DOI:10.25092/baunfbed. 1634534 *J. BAUN Inst. Sci. Technol., 28(1), 191-211, (2026)*

A boosted hierarchical clustering linkage algorithm: K-Centroid link supported with OWA approach

Alican DOĞAN[1*], Efendi NASİBOV[2]

[1]*Banadırma Onyedi Eylül Üniversitesi, Yönetim Bilişim Sistemleri Böl., Bandırma.*
[2]*Dokuz Eylül Üniversitesi Bilgisayar Bilimleri Böl., İzmir.*

Geliş Tarihi (Received Date): 06.02.2025
Kabul Tarihi (Accepted Date):05.10.2025

Abstract

The choice of linkage algorithm plays a crucial role in determining the quality of hierarchical clustering and therefore must be made carefully. This selection significantly influences the effectiveness of the clustering process. However, conventional linkage methods do not take into account the influence of records located near the cluster centers. Previous studies proposed the k-centroid link, a new cluster merging criterion that analyzes instances near cluster centers in greater detail to improve clustering quality. The k-centroid link computes the average distance among the k nearest data points to the central point within each cluster. In this study, we enhance the clustering capability of the k-centroid link by integrating the Ordered Weighted Averaging (OWA) approach. Specifically, OWA values of the average distances between the k nearest records to each cluster center are calculated using a constant-level weighted stress function across different α values, rather than relying solely on direct distance calculations. The proposed model was evaluated on 24 publicly available benchmark datasets specifically designed for clustering tasks. The results demonstrate that the k-centroid link can be significantly improved through the application of OWA-based approaches with different stress functions.

Keywords: *Machine learning, hierarchical clustering, linkage method, OWA*

OWA yaklaşımı ile desteklenen K-Centroid bağlantılı güçlendirilmiş hiyerarşik kümeleme bağlantı algoritması

Öz

Hiyerarşik kümeleme işleminin kalitesini belirlemede bağlantı algoritması önemli bir rol oynamaktadır. Bu nedenle, dikkatli bir şekilde seçilmelidir. Bu seçim, kümeleme sürecinin

[*]Alican DOĞAN, alicandogan@bandirma.edu.tr, http://orcid.org/0000-0002-0553-2888
Efendi Nasiboğlu, efendi.nasibov@deu.edu.tr, http://orcid.org/0000-0002-7273-1473

etkinliğini önemli ölçüde etkilemektedir. Bununla birlikte, geleneksel bağlantı türleri, kümelerin merkez noktalarına yakın olan çevresel kayıtların etkisini dikkate almamaktadır. Bu eksikliği gidermek amacıyla, önceki çalışmalarda k-centroid bağlantı adı verilen yeni bir küme birleştirme kriter modeli önerilmiştir. Bu model, kümeleme kalitesini artırmak için küme merkezlerine yakın örnekleri detaylı bir şekilde analiz etmektedir. K-centroid bağlantı, her küme içerisindeki merkez noktasına en yakın k veri noktasının ortalama uzaklığını hesaplamaktadır. Bu çalışmada, k-centroid bağlantı kümeleme yeteneği Sıralı Ağırlıklı Ortalama (OWA) yaklaşımı ile desteklenerek geliştirilmiştir. Küme merkezlerine en yakın k kayıt arasındaki ortalama mesafelerin OWA değerleri, doğrudan bu mesafeleri hesaplamak yerine farklı α değerleri için eş seviyeli ağırlıklı stres fonksiyonu kullanılarak hesaplanmıştır. Bu yeni model, kümeleme için tasarlanmış 24 farklı açık erişimli veri kümesi üzerinde değerlendirilmiştir. Sonuçlar, farklı stres fonksiyonları için OWA yaklaşımlarının desteğiyle k-centroid bağlantı modelinin önemli ölçüde geliştirilebileceğini göstermektedir.

Anahtar kelimeler: *Makine öğrenimi, hiyerarşik kümeleme, bağlantı yöntemi, OWA*

1. Introduction

Clustering plays a pivotal role in machine learning and data mining. The hierarchical clustering problem is a fundamental branch of clustering, and its performance is influenced by various factors [1,2]. Among these factors, feature selection and the choice of linkage type are particularly critical [3,4]. The goal of clustering is to create disjoint groups, each containing at least one instance, and to assign instances to these groups based on their similarity [5]. In hierarchical clustering, the original groups of instances are divided or agglomerated into subgroups according to specific criteria. Hierarchical algorithms provide valuable insights into the relationships among features.
However, they face certain challenges, such as the lack of a clear stopping criterion for merging or dividing clusters and the difficulty of determining the optimal number of clusters. Several fuzzy approaches have been proposed to address these issues and to optimize the parameters more effectively [6–9]. For example, [8] proposed the FJP algorithm, which can automatically determine the appropriate number of clusters. Subsequent studies have introduced more effective versions of this algorithm [6,7–10].

Hierarchical clustering can be performed using two main approaches: bottom-up (agglomerative) and top-down (divisive) [11]. Although conceptually similar, these approaches produce different solutions. They generate different solutions in spite of being conceptually similar [12-16]. While both agglomerative and divisive strategies are fundamental to hierarchical clustering, this study focuses exclusively on the agglomerative approach. The proposed OWA-based k-centroid linkage is specifically designed as an enhancement to agglomerative merging criteria and is not applied to divisive clustering [12,13].

The primary contribution of this study is the enhancement of the k-centroid linkage method by integrating an OWA strategy [10]. The effectiveness of the proposed approach is evaluated on 24 benchmark datasets, and the results demonstrate its ability to outperform conventional linkage methods as well as the standard k-centroid link [17].

It should be noted that while challenges such as defining an explicit stopping criterion and determining the optimal number of clusters are important research questions in hierarchical clustering, they are beyond the main scope of this study. Instead, our work focuses on improving the linkage criterion itself. However, the proposed OWA-based k-centroid linkage may indirectly mitigate the effects of these challenges by producing more reliable and accurate cluster structures.

The remainder of the article is structured as follows: Section 2 reviews related work on hierarchical clustering. Section 3 presents an overview of widely used linkage models, the fundamentals of bottom-up hierarchical clustering, and details of the k-centroid link. Section 4 reports the experimental results, including comparisons among distinct linkage methods, the k-centroid link, and the OWA-based k-centroid linkage method. Finally, Section 5 concludes the study and outlines possible directions for future research.

2. Related work

Research on hierarchical clustering dates back to the 1950s. It remains one of the most frequently used approaches in cluster analysis. The foundations of the single linkage clustering method were first introduced during this period. Hierarchical clustering has been applied across diverse domains, including transportation, healthcare [17], environmental studies [18], geology [19], and industry. In practice, hierarchical clustering is employed for numerous purposes such as image segmentation [20], information retrieval, outlier detection, pattern recognition [19–21], and sentiment analysis [22]. Although divisive hierarchical clustering methods are valuable and have been applied in various domains, they are outside the scope of this work. Our contribution is restricted to the agglomerative family of methods, where linkage criteria play a central role in determining cluster formation. Although divisive hierarchical clustering methods are valuable and have been applied in various domains, they are outside the scope of this work. Our contribution is restricted to the agglomerative family of methods, where linkage criteria play a central role in determining cluster formation.

[23] proposed a heuristic clustering approach to study epidemic propagation within complex networks, which remains an important research area. [24] applied hierarchical clustering to solve a cost optimization problem. Hierarchical clustering has also been applied to study the spread of human diseases, the diffusion of rumors on social networks, and the propagation of computer viruses. Additionally, [25] provided a comparative analysis of various practical hierarchical clustering algorithms. [26] utilized hierarchical clustering to identify community structures within networks.

Pattern recognition extensively relies on density-based and hierarchical information [27]. Some studies focus on analyzing topological characteristics, including the identification of bi-communities and isolated nodes within networks, as overlapping community detection in bipartite networks is particularly important. [28] employed a hierarchical clustering structure to analyze noisy data. Hierarchical clustering offers advantages over other clustering techniques, such as density-based methods, which group data points according to spatial density and distribution patterns [29]. The performance of clustering can be evaluated in various ways. A notable advantage of hierarchical clustering is its ability to detect nested clusters, which most other clustering methods cannot achieve.

One commonly used evaluation measure in clustering is purity [30], which assigns each cluster the label of its most frequent class. Hierarchical clustering produces a dendrogram, a hierarchical tree that illustrates the sequential formation of clusters at different stages. The Rand index, another evaluation measure, calculates the proportion of correct decisions, ranging from 0 to 1. Dendrograms are also useful for detecting anomalies within clusters. The Davies–Bouldin index evaluates the separation between distinct clusters. Dendrograms additionally allow instances to be traversed using a depth-first search mechanism.

In hierarchical clustering, the process can be terminated at any stage. This eliminates the need to specify the number of clusters in advance, which is a significant advantage since determining the optimal cluster count is often challenging [32]. Although divisive hierarchical clustering methods are valuable and have been applied in various domains [33–35], they are outside the scope of this work. Our contribution is restricted to the agglomerative family of methods, where linkage criteria play a central role in determining cluster formation.

The most commonly used hierarchical clustering linkage methods include single, complete, average, mean, centroid, and Ward [36]. Each method has distinct advantages and disadvantages, making the selection of the most appropriate approach highly problem-dependent. In recent years, several innovative linkage methods have also been introduced in the literature. Examples include versatile linkage, privacy-preserving record linkage, min-max linkage [37], coalition link (c-link and gain link), gravitational merging coefficient linkage, shortest linkage, and ordered weighted averaging (OWA) linkage [38].

OWA operators have been widely employed in clustering and related machine learning tasks due to their flexible aggregation capability. For instance, [39] introduced an OWA-based linkage method for hierarchical clustering in phylogenetic trees. [10] applied OWA to aggregate fuzzy similarity relations in water treatment applications. OWA has also been incorporated into classification algorithms such as k-nearest neighbor, where [40,41] proposed OWA-based distance measures to improve accuracy. More recently, [16] applied an OWA-based hierarchical clustering approach to analyze user lifestyles, and [44] utilized OWA-based clustering for hotel segmentation. These studies confirm the versatility of OWA in clustering tasks. However, to the best of our knowledge, no prior work has integrated OWA into the k-centroid linkage method, which is the central contribution of this study.

3. Materials and method

3.1. Hierarchical clustering

Hierarchical clustering constructs a tree-like structure in which data are organized into successive layers of partitions. In this study, Euclidean distance is employed as the similarity measure, and pairwise distances between all data points are computed as the initial step. These values are stored in a distance matrix, which represents the pairwise proximities between data points. The distance matrix is then used to guide decisions about merging or splitting clusters. In agglomerative hierarchical clustering, the choice of linkage method is the most influential factor in determining which clusters are merged [45]. In this study, we adopt the agglomerative (bottom-up) strategy exclusively, as the

proposed OWA-based k-centroid linkage is inherently designed for merging clusters rather than recursively splitting them.

Hierarchical clustering is widely used for organizing data into meaningful structures by constructing a hierarchy of nested clusters [46]. Agglomerative hierarchical clustering (bottom-up) and divisive hierarchical clustering (top-down) are the two main approaches. Agglomerative hierarchical clustering begins with each data point as an individual cluster and successively merges the nearest clusters until all points are combined into a single cluster. In contrast, divisive hierarchical clustering begins with all data points in a single cluster and recursively splits them until each data point forms its own cluster.

This study focuses on the agglomerative approach. In the initialization step, each data point is treated as an individual cluster. Thus, with *n* data points, the process begins with *n* clusters. Next, pairwise distances are calculated, and the two closest clusters are merged according to the chosen linkage criterion. In this study, this modified version is referred to as the OWA-based k-centroid link. Then, the distance matrix is updated and the previous steps are repeated until there is only one cluster left or a predetermined stopping rule is satisfied.

3.2. Hierarchical clustering linkage method

Consider a given dataset *S*, represented as $S = \{x_1, x_2, \dots, x_n\}$. To analyze the data, a hierarchical clustering algorithm is employed. The goal is to identify *t* clusters, where *t* is a positive integer ranging from 1 to the total number of elements in S. These clusters denoted as $C = \{C_1, C_2, \dots, C_t\}$, must satisfy certain conditions. Specifically, the union of all clusters should equal to the original dataset S, and each cluster C_i within *C* must be non-empty and a subset of S. Moreover, no two clusters C_i and C_j should have any common elements, meaning their intersection should be empty. The algorithm constructs a hierarchical structure of nested groups denoted as $C^* = \{C^{(0)}, C^{(1)}, \dots, C^{(u)}\}$. Each C_i represents a partition of *S* at the j_{th} level of the hierarchy, consisting of v clusters: $C^{(j)} = \{C_1^{(j)}, C_2^{(j)}, \dots, C_v^{(j)}\}$. The algorithm begins with n clusters, where every data point in the dataset initially forms its own individual cluster. In each iteration, the algorithm calculates denoted as $D(C_i, C_j)$, between any two clusters C_i and C_j. These clusters may contain p and q objects, respectively. The distance $D(C_i, C_j)$ is determined by the dissimilarity measure $d(x_i, x_j)$ which quantifies the dissimilarity between the instance, x_i from the cluster Ci and the instance xj from the cluster C_j. The function $d: S \times S \rightarrow [0, \infty]$ represents a pairwise distance metric, which could be the Jaccard, Euclidean, or Manhattan distance, among others, providing a numerical value that reflects the dissimilarity between two objects in the datasets.

The distance metric d must be symmetric, i.e., $d(x_i, x_j) = d(x_j, x_i)$. Also, the distance between the same elements should be *0*, i.e. $d(x_i, x_j) = 0$. The smallest distance is determined and the clusters producing this distance are merged in every repetitive step. These operations are repeated until there is only one cluster is left. If no explicit stopping criterion is specified, the algorithm completes in at most $n-1$ iterations, since each iteration merges two clusters. The desired number of clusters can be obtained by slicing the hierarchy at a specific level.

Consider two groups of objects denoted as C_i and C_j. The following are some of the widely adopted linkage techniques.

We propose an algorithm designed to identify and refine clusters of majority-class instances. By removing observations from high-density regions, the approach minimizes information loss compared to removing individual or low-density instances.

The proposed solution combines the benefits of prior methods by systematically removing the nearest neighbors of each majority-class instance. The main idea is to ensure an even elimination of majority class samples while concentrating on the nearest objects.

Single Link: The equation (1) takes into account the smallest distance between instances within each cluster.

$$D_s(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} \{d(x_i, x_j)\} \tag{1}$$

Complete Link: The equation (2) takes into account the largest distance between instances within each cluster.

$$D_c(C_i, C_j) = \max_{x_i \in C_i, x_j \in C_j} \{d(x_i, x_j)\} \tag{2}$$

Average Link: Equation (3) accounts for the mean separation between objects within clusters, considering the average distance among all pairs.

$$D_a(C_i, C_j) = \frac{1}{|C_i|}\frac{1}{|C_j|}\sum_{x_i \in C_i}\sum_{x_j \in C_j} d(x_i, x_j) \tag{3}$$

where $|C_i|$ and $|C_j|$ are the number of data points in the clusters C_i and C_j respectively.

Centroid Link: Equation (4) considers the distance between the centroid of each cluster as a factor.

$$D_g(C_i, C_j) = d(\mu_i, \mu_j) = d\left(\left(\frac{1}{|C_i|}\sum_{x_i \in C_i} x_i\right), \left(\frac{1}{|C_j|}\sum_{x_j \in C_j} x_j\right)\right) \tag{4}$$

where μ_i and μ_j are the centroids (mean) of the clusters C_i and C_j respectively.

Mean Link: Equation (5) incorporates the inter-cluster pairwise distances, which involve considering the distances between a specific data point and the remaining data points within the merged clusters.

$$D_m(C_i, C_j) = \frac{|C_i \cup C_j|(|C_i \cup C_j| - 1)}{2}\sum_{x_i \in (C_i \cup C_j)}\sum_{x_j \in (C_i \cup C_j)} d(x_i, x_j) \tag{5}$$

Ward Method: Equation (6) takes into account the least variance distance between data points belonging to different clusters.

$$D_w(C_i, C_j) = \frac{|C_i||C_i|}{|C_i|+|C_i|} d\left(\left(\frac{1}{|C_i|}\sum_{x_i \in C_i} x_i\right), \left(\frac{1}{|C_j|}\sum_{x_j \in C_j} x_j\right)\right) \tag{6}$$

3.3. The Proposed Method, k-centroid Link Supported with OWA

This study enhances the performance of the k-centroid link model [17] by incorporating an OWA strategy after selecting the k nearest objects. The OWA distance approach has previously been applied in k-Nearest Neighbor methods [40-41]. Traditionally, the k-centroid link calculates the average pairwise distance between the k nearest instances and the centroid of each cluster. At each iteration, it identifies the k closest instances to the centroids within each cluster and evaluates up to *k x k* distances to determine the similarity between clusters. By considering multiple centroid-neighbor instances during merging, the k-centroid link combines characteristics of both average-link and centroid-link methods.

Definition 1. The closest centroid neighbors

In a given a cluster *C*, an object *o* within *C* is called the closest centroid neighbor, denoted as *N(C)*, if its distance to the cluster centroid *(μC)* is the smallest compared to the distances of all other objects in *C* to the centroid. In simpler terms, this means it satisfies the condition $d(o, \mu c) \leq d(o', \mu c)$ for all objects o in *C* excluding *o*.

Definition 2. K-closest centroid neighbors

Given a cluster C and a positive integer k (where $k \leq |C|$), the k-closest centroid neighbors, denoted $Nk(C)$, are the k data points closest to the centroid of C.

Definition 3. K-centroid link

When considering two clusters, C_i and C_j, along with a positive integer k, the k-centroid link refers to a method of measuring the distance between the clusters. This method calculates the distance by averaging the distances between all pairs of the k-nearest centroid neighbors from the two clusters. The computation is represented by Equation (7). In essence, the k-centroid link provides a way to determine the distance between clusters based on their respective k-closest centroid neighbors.

$$D_k(C_i, C_j) = \frac{1}{\min(|C_i|, k)} \frac{1}{\min(|C_j|, k)} \sum_{x_i \in N_k(C_i)} \sum_{x_j \in N_k(C_j)} d(x_i, x_j) \tag{7}$$

Algorithm 1 presents the pseudocode for the k-centroid link algorithm. Initially, every instance in the dataset is treated as a separate cluster containing a single element. The algorithm proceeds for $n-1$iterations, where n is the total number of instances. During every iteration, two clusters are combined to create a larger cluster, referred to as a super-cluster.

In each step:

1. All possible pairs of clusters (C_i, C_j) are evaluated to determine their similarity (closeness).
2. The center (centroid) of each cluster is computed. Initially, since each cluster has a single element, the element itself serves as the center.
3. The centers are dynamic and are recalculated when clusters are updated with new elements.

4. For each cluster, the distances between the center and all instances in the cluster are computed, identifying the k-nearest instances (k-closest centroid neighbors) to the center.

5. The algorithm calculates the mean of distances between each k-closest data points in c_i and the k-closest objects in c_j, labeling this value as the "distance" of these two clusters.

6. The clusters with the shortest distance between them are chosen and merged into a new cluster. The center of this newly formed cluster is then recalculated.

7. This merging operation continues iteratively while there are more than one cluster.

Algorithm 1 demonstrates the pseudocode k-centroid link algorithm. In the beginning, every data point is a member of one cluster. Each cluster includes only one element. The method executes a maximum of n−1 iterations, where n denotes the total number of instances in the dataset, as each iteration merges two clusters into a single cluster. At each step, all potential cluster pairs (c_i, c_j) are evaluated, and the most similar (nearest) clusters are combined to create a new super-cluster. This model calculates the coordinates of the center of every cluster. At first, every instance becomes center points because each instance is a cluster. These centers are not constant. They differ in case of the presence of new elements. The distances from the central point to every data point within the current cluster are computed, allowing the identification of the k nearest instances to the center. Afterward, the average pairwise distance between the k closest objects in the first cluster (c_i) and the k closest objects in the second cluster (c_j) is calculated, and this value is labeled as the distance of those two clusters. Clusters with the smallest distance between them are merged, and the center of the newly formed cluster is recalculated. This process repeats iteratively until all data points are merged into a single cluster.

Algorithm 1: Merging with k-Centroid Link

Inputs: $S = \{x_1, x_2, ..., x_n\}$, the data including n instances
k, the number of neighbors of a center point
Output: $C=\{C_1, C_2, ..., C_t\}$, the produced clusters

```
for i in range (1,n)
    Ci = {xi}
for m in range (1, n-1)
 min = ∞
    for all pairs Ci, Cj ϵ C for i ≠ j
        μi = FindCentroid(Ci)
        μj = FindCentroid(Cj)
        foreach object o in Ci
            dist1[o] = d(o, μi)
        foreach object o in Cj
            dist2[o] = d(o, μj)
        Nk(Ci) = arg min(dist1.sort())
                  k
        Nk(Cj) = arg min(dist2.sort())
                  k
        foreach object o1 in Nk(Ci)
            foreach object o2 in Nk(Cj)
                total = total + d(o1,o2)
```

average = total / (min(k, $|N_k(C_i)|$) * min(k, $|N_k(C_j)|$))
if (average < min)
min = average
store $C_u = C_i$ and $C_v = C_j$
$C_y = C_u U C_v$
$C = (C - \{C_u\} - \{C_v\}) U C_y$

The OWA operator is a parameterized aggregation operator that generalizes the minimum, maximum, and arithmetic mean, forming a versatile class of mean-type operators [43]. Frequently applied in various decision-making problems [39-43], the OWA operator was introduced by Yager to create a general aggregation framework encompassing the min, max, and arithmetic mean. It is defined by a weight vector of nonnegative values that sum to one [25]. It is important to note that the OWA (Ordered Weighted Averaging) operator is not used merely to adjust the parameter kkk. While the constant-level stress function influences kkk values, OWA aggregates multiple stress contributions by weighting them according to their relative importance. This allows the clustering process to adaptively consider variations across the dataset, resulting in more robust merging and splitting decisions. This approach differs from the methods described in [17] and [39], where stress adjustments are performed more uniformly without adaptive aggregation.

The OWA operator is typically implemented in three steps:
(a) Sorting the input arguments in descending order.
(b) Determining the weights for the OWA operator.
(c) Aggregating the sorted arguments using the OWA weights.

Numerous methods have been developed to compute OWA weights [40-45]. Among these, Yager's stress function method stands out for its ability to characterize the structure of the OWA operator. This method allows for consistent weight vector generation across varying numbers of arguments while maintaining interpretability of the results.

In this study, a constant-level stress function was utilized. This function assigns equal weights to all data points, reflecting a behavioral character of 0.5. The parameter k defines the height of the stress function.

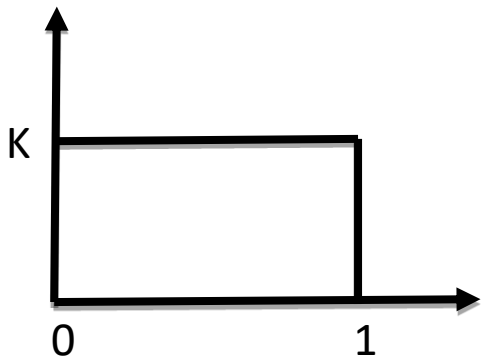

Figure 1. Constant level stress function

Figure 2 illustrates the constant-level stress function, which emphasizes weights of the lower-ranked values, assigning the smallest weight to the lowest input. *K* parameter is the value of height and α determines the highest threshold value of the data points. In this study, we have benefited from the constant level range stress function to determine OWA parameters and selected α parameter as 0.5.

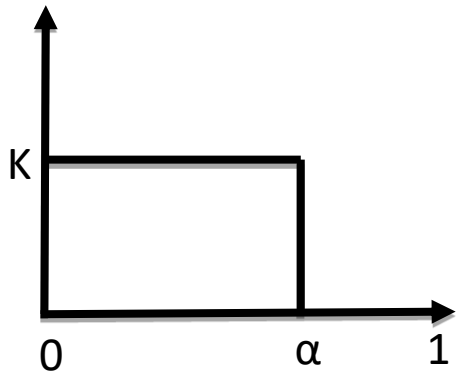

Figure 2. Constant level range stress function

In this study, the OWA-based k-centroid linkage was tested under nine configurations, with the α parameter varied from 0.1 to 0.9 in increments of 0.1. When α=1, the algorithm reduces to the standard k-centroid linkage, whereas at α=0, none of the k-nearest centroid neighbors are considered. Since the proposed linkage requires an additional input parameter (k), we compared multiple k values {1, 3, 5, 7, 9, √n} alongside traditional linkage methods. This setup ensures a comprehensive evaluation of the proposed approach under different parameter conditions.

4. Experimental research

Comprehensive experiments were conducted on 24 datasets to evaluate the clustering capability of the OWA-based k-centroid link. The model was compared with traditional linkage methods (Ward, single, mean, complete, centroid, average) as well as the standard k-centroid linkage method without OWA, focusing on clustering accuracy. Accuracy was measured using external cluster validation, which compares clustering results with predefined class labels

The proposed method was implemented in Java using the WEKA Agglomerative Hierarchical Clustering library [47]. Except for enabling Euclidean distance normalization, all other input parameters were left at their default values. In each experiment, the datasets were partitioned according to the number of clusters specified in their descriptions. The experiments focused on the k-centroid link model, with the parameter *k* set to the square root of the dataset size ($\sqrt{n}$).

4.1. Dataset description

The clustering experiments employed 24 publicly available datasets drawn from diverse domains, including healthcare and environmental sciences. These datasets were obtained from well-known repositories, including UCI (University of California at Irvine) and OpenML. Table 1 provides comprehensive details about the datasets, including the number of attributes, instances, clusters, and their respective domains.

Table 1. The overall attributes of the dataset

Dataset Name	**Number of Attributes**	**Number of Instances**	**Number of Clusters**	**Domain**
Colon32	32	62	2	Health
Wilt	5	4839	2	Environment
Wholesale Customers	7	440	3	Marketing
Tic Tac Toe	9	958	2	Game
Iris	4	150	3	Environment
Thyroid-newthyroid	5	215	3	Health
Haberman's Survival	3	306	2	Health
Breast Cancer	9	286	2	Health
Seismic-bumps	18	2584	2	Geology
Acute Inflammations	6	120	2	Health
Balloons-Yellow-Small+Adult-Stretch (BYSAS)	4	16	2	Psychology
Thoracic Surgery	16	470	2	Health
Hepatitis	20	155	2	Health
Balloons-Yellow-Small (BYS)	4	16	2	Psychology
Planning Relax	12	182	2	Health
Car	6	1728	4	Marketing
German Credit	20	1000	2	Banking
Thyroid-ann	21	3772	3	Health
Appendicitis	9	106	2	Health
Thyroid-sick-euthyroid	25	3163	2	Health
Blogger	5	100	2	Cyber Space
Balloons-Adult+Stretch (BAS)	4	16	2	Psychology
Blood Transfusion Service	5	748	2	Health
Zoo	18	101	7	Veterinary

4.2. Effect of the parameter

In this section, a compared evaluation is presented between OWA-based k-centroid link and the most popular merging criteria in agglomerative hierarchical clustering, including k-centroid link. Ward, single, mean, complete, centroid, and average linking are the examined linkage methods. The parameter k was set to the square root of the total number of data points in each dataset, as this choice generally produces clusters consistent with the dataset's size and structure. For the OWA stress function, the threshold parameter α was varied incrementally from 0.1 to 0.9 in steps of 0.1

The OWA-based k-centroid link method incorporates a user-defined parameter, α, which dictates the number of nearest objects to the centroid considered during clustering. This parameter is critical, as it determines how many central objects represent each cluster. Its flexibility allows researchers to adjust its value according to specific goals and requirements. This approach facilitates the selection of the subset of instances closest to the center from the k nearest objects. When α=1, all k nearest instances are considered,

whereas at α=0.1, only 10% of the k nearest instances are included. The selection is not random. The k nearest data points are first sorted by their distance to the cluster center, and then the closest instances up to the α threshold are chosen.

Table 2 reports the clustering accuracy (%) for different α values, ranging from 0.1 to 0.9 in increments of 0.1. Here, we examine the impact of α, as its value can influence clustering rate. When α is low, the selected subset may lack sufficient information to accurately represent cluster similarity. When α is high, the method tends to behave similarly to the standard k-centroid link, offering little to no improvement because it considers each distance between k pairs of instances of two distinct clusters. As a result, this parameter should be assigned to a reasonable value.

Table 2. The variations in clustering accuracy values corresponding to different α values

Dataset	**α=0,1**	**α=0,2**	**α=0,3**	**α=0,4**	**α=0,5**	**α=0,6**	**α=0,7**	**α=0,8**	**α=0,9**
Acute Inflammations	56.67	85	85	85	85	84.17	85	85	85
Appendicitis	78.31	78.31	78.31	78.31	88.31	78.31	78.31	78.31	78.31
Balloons-Adult+Stretch	70	70	50	50	70	50	60	50	55
Thyroid-sick-euthyroid	90.71	90.39	90.39	90.39	90.39	90.39	90.39	90.36	90.39
Balloons-Yellow-Small	70	60	50	50	60	50	50	60	50
Balloons-Yellow-Small+Adult-Stretch	68.75	68.75	56.25	56.25	68.75	56.25	56.25	56.25	56.25
Blogger	57	70	63	67	70	57	57	57	57
Blood Transfusion Service	76.48	76.48	76.48	76.48	76.48	76.48	76.84	76.74	76.74
Breast Cancer	69.59	70.63	69.59	69.59	70.63	70.63	70.63	69.59	69.59
Car	69.68	57.47	69.97	69.8	57.47	61.52	57.41	61.69	54.98
Colon32	67.75	72.59	66.13	72.59	72.59	72.59	69.36	69.36	69.36
German Credit	70	71.2	71.2	70.5	71.2	70.8	70.8	71.1	71.2
Haberman's Survival	73.53	73.86	73.86	73.86	73.86	73.53	73.86	73.53	73.86
Hepatitis	79.36	79.36	79.36	79.36	79.36	79.36	79.36	79.36	79.36
Iris	34.67	74	69.34	74.67	74	69.34	75.34	69.34	74
Planning Relax	71.43	70.88	70.88	70.88	70.88	70.88	70.88	70.88	70.88
Seismic-bumps	93.35	93.16	92.88	92.42	93.16	93.16	92.46	95.16	92.15
Thoracic Surgery	84.69	82.56	82.56	82.35	82.56	82.56	82.56	82.56	82.56
Thyroid-ann	92.37	92.26	92.4	92.4	92.26	92.32	92.32	92.37	92.37
Thyroid-newthyroid	70.7	72.1	70.7	70.7	72.1	70.7	73.96	73.96	72.56
Tic Tac Toe	65.14	60.55	65.14	64.51	60.55	64.72	50.53	61.17	58.56
Wholesale Customers	71.37	71.6	71.14	71.37	71.6	71.6	71.14	70.91	71.6
Wilt	94.57	94.59	94.59	94.59	94.59	94.59	94.59	94.59	94.59
Zoo	43.57	73.27	75.25	75.25	73.27	79.21	83.17	73.27	73.27

Table 2 shows that the optimal value of α varies across datasets. On average, the best clustering accuracy was obtained when α=0.3. Figure 3 demonstrates the number of experimented datasets where k-centroid supported with OWA approach for different α values produces greater and less clustering accuracies than normal k-centroid linkage. According to the results, when α is equal to 0.3, the clustering accuracy ratio became the best. The proposed method demonstrates a higher likelihood of effective clustering when α is set to 0.3 for the provided datasets. Except the condition that α = 0.1, k-centroid OWA surpasses the results obtained by normal k-centroid link for all tested stress functions. When α is equal to 0.3, k-centroid OWA produces better clustering accuracy for 13 datasets, worse clustering accuracy for 4 datasets and it produces the same performance with normal k-centroid linkage for 7 datasets out of 24 datasets in total. Figure 3 shows that the OWA-enhanced k-centroid method improved performance over the standard k-centroid linkage for at least 10 datasets across all tested α values.

4.3. Experimental results

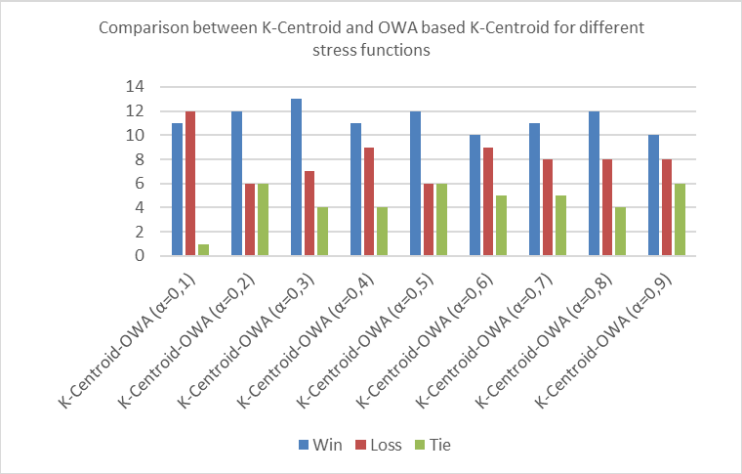

Figure 3. Comparison between K-Centroid and OWA based K-Centroid for different stress functions

Figure 3. presents a comparative analysis between the K-Centroid and OWA-based K-Centroid methods across different stress function parameters (α values). The vertical axis represents the number of datasets where the OWA-based K-Centroid outperforms (Win), underperforms (Loss), or results in a tie with the conventional K-Centroid approach.

Overall, the results reveal that the OWA-based k-centroid method consistently outperforms the standard k-centroid approach across different stress function settings. Specifically, the number of datasets where the OWA-based method achieves superior results (blue bars) remains consistently higher across different α values, peaking particularly around α = 0.3, α = 0.5, and α = 0.8. In contrast, the number of losses (red bars) remains moderate across all configurations, indicating that in most cases, the proposed approach offers advantages over the standard method. Additionally, the presence of ties (green bars) suggests that in some instances, the choice of α has minimal impact on clustering performance.

These findings confirm the effectiveness of integrating the OWA approach into k-centroid hierarchical clustering, highlighting its potential to enhance clustering performance under varying stress function conditions. The findings highlight that careful

selection of the α parameter is crucial for maximizing the benefits of this enhanced clustering methodology.

Table 3 displays the comparison results expressed as clustering accuracy ratios (in percentages). It is obvious that the output of the experiments by seem the k-centroid link supported with OWA better than most of the traditional linkage methods in the majority of the datasets. It also exceeds the clustering efficacy of original k-centroid link in 13 datasets out of all 24 datasets. The results painted with green under the k-centroid OWA column represents better accuracy than k-centroid. Likewise, red cells refer to worse accuracies and yellow cells refer to the results with the same performance compared to k-centroid OWA linkage. In the previous publication, k-centroid link was proved to perform much better than these traditional linkage types. In this study, experimental results show that k-centroid link can also be improved and more productive when OWA approaches are integrated.

Table 3. The comparison of the proposed approach with the existing approaches in terms of clustering accuracy (%)

Dataset	**Single**	**Complete**	**Average**	**Mean**	**Centroid**	**Ward**	***K*-Centroid**	**K-Centroid-OWA**
Acute Inflammations	50	91.67	84.17	84.17	84.17	84.17	84.17	85
Appendicitis	79.25	61.32	78.3	86.79	78.3	78.3	78.3	88.31
Balloons-Adult+Stretch	80	55	75	60	50	85	75	70
Balloons-Yellow-Small	70	80	80	60	80	80	80	70
Balloons-Yellow-Small+Adult-Stretch	56.25	50	68.75	68.75	50	68.75	68.75	68.75
Blogger	70	70	58	56	67	50	58	70
Blood Transfusion Service	76.74	76.74	76.74	72.59	76.74	57.35	76.74	76.84
Breast Cancer	70.63	55.59	69.93	53.15	70.63	66.43	69.24	70.63
Car	69.85	51.85	32.23	28.65	66.72	40.28	51.45	69.97
Colon32	66.13	83.87	66.13	82.26	66.13	83.87	72.59	72.59
German Credit	70.1	70.6	70.5	67.2	70.5	66	71.2	71.2
Haberman's Survival	73.86	73.53	73.53	54.9	73.86	56.86	73.86	73.86
Hepatitis	79.35	78.71	79.35	69.03	79.35	78.71	79.35	79.36
Iris	68	84	90.67	84	90.67	90	90.67	75.34
Planning Relax	70.88	64.84	70.88	54.4	70.88	61.54	70.88	71.43
Seismic-bumps	93.38	92.18	93.15	92.14	91.87	88.08	92.46	93.35
Thoracic Surgery	82.55	82.55	82.55	62.13	82.55	63.19	82.55	84.69
Thyroid-ann	92.42	88.23	92.24	54.83	92.45	40.43	92.32	92.4
Thyroid-newthyroid	70.23	71.63	73.95	46.51	72.09	69.3	72.56	73.96
Thyroid-sick-euthyroid	90.68	90.2	90.68	73.61	90.36	82.52	90.08	90.71
Tic Tac Toe	65.14	68.06	50.73	56.47	65.34	59.29	60.23	65.14
Wholesale Customers	71.36	70.23	70.91	47.27	70	36.14	70.91	71.6
Wilt	94.59	94.59	94.59	58.28	94.59	50.64	94.59	94.59
Zoo	67.33	75.25	75.25	77.23	57.43	64.36	72.28	83.17

We applied the Friedman test across the 24 datasets to evaluate the statistical significance of the performance differences among the eight clustering methods. The test yielded Q=20.79 with 7 degrees of freedom and a p-value of 0.0041, indicating a statistically significant difference among methods. The average ranks show that the proposed OWA-based k-centroid link achieved the best performance (average rank = 3.29), followed by the standard k-centroid link (average rank = 3.50). These findings confirm that the improvements of the proposed method are not only consistent across datasets but also statistically significant.

Table 3. Friedman test results

Method	**Avg. Rank**
K-centroid-OWA	3.29
K-centroid	3.50
Single	3.96
Average	4.46
Complete	4.75
Centroid	5.04
Ward	5.29
Mean	5.71

In addition to accuracy, we evaluated the computational cost of the proposed OWA-based k-centroid link. For each dataset, we measured the average runtime (seconds) and during clustering. Table 4 summarizes the results. While the OWA-based k-centroid requires additional computations to apply the OWA weights to the selected k nearest objects, the overhead is moderate. Across all datasets, the average runtime was 1.08 times that of the standard k-centroid method. This indicates that the proposed method offers significantly better accuracy at a reasonable computational cost, making it suitable for practical use in large-scale clustering tasks.

Table 4. Running time comparison (in seconds)

Datasets	**Single**	**Complete**	**Average**	**Mean**	**Centroid**	**Ward**	**K-Centroid**	**K-Centroid-OWA**
Colon32	0.0359	0.0371	0.0368	0.0362	0.038	0.0368	0.0392	0.0423
Wilt	5.867	5.7428	5.8193	6.0417	5.8544	5.7584	6.259	6.7305
Wholesale Customers	0.3793	0.3889	0.3909	0.3924	0.392	0.3858	0.4004	0.436
Tic Tac Toe	0.9274	0.9266	0.9715	0.9617	0.9492	0.9344	1.0032	1.0608
Iris	0.1072	0.1082	0.1109	0.1075	0.1106	0.1101	0.1143	0.1245
Thyroid-newthyroid	0.1667	0.1687	0.1667	0.1642	0.1647	0.1669	0.1762	0.1883
Haberman's Survival	0.2488	0.2582	0.2579	0.2557	0.2516	0.2505	0.2639	0.2868
Breast Cancer	0.2354	0.2297	0.2352	0.2299	0.2286	0.2334	0.2485	0.2654
Seismic-bumps	2.9157	2.9727	2.9692	2.9004	2.9684	2.9548	3.0753	3.3332
Acute Inflammations	0.0828	0.0855	0.0831	0.0827	0.0838	0.0836	0.0868	0.0952
Balloons-Yellow-Small+Adult-Stretch (BYSAS)	0.0065	0.0064	0.0065	0.0063	0.0064	0.0066	0.0067	0.0072
Thoracic Surgery	0.4211	0.414	0.4153	0.414	0.4221	0.4157	0.4416	0.472
Hepatitis	0.1141	0.1106	0.1127	0.1151	0.1113	0.1119	0.1169	0.1295

Balloons-Yellow-Small (BYS)	0.0065	0.0064	0.0064	0.0064	0.0063	0.0064	0.0068	0.0073
Planning Relax	0.1348	0.1337	0.1346	0.138	0.1376	0.139	0.145	0.1541
Car	1.8164	1.865	1.9014	1.8418	1.8609	1.8731	1.9358	2.1043
German Credit	0.987	0.9966	0.9823	0.9907	0.9803	1.0009	1.0584	1.144
Thyroid-ann	4.519	4.5327	4.5101	4.4375	4.4703	4.4223	4.685	5.0505
Appendicitis	0.0724	0.0704	0.0719	0.0709	0.0716	0.0706	0.0757	0.0799
Thyroid-sick-euthyroid	3.6574	3.66	3.7215	3.667	3.6931	3.7137	3.8281	4.1873
Blogger	0.0678	0.0653	0.0651	0.0673	0.0644	0.0672	0.069	0.0744
Balloons-Adult+Stretch (BAS)	0.0063	0.0063	0.0064	0.0063	0.0063	0.0065	0.0067	0.0073
Blood Transfusion Service	0.6973	0.7355	0.6905	0.7207	0.7247	0.7171	0.7522	0.7973
Zoo	0.0679	0.0683	0.0682	0.0665	0.0683	0.0661	0.0701	0.076

Visualizations were generated to demonstrate the clustering behavior of the proposed method to complement the numerical results. Figure 4 shows dendrograms obtained from the Iris dataset using the k-centroid linkage and the OWA-based k-centroid linkage.

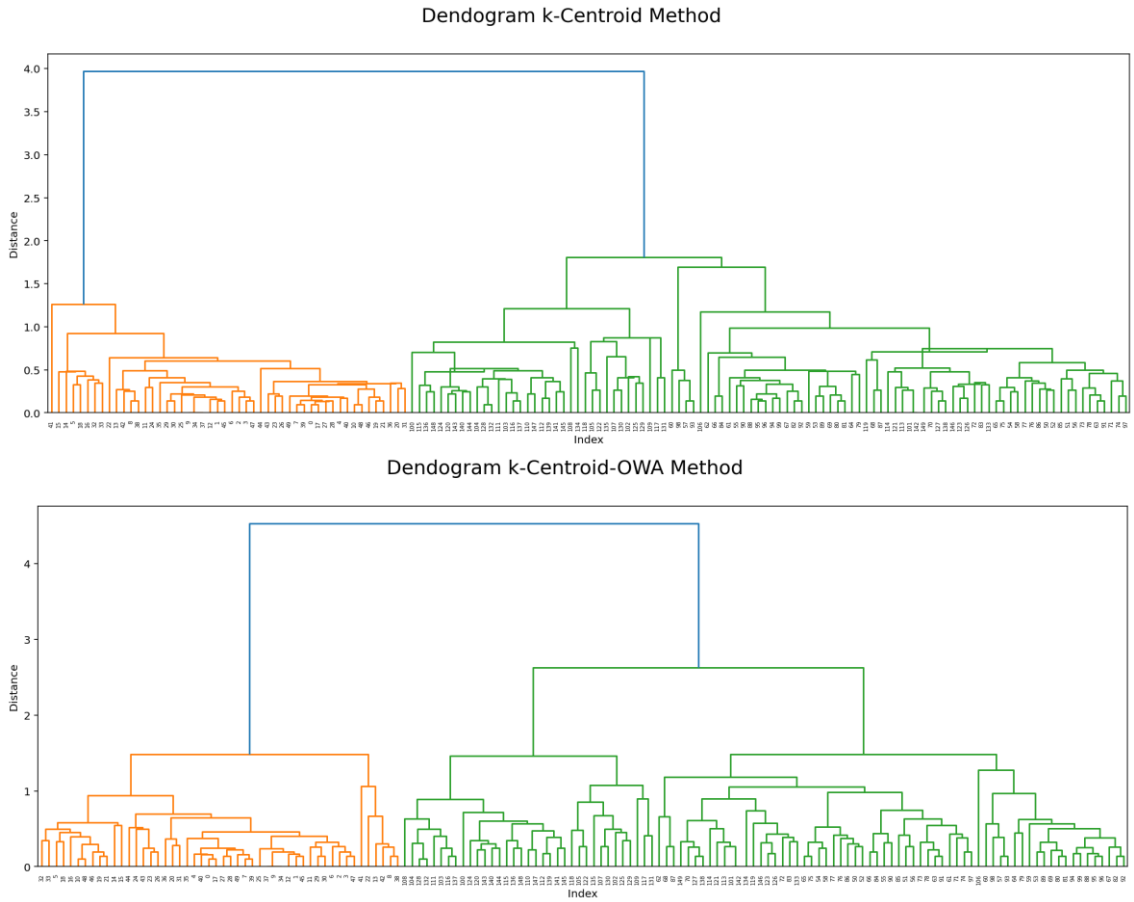

Figure 4. Dendogram comparison between K-Centroid and OWA based K-Centroid linkage methods for Iris dataset

In order to evaluate the performance of the proposed approach more comprehensively, we conducted additional experiments by comparing it with a variety of well-known clustering algorithms on the same benchmark datasets. KMeans is a partitioning-based method that minimizes within-cluster variance, while Gaussian Mixture Models (GMM) assume clusters follow Gaussian distributions. DBSCAN, on the other hand, is a density-based method that can discover clusters of arbitrary shapes, and DIANA is a divisive hierarchical clustering algorithm that recursively partitions data from top to bottom.

Table 5. The comparison of the proposed approach with different kinds of clustering approaches apart from agglomerative hierarchical clustering in terms of clustering accuracy (%)

Datasets	**KMeans**	**GMM**	**DBSCAN**	**DIANA**	**K-Centroid-OWA**
Acute Inflammations	85	83	80	82	85
Appendicitis	88.14	85.37	82.63	84.48	88.31
Balloons-Adult+Stretch	90	89	87	68	70
Balloons-Yellow-Small	92	91	89	80	70
Balloons-Yellow-Small+Adult-Stretch	93.50	92.25	90.50	81.25	68.75
Blogger	70	68	65	67	70
Blood Transfusion Service	85.26	84.72	82.35	73.81	76.84
Breast Cancer	76.38	75.54	73.21	64.92	70.63
Car	75.43	73.38	70.26	72.86	69.97
Colon32	72.03	70.28	73.63	69.41	72.59
German Credit	90	88	85	67	71.2
Haberman's Survival	71.38	69.92	66.76	68.26	73.86
Hepatitis	89.11	87.59	84.72	76.44	79.36
Iris	73.92	72.44	65.19	71.38	75.34
Planning Relax	82.06	80.76	77.36	79.14	71.43
Seismic-bumps	84.13	82.17	79.55	81.67	93.35
Thoracic Surgery	88.14	86.63	83.61	85.38	84.69
Thyroid-ann	91	89	87	88	92.4
Thyroid-newthyroid	92.14	90.86	88.36	89.75	73.96
Thyroid-sick-euthyroid	93.44	91.12	89.63	90.23	90.71
Tic Tac Toe	65.19	63.71	60.85	62.16	65.14
Wholesale Customers	80.33	78.24	75.79	77.42	71.6
Wilt	82.41	80.59	77.23	79.67	94.59

Table 5 presents the clustering accuracy results of the proposed K-Centroid-OWA method against these algorithms. The comparison reveals several important observations. First, our approach achieves competitive or superior performance in many datasets, particularly in cases such as Seismic-bumps and Wilt, where it significantly outperforms all other methods. Second, in datasets like Haberman's Survival, Iris, and Thyroid-ann, the proposed method shows clear improvements over classical approaches, demonstrating its robustness across different domains. However, in a few datasets such as the Balloons variants and Planning Relax, traditional algorithms (e.g., KMeans and GMM) slightly outperform K-Centroid-OWA, suggesting that data characteristics may influence which method is more suitable.

Overall, these results indicate that the proposed method provides a strong and versatile alternative to existing clustering algorithms, showing both competitive accuracy and adaptability across a wide range of real-world datasets.

5. Conclusion and future work

Agglomerative hierarchical clustering operates by progressively merging the most similar clusters into larger ones, with cluster similarity determined by a linkage method that measures inter-group distances. The selection of a linkage method plays a crucial role in the quality and performance of the clustering process. Commonly used linkage methods include single, complete, average, mean, centroid, and Ward's method. However, traditional approaches can occasionally yield suboptimal results, such as elongated chain-like or overly compact globular clusters. Additionally, their inability to account for the influence of objects surrounding cluster centers often leads to suboptimal clustering results. To address these limitations, this study introduces a new cluster merging criterion model: the OWA-based k-centroid linkage.

The OWA-based k-centroid linkage method calculates the mean of pairwise distances among a chosen set of k objects closest to the centers of the clusters being merged. The proposed method was tested through experimental evaluations on 24 publicly available benchmark datasets tailored for clustering tasks. A constant-level range stress function was employed to determine the group of nearest instances among the k closest objects to the cluster centers. This weighting mechanism, derived from the OWA approach, enhanced clustering performance without adding computational overhead. Various upper threshold values were tested to identify the optimal parameter for selecting the closest instances to the cluster centers. Additionally, numerous numerical experiments confirmed the efficacy of the proposed method. Future research could explore alternative stress functions beyond the constant-level range approach to further improve the hierarchical clustering capabilities of the k-centroid linkage method.

Although the proposed OWA-based k-centroid linkage does not directly provide a solution to the problems of defining a stopping criterion or automatically determining the optimal number of clusters, its enhanced clustering performance may reduce the negative effects of these unresolved issues. Future research could investigate the integration of this approach with adaptive stopping rules and model selection techniques, enabling the method to jointly improve both clustering quality and parameter determination.

References

[1] Jyothi, B., Lingamgunta, S., and Eluri, S. Intelligent deep learning-based hierarchical clustering for unstructured text data, **Concurrency and Computation: Practice and Experience**, 34, Article e7388, (2022).

[2] Ghasemkhani, B., Yilmaz, R., and Kut, A., Birant, D. Logistic Model Tree Forest for Steel Plates Faults Prediction, **Machines**, 11(7), 679, (2023).

[3] Senthilnath, J., Shreyas, P., Ritwik, R., Suresh, S., Sushant, K., and Benediktsson, J. Hierarchical clustering approaches for flood assessment using multi-sensor satellite images, **International Journal of Image and Data Fusion**, 10(1): 28-44, (2019).

[4] Alberto, F., and Sergio, G. Versatile Linkage: a family of space-conserving strategies for agglomerative hierarchical clustering, **Journal of Classification**, 37: 584-597, (2019).

[5] Jaroonchokanan, N., Termsaithong, T., and Suwanna, S. Dynamics of hierarchical clustering in stocks market during financial crises, **Physica A: Statistical Mechanics and Its Applications**, 607: 128183, (2022).

[6] Zhong, C., Wang, H., and Yang, Q. Hydrochemical interpretation of groundwater in Yinchuan basin using self-organizing maps and hierarchical clustering, **Chemosphere**, 309: 136787, (2022).

[7] Nasibov, E. A robust algorithm for solution of the fuzzy clustering problem on the basis of the fuzzy joint points method, **Cybernetics and System Analysis**, 44(1): 7–17, (2008).

[8] Atilgan, C., and Nasibov, E. A space-efficient minimum spanning tree approach to the fuzzy joint points clustering algorithm, **IEEE Transactions on Fuzzy Systems**, 27(6): 1317-1322, (2019).

[9] Nasibov, E., and Ulutagay, G. A new unsupervised approach for fuzzy clustering, **Fuzzy Sets and Systems**, 158(19): 2118-2133, (2007).

[10] Su, P., Shen, Q., Chen, T., and Shang, C. Ordered weighted aggregation of fuzzy similarity relations and its application to detecting water treatment plant malfunction, **Engineering Applications of Artificial Intelligence**, 66: 17-29, (2017).

[11] Nasibov, E., Atilgan, C., Berberler, M., and Nasiboglu, R. Fuzzy joint points-based clustering algorithms for large datasets, **Fuzzy Sets and Systems**, 270: 111-126, (2015).

[12] Tian, P., Shen, H., and Abolfathi, A. Towards efficient ensemble hierarchical clustering with MapReduce based clusters clustering technique and the innovative similarity criterion, **Journal of Grid Computing**, 20(4): 34, (2022).

[13] Gao, Y., Fang, H., Ni, K., and Feng, Y. Water clusters and density fluctuations in liquid water based on extended hierarchical clustering methods, **Scientific Reports**, 12(1): 8036, (2022).

[14] Maldonado, S., Saltos, R., Vairetti, C., and Delpiano, J. Mitigating the effect of dataset shift in clustering, **Pattern Recognition**, 134: 109058, (2023).

[15] Tang, C., Tsai, M., Chuang, S., Cheng, J., and Wang, W. Shortest-linkage-based parallel hierarchical clustering on main-belt moving objects of the solar system, **Future Generation Computer Systems**, 34: 26-46, (2014).

[16] Nguyen, J., Armisen, A., Sanchez-Hernandez, G., Casabayo, M., and Agell, N. An OWA-based hierarchical clustering approach to understanding users' lifestyles, **Knowledge-Based Systems**, 190: 105308, (2020).

[17] Dogan, A., and Birant, D. K-Centroid Link: A novel hierarchical clustering linkage method, **Applied Intelligence**, 52: 5537-5560, (2021).

[18] Ashton, J., Borca, F., Mossotto, E., Hang, T., Ennis, S., and Beattie, R. Analysis and hierarchical clustering of blood results before diagnosis in pediatric inflammatory bowel disease, **Inflammatory Bowel Diseases**, 26(3): 469-475, (2020).

[19] Zhou, N., Xie, B., and Wang, T. Aggregation of individual feature-based similarities and application to hierarchical clustering, **Chemical Engineering Transactions**, 46: 217-222, (2015).

[20] Unglert, K., Radic, V., and Jellinek, A. Principal component analysis vs. self-organizing maps combined with hierarchical clustering for pattern recognition in

volcano seismic spectra, **Journal of Volcanology and Geothermal Research**, 320: 58–74, (2016).

[21] Saltos, R., and Weber, R. Rough fuzzy support vector clustering with OWA operators, **Inteligencia Artificial**, 25(69): 42–56, (2022).

[22] Fooladi, M., Golmohammadi, M., Rahimi, I., Safavi, H., and Nikoo, M. Assessing the changeability of precipitation patterns using multiple remote sensory data and an efficient uncertainty method over different climate regions of Iran, **Expert Systems with Applications**, 221: 119788, (2023).

[23] Chen, M., Feng, Y., and Kong, M. Optimization of the construction plan of assembled concrete structures in courtyard buildings based on grey clustering, **Electronic Journal of Structural Engineering**, 22(3): 38–46, (2022).

[24] Mokarram, M., Pham, T., and Khooban, M. A hybrid GIS-MCDM approach for multi-level risk assessment and corresponding effective criteria in optimal solar power plant, **Environmental Science and Pollution Research International**, 29(56): 84661–84674, (2022).

[25] Dasgupta, I., and Griffiths, T. Clustering and the efficient use of cognitive resources, **Journal of Mathematical Psychology**, 109: 102675, (2022).

[26] Yang, K., Shu, L., and Yang, G. Complex intuitionistic fuzzy ordered weighted distance measure, **Computational and Applied Mathematics**, 41: 353, (2022).

[27] Son, C., Cho, S., and Yoo, J. Volume traffic anomaly detection using hierarchical clustering, **Proceedings of the 12th Asia-Pacific Network Operations and Management Symposium**, 23–25, (2009).

[28] Farinelli, A., Bicego, M., Ramchurn, S., and Zucchelli, M. C-Link: A hierarchical clustering approach to large-scale near-optimal coalition formation, **Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)**, 106–111, (2013).

[29] Hussain, W., Merigó, J., Gil-Lafuente, J., and Geo, H. Complex nonlinear neural network prediction with IOWA network, **Soft Computing**, 27: 4853–4863, (2023).

[30] Suresh, H., and Raj, S. A fuzzy-based hybrid hierarchical clustering model for Twitter sentiment analysis, **Communications in Computer and Information Science**, 776: 384–397, (2017).

[31] Jayasundara, P., Perera, S., Malavige, G., and Jayasinghe, S. Mathematical modelling and a systems science approach to describe the role of cytokines in the evolution of severe dengue, **BMC Systems Biology**, 11(1): 34, (2017).

[32] De La Hoz, E., Lopez-Carmona, M., Klein, M., and Ivan, M. Alternative social welfare definitions for multiparty negotiation protocols, **Studies in Computational Intelligence**, 535: 23–41, (2014).

[33] Huo, X., Xue, H., and Jiao, L. Risk management of retrofit projects in old residential areas under green development, **Energy and Buildings**, 279: 112708, (2023).

[34] Park, H., Kwon, K., Khiati, A., Lee, J., and Chung, I. Agglomerative hierarchical clustering for information retrieval using latent semantic index, **Proceedings of IEEE International Conference on Smart City/SocialCom/SustainCom**, (2015).

[35] Li, Y., Zhao, K., and Zhang, F. Identification of key influencing factors to Chinese coal power enterprises transition in the context of carbon neutrality: A modified fuzzy DEMATEL approach, **Energy**, 263: 125427, (2023).

[36] Roux, M. A comparative study of divisive and agglomerative hierarchical clustering algorithms, **Journal of Classification**, 35: 345–366, (2018).

[37] Patnaik, A., Bhuyan, P., and Krishna Rao, K. Divisive Analysis (DIANA) of hierarchical clustering and GPS data for level of service criteria of urban streets, **Alexandria Engineering Journal**, 55(1): 407–418, (2016).

[38] Bien, J., and Tibshirani, R. Hierarchical clustering with prototypes via Minimax Linkage, **Journal of the American Statistical Association**, 106(495): 1075–1084, (2011).

[39] Nasibov, E., and Cavas, C. OWA-based linkage method in hierarchical clustering: Application on phylogenetic trees, **Expert Systems with Applications**, 38(10): 12684–12690, (2011).

[40] Ulutagay, G., and Nasibov, E. OWA aggregation based CxK-nearest neighbor classification algorithm, **Proceedings of the 6th IEEE International Conference on Intelligent Systems**, 219–224, (2012).

[41] Ulutagay, G., and Nasibov, E. C × K-Nearest neighbor classification with ordered weighted averaging distance, **Studies in Computational Intelligence**, 586: 95–102, (2016).

[42] Shu, Z., Carrasco-Gonzalez, R., Garcia-Miguel, J., and Sanchez-Montanes, M. Clustering using ordered weighted averaging operator and 2-tuple linguistic model for hotel segmentation: The case of TripAdvisor, **Expert Systems with Applications**, 213: 118922, (2023).

[43] Zabeo, A., Basei, G., Tsiliki, G., Peijnenburg, W., and Hristozov, D. Ordered Weighted based grouping of nanomaterials with Arsinh and dose response similarity models, **NanoImpact**, 25: 100370, (2022).

[44] Nietto, P., and Nicoletti, M. Case studies in divisive hierarchical clustering, **International Journal of Innovative Computing and Applications**, 8(2): 102–112, (2017).

[45] Lall, U., and Sharma, A. A nearest neighbor bootstrap for resampling hydrologic time series, **Water Resources Research**, 32(3): 679–693, (1996).

[46] Nasiboglu, R., Tezel, B., and Nasibov, E. Learning the stress function pattern of ordered weighted average aggregation using DBSCAN clustering, **International Journal of Intelligent Systems**, 34: 477–492, (2019).

[47] Witten, I., Frank, E., Hall, M., and Pal, C. **Data Mining: Practical Machine Learning Tools and Techniques**, 4th ed., Morgan Kaufmann, (2016).