5-16-2024

High-Dimensional Data Analysis Using Parameter Free Algorithm Data Point Positioning Analysis

S. M. F. D. Syed Mustapha
Zayed University

applied sciences

MDPI

Article

High-Dimensional Data Analysis Using Parameter Free Algorithm Data Point Positioning Analysis

S. M. F. D. Syed Mustapha

College of Technological Innovation, Zayed University, Dubai P.O. Box 19282, United Arab Emirates; syed.duani@zu.ac.ae

Abstract: Clustering is an effective statistical data analysis technique; it has several applications, including data mining, pattern recognition, image analysis, bioinformatics, and machine learning. Clustering helps to partition data into groups of objects with distinct characteristics. Most of the methods for clustering use manually selected parameters to find the clusters from the dataset. Consequently, it can be very challenging and time-consuming to extract the optimal parameters for clustering a dataset. Moreover, some clustering methods are inadequate for locating clusters in high-dimensional data. To address these concerns systematically, this paper introduces a novel selection-free clustering technique named data point positioning analysis (DPPA). The proposed method is straightforward since it calculates 1-NN and Max-NN by analyzing the data point placements without the requirement of an initial manual parameter assignment. This method is validated using two well-known publicly available datasets used in several clustering algorithms. To compare the performance of the proposed method, this study also investigated four popular clustering algorithms (DBSCAN, affinity propagation, Mean Shift, and K-means), where the proposed method provides higher performance in finding the cluster without using any manually selected parameters. The experimental finding demonstrated that the proposed DPPA algorithm is less time-consuming compared to the existing traditional methods and achieves higher performance without using any manually selected parameters.

Keywords: clustering; parameter-free algorithm; unsupervised learning; data mining; DPPA

Citation: Syed Mustapha, S.M.F.D. High-Dimensional Data Analysis Using Parameter Free Algorithm Data Point Positioning Analysis. *Appl. Sci.* **2024**, *14*, 4231. https://doi.org/10.3390/app14104231

Academic Editors: João M. F. Rodrigues and Giacomo Fiumara

Received: 5 March 2023
Revised: 16 July 2023
Accepted: 21 July 2023
Published: 16 May 2024

1. Introduction

Clustering has become a useful technique in many fields, including science, engineering, medicine, and our everyday lives. The term "clustering" refers to the process of grouping a collection of elements so that they are more comparable to other elements in their own cluster (group) than those in other clusters. A clustering theory needs to provide a unified framework for various approaches. Firstly, it should have foundations and limits of applicability. Secondly, it has to show connections between ideas and methods inside and outside of clustering. Finally, it should offer a cornerstone for resolving the emerging challenges in data processing and developing applications.

In the last few decades, clustering techniques for data mining have come a long way. Researchers have developed several methods [1–3] to effectively cluster data. Despite some successful clustering method applications, there are still some challenges in data mining. For instance, most of the conventional clustering approaches function well with low-dimensional data but encounter difficulties with high-dimensional data. The term "dimensionality curse" refers to this phenomenon [4]. Moreover, the overfitting problem occurs due to the high dimensionality of the feature space, and the model being recorded is too complicated [5]. Furthermore, the available data could not accurately reflect the whole ground-truth model. When this occurs, learning algorithms tend to fit a model to the available data samples, overlooking the underlying structure. In other words, memorizing takes the place of learning in some way. Tasks involving practical signal processing and

learning must also contain noise and outliers. Non-Gaussian noise is particularly prevalent in applications involving measurements. On the other hand, outliers are incongruous observations with the overall population. These idiosyncrasies provide significant hurdles for issues involving both linear and non-linear systems, especially when coupled with specific output requirements. Effective investigations into the consequences of these extra concerns are found in several recent works, e.g., [6,7]. Moreover, the raw data are often in crude formats. Clustering techniques need a preprocessing phase to deal with high dimensions and undesirable sampling problems. In circumstances involving large dimensions, preprocessing strategies [8–10] have been proposed to improve performance. To observe the dataset more precisely and make it more suited for subsequent processing, these methods often restructure the sample space using transformations or eliminations. For example, the principal component analysis (PCA) converts sample characteristics into a form with the largest variance, making it more appropriate for classification tasks with the added bonus of decreased dimensions [11]. This idea may be immediately extended as feature transformation in general. It is crucial to remember that these strategies keep all traits and restructure them using (non-linear) combinations rather than discard unimportant ones. The expanded method of dimensionality reduction by feature selection eliminates unimportant features and considers the subset of features that seems to be the most crucial while adhering to predetermined optimality criteria. Future classification or clustering problems may benefit from using any of these strategies to provide more accurate representations.

Despite the fact that feature selection may be utilized as an efficient solution to high-dimensional challenges, the elimination process may result in a loss of critical information with great significance in diverse contexts. After considering the difficulties of handling high-dimensional data, it is vital to list a few methods for avoiding overfitting. To address the issue of high-dimensional datasets, researchers have proposed various techniques. For instance, Yan et al. [12] presented a novel supervised multi-view hash model that utilized deep learning and a multi-view stability evaluation method to enhance hash learning. The approach included multi-data fusion methods and a memory network to reduce computing resources during retrieval. The proposed method outperformed state-of-the-art single-view and multi-view hashing methods on three datasets, showing its potential for improving hash learning. In [13], the authors proposed a group-based nuclear norm and learning graph (GNNLG) model for image restoration, which exploits patch similarity and low-rank properties. An optimized learning strategy was developed to impose smoothing priors on the denoised depth image. The authors used the alternating direction method of multipliers to improve speed and convergence. Experimental results showed that the proposed method outperformed state-of-the-art denoising methods. In Reference [14], the authors proposed a task-adaptive attention module for image captioning that learned implicit non-visual clues and alleviated misleading problems. The authors also introduced diversity regularization to enhance the expression ability of the module. The experiments on the MSCOCO captioning dataset showed that using the proposed module with a vanilla Transformer-based image captioning model improved performance. The authors of [15] proposed a precise NR IQA scheme with two steps: distortion identification and targeted quality evaluation. They used Inception-ResNet-v2 to train a classifier and designed a specific approach to quantify image distortion. The method outperformed state-of-the-art NR IQA methods in the experiments, indicating its potential for practical applications. A popular strategy for dealing with overfitting in clustering is to reduce within-cluster variation. In the case of clustering, a common way to deal with overfitting is to minimize within-cluster variance [16].

The authors of [17] proposed a technique where they merged DBSCAN with affinity propagation (APSCAN), enabling users to benefit from both methods. The density contains two pieces of data: radius and number. Radius is the average distance from the exemplar to all points, and the number is the number of points that are located inside the radius neighborhood. Both parameters—radius and number—have been employed in the strategy as an alternative to Eps and MinPts. The evolutionary parameter-free clustering algorithm

(EPFC) was developed by Ding and Li [18]; the average value of nearest neighbor points is calculated based on all datasets. Depending on the computed distance between each data point and its closest neighbor, one of two decisions will be determined, i.e., to divide the group or combine the data points. In [19], a parameter-free approach was described, where the radius was determined automatically based on the window size and data distribution.

It may be more efficient to sample a group of solutions rather than search for the solution space entirely when faced with all obstacles at once, including high dimensionality, high nonlinearity, parameter interaction, and disturbances due to randomness. Most of the previously investigated clustering methods for data mining used manual parameter selection to find the cluster from the raw data. Moreover, the methods used in high-dimension datasets provided comparatively lower performances. To address this concern, this study developed a novel clustering method (DPPA) to find the cluster from the raw dataset. The proposed method has been tested using the well-reputed publicly available dataset.

The key contributions of this paper are as follows:

1. In this study, we propose a new non-parametric clustering algorithm (DPPA), which can calculate 1-NN and Max-NN by analyzing the positions of data in the dataset without any initial manual parameter assignment.
2. We use two well-known publicly available datasets to evaluate the proposed method to find clusters. In addition, the proposed method is not time-consuming because it reduces the dependence of analysis on the selection of artificial parameters. Finally, four popular algorithms (DBSCAN algorithm, K-means algorithm, affinity propagation algorithm, and Mean Shift algorithm) are implemented to compare the performance of the proposed model.

The rest of the manuscript is arranged as follows: an overview of existing clustering algorithms is presented in Section 2. A detailed description of the proposed algorithm is described in Section 3. Section 4 describes details about the dataset. The experimental performance is described in Section 5. Section 6 presents the conclusion of the present study.

2. Related Work

Numerous clustering algorithms have been introduced in past years, as evidenced by the literature [20,21]. For instance, in Reference [22], a method utilizing association rules is proposed to cluster customer transactions within a market database. The study conducted in [23] focused on exploring algorithms for clustering categorical databases using non-linear dynamical systems.

According to Rokach [24], the process of clustering splits data patterns into subsets in such a manner that patterns with comparable characteristics are grouped in the same cluster. Therefore, the patterns are organized into a well-formed assessment, which identifies the population from which the sample was selected. For clustering, the data, K-means algorithm was implemented in [25,26]. Likas et al. [27] presented a global k-means algorithm, an incremental concept to clustering that dynamically adds one cluster center at a time using a deterministic global search technique, consisting of the sizes of the dataset executions of the k-means algorithm from suitable initial positions. The authors of [28] presented an affinity propagation clustering technique for pattern recognition. In their approach, each data point exchanges signals (called responsibility and availability) in an iterative manner to identify acceptable examples. To ascertain the affinities of the exemplars in the network, certain preferences for criteria must be established. The success of this algorithm relies on well-defined criteria to attain its ideal aim, even if there are no beginning parameters that must be set. Its inability to handle groups of arbitrary forms is another issue.

The DNo, DNg, and DNs indices have been recognized as simple internal validity measures for clustering [29–31]. These indices assess the relationship between cluster size and inter-cluster distance. Specifically, they calculate the ratio of the minimum distance between two clusters to the size of the largest cluster, aiming to maximize the index values. The DB index, proposed by Davies and Bouldin [32], quantifies the average similarity between each cluster and its most similar one. It aims to maximize the distances between

clusters while minimizing the distance between the cluster centroid and other data objects. The silhouette value, introduced by Rousseeuw [33], measures the similarity of an object to its own cluster (cohesion) compared to other clusters (separation). Ranging from −1 to +1, a higher silhouette value indicates a better fit to the object's own cluster and a poorer fit to neighboring clusters. Positive and negative large silhouette widths (SWs) signify proper and improper clustering, respectively. Objects with a SW value close to zero indicate a lack of clear discrimination between clusters. Another cluster validity measure is the gap statistic, which is based on a statistical hypothesis test [34]. It assesses the change in within-cluster dispersion relative to the expected change under an appropriate reference null distribution.

In Reference [35], a comparative study was conducted to evaluate five different clustering algorithms using gene expression time series datasets of the Saccharomyces cerevisiae yeast. The experiments employed a k-fold cross-validation procedure to compare the performance of the algorithms. The results showed that k-means, dynamical clustering, and SOM consistently achieved high accuracy across all experiments. In Reference [36], the performances of k-means, single linkage, and simulated annealing (SA) were assessed using various partitions obtained through validation indexes. They introduced a novel validation index called the I index, which measured separation based on the maximum distance between clusters and compactness based on the sum of distances between objects and their respective centroids. The study concluded that the I index demonstrated the highest reliability among the considered indices, attaining its maximum value when the number of clusters was appropriately chosen.

It is worth noting that most existing clustering methods often require some form of supervision or expert knowledge, such as specifying the number of clusters or setting similarity thresholds and algorithm-specific hyperparameters. The research gap in the provided passage is that most previously investigated clustering methods for data mining relied on manual parameter selection to identify clusters from raw data. However, these methods often faced challenges in high-dimensional datasets and provided relatively lower performance. This suggests a need for a clustering method that can overcome these limitations and offer improved performance in high-dimensional settings.

To address this research gap, this study extends the work on developing the novel clustering method, called DPPA, developed by [37]. This method is non-parametric and can calculate 1-NN and Max-NN by analyzing the positions of data in the dataset, eliminating the need for initial manual parameter assignments. The proposed method is evaluated using well-known publicly available datasets and offers the advantage of being less time-consuming by reducing the dependence on artificial parameter selection. Furthermore, the performance of the proposed method is compared with four popular clustering algorithms (DBSCAN, K-means, affinity propagation, and Mean Shift) to assess its effectiveness.

In summary, the research gap pertains to the limitations of existing clustering methods in high-dimensional datasets, while the proposed solution is the development of the DPPA algorithm that addresses these limitations and provides improved performance.

3. Overview of Clustering Algorithms

3.1. DBSCAN

DBSCAN, which stands for density-based spatial clustering of applications with noise, is widely used in machine learning and data science [38]. The algorithm is capable of identifying clusters of points in a dataset based on their density and has the ability to handle noise and outliers effectively. DBSCAN is particularly useful for datasets with non-linear or complex structures and can detect clusters of different shapes and sizes.

The main idea behind DBSCAN is to identify regions in the dataset where the data points are densely packed together. The algorithm works by defining two key parameters, epsilon (ϵ) and minimum points (MinPts). The epsilon parameter defines the radius of a neighborhood around each data point, and the MinPts parameter specifies the minimum number of points required to form a dense region.

The DBSCAN algorithm starts by randomly selecting a data point that has not yet been assigned to a cluster. It then identifies all the neighboring points within the radius of ϵ and checks if the number of neighboring points is greater than or equal to the MinPts parameter. If the number of neighbors is greater than or equal to MinPts, the algorithm considers the data point as part of a cluster and explores its neighboring points to find all other data points that belong to the same cluster. This process continues until no more points can be added to the cluster, and then the algorithm selects a new unassigned point and repeats the process.

If the number of neighboring points is less than MinPts, the algorithm marks the data point as a noise point or an outlier, which means that it does not belong to any cluster. The algorithm then moves on to the next unassigned data point and repeats the process until all data points have been assigned to a cluster or marked as noise.

One of the advantages of DBSCAN is that it can detect clusters of different shapes and sizes, including clusters that are not linearly separable. The algorithm can also handle noise and outliers effectively, as they are simply marked as noise points and do not interfere with the clustering process. DBSCAN is also relatively easy to implement and requires only two input parameters, ϵ and MinPts.

However, DBSCAN also has some limitations. One of the main challenges is setting the appropriate values for the ϵ and MinPts parameters. Choosing the right values can be difficult, especially for large and complex datasets, and may require some trial and error. DBSCAN is also sensitive to the choice of the distance metric used to calculate the distance between data points, and some distance metrics may perform better than others, depending on the dataset.

In summary, DBSCAN is a powerful clustering algorithm that can detect clusters of different shapes and sizes in complex datasets. It is capable of managing noise and outliers effectively, making it particularly useful for real-world applications. However, choosing the appropriate values for the ϵ and MinPts parameters can be challenging, and the choice of distance metric can also impact the performance of the Algorithm 1.

Algorithm 1: DBSCAN clustering algorithm.

Input: X: dataset of n data points, EPS: radius of the neighborhood, MinPts: the minimum quantity of points necessary to produce a dense region
/ **Output:** labels: cluster assignments for each data point (0 = noise)
. Initialize all data points as unvisited and with cluster assignment 0; **for** *each unvisited data point p* **do**
  Mark p as visited; find all points within distance EPS of p and add them to the current cluster; **if** *the number of points is less than MinPts* **then**
    Mark p as a noise point;
  end
  else
    Mark all added points as visited and expand the cluster;
  end
end
Expand the cluster by adding points to it; **for** *each point q in the cluster* **do**
  Find all points within distance EPS of q and add them to the current cluster; **if** *The number of points is at least MinPt or higher* **then**
    Mark them as visited;
  end
  if *q has not already been assigned to a cluster* **then**
    Assign it to the current cluster;
  end
end

3.2. Affinity Propagation

The affinity propagation (AP) technique, mentioned in [28,39], is a distance-dependent approach that can identify exemplars in a dataset. AP can cluster facial images, locate genes, and cluster spatial datasets on a map, among other tasks [40–42]. Unlike K-means, AP selects existing points as exemplars and does not generate new points. It does not need a predetermined number of clusters or initial candidate exemplars. Instead, AP considers every point in the dataset and chooses exemplars through competitive iterations.

AP employs two types of "messages" to compute responsibility and availability, based on the distance between any two distinct points, x_i and x_k, in the dataset. Responsibility, represented by $r(i,k)$, is a "message" sent from x_i to its candidate exemplar point x_k and indicates how well x_k can represent x_i as an exemplar. Availability, represented by $a(k,i)$, is the "message" sent from x_k to x_i and represents the accumulated evidence for how suitable x_k is for being selected as an exemplar by x_i. AP initializes the values of these "messages" between any two distinct points, and then iteratively updates the values of responsibility and availability. Finally, AP selects the point $x_{k'}$ with the highest sum of responsibility $r(i,k')$ and availability $a(k',i)$ as the exemplar for each point x_i. Thus, AP clusters x_i and $x_{k'}$ into the same cluster and assigns x_i a label based on the sequence index number of $x_{k'}$ in the dataset. As a result, AP can identify exemplars and assign class labels to each point based on its exemplar's sequence index number.

Here is a detailed description of the affinity propagation algorithm:

1. Initialization: The algorithm begins by initializing two matrices: the "similarity matrix" and the "responsibility matrix". The similarity matrix contains the pairwise similarity scores between all data points. The responsibility matrix is a matrix of the same size as the similarity matrix and is initialized to 0.
2. Calculate responsibility matrix: In this step, the algorithm iteratively updates the responsibility matrix. The responsibility matrix represents how well-suited each data point is to be an "exemplar" or representative of a cluster. The update rule is as follows:

$$r(i,k) = s(i,k) - \max_{k' \neq k} a(i,k') + s(i,k') \tag{1}$$

   where $r(i,k)$ is the responsibility of point i to exemplar k, $s(i,k)$ is the similarity score between point i and exemplar k, and $a(i,k)$ is the "availability" of point i to exemplar k, which is updated in step 3.
3. Calculate availability matrix: In this step, the algorithm updates the availability matrix. The availability matrix represents how much "support" a data point receives from other data points for being an exemplar. The update rule is as follows:

$$a(i,k) = \min 0, r(k,k) + \sum_{i' \neq i, i' \neq k} \max 0, r(i',k) \tag{2}$$

   where $a(i,k)$ is the availability of point i to exemplar k, $r(k,k)$ is the self-responsibility of exemplar k, and the summation term represents the total responsibility that other points have assigned to exemplar k.
4. Calculate cluster exemplars: The algorithm uses the responsibility and availability matrices to calculate which data points are the best exemplars for each cluster. The exemplars are chosen as the data points with the highest sum of their responsibility and availability scores:

$$e(k) = \text{argmax}_i(r(i,k) + a(i,k)) \tag{3}$$

5. Assign data points to clusters: Finally, the Algorithm 2 assigns each data point to its nearest exemplar, which forms the final clusters.

Affinity propagation algorithm has advantages: Automatic cluster number determination and handling of non-spherical clusters. However, it can be computationally expensive and sensitive to the initial exemplar choice.

Algorithm 2: Affinity propagation algorithm.

```
Inputs:
- Similarity matrix s
- Damping factor damping
- Maximum number of iterations max_iter
- Tolerance tol
Outputs:
- Cluster assignments
Initialize:
- Responsibility matrix r = zeros(n_samples, n_samples)
- Availability matrix a = zeros(n_samples, n_samples)
for iter = 1 to max_iter do
    for i in range(n_samples) do
        max_val = -inf
        max_idx = -1
        for k in range(n_samples) do
            val = a[i, k] + s[i, k] if val > max_val then
                max_val = val
                max_idx = k
            end
        end
        for k in range(n_samples) do
            if k != max_idx then
                r[i, k] = damping * r[i, k] + (1 - damping) * (s[i, k] - max_val)
            end
        end
    end
    for k in range(n_samples) do
        max_val = -inf
        max_idx = -1 for i in range(n_samples) do
            if i != k then
                val = max(0, r[i, k]) if val > max_val then
                    max_val = val
                    max_idx = i
                end
            end
        end
        for i in range(n_samples) do
            if i != k then
                val = max(0, r[max_idx, k])
                a[i, k] = damping * a[i, k] + (1 - damping) * min(r[i, k], val)
            end
        end
    end
    if abs(r + a - s).max() < tol then
        break
    end
end
exemplars = argmax(r + a, axis=1)
clusters = [[] for _ in range(n_samples)]
for for i in range(n_samples) do
    clusters[exemplars[i]].append(i)
end
Output: Finally, the cluster output is a list of lists, where each sub-list contains the indices of the data
  points in the corresponding cluster.
```

3.3. Mean Shift Algorithm

The authors of [43,44] explored kernel-based clustering for a dataset X (where $X = \{x_1, \ldots, x_n\}$) in an s-dimensional Euclidean space R^s. This approach transforms the data space into a high-dimensional feature space F using a kernel function to represent inner products. Another kernel-based clustering method in the data space is the kernel

density estimation, which estimates the density over X and identifies modes that correspond to the densest regions [45]. These modes can serve as estimates of cluster centers. To identify the modes in kernel density estimation, the mean shift, a basic gradient approach, is used. In the following section, we will delve into the mean shift procedure.

3.3.1. Mean Shift Procedures

Mean shift techniques are utilized to identify the modes of the kernel density estimation. The kernel, denoted as $H : X \to R$, is defined by $H(x) = h(|x - x_j|^2)$. The estimation of the kernel density is derived using the following equation [46]:

$$\hat{\int} H(x) = \sum_{j=1}^{n} h(|x - x_j|^2) w(x_j) \tag{4}$$

In this equation, $w(x_j)$ is a weight function. Fukunaga and Hostetler [47] first introduced the statistical properties of the gradient of the density estimation using a uniform weight, which includes asymptotic unbiasedness, consistency, and uniform consistency.

$$\nabla \hat{\int}_H (x) = 2 \sum_{j=1}^{n} (x - x_j) h'(\|x - x_j\|^2) w(x_j) \tag{5}$$

Assume the existence of a kernel $K : X \to R$, such that $h'(r) = ck(r)$, where c is a constant. If kernel H is considered a shadow of kernel K, as defined in Ref. [48], then the following equation holds:

$$\begin{aligned} \nabla \hat{\int}_H (x) &= \sum_{j=1}^{n} k(\|x - x_j\|^2)(x_j - x) w(x_j) \\ &= \left[\sum_{j=1}^{n} k(\|x - x_j\|^2) w(x_j) \right] \times \left[\frac{\sum_{j=1}^{n} k(\|x - x_j\|^2) w(x_j) x_j}{\sum_{j=1}^{n} k(\|x - x_j\|^2) w(x_j)} \right] \\ &= \int' K(x)[m_K(x) - x] \end{aligned} \tag{6}$$

The generalized mean shift is expressed by the formula $m_K(x) - x = \nabla \hat{f}_H(x) / \hat{f}_K(x)$, which measures the estimated density gradient. This equation was first introduced in Reference [49] under the assumption of uniform weights. When the gradient estimator $\nabla \hat{f}_H(x)$ is zero, the mode estimation can be determined by:

$$x = m_K(x) = \frac{\sum_{j=1}^{n} k(\|x - x_j\|^2) w(x_j) x_j}{\sum_{j=1}^{n} k(\|x - x_j\|^2) w(x_j)} \tag{7}$$

Here, K is the kernel, and H is its shadow. Equation (6) is known as the sample weighted mean using kernel K". The mean shift can be implemented in three ways. The first method involves assigning initial values to each data point and updating each data point's x_j with $m_K(x_j)$, called blurring mean shift, where each data point and the density estimate $\hat{f}_H(x)$ are modified with each iteration. The second method, called non-blurring mean shift, updates only the data point x with $m_K(x_j)$, while keeping most data points and the density estimate unchanged. The third method, called general mean shift, involves selecting c starting values, which can be more or less than n, and updating the data point x with $m_K(x_j)$. Cheng [48] discussed the convergence properties of blurring mean shift using Equation (6), while Comaniciu and Meer [49] presented various mean shift properties for discrete data and their relationship with the Nadaraya–Watson estimator through kernel regression and the robust M-estimator.

If a kernel K has a shadow H, the mean shift method can identify the modes of a known density estimate $\hat{f}_H(x)$, allowing for direct identification of the estimated density shape modes. In cases where the shadow for the kernel K is unknown or does not exist,

the mean shift method can still be useful for estimating alternative modes, such as cluster centers, for a dataset with an unknown density function.

3.3.2. Some Special Kernels and Their Shadows

In this section, the main focus will be on investigating special kernels that possess shadows. More specifically, we delve into Gaussian kernels $G^p(x)$, which are the most frequently utilized kernels that possess their own shadows [46].

$$G^p(x) = [g(\|x - x_j\|^2)]^p = [exp\left\{-\|x - x_j\|^2/\beta\right\}]^p \tag{8}$$

with their shadows SG^p defined as

$$SG^p(x) = G^p(x), p > 0 \tag{9}$$

The process of mean shift, where x is reassigned as $m_{G^p}(x_j)$, is utilized to identify the modes of the density estimate $\hat{f}SG^p(x)$. Cheng conducted a study on the behavior of mean shift in cluster analysis by employing a Gaussian kernel. The maximum entropy clustering algorithm is a specialized weight function-based Gaussian kernel mean shift. Chen and Zhang employed a Gaussian kernel-induced distance measure to perform robust image segmentation using spatially constrained fuzzy c-means (FCM). Yang and Wu utilized $\hat{f}SG^p(x)$ as an objective function for total similarity and derived a similarity-based clustering method (SCM) that could autonomously organize the number and size of clusters based on the data structure [46].

$$C^p(x) = [c(\|x - x_j\|^2)]^p = [1 + \|x - x_j\|^2/\beta^{-1}]^p \tag{10}$$

The Cauchy kernels, which are obtained from the Cauchy probability density function $f(x) = (1/\pi)(1 + x^2)^{-1}$, where $-\infty < x < \infty$, are noteworthy kernels. Their shadows are defined in the following manner [46]:

$$SC^p(x) = C^{p-1}(x), p > 1 \tag{11}$$

The process of mean shift using $x \leftarrow m_{C^p}(x_j)$ is used to locate the modes of the density estimate $\hat{f}_{SC^p}(x)$. The use of Cauchy kernels is not as prevalent as other types of kernels. To address the limitations of FCM in noisy environments, Krishnapuram and Keller proposed the possibilistic c-means (PCM) clustering algorithm by relaxing the constraint of the fuzzy c-partition's summation to 1. The possibilistic membership functions were subsequently utilized as Cauchy kernels. This is the only instance of Cauchy kernels being utilized in clustering that we are aware of.

The flat kernel is the most basic kernel and is defined as follows [46]:

$$F(x) = \begin{cases} 1 & if \|x - x_j\|^2 \leqslant 1, \\ 0 & if \|x - x_j\|^2 > 1 \end{cases} \tag{12}$$

with the Epanechnikov kernel $E(x)$, as its shadows

$$E(x) = \begin{cases} 1 - \|x - x_j\|^2 & if \|x - x_j\|^2 \leqslant 1, \\ 0 & if \|x - x_j\|^2 > 1 \end{cases} \tag{13}$$

In addition, the Epanechnikov kernel's shadow is the biweight kernel $B(x)$.

$$B(x) = \begin{cases} (1 - \|x - x_j\|^2)^2 & if \|x - x_j\|^2 \leqslant 1, \\ 0 & if \|x - x_j\|^2 > 1 \end{cases} \tag{14}$$

To provide a more general explanation, we generalize the Epanechnikov kernel $E(x)$ into the form of generalized Epanechnikov kernels $K_E^p(x)$ with the parameter p defined as follows [46]:

$$K_E^p(x) = [k_E(\|x - x_j\|^2)]^p = \begin{cases} (1 - \|x - x_j\|^2/\beta)^p & if \|x - x_j\|^2 \leqslant \beta, \\ 0 & if \|x - x_j\|^2 > \beta \end{cases} \quad (15)$$

As a result, the generalized Epanechnikov kernels $K_E^p(x)$ have corresponding shadows $SK_E^p(x)$, defined as [46]:

$$SK_E^p(x) = K_E^{p+1}(x), \ p > 0 \quad (16)$$

When $\beta = 1$, functions $K_E^0(x)$, $K_E^1(x)$, and $K_E^2(x)$ become $F(x)$, $E(x)$, and $B(x)$, respectively. To identify the modes of the estimated density function $\hat{f}SKE^p(x)$, we employ the mean shift process using x replaced by $m_{K_E^p}(x_j)$. In total, we have three categories of kernels and their corresponding counterparts, namely Gaussian kernels $G^p(x)$ and their shadows $SG^p(x) = G^p(x)$, Cauchy kernels $C^p(x)$ and their shadows $SC^p(x) = C^{p-1}(x)$, and generalized Epanechnikov kernels $K_E^p(x)$ and their shadows $SK_E^p(x) = K_E^{p+1}(x)$. The corresponding density estimates can be found using the mean shift process with any of these three kernel categories. The performance of the mean shift procedure for kernel density estimation is significantly affected by the normalization and stabilization parameters, denoted by β and p, respectively. We will explore this topic in the next section.

3.4. K-Means

K-means clustering is a popular unsupervised machine learning algorithm used for clustering data points into groups based on their similarity [50,51]. The algorithm aims to minimize the sum of squared distances between the data points and their assigned cluster centers. Here are the main steps of the algorithm:

1. Choose the number of clusters K that you want to identify and randomly initialize K cluster centers.
2. Assign each data point to its nearest cluster center. This can be done using Euclidean distance or other distance measures.
3. Calculate the mean of the data points in each cluster to obtain the new cluster centers.
4. Repeat steps 2 and 3 until the cluster centers converge, i.e., when the assignments of the data points to clusters no longer change or change minimally.

Here are the equations for the steps:

Step 2: Assign each data point x_i to its nearest cluster center c_j:

$$\underset{j}{argmin} ||x_i - c_j||^2 \quad (17)$$

where $|.|$ denotes the Euclidean distance.

Step 3: Calculate the mean of the data points in each cluster to obtain the new cluster centers:

$$c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \quad (18)$$

where C_j is the set of data points assigned to cluster j.

Step 4: Repeat steps 2 and 3 until convergence.

The algorithm may converge to a local optimum rather than a global one, so it is often run multiple times with different random initializations to find the best result.

K-means clustering can be extended to handle different data types and distances, as well as more complex clustering problems, such as non-convex clusters or variable cluster sizes. However, the algorithm is sensitive to the choice of K and the initial cluster centers, and it may not work well with noisy or high-dimensional data.

4. Data Point Positioning Analysis Algorithm

The DPPA algorithm is required because clustering is an essential data analysis technique that is used in various fields, such as data mining, machine learning, and pattern recognition. Clustering helps to identify patterns, group similar data points, and extract useful information from large datasets. However, traditional clustering algorithms, like k-means, DBSCAN, Mean Shift, and affinity propagation, have some limitations that DPPA aims to address.

The k-means algorithm requires the number of clusters to be specified beforehand, which may not be known in advance. The DBSCAN algorithm assumes that the clusters have similar densities, which may not be valid in all cases. The Mean Shift algorithm is sensitive to the choice of the bandwidth parameter and may lead to overfitting or underfitting. The affinity propagation algorithm requires the selection of a damping factor, which can significantly affect the clustering results.

In contrast, the DPPA algorithm does not require any prior knowledge of the number of clusters or the density of the data. The algorithm works by calculating the distances between each pair of adjacent data points, and then using this information to identify neighborhoods of data points. For each data point, the algorithm counts the number of neighboring data points that fall within a specified distance range and builds a neighbor-link (Table 1) that includes information about the 1-NN and Max-NN of each data point. The algorithm then sorts the data points based on the number of neighbors in ascending order, and clusters the data points, starting with the data point with the fewest neighbors and following its Max-NN, until no more points can be added to the cluster. The algorithm repeats this process for all remaining data points until all have been assigned to a cluster. The Algorithm 3 presented below outlines the complete set of steps involved in the DPPA clustering algorithm:

1. $d(p_i, p_{i+1}) =$
   $(\sqrt[2]{(p_i^x - p_{i+1}^x)^2 + (p_i^y - p_{i+1}^y)^2 + (p_i^z - p_{i+1}^z)^2})$
2. For $\forall p_i$ determine $min(d(p_i, p_{i+1})), i = \{1 \ldots m\} \xrightarrow{stores} \phi(min(d(p_i)), \ldots, min(d(p_m)))$
3. Calculate the range of radius, denoted by λ, between the minimum and maximum values of a scalar value ϕ.
4. For each point $p_i \epsilon \delta$,
   (a) Compute n_{pi}, which is the count of neighboring data points within the specified radius range λ, for a given data point p_i. The calculation involves summing up a series of ones or zeros, depending on whether the distance between p_i and a particular neighboring data point is within or outside the specified range.
   (b) Construct a table (Table 1) that shows the nearest neighboring data points for each data point d_{pi}, including the nearest neighbor (1-NN) and maximum neighbor (Max-NN):

Table 1. Details about the nearest neighboring data points.

Data Point	1-NN	Max-NN
p_i	p_a where a is some index	$p_i \rightarrow p_a \rightarrow p_d \rightarrow p_e$
p_i	p_b where b is some index	
p_i	p_c where c is some index	
p_i	p_d	
p_i	p_e	

Note: In the context of this statement, 1-NN refers to a data point p_i, where the distance $d(p_i)$ falls within a certain range defined by the scalar values $min(\phi)$ and $max(\phi)$. This condition applies to each data point p_i, where i is an index ranging from 1 to m. Max-NN refers to a process of linking data points, starting with a particular point p_i, including the 1-NN of p_i, and continuing the linkage process until p_i has no 1-NN left to be included in the linkage.

5. Arrange the data points p_i in the neighbor-link table in a manner such that they are sorted in ascending order based on the value of their corresponding n_{pi}.
   (a) To form a cluster C_i, first place the data point p_i, and then add all the data points p_k that are in the Max-NN linkage of p_i.
   (b) Add all the data points p_j that are 1-NN (nearest neighbors) of p_i to the cluster C_i.
   (c) If the next data point p_{i+1} belongs to cluster C_k, then assign C_k to C_i and set p_i to p_{i+1}, and repeat the process, starting from step a.
   (d) Continue the process until there are no more data points left.

Algorithm 3: Data point positioning analysis algorithm.

Data: $D = x1, x2, x3..., x_n$
AllDataPoint=[Array declare]
AllMinValue=[Array declare]
for *each position i in D* **do**
  for *each position j in D* **do**
    if *i is not equal to j* **then**
      findPoint= $\sqrt[2]{(p_i^x - p_j^x)^2 + ... + (p_i^n - p_j^n)^2}$
      AllDataPoint.append([findPoint, i, j])
    end
  end
  findMin=min(D[findPoint]) AllMinValue.append(findMin)
end
minP=min(AllMinValue)
maxP=max(AllMinValue)
FinalData=[Array declare]
for *each position k in AllDataPoint* **do**
  findPoint=AllDataPoint[k][0]
  if $(minP \leq findPoint\ and\ maxP \geq findPoint)$ **then**
    $FinalData.append(AllDataPoint[k])$
  end
end
for *each position m in FinalData* **do**
  Find the relations based on i and j in FinalData with the data point value.
  There will be separate relations; every relation is a cluster.
end
Output: Cluster Labels (Class ID or "noise") of Dataset D

5. Dataset

5.1. Dataset 1

The dataset was collected by the University of California, Irvine (UCI), and was acquired by S. Moro, R. Laureano, and P. Cortez at the 2011 European Simulation and Modeling Conference [52]. The dataset can be downloaded from [53]. The data were obtained from a Portuguese financial institution and relate to their direct marketing campaigns, which relied on phone calls to potential customers. Multiple calls were often required to determine whether a customer had subscribed to the bank term deposit product or not. The dataset contains 45,211 samples, each with 17 characteristics and no missing values [52].

Table 2 displays the features of the dataset, which are classified into three types of attributes: numeric attributes that have a range, such as balance, campaign, Age, Pdays, day, duration, and previous; categorical attributes that are classified into sets, such as marital, outcome, job, contact, month, and education; and binary-categorical features that have only two classes, represented as yes or no, such as job, marital, education, contact, month, outcome (default, housing, loan, output).

Table 2 shows the number of classes associated with each attribute's name. For example, the second attribute, titled "Job", has several types of employment, including unknown, management, technician, entrepreneur, self-employed, student, blue-collar, admin, unemployed, retired, housemaid, and services. The "Marital" attribute has three classes, which are represented by single, divorced, and married with divorce, representing those who are divorced or widowed. The "Education" attribute has four classes, namely unknown, secondary, primary, and tertiary. However, the "Default", "output Loan", and "Housing" attributes have only two classes each. The "Contact" attribute has three classes, including unknown, telephone, and cellular. The "Month" attribute has classes that correspond to the names of the months, such as January and February. The "outcome" attribute reflects the outcome of the previous marketing campaign, such as unknown, other, failure, and success. The last column of Table 2 shows the duration for each range of numerical-type attributes, such as the "Age" attribute, which has a duration of (18:95), indicating that all customers or sample ages lie between 18 and 95 years.

Table 2. Details about the bank marketing dataset.

Attributes	Type	Kind	Attributes illustration
Age	Range	Numeric	
Job	Set	Categorical	("admin", "unknown", "unemployed", "management", "housemaid", "entrepreneur", "student", "blue-collar", "self employed", "retired", "technician", "services")
Marital	Set	Categorical	marital status ("married", "divorced", "single"; note: "divorced" means divorced or widowed)
Education	Set	Categorical	("unknown", "secondary", "primary", "tertiary")
Default	Flag	Binary (Categorical)	Has defaulted on credit? (Yes/No in binary)
Balance	Range	Numeric	Typical annual balance, in euros
Housing	Flag	Binary (Categorical)	Has a mortgage loan? (Yes/No in binary)
Loan	Flag	Binary (Categorical)	Individual loan? (Yes/No in binary) # pertaining to the most recent campaign interaction.
Contact	Set	Categorical	Kind of contact communication (categorical: "unknown", "telephone", "cellular")
Day	Range	Numeric	Last contact day of the month
Month	Set	Categorical	Year's last month of contact (categorical: "jan", "feb", "mar", . . . , "nov", "dec")
Duration	Range	Numeric	Length of the last contact, in seconds
Campaign	Range	Numeric	Number of contacts made for this customer during this campaign (includes the last contact)
Pdays	Range	Numeric	Number of days since the last contact with the client from a prior campaign (−1 means the client was not previously contacted)
Previous	Range	Numeric	Number of contacts performed for this customer prior to this campaign.
Poutcome	Set	Categorical	Result of the preceding marketing effort (categorical: "unknown", "other", "failure", "success")
Output	Flag	Binary (Categorical)	Output variable (desired outcome): y—has the customer made a term deposit? (Yes/No in binary)

Likewise, the dataset includes attributes, such as the day of the month, which ranges from 1 to 31, and the duration of the last contact made with the client, which ranges from 0 to 4918 s. However, the Pdays attribute has a domain that ranges from −1 to 871, indicating the number of days that have passed since the client was last contacted during a previous marketing campaign (−1 indicates that the client has not been contacted). Lastly, the previous attribute shows the number of previous contacts made with the client, ranging from 0 to 275.

Typically, there are two primary types of methods used for analyzing and modeling data: supervised and unsupervised learning. In supervised learning, the data input must contain both independent predictor characteristics as well as a dependent target attribute whose value must be estimated. The method learns how to create a model that can predict the value of the target attribute based on the predictor attributes. Decision trees and neural networks are some examples of supervised learning. Generally, supervised learning is most suitable for studies where the goal is to predict a specific attribute [54].

Unsupervised learning differs from supervised learning as it does not aim to predict a specific target attribute, but instead treats all attributes equally. The objective of unsupervised learning is to recognize patterns, clusters, or other approaches for differentiating the data, which may help uncover the relationships between the data rather than predict the value of a target attribute. Unsupervised learning involves techniques such as correlation analysis, statistical measures, and cluster analysis [54].

5.2. Dataset 2

The subsequent dataset was obtained from Northwestern University's Center for Ultra-scale Computing and Information Security (CUCIS). The synthetic-cluster datasets were created using the IBM synthetic data generator mentioned in previous studies [55,56]. The dataset can download from [57]. In these datasets, a predetermined number of random points was initially selected as distinct clusters, followed by the random addition of points to these clusters. The dataset comprises 500,000 data points distributed across 10 dimensions.

6. Results and Discussion

In this section, the experimental outcomes of the proposed data point positioning analysis (DPPA) method are presented. Furthermore, four popular clustering methods (Mean Shift, K-means, affinity propagation, DBSCAN) are implemented to validate and compare the effectiveness of the proposed approach. The implemented methods are tested on higher-dimensional datasets to examine the ability of DPPA to find the clusters. The experimental results are illustrated as follows.

6.1. Proposed Data Point Positioning Analysis

This experiment is conducted on two publicly available datasets to validate the proposed model. The performance of the proposed model is illustrated in Figures 1 and 2. To visualize the outcomes of the proposed model, the t-SNE Python library is used. Both figures show the number of clusters achieved with the proposed method, with different colors representing different clusters. Figure 1 displays the experimental outcomes of the proposed model with dataset 1, while Figure 2 displays the experimental outcomes with dataset 2. The proposed method successfully identifies a total of 8 clusters for dataset 1 and 50 clusters for dataset 2, demonstrating its ability to handle higher-dimensional datasets, handle noisy data, and identify the correct number of clusters. Moreover, we calculate the silhouette coefficient for performance evaluation. With the proposed approach, a maximum coefficient value of 0.288 is achieved for dataset 1, while a coefficient value of 0.832 is achieved for dataset 2. We also compare the outcome of the proposed method with the popular clustering methods. From Figure 3 and Table 3, it is evident that our proposed method outperforms the other clustering algorithms.

Furthermore, the proposed method does not require manual parameter selection, as it automatically determines the optimal cluster from the dataset without analyzing different parameter combinations. To compare and validate the performance of the proposed algo-

rithm, this study also implements four popular algorithms with the higher-dimensional dataset. The experimental outcomes of these algorithms are presented below.

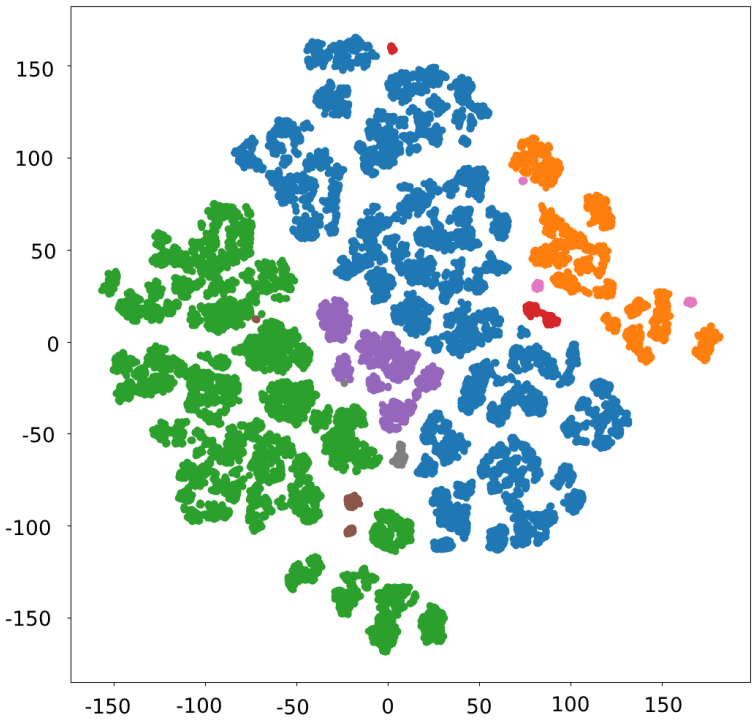

Figure 1. Data point positioning analysis clustering result for dataset 1.

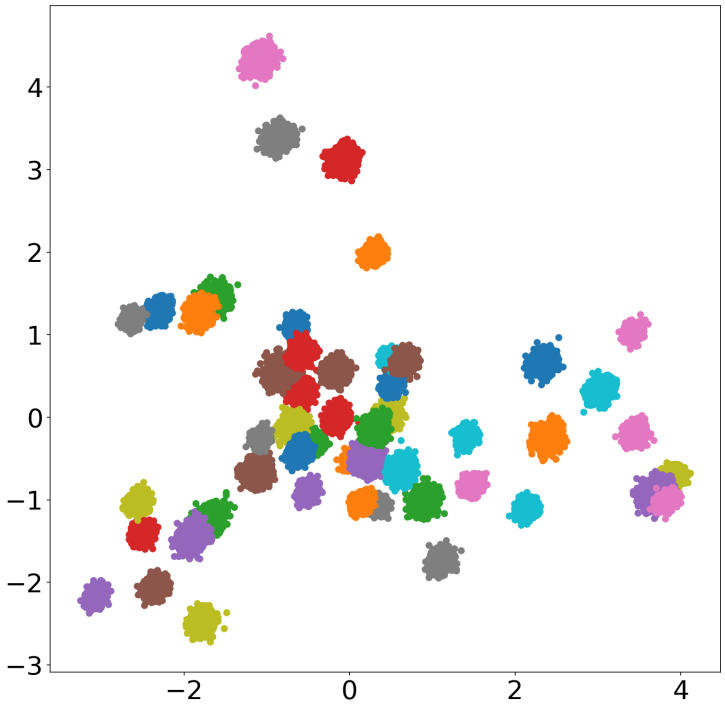

Figure 2. Data point positioning analysis clustering result for dataset 2.

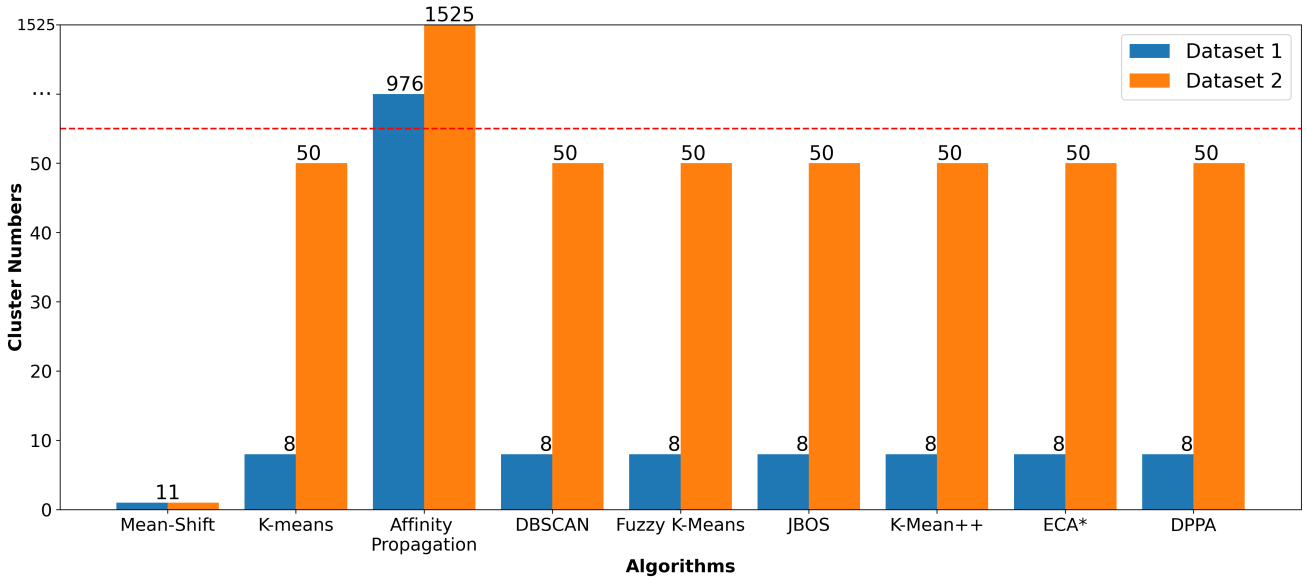

Figure 3. Performance comparison of the algorithms.

Table 3. Performance evaluation of the algorithms for two datasets.

	Dataset 1		**Dataset 2**	
Methods	Cluster Number	Silhouette Coefficient	Cluster Number	Silhouette Coefficient
Mean Shift	1	N/A	1	N/A
K-means	8	0.226	50	0.832
Affinity Propagation	976	0.195	1525	0.0745
DBSCAN	8	0.288	50	0.832
Fuzzy K-means	8	0.158	50	0.832
JBOS	8	0.259	50	0.584
K-Mean++	8	0.262	50	0.60
ECA*	8	0.001	50	−0.12
DPPA	**8**	**0.288**	**50**	**0.832**

6.2. Comparison with Four Popular Methods

6.2.1. DBSCAN

In this experimental analysis, we implement the popular DBSCAN algorithm with two higher-dimensional datasets. The performance of DBSCAN depends on two supplied parameters (Eps and MinPts). It is evident that a small change in Eps or MinPts can result in a different number of clusters and a varying amount of noisy data. When Eps is too small or MinPts is too large, DBSCAN is unable to separate nearby clusters (e.g., MinPts = 3, Eps = 0.5 vs MinPts = 3 and Eps = 0.7). Therefore, it is necessary to search for different parameter values to find the most optimal ones, where the algorithm provides the optimal number of clusters without noise from the dataset. Initially, Eps is set to 0.5, and MinPts is set to 3. In this case, DBSCAN produces a total of 89 clusters, but the noise is high (440). For (MinPts = 4 and Eps = 0.5), DBSCAN provides 81 clusters, but the noise remains high. Hence, it is crucial to analyze different parameter combinations to achieve optimal noise-free clusters. For example, when the parameters (MinPts = 3 and Eps = 1.0) are used, the algorithm produces a total of 8 clusters with no noise. In some cases, the noise is also zero, but the algorithm fails to determine the clusters from the dataset (e.g., Eps = 1.1 and MinPts = 3). In a similar way, we obtain a total of 50 clusters for dataset 2. In addition, we evaluate the performance using the silhouette coefficient. By employing DBSCAN, we obtain a coefficient value of 0.288 for dataset 1, whereas, for dataset 2, the coefficient value reaches 0.832, which is the same as the proposed method. Although the algorithm provides optimal outcomes with the dataset, searching and analyzing different parameter values can

be challenging and time-consuming. In our experiment, we evaluated nearly 40 parameter combinations to achieve optimal noise-free clusters.

6.2.2. Affinity Propagation

The affinity propagation (AP) algorithm, known for its power in clustering data, operates by considering similarities between pairs of data points and simultaneously evaluating all data points as potential exemplars (cluster centers). It employs an iterative technique to recursively search for clusters, where real-valued data are exchanged between data points until a set of high-quality exemplars and their associated clusters are established. In our experimental analysis, we implemented the AP algorithm to cluster higher-dimensional datasets. The algorithm yielded a total of 976 clusters for dataset 1 and 1525 clusters for dataset 2. In addition, we assess the performance using the silhouette coefficient. Employing affinity propagation, dataset 1 achieves a maximum coefficient value of 0.195, whereas dataset 2 attains a coefficient value of 0.0745.

6.2.3. Mean Shift

Mean Shift is a powerful unsupervised clustering algorithm (non-parametric). The Mean Shift algorithm distributes data points to clusters repeatedly by shifting data points toward clusters with the highest data point density. This algorithm has been implemented in this experimental analysis and is being tested with the higher-dimensional dataset. However, in this analysis, Mean Shift is unable to find clusters from the raw higher-dimensional datasets. For both datasets, this algorithm considers the entire dataset as a single cluster. Since this algorithm provides only one cluster with both datasets, it is not possible to calculate the silhouette coefficient.

6.2.4. K-Means

K-means is one of the simplest and most well-established unsupervised learning algorithms that solves the clustering problem. It follows a basic and straightforward method for classifying a given dataset into a defined number of clusters (assuming k clusters). Therefore, it requires providing how many clusters are needed from the dataset. The central concept is to identify k centroids, one for each cluster. These centroids are positioned ingeniously as their placements affect the outcome. With this algorithm, we achieve a total of 8 clusters for dataset 1 and 50 clusters for dataset 2. Here, we set the values of K to 8 and 50, respectively. Since it provides the required clusters, prior knowledge of the dataset is required. Therefore, finding the optimal number of clusters from a new dataset is also very challenging. As each iteration produces different results, the random initial center clustering outcomes are the average of 100 iterations. Furthermore, the evaluation of performance is conducted by utilizing the silhouette coefficient. When applying K-means, dataset 1 demonstrates a relatively low coefficient of 0.226, whereas dataset 2 achieves a significantly higher coefficient of 0.832. Although the maximum coefficient value is achieved with dataset 2, the parameter of K-means (K value = 50) is set manually.

6.2.5. Performed on Other Methods

In this study, we propose a novel clustering algorithm and compare it with four popular clustering methods: DBSCAN, affinity propagation, Mean Shift, and K-means. These methods are widely used by researchers to compare their algorithms. Additionally, we implemented four state-of-the-art algorithms: evolutionary clustering algorithm star (ECA*) [58,59], fuzzy K-means [60], junction based on object similarity (JBOS) [61], and K-means++ [62]. However, these algorithms have some drawbacks, such as the need for manual parameter selection and instability in producing different cluster points each time they are run. The fuzzy K-means method may also result in a large number of missing data points for dataset 1, although it achieves the maximum silhouette coefficient value for dataset 2. The number of clusters and silhouette coefficients are shown in Table 3. JBOS, ECA*, and K-means++ methods provide comparatively lower silhouette coefficient values for both datasets. The main advantage of

using our proposed algorithm is that it is completely parameter-selection-free, eliminating the need for prior knowledge about the datasets.

7. Conclusions

This study proposed a novel clustering algorithm (DPPA) to efficiently cluster the higher-dimensional dataset. The proposed algorithm does not depend on any predefined parameter value, meaning no need to analyze the values of the different parameters, like DBSCAN. For this purpose, this approach calculates 1-NN and Max-NN without requiring the initial manual parameter assignment by examining the positions of data points in the dataset. Therefore, it is no longer required to define the appropriate radius Eps and the minimum number of points MinPts, since the radius range, λ, is automatically computed by evaluating the position and distance of the data points. The proposed algorithm was validated using two publicly available higher-dimensional datasets. To compare the performance of the proposed model, this study also implemented four popular clustering algorithms. The experimental outcomes demonstrated that the proposed algorithm could identify the right number of clusters from the higher-dimensional dataset.

The following are the main benefits of our proposed method over previous non-parameter-free techniques:

- This study proposed a novel approach for clustering data, called data point positioning analysis (DPPA), to enhance the efficiency of high-dimensional dataset clustering.
- In this method, there is no need to pre-specify the number of clusters; whereas traditional clustering methods often require the number of clusters to be determined beforehand. This makes parameter-free methods more flexible and adaptable to different datasets.
- This proposed parameter-free clustering algorithm is better able to handle noisy or outlier data points since it uses density-based clustering techniques that do not depend on distance measures alone.
- The study compared the proposed method to four popular clustering algorithms and demonstrated that the proposed method achieves superior performance in identifying clusters.

In the future, we intend to compare the performance of the proposed method with various datasets. Future research directions for decision-dependent uncertainty in clustering algorithms in critical fields, such as electricity grids and petroleum offshore platforms, include exploring ensemble clustering techniques to improve the reliability of results, and incorporating prior knowledge or domain expertise to enhance the accuracy and effectiveness of the results. Developing methods for quantifying and evaluating uncertainty, as well as investigating the impacts of different evaluation metrics on decision-making processes, are also important areas of future work. Furthermore, applying clustering in other critical fields and evaluating its effectiveness would contribute to the advancement of clustering algorithms and improve decision-making processes in these fields.

Funding: The work is funded by Zayed University under the start-up grant—R21043.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Mirkin, B. *Clustering for Data Mining: A Data Recovery Approach*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2005.
2. Huang, F.; Zhu, Q.; Zhou, J.; Tao, J.; Zhou, X.; Jin, D.; Tan, X.; Wang, L. Research on the parallelization of the DBSCAN clustering algorithm for spatial data mining based on the spark platform. *Remote Sens.* **2017**, *9*, 1301. [CrossRef]
3. Sabor, K.; Jougnot, D.; Guerin, R.; Steck, B.; Henault, J.M.; Apffel, L.; Vautrin, D. A data mining approach for improved interpretation of ERT inverted sections using the DBSCAN clustering algorithm. *Geophys. J. Int.* **2021**, *225*, 1304–1318. [CrossRef]
4. Parsons, L.; Haque, E.; Liu, H. Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 90–105. [CrossRef]
5. Hawkins, D.M. The problem of overfitting. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1–12. [CrossRef]
6. Stojanovic, V.; Nedic, N.; Prsic, D.; Dubonjic, L. Optimal experiment design for identification of ARX models with constrained output in non-Gaussian noise. *Appl. Math. Model.* **2016**, *40*, 6676–6689. [CrossRef]

7. Stojanovic, V.; Nedic, N. Identification of time-varying OE models in presence of non-Gaussian noise: Application to pneumatic servo drives. *Int. J. Robust Nonlinear Control* **2016**, *26*, 3974–3995. [CrossRef]
8. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
9. Yan, S.; Xu, D.; Zhang, B.; Zhang, H.J.; Yang, Q.; Lin, S. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *29*, 40–51. [CrossRef]
10. Baudat, G.; Anouar, F. Generalized discriminant analysis using a kernel approach. *Neural Comput.* **2000**, *12*, 2385–2404. [CrossRef]
11. Jolliffe, I.T.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2016**, *374*, 20150202. [CrossRef]
12. Yan, C.; Gong, B.; Wei, Y.; Gao, Y. Deep multi-view enhancement hashing for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 1445–1451. [CrossRef]
13. Yan, C.; Li, Z.; Zhang, Y.; Liu, Y.; Ji, X.; Zhang, Y. Depth image denoising using nuclear norm and learning graph model. *ACM Trans. Multimed. Comput. Commun. Appl. TOMM* **2020**, *16*, 122. [CrossRef]
14. Yan, C.; Hao, Y.; Li, L.; Yin, J.; Liu, A.; Mao, Z.; Chen, Z.; Gao, X. Task-adaptive attention for image captioning. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 43–51. [CrossRef]
15. Yan, C.; Teng, T.; Liu, Y.; Zhang, Y.; Wang, H.; Ji, X. Precise no-reference image quality evaluation based on distortion identification. *ACM Trans. Multimed. Comput. Commun. Appl. TOMM* **2021**, *17*, 110. [CrossRef]
16. Demiriz, A.; Bennett, K.P.; Embrechts, M.J. Semi-supervised clustering using genetic algorithms. In Proceedings of the Artificial Neural Networks in Engineering (ANNIE-99), St. Louis, MO, USA, 7–10 November 1999; pp. 809–814.
17. Chen, X.; Liu, W.; Qiu, H.; Lai, J. APSCAN: A parameter free algorithm for clustering. *Pattern Recognit. Lett.* **2011**, *32*, 973–986. [CrossRef]
18. Ding, Z.; Xie, H.; Li, P. Evolutionary Parameter-Free Clustering Algorithm. In Proceedings of the 2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML), Chengdu, China, 16–18 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 33–38.
19. Mustapha, S.S.; Theruvil, B.; Madanan, M. Visual comparison of clustering using link-based clustering method (Lbcm) without predetermining initial centroid information. *ICIC Express Lett. Part B Appl. Int. J. Res. Surv.* **2021**, *12*, 317–323.
20. Chang, C.H.; Ding, Z.K. Categorical data visualization and clustering using subjective factors. *Data Knowl. Eng.* **2005**, *53*, 243–262. [CrossRef]
21. He, Z.; Xu, X.; Deng, S. A cluster ensemble method for clustering categorical data. *Inf. Fusion* **2005**, *6*, 143–151. [CrossRef]
22. Han, E.; Karypis, G.; Kumar, V.; Mobasher, B. *Clustering Based on Association Rule Hypergraphs*; University of Minnesota: Minneapolis, MN, USA, 1997.
23. Gibson, D.; Kleinberg, J.; Raghavan, P. Clustering categorical data: An approach based on dynamical systems. *VLDB J.* **2000**, *8*, 222–236. [CrossRef]
24. Rokach, L.; Maimon, O. Clustering methods. In *Data Mining and Knowledge Discovery Handbook*; Springer: Berlin, Germany, 2005; pp. 321–352.
25. San, O.M.; Huynh, V.N.; Nakamori, Y. An alternative extension of the k-means algorithm for clustering categorical data. *Int. J. Appl. Math. Comput. Sci.* **2004**, *14*, 241–247.
26. Wu, F.X. Genetic weighted k-means algorithm for clustering large-scale gene expression data. *BMC Bioinform.* **2008**, *9*, S12. [CrossRef] [PubMed]
27. Likas, A.; Vlassis, N.; Verbeek, J.J. The global k-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [CrossRef]
28. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976. [CrossRef] [PubMed]
29. Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* **1973**, *3*, 32–57. [CrossRef]
30. Pal, N.R.; Biswas, J. Cluster validation using graph theoretic concepts. *Pattern Recognit.* **1997**, *30*, 847–857. [CrossRef]
31. Ilc, N. Modified Dunn's cluster validity index based on graph theory. *Prz. Elektrotech.* **2012**, *88*, 126–131.
32. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [CrossRef]
33. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
34. Tibshirani, R.; Walther, G.; Hastie, T. Estimating the number of clusters in a dataset via the gap statistic. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2001**, *63*, 411–423. [CrossRef]
35. Costa, I.G.; de Carvalho, F.d.A.; de Souto, M.C. Comparative analysis of clustering methods for gene expression time course data. *Genet. Mol. Biol.* **2004**, *27*, 623–631. [CrossRef]
36. Moulick, S.; Mal, B.; Bandyopadhyay, S. Prediction of aeration performance of paddle wheel aerators. *Aquac. Eng.* **2002**, *25*, 217–237. [CrossRef]
37. Mustapha, S.S. An Alternative Parameter Free Clustering Algorithm Using Data Point Positioning Analysis (DPPA): Comparison with DBSCAN. *Int. J. Innov. Comput. Inf. Control* 2024, *in press* .
38. Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Min. Knowl. Discov.* **1998**, *2*, 169–194. [CrossRef]

39. Frey, B.J.; Dueck, D. Mixture modeling by affinity propagation. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 5–8 December 2005; Volume 18.
40. Dueck, D.; Frey, B.J. Non-metric affinity propagation for unsupervised image categorization. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio De Janeiro, Brazil, 14–21 October 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–8.
41. Liu, Y.; Wu, F. Multi-modality video shot clustering with tensor representation. *Multimed. Tools Appl.* **2009**, *41*, 93–109. [CrossRef]
42. Zhang, X.; Gao, J.; Lu, P.; Yan, Y. A novel speaker clustering algorithm via supervised affinity propagation. In Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA, 31 March–4 April 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 4369–4372.
43. Camastra, F.; Verri, A. A novel kernel method for clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 801–805. [CrossRef]
44. Girolami, M. Mercer kernel-based clustering in feature space. *IEEE Trans. Neural Netw.* **2002**, *13*, 780–784. [CrossRef]
45. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; Routledge: Abingdon, UK, 2018.
46. Wu, K.L.; Yang, M.S. Mean shift-based clustering. *Pattern Recognit.* **2007**, *40*, 3035–3052. [CrossRef]
47. Fukunaga, K.; Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory* **1975**, *21*, 32–40. [CrossRef]
48. Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 790–799. [CrossRef]
49. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [CrossRef]
50. MacQueen, J. Classification and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Los Angeles, CA, USA, 21 June–18 July 1965; 27 December 1965–7 January 1966; University of California: Los Angeles, CA, USA, 1967; pp. 281–297.
51. Wang, Q.; Wang, C.; Feng, Z.; Ye, J.f. Review of K-means clustering algorithm. *Electron. Des. Eng.* **2012**, *20*, 21–24.
52. Moro, S.; Laureano, R.; Cortez, P. *Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology*; EUROSIS-ETI: Ostend, Belgium, 2011.
53. PSVishnu. Bank Direct Marketing Dataset. Available online: https://www.kaggle.com/datasets/psvishnu/bank-direct-marketing (accessed on 20 July 2023).
54. Ayetiran, E.F.; Adeyemo, A.B. A data mining-based response model for target selection in direct marketing. *IJ Inf. Technol. Comput. Sci.* **2012**, *1*, 9–18.
55. Pisharath, J.; Liu, Y.; Liao, W.; Choudhary, A.; Memik, G.; Parhi, J. *NU-MineBench 3.0*; Technical Report CUCIS-2005-08-01; Northwestern University: Evanston, IL, USA, 2010.
56. Agrawal, R.; Srikant, R. *Quest Synthetic Data Generator*; Technical Report; IBM Almaden Research Center: San Jose, CA, USA, 1994.
57. CUCIS—Northwestern University. Clustering Benchmark Datasets. Available online: http://cucis.ece.northwestern.edu/projects/Clustering/download_data.html (accessed on 20 July 2023).
58. Hassan, B.A.; Rashid, T.A.; Mirjalili, S. Performance evaluation results of evolutionary clustering algorithm star for clustering heterogeneous datasets. *Data Brief* **2021**, *36*, 107044. [CrossRef] [PubMed]
59. Hassan, B.A.; Rashid, T.A. A multidisciplinary ensemble algorithm for clustering heterogeneous datasets. *Neural Comput. Appl.* **2021**, *33*, 10987–11010. [CrossRef]
60. Kumar, C.A.; Srinivas, S. Concept lattice reduction using fuzzy K-means clustering. *Expert Syst. Appl.* **2010**, *37*, 2696–2704. [CrossRef]
61. Dias, S.M.; Vieira, N. Reducing the Size of Concept Lattices: The JBOS Approach. In Proceedings of the CLA, Sevilla, Spain, 19–21 October 2010; Volume 672, pp. 80–91.
62. Arthur, D.; Vassilvitskii, S. K-means++ the advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.