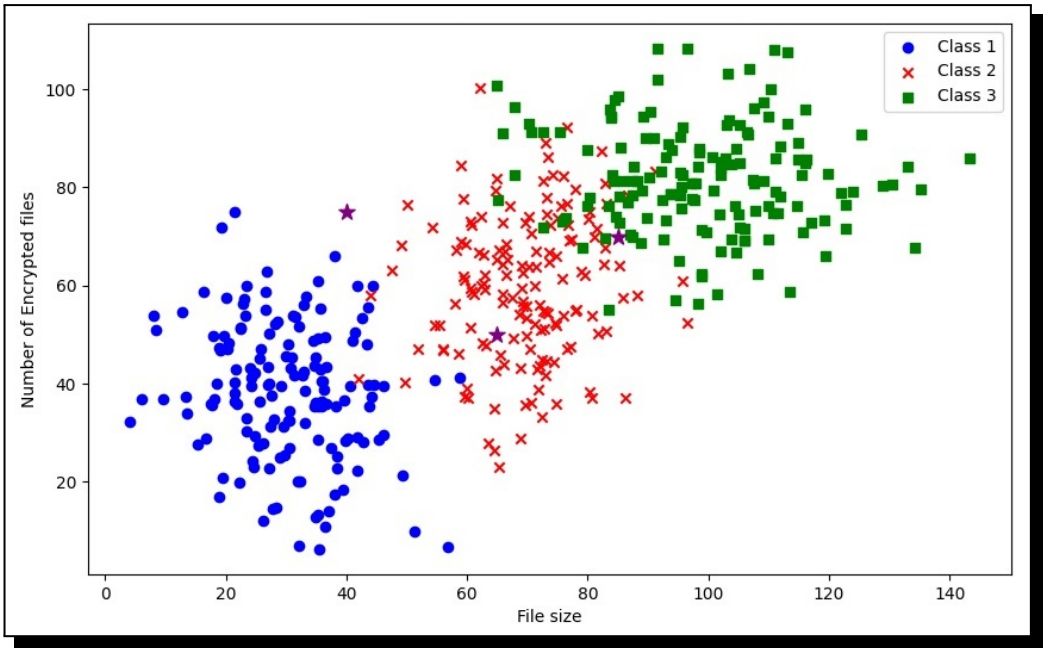


**Figure 2.** KNN algorithm for training data

We have computed the distance between the nearest data in three different cases by assuming the different values of $k$ ($k = 3$, $k = 5$ and $k = 7$).

- *Identified Unlabeled Data Points*: we have a set of 3 data points that do not yet have assigned labels. These are the unlabeled data points.
- *Objective*: The goal is to assign labels to these unlabeled data points using the K-Nearest Neighbors (K-NN) algorithm. K-NN uses the distances between these unlabeled data points and the nearest labeled data points to determine the most likely class or label for each of the unlabeled data points.

- *Calculating Euclidean Distances*: To do this, we will calculate the Euclidean distances between each of the unlabeled data points and the nearest labeled data points. The Euclidean distance measures the straight-line distance between two points in a multi-dimensional space. This distance serves as a measure of similarity between data points.
- *Numerical Calculation*: For each unlabeled data point, we will numerically compute the Euclidean distance to the nearest data points from your existing dataset, which includes both labeled data and outliers.
- *Different Cases with Varying $k$*: To analyze the effect of different scenarios, you are performing this distance calculation for three different cases, each with a different value of $k$:

  $k = 3$: This means we will consider the 3 nearest data points to each unlabeled data point.

  $k = 5$: This involves finding the 5 nearest data points.

  $k = 7$: Lastly, we will compute distances to the 7 nearest data points.
- *Interpreting Results*: The resulting distances for each outlier-unlabeled data point pair will show how close or far they are from each other in the given dataset. Generally, a smaller distance indicates higher similarity.

By computing these distances for varying values of $k$, we are essentially quantifying how similar the unlabeled data points are to the labeled data points in different scenarios. This similarity information will be used to classify the unlabeled data points using KNN, where the class assigned to an unlabeled point is determined by the majority class among its K-Nearest Neighbors. The values of $k$ will influence the level of granularity in the classification process.

## 6. Discussion

In the present analysis, we conducted computations to ascertain the Euclidean distances between each outlier and its three nearest data points. These calculations were executed using the provided dataset, encompassing classes and outlier data. The Euclidean distance, a pivotal metric measuring the direct spatial gap between two points within a Euclidean space, was employed. For a pair of points represented as $(x_1, x_2)$, the Euclidean distance determines the straight-line distance between them. Observing the graph presented below, it becomes apparent that the Euclidean distances manifest variations in three distinct scenarios, when the value of $k$ is set at 3, 5, and 7, with regard to each outlier and its closest data points. These distances hold significance as they can be employed as a measure of resemblance between data points. Ordinarily, outliers tend to exhibit larger distances in comparison to their nearest neighbors, while data points originating from the same class demonstrate relatively smaller distances as represented in Figure 3, Figure 4 and Figure 5.

These shorter distances imply a higher degree of resemblance between them. Euclidean distances find wide-ranging applicability in domains such as machine learning, data analysis, and clustering. They function as a tool to quantify the associations between various data points. These distances furnish valuable insights into the geometric relationships and underlying patterns within the dataset, thereby aiding in the comprehension of its inherent structure.

**Table 1.** Distance table of outlier

| Class | $K = 3$ | | | $K = 5$ | | | $K = 7$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Class 1 | Outlier1 | Outlier2 | Outlier3 | Outlier1 | Outlier2 | Outlier3 | Outlier1 | Outlier2 | Outlier3 |
| | 35.35 | 63.25 | 35.08 | 35.35 | 59.02 | 35.02 | 35.35 | 63.45 | 35.35 |
| | | | | | 48.52 | 39.78 | | 68.70 | 37.91 |
| | | | | | 30.87 | | | 72.88 | |
| Class 2 | | 18.97 | 41.23 | 20.86 | 14.02 | 15.05 | 24.00 | 18.97 | 38.15 |
| | | | | | | | 18.27 | 22.99 | 42.85 |
| | | | | | | | 15.27 | | |
| Class 3 | 9.22 | 56.35 | 22.36 | 25.98 | 26.25 | 74.45 | 9.22 | 18.87 | 22.36 |
| | 25.00 | | | 36.80 | | 28.98 | 24.05 | 24.07 | 28.18 |
| | | | | 9.82 | | | 18.71 | | 36.48 |

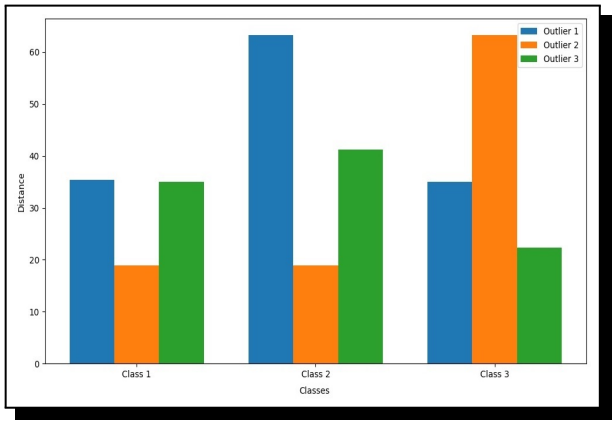


**Figure 3.** Distances outliers to the nearest classes ($k = 3$)

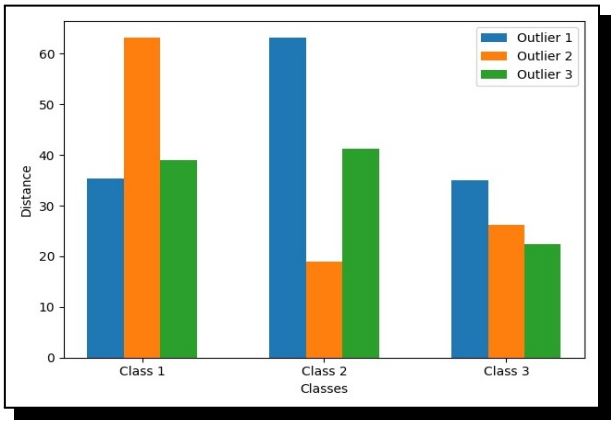


**Figure 4.** Distances outliers to the nearest classes ($k = 5$)

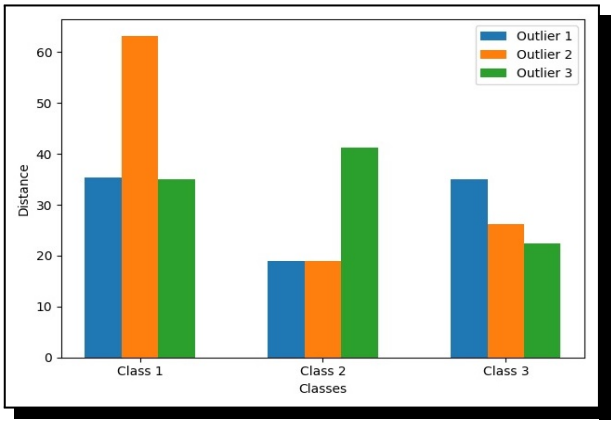


**Figure 5.** Distances outliers to the nearest classes ($k = 7$)

## 7. Conclusion

This research introduces an adaptable KNN classification methodology and conducts a comprehensive evaluation of its efficiency, specifically comparing scenarios involving $k = 3$,

$k = 5$, and $k = 7$. Through the customization of neighborhood size and distance metrics, the study achieves a nuanced comprehension of the performance of our proposed approach. The calculated distances between outliers and their nearest classes yield crucial insights into the adaptability and dependability of the approach. The experimental findings highlight the potential for improved classification accuracy at $k = 7$ as opposed to $k = 3$. As evidenced in Table 1, our research establishes a robust framework for refining KNN-based classification techniques. Additionally, the research emphasizes the significance of tailoring neighborhood parameters, emphasizing that an adaptive strategy can yield more dependable and accurate outcomes.

## Acknowledgement

The authors would like to thank REVA University for their encouragement and support in carrying out this research work.

### Competing Interests

The authors declare that they have no competing interests.

### Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

## References

**[1]** Y.-L. Cai, D. Ji and D. F. Cai, A KNN research paper classification method based on shared nearest neighbor, in: *Proceedings of NTCIR-8 Workshop Meeting*, June 15–18, 2010, Tokyo, Japan, pp. 336 – 340 (2010), URL: https://research.nii.ac.jp/ntcir/workshop/OnlineProceedings8/NTCIR/07-NTCIR8-PATMN-CaiY.pdf.

**[2]** C.-R. Chen and U. T. Kartini, $k$-Nearest neighbor neural network models for very short-term global solar irradiance forecasting based on meteorological data, *Energies* **10**(2) (2017), 186, DOI: 10.3390/en10020186.

**[3]** J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao and H. Yang, A generalized mean distance-based $k$-nearest neighbor classifier, *Expert Systems with Applications* **115** (2019), 356 – 372, DOI: 10.1016/j.eswa.2018.08.021.

**[4]** J. M. Keller, M. R. Gray and J. A. Givens, A fuzzy K-nearest neighbor algorithm, *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-15**(4) (2016), 580 – 585, DOI: 10.1109/TSMC.1985.6313426.

**[5]** A. Lamba and D. Kumar, Survey on KNN and its variants, *International Journal of Advanced Research in Computer and Communication Engineering* **5**(5) (2016), 430 – 435.

**[6]** D. Lopez-Bernal, D. Balderas, P. Ponce and A. Molina, Education 4.0: Teaching the basics of KNN, LDA and simple perceptron algorithms for binary classification problems, *Future Internet* **13** (2021), 193, DOI: 10.3390/fi13080193.

**[7]** Z. Pan, Y. Wang and Y. Pan, A new locally adaptive $k$-nearest neighbor algorithm based on discrimination class, *Knowledge-Based System* **204** (2020), 106185, DOI: 10.1016/j.knosys.2020.106185.

**[8]** B. Wang, X. Gan, X. Liu, B. Yu, R. Jia, L. Huang and H. Jia, A novel weighted KNN algorithm based on RSS similarity and position distance for Wi-Fi fingerprint positioning, *IEEE Access* **8** (2020), 30591 – 30602, DOI: 10.1109/access.2020.2973212.

**[9]** T. Xiao, F. Cao, T. Li, G. Song, K. Zhou, J. Zhu and H. Wang, KNN and re-ranking models for English patent mining at NTCIR-7, in: *Proceedings of the 7th NTCIR Workshop Meeting*, December 16–19, 2008, Tokyo, Japan, pp. 333 – 340, URL: https://research.nii.ac.jp/ntcir/workshop/OnlineProceedings7/pdf/NTCIR7/C3/PATMN/02-NTCIR7-PATMN-XiaoT.pdf.

**[10]** S. Zhang, X. Li, M. Zong, X. Zhu and D. Cheng, Learning K for KNN classification, *ACM Transactions on Intelligent Systems and Technology* **8** (2017), Article number 43, 1 – 19, DOI: 10.1145/2990508.