Acta Polytechnica **61**(2):364–377, 2021
DOI:10.14311/AP.2021.61.0364

available online at https://ojs.cvut.cz/ojs/index.php/ap

MODIFIED POSSIBILISTIC FUZZY C-MEANS ALGORITHM FOR CLUSTERING INCOMPLETE DATA SETS

Rustam[a,*], Koredianto Usman[a], Mudyawati Kamaruddin[b], Dina Chamidah[c], Nopendri[d], Khaerudin Saleh[a], Yulinda Eliskar[a], Ismail Marzuki[e]

[a] *Telkom University, School of Electrical Engineering, Department of Telecommunication Engineering, Jl. Telekomunikasi No.1 Dayeuh Kolot, 40257 Kabupaten Bandung, Jawa Barat, Indonesia*

[b] *Universitas Muhammadiyah Semarang, Faculty of Health Sciences, Semarang, Jawa Tengah, Indonesia*

[c] *Universitas Wijaya Kusuma Surabaya, Faculty of Language and Science, Department of Biology Education, Surabaya, Jawa Timur, Indonesia*

[d] *Telkom University, School of Industrial Engineering, Department of Industrial Engineering, Jawa Barat, Indonesia*

[e] *Fajar University, Department of Chemical Engineering, Makassar, Sulawesi Selatan, Indonesia*

[*] corresponding author: rustam@telkomuniversity.ac.id

Abstract. A possibilistic fuzzy c-means (PFCM) algorithm is a reliable algorithm proposed to deal with the weaknesses associated with handling noise sensitivity and coincidence clusters in fuzzy c-means (FCM) and possibilistic c-means (PCM). However, the PFCM algorithm is only applicable to complete data sets. Therefore, this research modified the PFCM for clustering incomplete data sets to OCSPFCM and NPSPFCM with the performance evaluated based on three aspects, 1) accuracy percentage, 2) the number of iterations, and 3) centroid errors. The results showed that the NPSPFCM outperforms the OCSPFCM with missing values ranging from $5\,\% - 30\,\%$ for all experimental data sets. Furthermore, both algorithms provide average accuracies between $97.75\,\% - 78.98\,\%$ and $98.86\,\% - 92.49\,\%$, respectively.

Keywords: Incomplete data, fuzzy clustering, possibilistic clustering, missing values imputation.

1. Introduction

Incomplete data sets are commonly found in the real world due to failures during the collection, merging, cleaning, and transfer of data from one source to another [1]. The main problem faced when trying to cluster incomplete data sets is the inability of the existing algorithm to carry out the process. This is because popular clustering algorithms comprises *fuzzy c-means* (FCM) [2] and *possibilistic c-means* (PCM) [3], which are used for complete data sets. Bezdek and Hathaway [4] developed the FCM algorithm to deal with the problem of clustering data sets with missing values. They proposed *whole data strategy fuzzy c-means* (WDSFCM) to deal with the problems associated with the incomplete data set clustering by removing features that contain missing values and running standard FCM algorithms, thereby making the remaining data complete. However, the WDSFCM produces biased clustering results when the missing values are large.

Dixon [5] proposed the *partial distance strategy* (PDS) algorithm to deal with incomplete clustering data sets by calculating a partial distance (squared euclidean). The available data points were used to determine the missing values with the quantity scaled using the reciprocal of the component proportion. Bezdek and Hathaway [4] modified the FCM using the PDS in order to deal with the problems associated with clustering incomplete data sets known as the PDSFCM algorithm. The WDSFCM and PDSFCM algorithms do not impute missing values, therefore they are unable to ascertain the missing values after the clustering process.

The following algorithms, proposed by Bezdek and Hathaway [4] imputed missing values. Furthermore, they modified the FCM algorithm using the *optimal completion strategy* (OCS) and the *nearest prototype strategy* (NPS), each of which is referred to as the OCSFCM and NPSFCM algorithms. The OCSFCM algorithm estimates missing values by considering missing values as an additional variable and partitioning the data while optimizing the value of the FCM objective function. The NPSFCM algorithm estimates missing values using the closest prototype cluster in each iteration step. Therefore, the difference between the OCSFCM and the NPSFCM algorithms lies in the technique used to update the imputation for missing values at each iteration step.

In another research, Bezdek et al. [6] introduced the *possibilistic fuzzy c-means* (PFCM) algorithm, which corrects the shortcomings of the FCM and PCM by overcoming noise sensitivity and the occurrence of coincidental clusters. However, there are some disadvantages associated with the PFCM algorithm, it can be used only for clustering complete data sets. In addition, some recent studies proposed clustering

algorithms only for complete data sets, including [7–12].

This research describes the PFCM algorithm for clustering complete data sets in Section 2. Section 3 explains the PFCM algorithm for clustering incomplete data sets, while Section 4 describes the experimental setup. In Section 5, the experimental results of the real world and artificial data sets are shown with the results analysed. Finally, Section 6 concludes the research.

2. Possibilistic Fuzzy C-Means (PFCM) Algorithm of Complete Data Sets

Suppose unlabeled data sets $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^d$ clustered into a fuzzy subset of c $(1 < c < n)$ clusters. Here, n and d state the number and dimension of each data point, respectively. The $\mathbf{X}$ is clustered into c by minimizing the following objective function [6].

$$J_{m,\tau}(U, T, \mathbf{V}; \mathbf{X}) = \sum_{k=1}^{n}\sum_{i=1}^{c}(\alpha u_{ik}^{m} + \beta t_{ik}^{\tau})d_{ik}^{2} + \sum_{i=1}^{c}\delta_i \sum_{k=1}^{n}(1 - t_{ik})^{\tau}. \tag{1}$$

Here, $\alpha(\alpha > 0)$ denotes the importance level of fuzzy membership degree (u_{ik}). Equation (1) is subject to $\sum_{i=1}^{c} u_{ik} = 1$ constraints. Krishnapuram and Keller [3] relaxed this constrain to become $\sum_{i=1}^{c} u_{ik} \geq 1$, therefore, it is better in reflecting clusters $\mathbf{x}_k$ to the i-th. t_{ik} denotes a possibilistic membership degree of $\mathbf{x}_k$ to the i-th cluster and $\beta(\beta > 0)$ denotes the importance level of t_{ik}. $d_{ik}^2 = \|\mathbf{x}_k - \mathbf{v}_i\|^2$ denotes the Euclidean distance of the j-th data point to i-th cluster centre. $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_c\}$ denotes the centre of the cluster set, where $\mathbf{v}_i \in \mathbb{R}^d$ and $\delta_i > 0$ is the typical of possibilistic. Where $m > 1$ and $\tau > 1$ are fuzzy parameter and possibilistic parameter, respectively.

Basically, u_{ik}, t_{ik}, and $\mathbf{v}_i$ are determined simultaneously. However, in this research, these values were determined numerically using the recursive method. Therefore, the initially values to be calculated are chosen as follows: initiate $\mathbf{v}_i$ to calculate u_{ik} and t_{ik}.

2.1. Possibilistic Fuzzy C-Means (PFCM) Algorithm

In this section, the complete data sets are clustered using the *possibilistic fuzzy c-means* (PFCM) algorithm [6]. The PFCM algorithm is described as follows.

Step I: Fix $m > 1$, $\tau > 1$, $\varepsilon > 0$ and $1 < c < n$. Pick $\mathbf{v}^{(0)} \in \mathbb{R}^d$, $\mathbf{v}^{(0)}$ can be chosen randomly from $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^d$. Then at step l, $l = 1, 2, \cdots$

Step II: Calculate fuzzy membership degree (u_{ik}) which minimize the objective function $J_{m,\tau}$ using the following

$$u_{ik}^{(l)} = \left(\sum_{j=1}^{c}\left(\frac{d_{ik}^2}{d_{jk}^2}\right)^{\frac{1}{m-1}}\right)^{-1}, \tag{2}$$

$$\text{where } d_{ik}^2 = \left\|\mathbf{x}_k - \mathbf{v}_i^{(l-1)}\right\|^2,$$

for $1 \leq i \leq c$ and $1 \leq k \leq n$.

Step III: Calculate the possibilistic typical (δ_i), which minimizes the objective function $J_{m,\tau}$ using the following

$$\delta_i^{(l)} = K\frac{\sum_{k=1}^{n}\left(u_{ik}^{(l)}\right)^m d_{ik}^2}{\sum_{k=1}^{n}\left(u_{ik}^{(l)}\right)^m}, \tag{3}$$

$$\text{where } d_{ik}^2 = \left\|\mathbf{x}_k - \mathbf{v}_i^{(l-1)}\right\|^2.$$

Here K, is always chosen to be 1 [6].

Step IV: Calculate the possibilistic membership degree (t_{ik}), which minimizes the objective function $J_{m,\tau}$ using the following

$$t_{ik}^{(l)} = \left(1 + \left(\frac{\beta}{\delta_i^{(l)}} d_{ik}^2\right)^{\frac{1}{\tau-1}}\right)^{-1}, \tag{4}$$

$$\text{where } d_{ik}^2 = \left\|\mathbf{x}_k - \mathbf{v}_i^{(l-1)}\right\|^2,$$

for $1 \leq i \leq c$ and $1 \leq k \leq n$.

Step V: Update the cluster centre $(\mathbf{v}_i)$, which minimizes the objective function $J_{m,\tau}$ using the following

$$\mathbf{v}_i^{(l)} = \frac{\sum_{k=1}^{n}\left(\left(\alpha u_{ik}^{(l)}\right)^m + \left(\beta t_{ik}^{(l)}\right)^{\tau}\right)\mathbf{x}_k}{\sum_{k=1}^{n}\left(\left(\alpha u_{ik}^{(l)}\right)^m + \left(\beta t_{ik}^{(l)}\right)^{\tau}\right)}, \tag{5}$$

for $1 \leq i \leq c$.

Step VI: Compare $\mathbf{v}_i^{(l)}$ to $\mathbf{v}_i^{(l-1)}$ using $\left\|\mathbf{v}_i^{(l)} - \mathbf{v}_i^{(l-1)}\right\| < \varepsilon$. If true, then stop. Otherwise, set $l = l + 1$ and return to **Step II**.

The clustering result of the complete data sets will be a base for evaluating the performance of the OCSPFCM and NPSPFCM (see Section 4).

3. Possibilistic Fuzzy C-Means (PFCM) Algorithm of Incomplete Data Sets

Given incomplete data sets $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\} \subset \mathbb{R}^d$ and $\mathbf{y}_2 = (2.35, ?, 1.32, ?, 3.44)^T \in \mathbb{R}^5$. y_{22} and y_{24} are missing values. The question is how to cluster $\mathbf{Y}$? To answer this question, we propose the OCSPFCM and NPSPFCM for clustering incomplete data sets such as $\mathbf{Y}$. The notation used throughout this article is as follows. Let

$\mathbf{y}_k = k^{th}$ d-dimensional datapoint (data vector), for $1 \leq k \leq n$;
$y_{kj} = j^{th}$ feature value of the k^{th}data point, for $1 \leq j \leq d, 1 \leq k \leq n$;
$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\} \subset \mathbb{R}^d$;
$\mathbf{Y}_C = \{\mathbf{y}_k \in \mathbf{Y} | \, \mathbf{y}_k \text{ is a complete data point}\}$;
$\mathbf{Y}_P = \{y_{kj}$ for $1 \leq j \leq d, 1 \leq k \leq n |$ the value for y_{kj} is present in $\mathbf{Y}\}$;
$\mathbf{Y}_M = \{y_{kj} = ?$ for $1 \leq j \leq d, 1 \leq k \leq n |$ the value for y_{kj} is missing in $\mathbf{Y}\}$.

3.1. Optimal Completion Strategy Possibilistic Fuzzy C-Means (OCSPFCM) Algorithm

The explanation of the OCSPFCM algorithm is as follows.

Step I: Fix $m > 1$, $\tau > 1$, $\varepsilon > 0$ and $1 < c < n$. Initiate $\mathbf{Y}_M^{(0)}$, for each $y_{kj} \in \mathbf{Y}_M$, with picking a random available value in $\mathbf{Y}_P$. Then pick $\mathbf{v}^{(0)} \in \mathbb{R}^d$, $\mathbf{v}^{(0)}$ can be chosen randomly from the $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\} \subset \mathbb{R}^d$. Then at step l, $l = 1, 2, \cdots$

Step II: Calculate the fuzzy membership degree (u_{ik}), which minimizes the objective function $J_{m,\tau}$ using the following

$$u_{ik}^{(l)} = \left(\sum_{j=1}^{c} \left(\frac{d_{ik}^2}{d_{jk}^2} \right)^{\frac{1}{m-1}} \right)^{-1}, \quad (6)$$

$$\text{where } d_{ik}^2 = \left\| \mathbf{x}_k - \mathbf{v}_i^{(l-1)} \right\|^2,$$

for $1 \leq i \leq c$ and $1 \leq k \leq n$.

Step III: Calculate the possibilistic typical (δ_i), which minimizes the objective function $J_{m,\tau}$ using the following

$$\delta_i^{(l)} = \frac{\sum_{k=1}^{n} \left(u_{ik}^{(l)} \right)^m d_{ik}^2}{\sum_{k=1}^{n} \left(u_{ik}^{(l)} \right)^m}, \quad (7)$$

$$\text{where } d_{ik}^2 = \left\| \mathbf{x}_k - \mathbf{v}_i^{(l-1)} \right\|^2.$$

Step IV: Calculate the possibilistic membership degree (t_{ik}), which minimizes the objective function $J_{m,\tau}$ using the following

$$t_{ik}^{(l)} = \left(1 + \left(\frac{\beta}{\delta_i^{(l)}} d_{ik}^2 \right)^{\frac{1}{\tau-1}} \right)^{-1}, \quad (8)$$

$$\text{where } d_{ik}^2 = \left\| \mathbf{x}_k - \mathbf{v}_i^{(l-1)} \right\|^2,$$

for $1 \leq i \leq c$ and $1 \leq k \leq n$.

Step V: Update the cluster centre ($\mathbf{v}_i$), which minimizes the objective function $J_{m,\tau}$ using the following

$$\mathbf{v}_i^{(l)} = \frac{\sum_{k=1}^{n} \left(\left(\alpha u_{ik}^{(l)} \right)^m + \left(\beta t_{ik}^{(l)} \right)^\tau \right) \mathbf{y}_k}{\sum_{k=1}^{n} \left(\left(\alpha u_{ik}^{(l)} \right)^m + \left(\beta t_{ik}^{(l)} \right)^\tau \right)}, \quad (9)$$

for $1 \leq i \leq c$.

Step VI: Compare $\mathbf{v}_i^{(l)}$ to $\mathbf{v}_i^{(l-1)}$ using $\left\| \mathbf{v}_i^{(l)} - \mathbf{v}_i^{(l-1)} \right\| < \varepsilon$. If true, then stop. Otherwise, go to **Step VII**.

Step VII: Calculate $\mathbf{Y}_M$, which minimizes the objective function $J_{m,\tau}$, for all $y_{kj} \in \mathbf{Y}_M$ using the following

$$y_{kj}^{(l)} = \frac{\sum_{k=1}^{n} \left(\left(\alpha u_{ik}^{(l)} \right)^m + \left(\beta t_{ik}^{(l)} \right)^\tau \right) v_{ij}^{(l)}}{\sum_{k=1}^{n} \left(\left(\alpha u_{ik}^{(l)} \right)^m + \left(\beta t_{ik}^{(l)} \right)^\tau \right)}. \quad (10)$$

Now, set $l = l + 1$ and return to **Step II**.

We update the missing values imputation on **Step VII** using the sum of fuzzy with possibilistic membership degress multiplied by the values existing on the cluster centre as shown in Equation 10.

3.2. Nearest Prototype Strategy Possibilistic Fuzzy C-Means (NPSPFCM) Algorithm

The difference between the OCSPFCM and the NPSPFCM lies in **Step VII**. The imputation of the missing values was updated by the value available in the nearest cluster center. **Step VII** of the NPSPFCM algorithm is defined as follows.

Step VII: Calculate $\mathbf{Y}_M^{(l)}$, which minimizes the objective function $J_{m,\tau}$, for all $y_{kj} \in \mathbf{Y}_M$ using the following

$$y_{kj}^{(l)} = v_{ij}^{(l)}, \quad (11)$$

where $d_{ik}^2 = \min\left\{d_{1k}^2, d_{2k}^2, ..., d_{ck}^2\right\}$ and $d_{ck}^2 = \|\mathbf{x}_k - \mathbf{v}_c\|^2$. Now, set $l = l + 1$ and return to **Step II**.

Literature [13] presents the time complexity of the OCS and NPS. The time complexity for the OCS and NPS is $O(nc^2d)$, respectively, where n is the number of data ponts, c is the number of clusters, and d is the dimension of data points. The OCSPFCM and NPSPFCM algorithms proposed in this research were adapted from the OCS and NPS with the same time complexity, namely $O(nc^2d)$.

4. Experimental Setup

This study evaluated and demonstrated the potential of the OCSPFCM and NPSPFCM for clustering incomplete data sets. The experiments were carried out in the following stages. Firstly, the complete data sets were clustered using the PFCM algorithm to obtain the distribution of data points in the actual cluster. The result of this stage is used as a base in evaluating the performance of the OCSPFCM and NPSPFCM

algorithms. In addition, the cluster validity index was used to obtain the optimal number of clusters in complete data sets. The cluster validity index used is the Xie-Beni index shown in Equation (12), with a validity index used to measure and determine the optimal number of clusters. Furthermore, the Xie-Beni index was used because the *partition coefficient* (PC) and *classification entropy* (CE) indexes eliminate the cluster centre and data in the index calculation. Meanwhile, the cluster centre and data are two basic attributes involved in a data clustering process based on the fuzzy rule [14]. The optimal number of clusters is indicated by the smallest Xie-Beni index value. Xie and Beni [15] proposed the cluster validity index as follows.

$$\mathbf{XB}(U,V;\mathbf{X}) = \frac{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^2 \left\|\mathbf{x}_k - \mathbf{v}_i\right\|^2}{n \cdot \min\limits_{i\neq j} \left\|\mathbf{v}_i - \mathbf{v}_j\right\|^2}. \quad (12)$$

The performance of the OCSPFCM and NPSPFCM algorithms is evaluated on the real-world and artificial data sets. The real-world data sets are iris [16] and wine [17] downloaded from `http://archive.ics.uci.edu/ml` [18]. Iris data sets consist of 150 data points with 4 features, with a data size of [150×4] in the matrix form. Wine data sets consist of 178 data points with 13 features, with a data size of [178×13]. The artificial data sets I and II used were generated from the Gaussian mixture distribution rule with two clusters. The artificial data set I consist of 1000 data points with 2 features and a size of [1000×2]. A scatter plot of the artificial data set I is shown in Figure 1. The artificial data set II consist of 1000 data points with 14 features with a size of [1000×14]. The authors also evaluated the performance of the OCSPFCM and NPSPFCM on larger data sets, namely the artificial data set III, which consist of 5000 data points with 7 features with a size of [5000×7]. The artificial data set III consists of five clusters. The row and column of the matrix represent the number of data points and features, respectively.

After the clustering, the complete data sets were made into incomplete data sets or, in other words, data sets contain missing values. Each data set consists of missing values with predetermined percentages of 5%, 10%, 15%, 20%, 25%, and 30%. Furthermore, the missing values were randomly determined in the matrix column direction of the complete the data sets.

The third stage examined the performance of the OCSPFCM and NPSPFCM algorithms for clustering incomplete data sets. The evaluation is based on three aspects, the percentage accuracy, the number of iterations, and centroid errors. The formula used to calculate the percentage accuracy is as follows [19].

$$\%\, accuracy = \frac{a}{n}100\% \quad (13)$$

Where a is the number of data points clustered correctly and n is the total number of data points. In this

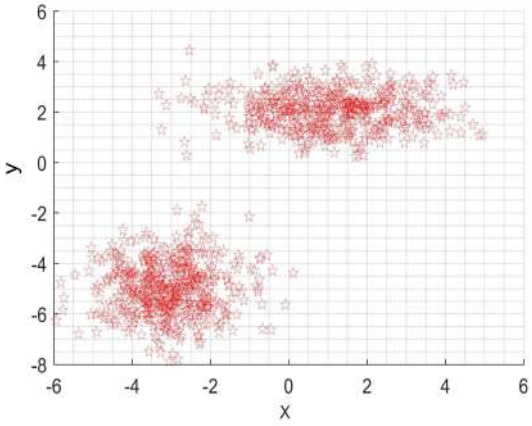

FIGURE 1. Artificial data set I

study, centroid errors are the magnitude of the cluster centre error for an incomplete data set clustered using the OCSPFCM and NPSPFCM algorithms when compared to the cluster centre of a complete data set clustered using the PFCM. In some applications, knowing the cluster centres is important to determine the partitioning of data points [1]. Therefore, this research evaluates the two algorithms by calculating the centroid errors at each level of the missing values. The Euclidean distance formula is used to calculate the centroid errors with the centroid errors (e) averaged using the following formula

$$e = \frac{\sum_{i=1}^{c} e_i}{c}. \quad (14)$$

Where e_i is the i-th centroid error and c is the number of clusters.

5. Experimental Results and Discussions

When carrying out the experiment, the first thing to do is to cluster the complete data sets using the PFCM algorithm with the Xie-Beni index as the validity index. The result showed that the smallest Xie-Beni index value for iris, wine, artificial data sets I, and II is in two clusters. This is in line with Pakhira et al. [20] using the Davies-Bouldin (DB) and the Dunns indexes [21] as the cluster validity index. For the complete wine data sets, the optimal number of clusters was obtained with two clusters. This is in line with Zhang et al. [22] using the MPC and MPA indexes [23] as the cluster validity index. For artificial data sets I and II, the optimal number of clusters obtained is also two clusters. Meanwhile, for the artificial dataset III, the optimal number of clusters obtained is five clusters.

For iris data sets, the first and second clusters consist of 50 and 100 data points. For wine data sets, the first and second clusters consist of 78 and 100 data point members. For artificial data sets I, the first and second clusters comprise 494 and 506 data points members, respectively. While, for artificial data

sets II, the first and second clusters consists of 510 and 490 data points. For artificial dataset III, 972, 1081, 1016, 962, and 969 data points were members of the first, second, third, fourth, and fifth clusters, respectively. These results are a base to the evaluation of the performance of the proposed OCSPFCM and NPSPFCM. Due to fluctuating results of percentage accuracy, the number of iterations and centroid errors in each experiment, this deficiency was addressed by conducting 30 experiments with each data set. The mean of the 30 values is used for percentage accuracy, the number of iterations, and centroid errors.

This study also compared the performance of the OCSPFCM and NPSPFCM with three clustering algorithms for the incomplete data sets, namely the OCSFCM, NPSFCM [4], and KFCM [24]. Our comparison with the KFCM algorithm uses the Gaussian kernel function with $\sigma = 1$. The sigma value ($\sigma = 1$) used is in line with Zhang and Chen as initiators of the KFCM algorithm [24]. Here, we use the computational condition: $\varepsilon = 0.00001$, maximum number of iterations = 100, $\alpha = 1$, $\beta = 1$, $m = 2$, and $\tau = 2$.

5.1. Experiment on Iris Data Sets

The results of the complete iris data set clusters by the PFCM algorithm are used as a base to evaluate the performance of the OCSPFCM and NPSPFCM on incomplete iris data sets.

Table 1 shows the average accuracy percentage for iris data sets using the OCSPFCM and NPSPFCM algorithms. For missing values below 15%, the OCSPFCM algorithm had the accuracy percentage above 90%. However, for missing values between 20% to 30%, the OCSPFCM algorithm had an accuracy percentage above 80% with a maximum of 86%. The NPSPFCM algorithm has an accuracy percentage above 90% for all tested missing values except for 30%, with an accuracy of 89.13%. The percentage of accuracy shows a significant difference above 20% of the total missing values. Table 1 also shows that the greater the number of missing values, the lower the accuracy percentage. Furthermore, the decrease in the accuracy percentage is due to the updated missing values imputation, which falls far from the actual value. Therefore, the data points that contain the missing values become members of the inappropriate cluster. An accuracy percentage of 80% in the case of the OCSPFCM algorithm means that there are 130 data points out of a total of 150 members of the appropriate cluster with a 30% missing values. In addition, there are 20 data points that are members of the inappropriate cluster. While 89.13% of accuracy percentage on the NPSPFCM algorithm means 134 data points are members of the appropriate cluster. In contrast, there are 16 data points that are members of the inappropriate cluster. From Table 1, it can be seen that the OCSFCM has the best performance. Our OCSPFCM outperforms the NPSFCM and our NPSPFCM has almost the same performance as the NPSFCM and better than KFCM.

Table 2 shows the behaviour of the OCSPFCM and NPSPFCM that is inversely proportional to the accuracy percentage and the number of iterations. Furthermore, the percentage of accuracy inversely decreases with an increase in the number of iterations needed. An increase in the number of missing values led to a rise in the number of iterations. In other words, the greater the number of missing values, the more iterations needed for the termination. The OCSPFCM requires more iterations than the NPSPFCM. Meanwhile, the OCSFCM and NPSPFCM provide almost the same and better iteration performance than others.The NPSPFCM has a better iteration performance than the OCSPFCM and KFCM.

Table 3 shows the difference in centroid errors between the OCSPFCM and the NPSPFCM, which starts to be significant at 20% missing values. The shift of the cluster center is closely related to the process of updating the missing values. The shift of the clustre centre is closely related to the process of updating the missing values. The algorithm-updated missing values imputation falls far from the actual value and a cluster centre error occurs. Table 3 also shows the process of updating the missing values imputation by the NPSPFCM algorithm comprising of smaller cluster centre (centroid) errors compared to the OCSPFCM algorithm, and it outperforms the KFCM. Conversely, the OCSFCM has smaller centroid errors than the NPSPFCM and is better than other algorithms. From Table 1, 2, and 3, it is found that the NPSPFCM's performance is always better than that of the OCSPFCM and KFCM, but not OCSFCM.

5.2. Experiment on Wine Data Sets

In the wine data sets, evaluations related to the percentage accuracy, number of iterations, and centroid errors are, respectively, shown in Tables 4, 5, and 6.

Table 4 shows the accuracy percentage of the OCSPFCM, above 90% (5% and 10% missing values), above 80% (15%, 20%, and 25% missing values), and 74.72% (30% missing values). Meanwhile, the NPSPFCM algorithm as an accuracy percentage above 90% for all levels of missing values, except for the 30% with an accuracy of 89.89%. These algorithms produced a percentage of accuracy that decreases with the missing value. For the OCSPFCM algorithm, the percentage of accuracy is 74.72%, which means that with 30% missing values, there are 133 data points out of a total of 178 in the appropriate cluster. Conversely, there are 45 data points that are members of an inappropriate cluster. For the NPSPFCM algorithm, the percentage of accuracy is 89.89%, which means that there are 160 data points out of a total of 178 members of the appropriate cluster at the 30% missing values level. Conversely, 18 data points are members of an inappropriate cluster. In the wine data

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	96.00	98.27	99.2	98.73	96.33
10	93.33	96.44	98.8	96.87	93.00
15	90.00	93.96	95.87	94.44	91.33
20	86.00	92.44	94.33	94.27	89.12
25	82.00	90.76	93.80	91.47	86.42
30	80.00	89.13	92.27	90.80	81.00

TABLE 1. The average accuracy percentage for iris data sets

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	14.03	13.90	12.90	11.30	28.50
10	17.03	16.50	14.40	12.20	33.10
15	20.63	19.03	17.40	13.22	45.00
20	23.40	20.77	16.70	14.00	70.20
25	30.33	25.80	18.30	16.90	82.00
30	35.07	29.77	20.90	19.20	83.00

TABLE 2. The average iterations for iris data sets

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	0.014	0.002	0.002	0.002	0.0556
10	0.029	0.005	0.002	0.006	0.2201
15	0.060	0.015	0.016	0.019	0.5428
20	0.106	0.023	0.024	0.018	1.0256
25	0.185	0.031	0.023	0.040	6.9662
30	0.228	0.043	0.034	0.070	15.5065

TABLE 3. The average centroid errors for iris data sets

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	98.31	97.38	97.53	97.36	96.62
10	93.26	96.63	95.89	95.38	93.82
15	89.36	94.38	89.32	89.32	88.76
20	84.27	92.88	85.17	84.27	85.95
25	80.82	90.64	79.27	78.09	84.26
30	74.72	89.89	74.72	74.71	78.08

TABLE 4. The average accuracy percentage for wine data sets

sets experiment, the NPSPFCM accuracy percentage outperforms other algorithms.

Table 5 shows that the NPSPFCM algorithm provided a number of more efficient iterations than the OCSPFCM and KFCM. However, the OCSPFCM and NPSPFCM are no more efficient compared to the OCSFCM and NPSFCM.

Table 6 shows the average centroid errors in wine data sets using the OCSPFCM and NPSPFCM algorithms. Furthermore, there is a larger centroid error in the case of the OCSPFCM algorithm than in the case of the NPSPFCM at all levels of the total missing values, except at 15%, where the NPSPFCM gives larger centroid errors. The NPSPFCM has smaller centroid errors than the OCSPFCM and KFM. Meanwhile, the OCSFCM and NPSFCM have smaller centroid errors than the other three algorithms.

5.3. Experiment on Artificial Data Sets I

In the artificial data set I, evaluations related to the percentage of accuracy, number of iterations, and centroid errors are, respectively, shown in Tables 7, 8, and 9.

Table 7 shows that the OCSPFCM below 20% of the missing values gives an accuracy percentage above 90%. The NPSPFCM algorithm comprises of an accuracy performance above 90%, with 30% missing values. The NPSPFCM algorithm produces an accuracy percentage above 90%, except for 30% missing values, which has 88.86%. This means that with 30% missing values, there are 886 data points out of a total of 1000 members of the appropriate cluster and 114 data points in the inappropriate cluster. While in the case of the OCSPFCM algorithm, 84.90% means that there are 849 data points out of a total of 1000 members of an appropriate cluster with 151 in an inappropriate cluster. Table 7 shows that the OCSFCM has a better algorithm performance than others and the NPSPFCM has a better accuracy percentage than the OCSPFCM and KFCM, and has almost the same performance as the NPSFCM.

Table 8 shows the average number of iterations needed for the termination. In general, the number of iterations required by the OCSPFCM and NPSPFCM are relatively similar at each level of the missing values. Table 8 also shows that the NPSFCM has the most efficient iteration, while the OCSPFCM and NPSPFCM are more efficient than the OCSFCM and KFCM.

Table 9 shows the average centroid errors for artificial data set I with the NPSPFCM having smaller centroid errors than the OCSPFCM. This is the implication of the process of updating the imputation of missing values by the NPSPFCM. In other words, the cluster centre generated by the NPSPFCM algorithm is closer to the centre base used in the complete artificial data set I. Table 9 also shows the OCSFCM, which has the smallest centroid errors compared to others, and our NPSPFCM having centroid errors smaller than the NPSFCM and KFCM.

5.4. Experiment on Artificial Data Set II

For the artificial data set II, evaluations related to the percentage of accuracy, the number of iterations, and the centroid errors are, respectively, shown in Tables 10, 11, and 12.

Table 10 shows the average accuracy percentage for the artificial data set II. The OCSPFCM and NPSPFCM provide accuracy percentages above 95% for all missing values percentage levels. For the 30% missing values, the OCSPFCM and NPSPFCM give an accuracy percentage of 95.30% and 97.57%, respectively. This means that in the case of the OCSPFCM, there are 953 data points out of a total of 1000 members in the appropriate cluster, with 47 in an inappropriate cluster. Meanwhile, the case of in the NPSPFCM algorithm, there are 975 data points in the appropriate cluster and 25 in an inappropriate cluster. Table 10 also shows the advantages of the OCSFCM and the performance of the NPSPFCM and NPSFCM, which is almost the same but better than that of the OCSPFCM and KFCM.

Table 11 shows that the NPSPFCM and OCSFCM have a similar number of iteration and are more efficient than the OCSPFCM and KFCM. The NPSPFCM also provides a higher accuracy percentage as shown in Table 10. Furthermore, Table 12 shows the smallest centroid errors given by our NPSPFCM and it outperforms all others.

5.5. Experiment on Artificial Data Set III

Experiment results on artificial data set III are shown in Tables 13, 14, and 15, respectively.

Table 13 shows the inaccuracy of the OCSPFCM, which is still superior to the KFCM. The accuracy percentage of the OCSPFCM decreased dramatically at 20% missing values and above. However, the shortcomings of the OCSPFCM are covered by the NPSPFCM, which outperforms all existing algorithms. Table 14 shows the number of the most efficient iterations provided by the NPSPFCM. Likewise, in Table 15, the smallest centroid errors are given by our NPSPFCM.

Artificial data set III is a data set consisting of five clusters. The experiments conducted on the iris data sets, wine data sets, artificial data set I, and artificial data set II, consist of two clusters each. The comparison of the performance of the OCSPFCM and NPSPFCM on data sets with two clusters and five clusters is shown in Figures 2, 3, and 4 respectively.

Figure 2 shows the performance of the OCSPFCM and NPSPFCM in terms of the accuracy percentage on data sets with two clusters and five clusters. In the case of the artificial data set III, which consisted of five clusters, the OCSPFCM showed a lower performance compared to other data sets which consisted

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	37.03	33.73	26.10	25.80	100
10	55.10	35.50	30.40	27.33	100
15	61.90	37.87	43.10	42.60	100
20	72.53	48.17	46.70	48.00	100
25	92.47	55.63	47.80	47.20	100
30	99.00	64.00	51.80	51.20	100

TABLE 5. The average iterations for wine data sets

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	11.41	9.02	3.69	2.76	467.25
10	37.11	15.48	8.85	8.75	464.77
15	46.37	73.58	13.02	13.08	478.59
20	102.34	92.75	24.07	24.79	503.25
25	153.55	98.98	28.72	28.65	658.78
30	185.21	113.36	31.37	31.60	665.07

TABLE 6. The average centroid errors for wine data sets

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	98.10	98.71	99.38	98.29	98.00
10	95.80	96.57	98.62	96.67	95.50
15	92.50	94.29	96.89	95.18	93.10
20	91.40	93.05	95.46	92.90	89.20
25	87.70	92.11	95.37	89.83	88.10
30	84.90	88.86	92.78	88.36	85.20

TABLE 7. The average accuracy percentage for artificial data set I

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	13.80	13.77	17.10	8.90	25.10
10	14.83	14.23	15.00	10.00	26.70
15	17.77	15.63	22.70	11.70	30.10
20	18.83	19.87	25.40	13.40	33.30
25	21.13	21.83	21.00	15.90	34.00
30	27.03	24.60	35.80	17.90	38.00

TABLE 8. The average iterations for artificial data set I

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	0.01	0.004	0.001	0.005	0.0301
10	0.08	0.015	0.003	0.015	0.1091
15	0.24	0.039	0.011	0.036	0.2484
20	0.34	0.059	0.012	0.069	0.4403
25	0.80	0.072	0.014	0.157	0.7099
30	1.43	0.124	0.035	0.182	1.0799

TABLE 9. The average centroid errors for artificial data set I

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	99.80	99.95	100	99.93	97.30
10	99.30	99.85	100	99.86	95.00
15	98.50	99.50	100	99.64	91.70
20	97.70	99.04	99.96	99.24	88.00
25	96.70	98.51	99.95	98.48	73.40
30	95.30	97.57	99.62	97.17	50.30

Table 10. The average accuracy percentage for artificial data set II

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	9.43	8.62	8.60	6.90	30.80
10	10.77	10.03	9.30	8.70	47.34
15	12.77	11.07	10.30	9.90	57.03
20	13.77	13.00	13.10	11.60	61.20
25	16.87	14.21	15.00	12.20	68.34
30	20.40	16.28	16.90	13.20	79.34

Table 11. The average iterations for artificial data set II

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	0.02	0.0002	0.0002	0.0002	0.4425
10	0.08	0.0004	0.0005	0.0004	2.9061
15	0.21	0.0008	0.0005	0.0007	13.108
20	0.40	0.0012	0.0005	0.0015	17.459
25	0.72	0.0024	0.0011	0.0031	22.806
30	1.09	0.0063	0.0014	0.0095	30.112

Table 12. The average centroid errors for artificial data set II

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	96.52	100	100	100	91.58
10	93.64	99.97	99.64	99.96	85.02
15	85.77	99.85	99.36	99.59	80.04
20	73.83	98.99	98.34	98.98	70.82
25	65.05	97.72	96.64	97.58	63.42
30	55.50	96.99	95.09	96.43	51.16

Table 13. The average accuracy percentage for artificial data set III

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	8.64	8.00	8.60	9.40	100
10	10.82	10.00	12.80	13.20	100
15	39.10	11.00	15.00	16.20	100
20	44.00	13.20	17.40	17.40	100
25	65.34	15.00	19.40	20.00	100
30	75.00	17.60	21.30	24.80	100

Table 14. The average iterations for artificial data set III

Missing Values (%)	OSCPFCM (%)	NPSPFCM (%)	OCSFCM (%)	NPSFCM (%)	KFCM (%)
5	0.0251	0.00006	0.00009	0.00009	0.0058
10	0.1736	0.00016	0.00013	0.00014	0.0343
15	0.1769	0.00025	0.00030	0.00025	0.0895
20	0.1934	0.00056	0.00067	0.00068	0.1142
25	0.2640	0.00095	0.00135	0.00098	0.1455
30	0.3494	0.00113	0.00120	0.00238	0.1637

TABLE 15. The average centroid errors for artificial data set III

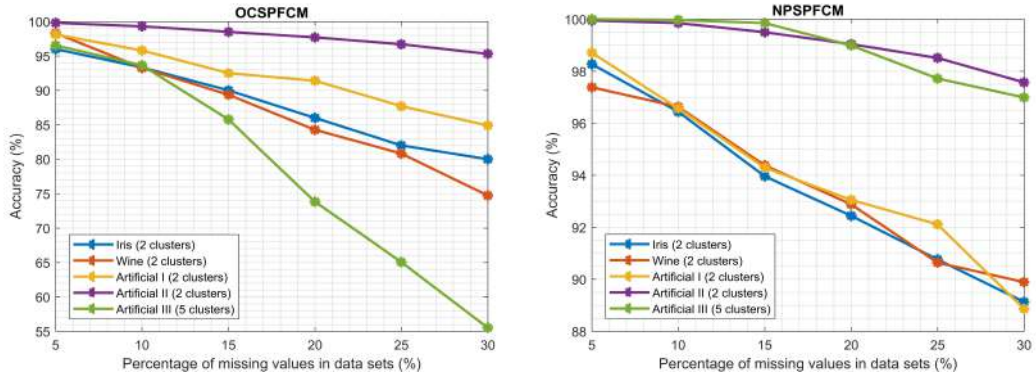

FIGURE 2. Comparison of the percentage accuracy of data sets with two-cluster and five-cluster structure

of two clusters. In contrast, the percentage accuracy of the NPSPFCM on artificial dataset III (five clusters) outperformed other datasets consisting of two clusters, except for artificial data set II, where the NPSPFCM has almost the same accuracy percentage. In general, the NPSPFCM is more reliable than the OCSPFCM for both the data set with two clusters and five clusters.

Figure 3 shows the performance of the number of iterations of the OCSPFCM and NPSPFCM on data sets with two clusters and five clusters. The OCSPFCM requires more iterations for data sets with five clusters than two clusters. However, the number of iterations needed for the wine data sets (two clusters) is higher than the one for artificial data sets III (five clusters). Meanwhile, the NPSPFCM provides an efficient number of iterations, for both the data sets with a two-cluster structure and the data sets with a five-cluster structure. These results indicate the NPSPFCM outperformed the OCSPFCM in the number of iterations for both the two-cluster and the five-cluster data sets.

Figure 4 shows the centroid errors of the OCSPFCM and NPSPFCM for data sets with two clusters and five clusters. Centroid errors for wine data sets (two clusters) for the OCSPFCM and NPSPFCM are not displayed because they are in the order of 10^2, so it would cause the other data sets (two clusters and five clusters) with the order of 10^{-2} to not be visible. In Figure 4, we can see that the OCSPFCM has a relatively smaller error centroid for the artificial data set III (five clusters) than for the artificial data sets I and II (two clusters), however, not smaller than for the iris data sets (two clusters). Meanwhile, the NPSPFCM has the smallest centroid errors for the artificial dataset III (five clusters) as compared to other data sets with a two-cluster structure.

From the experiments that have been carried out on the aforementioned data sets, it can be concluded that our algorithm is robust, specifically for data sets with a two-cluster and five-cluster structure.

We also plot the objective value of our algorithm at each iteration as shown in Figures 5 and 6. For each data set the objective value decreases monotonically and converges over few iterations. Subsequently, the average running time of the OCSPFCM for the iris, wine, artificial data sets I, II, and III are 3.58 second, 4.17 second, 4.49 second, 4.86 second, and 13.86 second, respectively. Meanwhile, the average running time of the NPSPFCM for the iris, wine, artificial data sets I, II and III are 3.70 second, 4.39 second, 4.61 second, 4.79 second, and 12.87 second, respectively. There is no significant difference in the running time for the OCSPFCM and NPSPFCM.

The experiments carried out on the five data sets showed that the proposed algorithms have the potential and ability to cluster incomplete data sets. The results also showed that in the terms of the percentage accuracy, the NPSPFCM always outperformed the OCSPFCM. The authors also performed comparisons with some existing incomplete data clustering algorithms including the KFCM, OCSFCM, and NPSFCM. The result showed that in the case of iris data sets, the OCSFCM and NPSFCM outperformed the NPSPFCM, while in the case of wine data sets, the NPSPFCM outperformed others. In the case of arti-

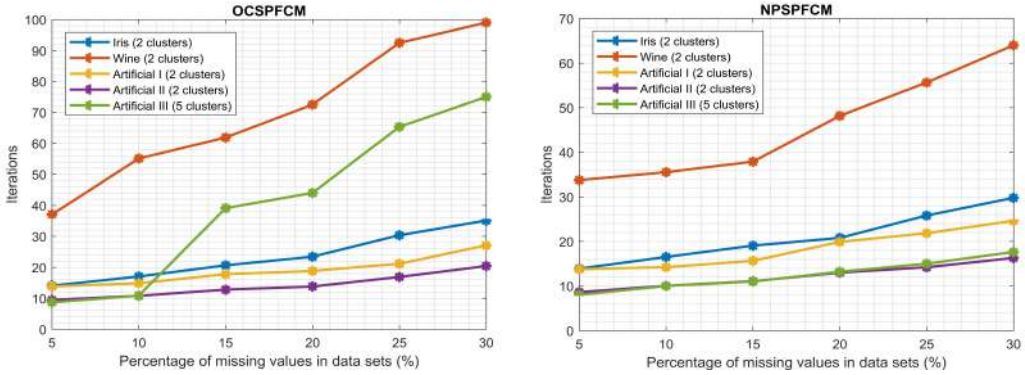

Figure 3. Comparison of the number of iterations of data sets with two-cluster and five-cluster structure

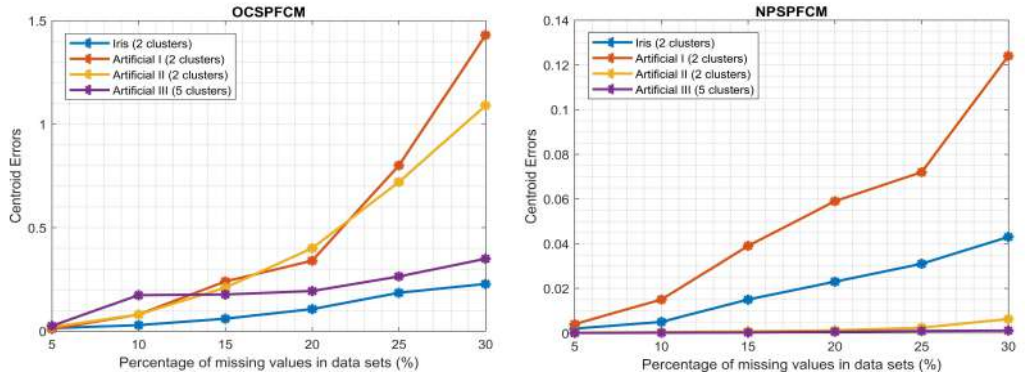

Figure 4. Comparison of the centroid errors of data sets with two-cluster and five-cluster structure

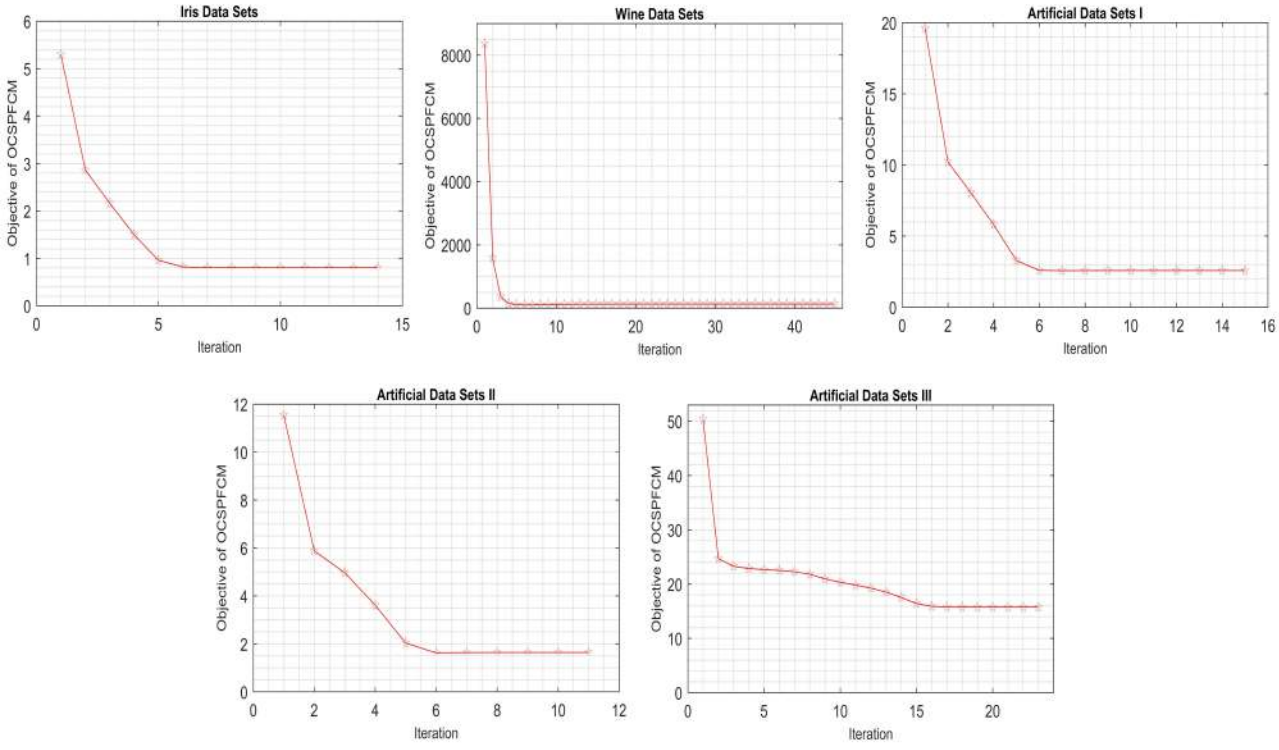

Figure 5. The objective values of the OCSPFCM algorithm

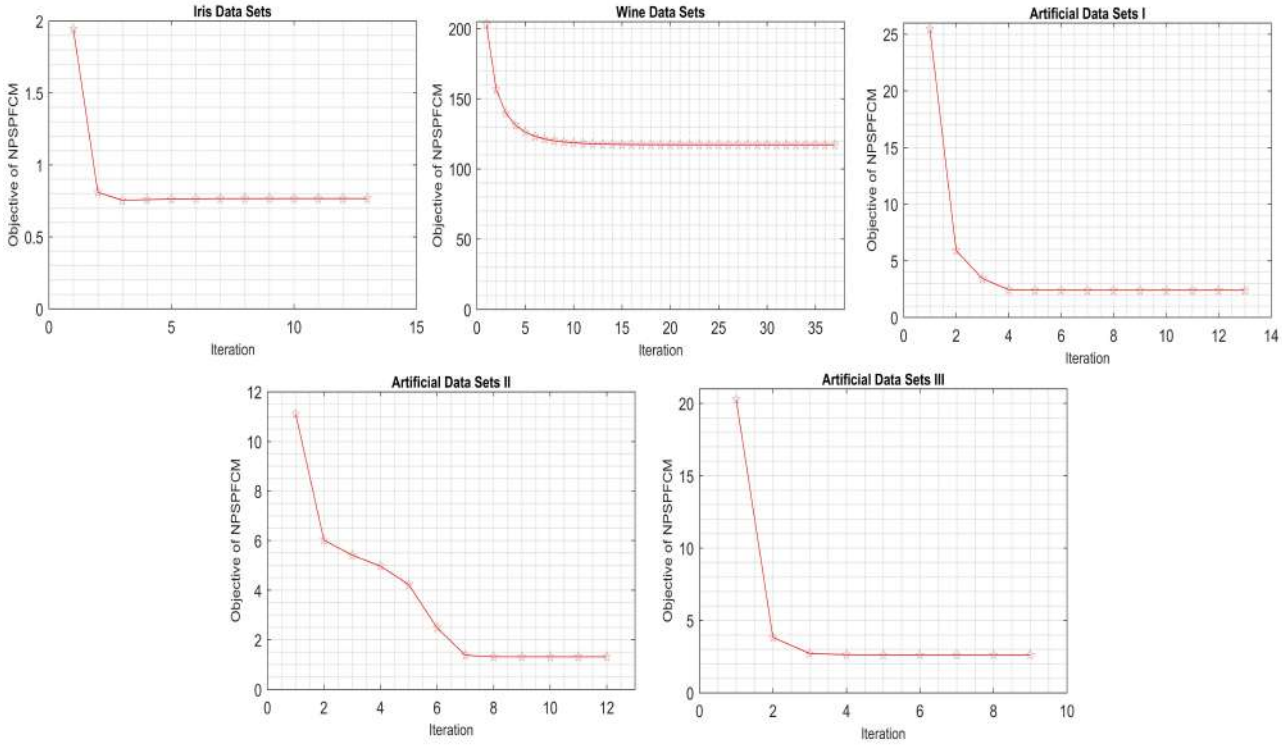

FIGURE 6. The objective values of the NPSPFCM algorithm

ficial data sets I and II, the OCSFCM outperformed NPSPFCM, which outperformed the NPSFCM and KFCM. In the case of artificial data sets III, containing larger data sets, the NPSPFCM outperformed all others, thereby making it the best tool for larger incomplete data sets. By taking the average percentage accuracy for all experiments data sets, it was found that the OCSPFCM and NPSPFCM provide an average accuracy between 97.75% -78.98% and 98.86% -92.49%, respectively.

The performance of the NPSPFCM algorithm in terms of the number of iterations for all data sets is better than that of the OCSPFCM, except for the artificial data set I, in this case, the OCSPFCM outperformed the NPSPFCM. However, in general, the number of iterations in the case of the artificial data set I was relatively equal between the OCSPFCM and NPSPFCM algorithms. The efficient number of iterations of the NPSPFCM was due to the fact that the missing values imputation are updated using values available at the nearest cluster center. This accelerates the convergence of cluster centres directly, or the condition $\left\|\mathbf{v}_i^{(l)} - \mathbf{v}_i^{(l-1)}\right\| < \varepsilon$ is achieved faster by the NPSPFCM algorithm than the OCSPFCM algorithm. Meanwhile, in case of the OCSPFCM algorithm, the imputation of the missing values was updated using the sum of fuzzy and possibilistic membership degrees, which is multiplied by available values in the cluster centre. Therefore, it causes the slower convergence of the OCSPFCM.

Finally, the performance of the OCSPFCM and NPSPFCM algorithms was evaluated on centroid errors in each data set. The results showed that the centroid errors of the NPSPFCM algorithm for all data sets are smaller than that of the OCSPFCM algorithm. The smaller centroid errors of the NPSPFCM algorithm can be attributed to its ability to produce cluster centres for incomplete data sets with a location that is not far from cluster centres of the base data sets. This is the implication of the process of updating the imputation of the missing values, where the NPSPFCM algorithm produces values close to the actual value.

6. Conclusions

In conclusion, this study analysed the potential and performance modification of the PFCM algorithm for clustering incomplete data sets. These modifications, which emerge from the PFCM, as OCSPFCM and NPSPFCM are associated with incomplete data sets clustering. Therefore, this research is divided into three stages. In the first stage, a clustering of complete data sets was carried out using the PFCM algorithm. The cluster results obtained at this stage are the base for evaluating the performance of the OSCPFCM and NPSPFCM algorithms. Furthermore, the performance of the OCSPFCM and NPSPFCM was analysed based on three parameters, accuracy percentage, the number of iterations, and centroid errors. In the second stage, the complete data sets were made incomplete with missing values at predetermined percentages. In the third stage, the incomplete data sets were clustered using the OCSPFCM and NPSPFCM. The results showed that both algorithms have the

potential to cluster incomplete data sets. However, the NPSPFCM outperforms the OCSPFCM based on the three evaluated processes. The NPSPFCM outperforms the OCSPFCM with the missing values ranging from 5%-30% for all experimental data sets. Therefore, this research recommends the use of the NPSPFCM for clustering incomplete data sets. Furthermore, the modification of the PFCM proposed in this research has enriched the reference of the incomplete data set clustering algorithm.

Acknowledgements

The authors would like to express their gratitude to Telkom University for its financial support.

List of symbols

$\mathbb{R}^d$ Space of real d-vectors
$\mathbf{x}_k$ Data vector (data point)
$\mathbf{v}_i$ Cluster centre
d_{ik} uclidean distance between $\mathbf{x}_k$ and $\mathbf{v}_i$
u_{ik} Fuzzy membership degree
t_{ik} Possibilistic membership degree
α Importance level of u_{ik}
β Importance level of t_{ik}
δ_i Possibilistic typicality
m Fuzzy parameter
τ Possibilistic parameter

References

[1] L. Himmelspach. *Fuzzy clustering of incomplete data.* Ph.D. thesis, 2016.

[2] J. C. Bezdek, R. Ehrlich, W. Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences* **10**(2-3):191–203, 1984. doi:10.1016/0098-3004(84)90020-7.

[3] R. Krishnapuram, J. M. Keller. A possibilistic approach to clustering. *IEEE transactions on fuzzy systems* **1**(2):98–110, 1993. doi:10.1109/91.227387.

[4] R. J. Hathaway, J. C. Bezdek. Fuzzy c-means clustering of incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **31**(5):735–744, 2001. doi:10.1109/3477.956035.

[5] J. K. Dixon. Pattern recognition with partly missing data. *IEEE Transactions on Systems, Man, and Cybernetics* **9**(10):617–621, 1979. doi:10.1109/TSMC.1979.4310090.

[6] N. R. Pal, K. Pal, J. M. Keller, J. C. Bezdek. A possibilistic fuzzy c-means clustering algorithm. *IEEE transactions on fuzzy systems* **13**(4):517–530, 2005. doi:10.1109/TFUZZ.2004.840099.

[7] Y. Jiang, K. Zhao, K. Xia, et al. A novel distributed multitask fuzzy clustering algorithm for automatic mr brain image segmentation. *Journal of medical systems* **43**(5):118, 2019. doi:10.1007/s10916-019-1245-1.

[8] T. Ren, H. Wang, H. Feng, et al. Study on the improved fuzzy clustering algorithm and its application in brain image segmentation. *Applied Soft Computing* **81**:105503, 2019. doi:10.1016/j.asoc.2019.105503.

[9] N. X. Thao, M. Ali, F. Smarandache. An intuitionistic fuzzy clustering algorithm based on a new correlation coefficient with application in medical diagnosis. *Journal of Intelligent & Fuzzy Systems* **36**(1):189–198, 2019. doi:10.3233/JIFS-181084.

[10] Y. Li, J.-c. Fan, J.-S. Pan, et al. A novel rough fuzzy clustering algorithm with a new similarity measurement. *Journal of Internet Technology* **20**(4):1145–1156, 2019. doi:10.3966/160792642019072004014.

[11] I. Škrjanc, S. Blažič, E. Lughofer, D. Dovžan. Inner matrix norms in evolving cauchy possibilistic clustering for classification and regression from data streams. *Information Sciences* **478**:540–563, 2019. doi:https://doi.org/10.1016/j.ins.2018.11.040.

[12] A. Koutsibella, K. D. Koutroumbas. Stochastic gradient descent possibilistic clustering. In *11th Hellenic Conference on Artificial Intelligence*, pp. 189–194. 2020. doi:10.1145/3411408.3411436.

[13] L. Zhang, W. Lu, X. Liu, et al. Fuzzy c-means clustering of incomplete data based on probabilistic information granules of missing values. *Knowledge-Based Systems* **99**:51–70, 2016. doi:10.1016/j.knosys.2016.01.048.

[14] Rustam, A. Y. Gunawan, M. T. A. P. Kresnowati. The hard c-means algorithm for clustering indonesian plantation commodity based on metabolites composition. In *Journal of Physics: Conference Series*, vol. 1315, p. 012085. IOP Publishing, 2019. doi:10.1088/1742-6596/1315/1/012085.

[15] X. L. Xie, G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (8):841–847, 1991. doi:10.1109/34.85677.

[16] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics* **7**(2):179–188, 1936. doi:10.1111/j.1469-1809.1936.tb02137.x.

[17] M. Forina, S. Lanteri, C. Armanino, et al. Parvus-an extendible package for data exploration, classification and correlation, institute of pharmaceutical and food analysis and technologies, via brigata salerno, 16147 genoa, italy (1988). *Av Loss Av O set Av Hit-Rate* 1991. doi:10.1002/cem.1180040210.

[18] D. Dua, C. Graff. UCI machine learning repository 2017. `http://archive.ics.uci.edu/ml`.

[19] Rustam, A. Y. Gunawan, M. T. A. P. Kresnowati. Artificial neural network approach for the identification of clove buds origin based on metabolites composition. *Acta Polytechnica* **60**(5):440–447, 2020. doi:10.14311/AP.2020.60.0440.

[20] M. K. Pakhira, S. Bandyopadhyay, U. Maulik. Validity index for crisp and fuzzy clusters. *Pattern recognition* **37**(3):487–501, 2004. doi:10.1016/j.patcog.2003.06.005.

[21] D. L. Davies, D. W. Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* (2):224–227, 1979. doi:10.1109/TPAMI.1979.4766909.

[22] D. Zhang, M. Ji, J. Yang, et al. A novel cluster validity index for fuzzy clustering based on bipartite modularity. *Fuzzy Sets and Systems* **253**:122–137, 2014. doi:10.1016/j.fss.2013.12.013.

[23] R. N. Dave. Validating fuzzy partitions obtained through c-shells clustering. *Pattern recognition letters* **17**(6):613–623, 1996. DOI:10.1016/0167-8655(96)00026-8.

[24] D.-Q. Zhang, S.-C. Chen. Clustering incomplete data using kernel-based fuzzy c-means algorithm. *Neural processing letters* **18**(3):155–162, 2003. DOI:10.1023/B:NEPL.0000011135.19145.1b.