

Real World Data Clustering using a Hybrid of Normalized Particle Swarm Optimization and Density-Sensitive Distance Measure

Temitayo Fagbola, Olugbara Oludayo and Surendra Thakur
ICT and Research Society Research Group,
Department of Information Technology, Durban University of Technology,
4000 Durban, South Africa
temitayo.fagbola@gmail.com, oludayoo@dut.ac.za, thakur@dut.ac.za

Abstract: k-means is among the most widely used classical partitioned clustering algorithms mainly because of its quick convergence rate, adaptability nature to sparse data and simplicity of implementation. However, it only guarantees convergence of sum of square's objective function to a local minimum while its convergence to global optimum appears NP-hard when introduced to large, noisy and non-convex structures. This in turn maximizes its error margin. Most currently existing improvements on k-means adopt techniques which further introduce additional challenges including inaccurate clustering results, high space and time complexities and sometimes premature convergence on k-means. However, high accuracy with large datasets, robustness to noisy data, low clustering time and low sum-of-squared error are sought-after capabilities of good clustering algorithms. In this study, a hybrid Normalized Particle Swarm Optimized-Density Sensitive (NPSO-DS) k-means algorithm is developed to manage the aforementioned limitations of k-means. The proposed NPSO-DS k-means algorithm combines the global stability feature of the normalized Particle Swarm Optimization (PSO) technique incorporating a min-max technique and a clustering error as objective function with the stable properties of a density-sensitive k-means to realize convergence of particles to global optimum with large and noisy real-world datasets. Using clustering accuracy, sum-of-squared error and clustering time as performance metrics, the experimental evaluation results obtained when the developed algorithm was tested on Educational Process Mining (EPM) and wine datasets indicate that it is significantly capable of consistently yielding high quality results. Furthermore, the developed NPSO-DS k-means algorithm could identify non-convex clustering structures and offers appreciable robustness to noisy data, thus, generalizing the application areas of the baseline k-means algorithm.

Key words: k-means, normalized particle swarm optimization, clustering, real world dataset, density-sensitive distance metric, min-max normalization

INTRODUCTION

Clustering is a data mining technique that involves the grouping of a set of data objects into multiple groups called clusters such that each object in a cluster share very close similarity attributes that distinguish them from distinct objects in the other clusters (Joshi and Kaur, 2013; Padhy *et al.*, 2012). The attribute values of each object are distinctive characteristics used to assess the level of dissimilarities and similarities that uniquely differentiate one object from the others. Clustering algorithms have been applied to a number of scientific problem domains including exploratory data analysis, image segmentation, recommender systems, web handling, pattern recognition, medical imaging analysis and mathematical programming have been developed using Chen and Zhang (2007), Han and Kamber (2006) and

Romero and Ventura, 2007). Owing to the large volume of data collected in databases, analysis of clusters has recently become a major research area of interest to many researchers. There are several applications where it is paramount to cluster a large collection of patterns. For example, in document retrieval, millions of instances with high dimensionality spanning beyond 100 have to be clustered to achieve data abstraction (Adebisi *et al.*, 2012). Similarly, the vagueness that characterizes the border of region of most real-world data makes accurate clustering very difficult. Therefore, clustering algorithms are expected to yield high quality outputs especially with large and noisy real world datasets.

k-means is among the most widely used classical partitioned clustering algorithm because of its quick convergence rate, adaptability nature to sparse data and simplicity of implementation (Mahmood *et al.*, 2015). It is

characterized by Euclidean distance, a default non-convex objective function which often fails in an attempt to obtain correct clusters for data points with convex distribution (Wang *et al.*, 2012). Since, global consistency of data is pertinent to accurate clustering, Euclidean Distance Measure (EDM) is highly undesirable especially when clusters have such complex structure and random distributions (Su and Chou, 2001). Consequently, the error gap in k-means performance becomes widened as k-means could only converge to local minima due to its associated EDM. In addition, k-means has a strong sensitivity to noisy data (Adigun *et al.*, 2014). If there is a certain amount of noise associated with a dataset, the final clustering outputs by k-means become impaired with errors (Zhou *et al.*, 2004). In the same vein, potential errors that may evolve when k-means is used to cluster certain real-world critical datasets emerging from medical, security and finance sectors can be highly expensive. This makes k-means less suitable for clustering noisy and large real-world datasets (Zheng *et al.*, 2014; Vrma and Kuma, 2014).

However, most currently existing improvements on k-means adopt techniques including genetic algorithm (Jeong and Gautam, 2012), principal component analysis (Sethi and Mishra, 2013), expectation maximization (Adigun *et al.*, 2014), MapReduce and grid (Zheng *et al.*, 2015) to improve the performance of k-means. However, these adopted techniques often induce some additional performance drawbacks including longer steps before convergence, curse of dimensionality inaccurate clustering results, high space and time complexities as well as premature convergence. Most of these works were tested only on controlled and limited dataset size. Emphatically insensitivity to noisy data, high accuracy obtainable from large datasets, low clustering time and low sum-of-squared error are sought-after capabilities of good clustering algorithms (Mathew *et al.*, 2013). As a result, a modified k-means that could offer global convergence with quality results in the face of large and noisy real-world datasets is highly desirable.

Particle Swarm Optimization (PSO) is considered as a leading and effective metaheuristic method that could offer improved precision, runtime efficiency and robustness of results (Olugbara *et al.*, 2015; Shinde and Gunjal, 2012) in lieu of its robustness to noise and its ability to efficiently find an optimal set of feature weights in large-dimensional complex features (Ayodele *et al.*, 2016) via a global search. It is an evolutionary algorithm that mimics the schooling and the flocking social behaviors of fishes and birds, respectively (Kennedy and Eberhart, 1995). Characteristically, it is fast, very quick to implement and understand, requires very few parameter

settings and computationally efficient. Furthermore, it has been adopted widely to optimize the performance of other algorithms for solving clustering problems (Chen and Zhang, 2017; Niu and Huang, 2011; Sun *et al.*, 2006), scheduling problems (Weijun *et al.*, 2004; Koay and Srinivasan, 2003), medical imaging (Kaur and Bal, 2017; Keshtkar and Geaieb, 2006) and anomaly detection problems (Karami and Guerrero-Zapata, 2015; Adigun *et al.*, 2014) among others. In this study, a Normalized PSO (NPSO) based on min-max technique and an integrated clustering error as the objective function was developed for prior pre-processing of complex, noisy and large datasets before final clustering by k-means. The euclidean distance measure in k-means was replaced with a density-sensitive distance metric to maximize the speed and improve the tendency of k-means to attain global convergence. Finally, a hybrid Normalized Particle Swarm Optimized-Density Sensitive (NPSO-DS) k-means algorithm is developed as a major improvement over the baseline k-means and its existing modifications.

The three major contributions of this study are mentioned as follows: developed a modified Particle Swarm Optimization (PSO) algorithm leveraging on min-max normalization technique and termed Normalized PSO (NPSO) that uses clustering error as the objective function. This algorithm could be used as a dimensionality reduction technique capable of eliminating noise, managing the inherent curse of dimensionality associated with most real-world datasets and evaluating particle's fitness for optimal selection of feature sets in classical data mining problems domain.

Developed a hybrid algorithm from NPSO and density-sensitive k-means. This algorithm can be easily applied to solving any feature selection and dimensionality reduction problem characterized by large and noisy data with complex structures. It can also be integrated seamlessly with any classification system to improve its quality.

The developed hybrid Normalized Particle Swarm Optimized-Density Sensitive (NPSO-DS) k-means algorithm was evaluated quantitatively on the public Educational Process Mining (EPM) and wine datasets using clustering accuracy sometimes referred to as rand index, sum of squared error and clustering time as metrics.

Literature review: Clustering is a common approach for statistical machine learning-based data analytics that has been widely employed in a number of challenging domains like pattern recognition, medical imaging, bioinformatics and social media analytics among others

(Su and Chou, 2001). Actually, clustering is an unsupervised systematic learning approach that attempts to group a finite set of closely related samples into one group called a cluster. Given an untagged dataset, it is required to put like samples in a cluster such that each cluster possesses maximum intracluster and minimum intercluster similarities based on some indices (Joshi and Kaur, 2013). However, finding high-dimensional clusters in spaces is computationally expensive and may reduce the learning performance of most learning systems.

k-means clustering algorithm: k-means is a commonly used algorithm in the field of data mining to solve various clustering problems. k-means is a partitioning clustering technique that allows for the formation of clusters with centroids. With the centroids, clusters vary with different iterations (Zhu, 2006). Moreover, data elements can be re-assigned to a different cluster as the due to randomness of the initial centroids. That is the choice of the initial centroids determines how clusters are formed. An arbitrary number of K data elements is chosen as initial centers, then Euclidean distance measure is used to calculate the distances of the data elements. Data elements are mapped to the appropriate clusters based on their proximities to the centroids iteratively until no more changes is observed. The clusters generated are non-hierarchical in nature, requires large initial size of data elements to proceed and with the possibility of not converging (Twinkle *et al.*, 2014). It is simple to implement and modify its objective function by optimizing the intra-cluster similarity. k-means is applicable only when the mean is defined and it terminates at a local optimum because it depends on gradient descent algorithm (Shen, *et al.*, 2010). This limited its ability to handle noise and outliers. The pseudocode description of k-means is presented in algorithm 1. k-means performs well with super-sphere data distributions using euclidean distance but often fails given data characterized by more difficult and rare shapes indicating the inappropriateness of euclidean distance measure for data elements with random distributions. In this case, there arises a need for a more intuitive objective function for k-means, on one hand to realize high intra-cluster (within-cluster) similarity and low between-cluster (inter-cluster) similarity and for robustness to noisy, complex and large datasets with arbitrary shaped clusters.

Algorithm 1; Conventional k-means (Arthur and Vassilvitskii, 2007):

Input: Number of initial centroids K

Output: K clusters

Let $D = \{d_1, d_2, \dots, d_n\}$ be set of data objects

- (1) Specify the size of K for D
- (2) Randomly select k centroids in the dataset, D or select first k instances
- (3) Calculate the arithmetic means of all data points to the centroids in D

- (4) Distribute each data point to its nearest cluster using the closest Euclidean distance
- (5) Re-compute new centroids by taking the mean of the observations distributed to a cluster
- (6) Repeat steps 3-5 until no more change is observed or convergence condition is satisfied
- (7) Stop

Feature selection for clustering: Feature selection problem is pervasive in all domains of application of machine learning and data mining including but not limited to product image classification, robotics and pattern recognition, text categorization and medical applications especially for diagnosis, prognosis and drug discovery (Temitayo *et al.*, 2012; Guyon, 2008). Feature subset selection is a probabilistic or randomized selection of inputs with the quest to search for near-optimal or optimal subset of features (highly discriminating features) based on some specified criteria. In other words, features that are capable of discriminating samples belonging to different classes are identified and selected. This is an important step to realizing effective utilization of computational resources and some cost savings. It often provides increased understanding of the data, the model and prediction performance (Temitayo *et al.*, 2012). Feature Selection Algorithms (FSA) seek for discriminating features that can potentially reduce the dimensionality size of the feature space with the no or little effect on classification accuracy. Meaning with a set of features, D, a subset of size $d < D$ having high discrimination power is chosen by the algorithm. This process is often tagged as a NP-hard problem. Thus with large input spaces, the high computational load of optimal methods necessitates the use of heuristic techniques to find near-optimal subsets in relatively reduced computational times.

An exploratory study of some widely used FSA including variants of Sequential Forward/Backward Selection (SFS/SBS) and relaxed branch and bound was conducted by Kudo and Sklansky. Other approaches include Genetic algorithms (Siedlecki and Sklansky, 1988), floating search (Pudil *et al.*, 1994), the Tabu search metaheuristic (Zhang and Sun, 2002), simulated annealing (Siedlecki and Sklansky, 1988) and Particle Swarm Optimization (PSO) (Shinde and Gunjal, 2012). However, PSO emerged as a leading metaheuristic technique for feature selection and multi-thresholding because of its ability to effectively find an optimal class of feature weights that improve precision, runtime efficiency and robustness of results (Olugbara *et al.*, 2015; Shinde and Gunjal, 2012).

By description, PSO is a stochastic, population-based evolutionary algorithm for devising efficient solutions to numerous general optimization problems. PSO simulates the shared behavior happening among the flocking birds

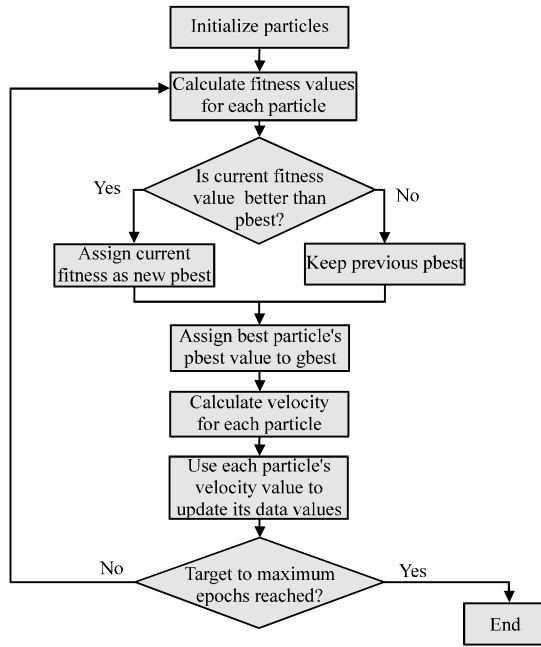


Fig. 1: Flow diagram depicting the behaviour of PSO algorithm (Oludare *et al.*, 2014)

and schooling fishes (Pavlik *et al.*, 2008). It is computationally cheap due to its low memory and CPU requirements and can easily be implemented (Bai, 2010; Eberhart *et al.*, 1996). Additionally, it is robust to overfitting often encountered by most evolutionary algorithms (Kennedy and Eberhart, 1995). The search can be conducted using the speed of the particle. It also depends on a population of individuals to discover favorable regions of the search boundary. Each data element is referred to as a particle and the group of all elements is known as a swarm. The goal of PSO is to locate the particle position that produces the best evaluation for a given objective (fitness) function. The flow diagram of a typical PSO algorithm is shown in Fig. 1. PSO searches the problem space via a manipulation of the moving point trajectories within a multidimensional space. The movement of the particles towards an optimal solution space is managed by the velocity and position of each individual with known previous best performance and that of their neighbors. All particles receive the broadcast of all best positions known to all swarm particles. The relationship among all particles are often conceptualized as a graph $G = \{V, E\}$ where, V depicts a swarm particle and E as an edge that connects the particles together. Generally, the baseline PSO algorithm is composed of three steps which are the generation of particle's positions and velocities, velocity update and position update (Oludare *et al.*, 2014). First, the positions

x_{id} and Velocities V_{id} of the swarm of particles are randomly initialized and obtained using the lower and upper bounds of the search variable values, LB and UB, expressed as:

$$x_{id} = LB + \text{rand}(UB - LB) \quad (1)$$

$$V_{id} = \frac{LB + \text{rand}(UB - LB)}{\Delta t} \quad (2)$$

In Eq. 1 and 2, depicts a uniformly distributed random variable with a value between 0 and 1. This process of initialization allows for random distribution of the swarm particles across the search space. Afterwards, swarm updates its best value at every cycle in order to find the optimized solution after several iterations using (Eberhart and Shi, 2001):

$$V_{id}(t+1) \leftarrow w * V_{id}(t) + V_{id}(t+1) \leftarrow w * V_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)) \quad (3)$$

And:

$$x_{id}(t+1) \leftarrow x_{id}(t) + V_{id}(t+1) \quad (4)$$

where, $V_{id}(t)$ is the velocity of the particle i in the time point t in the search region along the dimension d . $p_{id}(t)$ is the best position that previously offered the particle a high fitness value, p_{best} , $x_{id}(t)$ is the immediate position of the particle i in the search region, r_1 and r_2 are generated randomly with values within a $[0, 1]$ range, $p_{gd}(t)$ is the all-round best position offering a particle the best fitness value, g_{best} , c_1 and c_2 are basically the acceleration parameters while w is inertia weight whose value linearly decreases from 0.9 to 0.4. Furthermore, $x_{id}(t+1)$ is the new position the particle is expected to move to, x_{id} is the current trajectory of the particle and $V_{id}(t+1)$ is the new velocity of the particle that actually indicates the new trajectory location of the particle (Pavlik *et al.*, 2008). Position and velocity updates as well as fitness calculations are all repeated iteratively until the criterion for convergence is fulfilled.

Density-Sensitive Distance Metric (DSDM): Given a density-adjusted Length of Line Segment (LoLS) defined as (Wang *et al.*, 2012):

$$L(x_i, x_j) = \rho^{\text{dist}(x_i, x_j)} - 1 \quad (5)$$

where, $\text{dist}(x_i, x_j)$ is the euclidean distance between x_i and x_j whilst $\rho > 1$ is the flexing factor, the LoLS between two points can be adjusted by re-tuning the flexing factor ρ .

To define a global consistency of particles, let particles be the nodes of graph $G = (V, E)$ and $p \in V^l$ be a path of length $l = |p|$ connecting the nodes p_1 and $p_{|p|}$ in which $(p_k, p_{k+1}) \in E, 1 \leq k < |p|$. With P_{ij} denoting the set of all edges connecting nodes x_i and x_j , the DSDM between two nodes can be defined as (Wang *et al.*, 2012):

$$D_{ij} = \min_{p \in P_{ij}} \sum_{k=1}^{|p|-1} L(p_k, p_{k+1}) \quad (6)$$

Such that, D_{ij} satisfies the four conditions for a metric, that is:

$$D_{ij} = D_{ji} : D_{ij} \geq 0; D_{ij} \leq D_{ik} + D_{kj} \text{ for all } x_i, x_j, x_k \text{ and } D_{ij} = 0 \text{ if } x_i = x_j \quad (7)$$

With these conditions satisfied, the DSDM could measure the geodesic proximity along the surface, possibly producing any two arbitrary points with high density in the same region connected by several shorter edges whilst any two points of high density in a different region are connected by a longer edge through a low density region. That is the proximity of a pair of points is measured by seeking for the least path in the graph, G . This achieves the essence of increasing the distance among data points of disparate high density regions and instantaneously reducing that in the same high density region (Wang *et al.*, 2012). Hence, this distance metric can help converge complex and unstructured data to global optimum.

Min-max data normalization: Normalization is employed to standardize the characteristics of a dataset using a specified preferred criterion, so that, non-discriminating and noisy objects is eliminated and only reliable and discriminating data which can enhance the quality of results are used. Normalization is sometimes used to enhance specific feature measurement methods rather than fix problems. Data normalizations techniques include min-max, Z-score and decimal scaling (Patel and Metha, 2011). However, min-max technique is chosen for this study because it is highly robust to noise (Wang and Bai, 2016). Min-max normalization performs a direct modification of the original data. Suppose that \min_a and \max_a are minimum and maximum values for attribute A . Min-max normalization assigns a value v of Av' to within $[\min_a, \max_a]$ by computing:

$$v' = \frac{v - \min_a}{\max_a - \min_a} \quad (8)$$

Where:

- v' = new value for variable v
- v = The current value for variable

v and \max_a = The minimum and maximum values in the dataset, respectively

Trends of improvement and modifications of k-means clustering algorithm: Over time, several significant modifications to k-means are evident. Hung *et al.* (2005) improved k-means clustering algorithm with a simple partitioning method. The researchers argued that expensive calculation of centroid distances is required, if convergence will be achieved as characterized most modifications to k-means. In their research, binary splitting was used to split up the main dataset into blocks. Each Block Unit (UB) with at least one pattern has its centroid (CUB) determined via a simple calculation in order to generate dominant data subset as a representative of the main dataset. The subset was then applied to determine the final centroid of the main dataset. Each UB was examined on the perimeter of prospective clusters to locate the closest final centroid for each pattern in the UB. In this manner, time to estimate the final converged centroids was dramatically reduced. It was claimed that the algorithm showed better performance with regard to total execution time, number of distance calculations and the efficiency for clustering than other k-means algorithms. However, the improved k-means need more iterations to achieve the k centroids, sometimes even spending the maximum number of iterations do not achieve convergence.

Bolelli *et al.* (2007) developed a K-SV Means for multi-type interrelated datasets by combining SVM and k-means clustering. In a bid to eliminate the use of labeled instances to make SVM learn, the cluster assignments of k-means are used to train an online SVM with an arbitrary secondary data while the SVM affirms k-mean's clustering decisions in the primary clustering margin. This heterogeneous clustering process effectively increases the clustering performance compared to clustering using a single homogeneous data source. The researchers reported results for euclidean and spherical k-means averaged over ten runs. k-means makes assigns clusters based on the euclidean distances between the document vectors while the spherical k-means uses the cosine distances between documents as the similarity metric. The hypothetical results on analysis of newsgroup and citeseer real-world datasets reveal the success of K-SVMeans to discovering topical document clusters and providing more optimal clustering solutions than the homogeneous k-means algorithm. However, K-SVMeans suffers from high computational effort and can give inaccurate result, if the initial dataset adopted for training SVM is very

large. The computational demand for automated SVM parameter settings and training is also a great challenge.

Mary *et al.* (2010) used the Ant Colony Optimization (ACO) to enhance k-means clustering performance. The researchers improved the cluster quality after grouping via a two-phased process. The resultant technique uses Euclidean distance and remains highly responsive to the changes in the initial value of k. This makes it less applicable for clustering real world datasets. Niu and Huang (2011) developed an enhanced k-means algorithm using Kruskal algorithm and tested using gene expression data. Firstly, a Minimum Spanning Tree (MST) of the grouped objects was obtained by Kruskal algorithm. Then in declining order, K-1 edges are deleted depending on the weights. Finally, the mean values of the objects containing the K-connected graphs of the last two steps represent the initial clustering centers to cluster. Results showed that this method is less affected by the initial choice of K than the baseline k-means algorithm and increased the stability and accuracy of clusters. However, the developed k-means algorithm is only suitable for small datasets. When tested on large, complex, vast and real-time datasets, it suffers from high time complexity and high program difficulty.

Adigun *et al.* (2014) developed a hybrid k-means Expectation Maximization (KEM) algorithm to manage the limitations of k-means and Expectation Maximization (EM) algorithms. k-means converges only to local minima after several trials while EM converges prematurely. The hybrid KEM algorithm was developed for both initialization and iterative stages. In the initialization stage, the weighted average variation of k-means was employed to partition the data into desired number of clusters. At the iterative stage, a large number M, of uniformly distributed random cluster point vectors for the cluster centers were selected. Any cluster point vectors that are too close to other cluster point vectors were eliminated and M is reduced accordingly, until the clusters generated equal to the number of clusters wanted. This was achieved by computing the distances between all clusters and eliminating the clusters with distances lesser than a presumed magnitude value. Assigning each feature vector to the closest random cluster point vector was the next step achieved by computing and comparing the proximity of each feature vector with other cluster point vectors. The feature vector was assigned to that cluster point vector with the least proximity. The hybrid algorithm showed computationally efficient and improved accuracy improvements over k-means and EM when tested on real world educational dataset. However, the hybrid KEM still converges to local minima because the k-means

component used Euclidean distance metric and as such not suitable for clustering large real world dataset. The hybrid KEM was not developed to handle noise which characterizes the real world datasets.

Momin and Yelmar (2012) developed a Rough Fuzzy Possibilistic k-means (RFPKM). An overlapping of clusters with lower and upper estimations from rough sets handles uncertainty, vagueness and incompleteness via the membership function of the fuzzy sets. Possibilistic membership functions generate memberships which are compatible with the class center and not coupled with centers of other classes. RFPKM can group categorical data by using probability distribution of categorical values. The evaluation results obtained showed that RFPKM gives reduced value of objective function for categorical data than the baseline k-means and variants considered. However, it produced inaccurate classification with noisy and large datasets. Furthermore, Jeong and Gautam *et al.* (2012) developed a hybrid technique, GAKM by combining k-means and Genetic Algorithm (GA). The objective of GAKM is to learn the cluster centroids and optimal weights of attributes required to partition the dataset. An optimal solution is generated by GA using reproduction, crossover and mutation operators. In GAKM, k-means output was used to adjust the GA parameters. If fitness value is satisfied, the ideal solution is obtained, otherwise, the GA parameters are recombined and re-evaluated to generate an optimal number of clusters. The research did not present any evaluation result. However, it was reported that the developed GAKM performed better than k-means on categorical data. This improvement is at the expense of additional computational overhead. This is because, GA used longer execution steps to obtain optimal number of clusters. Overfitting is also a challenge of the developed GAKM because baseline k-means was implemented using euclidean distance.

Shanmugapriya and Punithavalli (2012) developed an improved projected k-means algorithm using an Effective Distance Measure (EDM) that iteratively enhances an exhaustive objective function. In the objective function of this developed algorithm, the EDM makes use of local and non-local information to provide improved clustering outputs in high dimensional data. Virtual dimensions are often incorporated into the objective function in order to maintain the optimality of the objective function from reducing when dimensions are excluded. It only works efficiently in principle as the developed algorithm was not evaluated. Elbatta and Ashour (2013) investigated issues of prototypes with random initialization and the demand for a pre-determined number of centroids for a dataset with the baseline k-

means. These random initializations force prototypes to local convergence. Based on this rationale, Efficient Data Clustering Algorithm (EDCA) was developed. This algorithm uses definition of density computation of data points by k-nearest neighbor method to calculate the initial number of clusters. Furthermore, noise and outliers which affect k-means strongly were detected. Their result showed slight improvement over the baseline k-means algorithm. EDCA could detect clusters with different non-convex shapes, different sizes and densities. This solution suffers from high computational resources demand especially with highly complex data.

An incremental k-means algorithm that could assign any random data point to the first group of a given set of data points was developed by Gupta and Ujjwal (2013). After selecting the next random object, the proximity between selected object and centroids of existing clusters was determined. This distance was matched with the boundary limit so as to be able to group the data point into any of the existing group or form a new group for it. Computational results showed that the developed algorithm produced clusters in lesser computation time but only with small and noise-free dataset. It cannot handle large and noisy dataset more efficiently because of the rigid nature of the incremental approach used. Zhang and Fang (2013) modified the baseline k-means by improving on the initial convergence point and process of determining the value of K. The centroid was initialized and adjusted. Euclidean distance of the various data points from each centroid was calculated and the square error criterion function was determined to ascertain, if convergence is reached. The improved clustering algorithm added weight of data point to the centroid, so as to eliminate noise level and its impact on the data points. The resulting clustering output of the improved k-means showed improved performance over some variants when evaluated. The developed technique was tested on small-sized dataset with controlled noise and thus not appropriate for real-world data clustering.

Sethi and Mishra (2013) developed a linear Principal Component Analysis (PCA)-based hybrid k-means PSO algorithm for large dataset partitioning. Covariance matrix in PCA was used for dimensionality reduction and centroid locations were estimated using euclidean distance. However, PSO was used to generate the final ideal solution. More generally, PSO could conduct a global search for optimal solution but requires a large number of iteration. The PSO was assisted by k-means to start with good initial centroid position that converge faster thereby yielding a more compact result. k-means output was treated as the initial

input to PSO to find an ideal solution by a globalized search to avoid high computational time complexity. Improved solution was achieved with PCA-based HYBRID (K-PSO) algorithm when compared with PSO only. The hybrid system is complex, converged to local minima given clusters with wide variation in size and shape incurred high computational complexity and was not evaluated with other improved k-means variants.

Furthermore, Zheng *et al.* (2014) used grid and MapReduce to enhance the performance of k-means. In their method, the size of the data point is used to determine which data point will be allocated to a corresponding grid in space. In each grid, the number of data points is counted, then, $M(M > K)$ grids composed of all data points are selected and the centroid calculated. The value of K is determined by the number of centroids in M based on the clustering output. Also, the maximum value in M was included in K such that noisy data are identified as data in the grid should the total data points in the grid be lesser than the threshold. In order to realize robustness to large data, k-means was paralleled and merged with MapReduce. Results obtained reflects a significant improvement of the new method over the baseline k-means with lesser iterations and good stability.

Mahmood *et al.* (2015) argued that the current minimum distance in traditional k-means might not always be the correct minimum distance because the proximity of a centroid to each data point is calculated at every iteration. The number of computations increases and the algorithm becomes more complex. In the improvement suggested by the researchers, a checkpoint value was added to store the center point of the proximity of two centroids and to deduce the cluster an object will be assigned to. This checkpoint value reduced the possibility of error during the clustering process. The researchers reported that the new method can offer higher accuracy at fewer iterations and computational resource demands than the baseline approaches. However, shortage of available resources and time limited the work. The proposed method was not tested on large, complex, vast and real-world datasets.

Wei *et al.* (2015) clustered disparate data using k-means via. Mutual Information-based Unsupervised Feature Transformation (MI-UFT). The research addressed the computational complexities of k-means for large datasets and its sensitivity to outliers. The researchers integrated the MI-UFT which could convert non-numerical features into numerical features with the baseline k-means to partition disparate data. Simulation results indicated that the developed UFT-k-means algorithm improved over other clustering algorithms with

considerable number of clusters for a real-world dataset and five real-world benchmark datasets. However, the developed algorithm is parameter-dependent and computationally highly inefficient. Summarily, most existing improvements on k-means do not suffice its ability to cluster noisy and large data accurately and in highly efficient way while some others are characterized by high computational overhead. Sequel to these, clustering large and noisy dataset using k-means in a computationally-efficient and accurate manner still remains largely an open problem which is addressed in this study.

MATERIALS AND METHODS

The experimental architecture for the hybrid NPSO-DS k-means algorithm is presented in 3 developmental stages including the real-world dataset acquisition, development, development of a Normalized Particle Swarm Optimization and integration of NPSO into a density-sensitive k-means algorithm.

Real-world dataset acquisition: UCI Educational Process Mining (EPM) and wine datasets are the most widely used real world datasets in literatures. These datasets can be accessed and downloaded from <https://archive.ics.uci.edu/ml/datasets>. However, the description of the datasets is presented in Table 1. However, sample EPM and wine datasets are shown in Fig. 2 and 3, respectively.

Table 1: Analysis of the EPM and wine real world datasets

UCI datasets	Instances	Number of attributes	Type of attribute
EPM	230318	13	integer
Wine	178	13	Integer and real

UCI Educational Process Mining (EPM) dataset: This is a publicly-available learning analytics dataset from smartlab located in Italy. It was collected in 2015 and contains the time series of student's activities during 6 laboratory sessions of a course on digital electronics. The student's data per session are contained in 6 folders with 99 csv files each peculiar to each student log for that session. The number of files in each folder changes due to the number of students present in each session. However, the files contain 230318 instances and 13 integer attributes each.

UCI wine dataset: The data contained are the outputs of a chemical examinations of wines brewed in Italy from three different varieties. The volumes of 13 components in each of the wines is informed by the inspection.

The developed normalized PSO algorithm: PSO technique was introduced for dimensionality reduction of the particles to be clustered by k-means. The conventional PSO was modified such that it incorporates the Clustering Error measure (CE) as the objective function. The Clustering Error (CE) can be defined as (Wang *et al.*, 2012):

$$CE(\Delta, \Delta^{\text{true}}) = \frac{1}{n} \sum_{i=1}^{k_{\text{true}}} \sum_{j=1, j \neq i}^k \text{Confusion}(i, j) \quad (9)$$


Where the clustering produced, Δ is given by:

$$\Delta = \{C_1, C_2, \dots, C_k\} \quad (10)$$

The true clustering, Δ^{true} expressed as:

Student ID	ES 1.1 (2 points)	ES 1.2 (3 points)	ES 2.1 (2 points)	ES 2.2 (1 points)	ES 3.1 (2 points)	ES 3.2 (2 points)	ES 3.3 (2 points)	ES 3.4 (3 points)	ES 3.5 (3 points)	ES 4.1 (15 points)	ES 4.2 (10 points)	ES 5.1 (2 points)	ES 5.2 (10 points)	ES 5.3 (3 points)	ES 6.1 (25 points)	ES 6.2 (15 points)	TOTAL (100)
3	2.0	3.0	1.0	2.0	1.0	2.0	2.0	2.0	3.0	15.0	10.0	1.0	5.0	3.0	18.0	15.0	85.0
6	2.0	3.0	2.0	3.0	1.0	2.0	2.0	0.0	3.0	15.0	7.0	2.0	9.0	3.0	13.0	15.0	82.0
7	2.0	3.0	1.0	1.5	1.0	2.0	0.0	0.0	3.0	5.0	4.0	0.0	0.0	3.0	17.0	10.0	52.5
10	2.0	3.0	2.0	1.5	1.0	2.0	0.0	2.0	3.0	11.0	1.0	2.0	10.0	1.5	7.0	10.0	59.0
13	2.0	3.0	2.0	1.5	1.0	2.0	2.0	2.0	3.0	14.5	10.0	2.0	2.0	3.0	25.0	15.0	90.0
15	2.0	3.0	1.0	2.0	1.0	2.0	2.0	2.0	3.0	15.0	10.0	2.0	4.0	1.5	2.0	15.0	67.5
16	2.0	3.0	1.0	0.0	1.0	2.0	2.0	2.0	3.0	3.0	9.0	1.0	0.0	3.0	20.0	15.0	67.0
17	2.0	3.0	1.0	2.0	1.0	2.0	2.0	2.0	3.0	15.0	10.0	2.0	10.0	3.0	24.0	15.0	97.0
18	1.0	3.0	2.0	3.0	1.0	2.0	2.0	2.0	3.0	15.0	7.0	2.0	2.5	1.5	5.0	10.0	62.0
20	2.0	3.0	2.0	0.0	1.0	2.0	2.0	2.0	1.5	15.0	10.0	0.0	0.0	3.0	5.0	10.0	58.5
24	2.0	3.0	1.0	1.5	1.0	2.0	2.0	2.0	3.0	11.0	1.0	0.0	0.0	3.0	0.0	0.0	32.5
27	2.0	3.0	1.0	2.0	1.0	2.0	0.0	2.0	3.0	15.0	9.0	0.0	0.0	0.0	24.5	10.0	74.5
28	2.0	3.0	1.0	2.0	1.0	2.0	2.0	2.0	0.0	15.0	8.5	2.0	4.0	3.0	19.0	13.0	79.5
29	2.0	3.0	2.0	0.0	1.0	2.0	2.0	2.0	3.0	15.0	10.0	2.0	10.0	3.0	25.0	15.0	97.0
30	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	2.0	0.0	0.0	0.0	0.0	0.0	7.0
32	2.0	3.0	2.0	1.5	1.0	2.0	2.0	2.0	3.0	15.0	10.0	2.0	10.0	3.0	12.0	5.0	75.5
36	2.0	3.0	1.0	0.0	1.0	1.0	0.0	2.0	3.0	13.0	9.0	2.0	10.0	3.0	16.0	13.0	79.0
37	2.0	3.0	1.0	0.0	0.0	2.0	0.0	2.0	0.0	8.0	1.0	0.0	0.0	0.0	0.0	0.0	19.0
39	2.0	3.0	2.0	0.0	0.0	0.0	0.0	2.0	0.0	3.0	4.0	0.0	0.0	0.0	10.0	15.0	41.0
41	2.0	3.0	2.0	3.0	1.0	2.0	2.0	2.0	3.0	13.0	10.0	2.0	3.0	25.0	15.0	15.0	98.0

Fig. 2: Sample data of EPM dataset


<https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data>

1	14.23	1.71	2.43	15.6	127	2.8	3.06	.28	2.29	5.64	1.04	3.92	1065	
1	13.2	1.78	2.14	11.2	100	2.65	2.76	.26	1.28	4.38	1.05	3.4	1050	
1	13.16	2.36	2.67	18.6	101	2.8	3.24	.3	2.81	5.68	1.03	3.17	1185	
1	14.37	1.95	2.5	16.8	113	3.85	3.49	.24	2.18	7.8	.86	3.45	1480	
1	13.24	2.59	2.87	21	118	2.8	2.69	.39	1.82	4.32	1.04	2.93	735	
1	14.2	1.76	2.45	15.2	112	3.27	3.39	.34	1.97	6.75	1.05	2.85	1450	
1	14.39	1.87	2.45	14.6	96	2.5	2.52	.3	1.98	5.25	1.02	3.58	1290	
1	14.06	2.15	2.61	17.6	121	2.6	2.51	.31	1.25	5.05	1.06	3.58	1295	
1	14.83	1.64	2.17	14	97	2.8	2.98	.29	1.98	5.2	1.08	2.85	1045	
1	13.86	1.35	2.27	16	98	2.98	3.15	.22	1.85	7.22	1.01	3.55	1045	
1	13.73	2.2	2.3	18	105	2.95	3.32	.22	2.38	5.75	1.25	3.17	1510	
1	14.12	1.48	2.32	16.8	95	2.2	2.43	.26	1.57	5	1.17	2.82	1280	
1	13.75	1.73	2.41	16	89	2.6	2.76	.29	1.81	5.6	1.15	2.9	1320	
1	14.75	1.73	2.39	11.4	91	3.1	3.69	.43	2.81	5.4	1.25	2.73	1150	
1	14.38	1.87	2.38	12	102	3.3	3.64	.29	2.96	7.5	1.2	3.15	1547	
1	13.63	1.81	2.7	17	2	112	2.85	.29	1.46	7.3	1.1	2.8	88	
1	14.3	1.92	2.72	20	120	2.8	3.14	.33	1.97	6.2	1.07	2.65	1280	
1	13.83	1.57	2.62	20	115	2.95	3.4	.4	1.72	6.6	1.13	2.57	1130	
1	14.19	1.59	2.48	16.5	108	3.3	3.93	.32	1.86	8.7	1.23	2.82	1680	
1	13.64	3.1	2.56	15.2	116	2.7	3.03	.17	1.66	5.1	.96	3.36	845	
1	14.06	1.63	2.28	16	126	3	3.17	.24	2.1	5.65	1.09	3.71	780	
1	12.99	3.8	2.65	18.6	102	2.41	2.41	.25	1.93	4.5	1.03	3.52	770	
1	13.71	1.86	2.36	16.6	101	2.61	2.88	.27	1.69	3.8	1.11	4	1035	
1	12.85	1.6	2.52	17.8	95	2.48	2.37	.26	1.46	3.93	1.09	3.63	1015	
1	13.5	1.81	2.61	20	96	2.53	2.61	.28	1.66	3.52	1.12	3.82	845	
1	13.05	2.05	3.22	25	124	2.63	2.68	.47	1.92	3.58	1.13	3.2	830	
1	13.39	1.77	2.62	16	1	93	2.85	.29	1.45	4.8	.92	3.22	1195	
1	13.3	1.72	2.14	17	94	2.4	2.19	.27	1.35	3.95	1.02	2.77	1285	
1	13.87	1.9	2.8	19.4	107	2.95	2.97	.37	1.76	4.5	1.25	3.4	915	
1	14.02	1.68	2.21	16	96	2.65	2.33	.26	1.98	4.7	1.04	3.59	1035	
1	13.73	1.5	2.7	22.5	101	3	3.25	.29	2.38	5.7	1.19	2.71	1285	
1	13.58	1.66	2.36	19	1	106	2.86	.3	1.9	6.9	1.09	2.88	1515	
1	13.68	1.83	2.36	17	2	104	2.42	.26	1.9	3.84	1.23	2.87	990	
1	13.76	1.53	2.7	19	5	132	2.95	.2	1.35	5.4	1.25	3	1235	
1	13.51	1.8	2.65	19	110	2.35	2.53	.29	1.54	4.2	1.1	2.87	1095	
1	13.48	1.81	2.41	20	5	100	2.7	2.98	.26	1.86	5.1	1.04	3.47	920
1	13.28	1.64	2.84	15.5	110	2.6	2.68	.34	1.36	4.6	1.09	2.78	880	
1	13.05	1.65	2.55	18	98	2.45	2.43	.29	1.44	4.25	1.12	2.51	1105	
1	13.0	1.5	2.1	15	5	98	2.4	2.64	.28	1.37	3.7	1.18	2.69	1020
1	14.22	3.93	2.9	13	2	132	3	3.04	.2	2.08	5.1	.89	3	760
1	13.56	1.71	2.31	16	2	117	3.15	3.29	.34	2.34	6.13	.95	3.38	795
1	13.41	3.84	2.12	18	8	90	2.45	2.68	.27	1.48	4.28	.91	3	1035
1	13.88	1.89	2.59	15	101	3	3.25	.35	1.7	1.7	5.43	.88	3.56	1095
1	13.24	3.98	2.29	17	5	103	2.64	2.63	.32	1.66	4.36	.82	3	680

Fig. 3: Sample data of wine dataset

$$\Delta^{\text{true}} = \{C_1^{\text{true}}, C_2^{\text{true}}, \dots, C_{k_{\text{true}}}^{\text{true}}\} \quad (11)$$

$$p = \{p^1, p^2, p^3, \dots, p^n\} \quad (12)$$

And n being the total number of data points. Thus, $\forall i \in [1, \dots, k_{\text{true}}] \quad j \in [1, \dots, k]$ confusion (i, j) denotes the number of same data points both in the true Cluster C_i^{true} and in the Cluster C_j produced. However, there exists a reordering/realigning problem. That is a cluster with true clustering might be assigned to a different cluster when a new cluster is formed. To counter that, the CE is calculated for all possible reordering of new clusters formed and the least of all those is taken. The best clustering performance is such with the smallest CE. The flowchart and use-case diagram of the developed NPSO are presented in Fig. 4 and 5, respectively. Given a group of data points as input, the normalized PSO is expected to return a limited number of discriminant particles. In this study, (5) major steps were followed to develop the NPSO technique for a given set of inputs which are $\{x_i\}_{i=1}^n$ maximum iteration number t_{max} and stop threshold ϵ .

Step 1: Initialization of particles via. random generation to form an initial population where each particle depicts a feasible cluster solution. The number of particles is taken as a product of dataset features and number of clusters to be generated. The dataset represents a swarm and the constituent elements represent the particles. Analytically, swarm consists of a group of particles:

where, n is the features of the dataset.

Step 2: The position and velocity of the particles are initialized, such that, at any time step t the particle p^i has two vectors, position, $x^i(t)$ and velocity, $V^i(t)$ associated. Each candidate solution possesses a position which represents the solution in search space and velocity for the movement of particles for finding global optimal solution. The particle's position and velocity were initialized as in Eq. 1 and 2, respectively.

Step 3: Evaluation of particle's fitness: the objective criterion of each particle was calculated using the clustering error as shown in Eq. 8. However, at each generation, best fitness values were updated using (Saini and Kaur, 2014):

$$P_i(t+1) = \begin{cases} P_i(t) & f(X_i(t+1)) \leq f(X_i(t)) \\ X_i(t+1) & f(X_i(t+1)) > f(X_i(t)) \end{cases} \quad (13)$$

Where:

- f = Depicts the fitness criterion (clustering error)
- $P_i(t)$ = Represents the optimal fitness values and the coordinates where the value was calculated
- $X_i(t)$ = Represents the current position
- t = Symbolizes the generation step

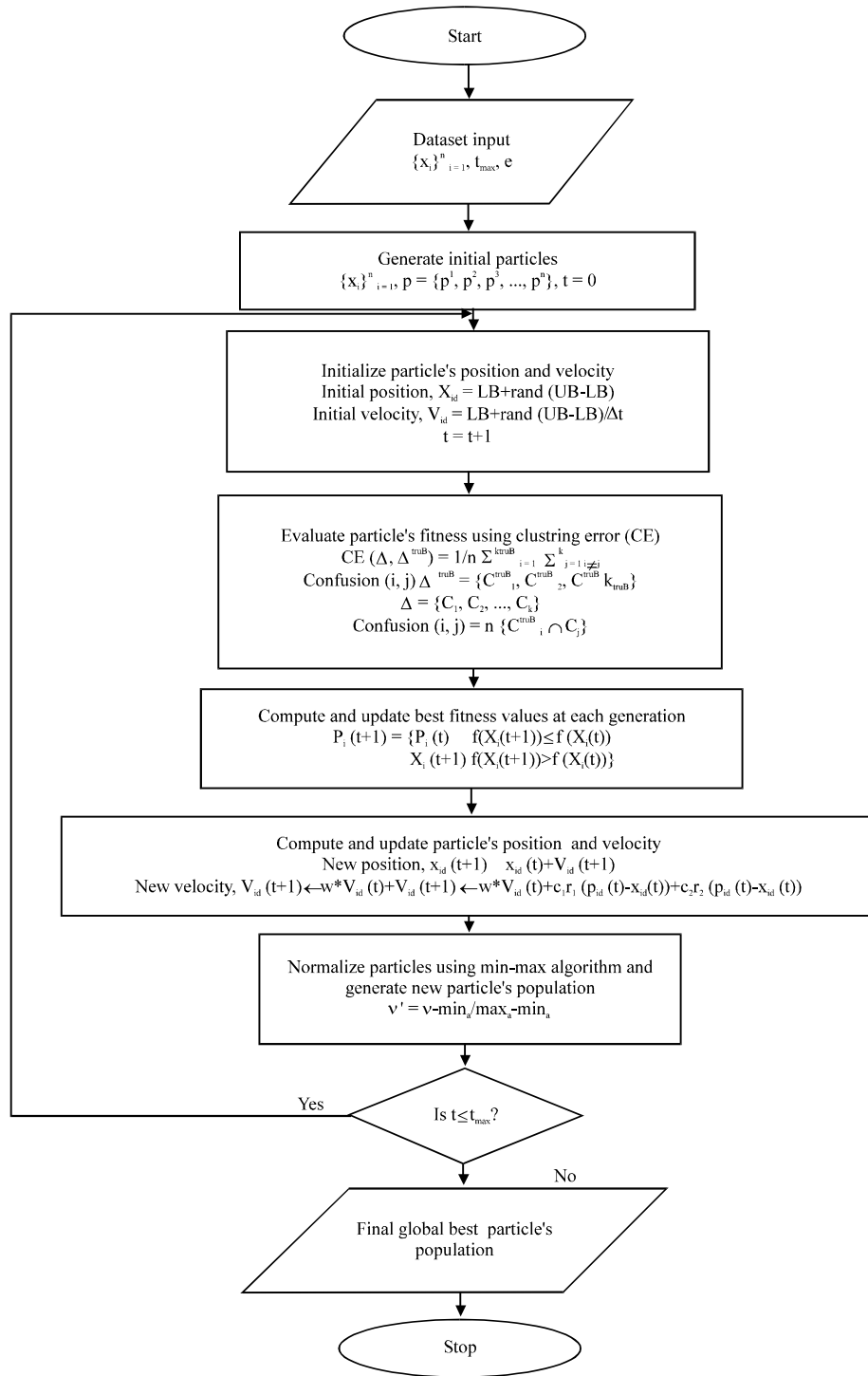


Fig. 4: Flowchart of the developed NPSO

Step 4: Position and velocity update: the exploit for the global optimal solution was carried out through a dynamic update of the particles in swarm. Equation 3 is utilized to update the velocity as a derivative of the initial

velocity, the particle own best performance and the swarm best performance. Position update was achieved by adding incremental change in position at each step using Eq. 4.

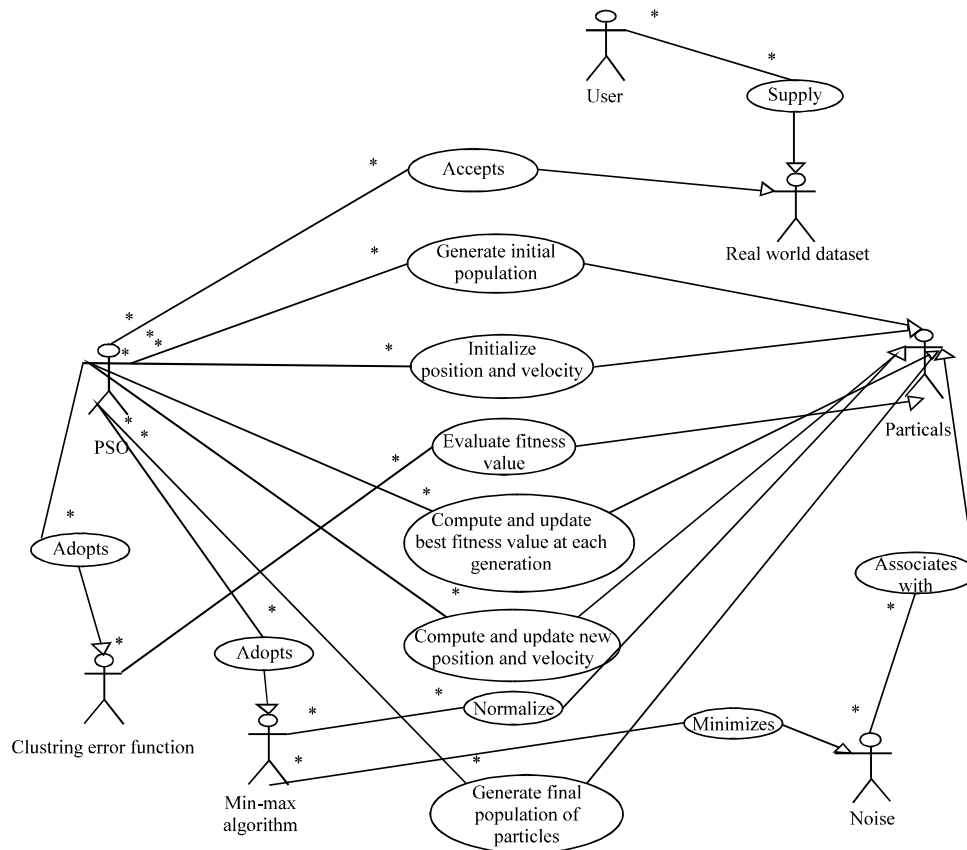


Fig. 5: Use cases diagram for the NPSO

At this step in the conventional PSO, some particles usually move out of search space boundary which often lead to errors and in turn affects the overall output accuracy. This is sometimes because of the presence of noisy data in the dataset (Saini and Kaur, 2014).

In this study, the devastating effect of the noisy data was addressed by forcing relevant particles to remain within the boundary or reset to the boundary value by using min-max normalization function defined in Eq. 7.

Step 5: Steps 2-4 is repeated until one of following termination conditions is satisfied.

- The maximum number of iterations is reached
- The mean change in centroid vectors is less than a predetermined value

After the completion of step 5, the expected output is m data points $\{x_i\}_{i=1}^m$: $m \ll n$.

The hybrid NPSO-Density sensitive k-means algorithm:

The hybrid NPSO-density sensitive k-means is the product of integrating the NPSO algorithm into a density sensitive k-means algorithm.

The corresponding conceptual flow and use cases diagrams are shown in Fig. 6 and 7, respectively. As presented in algorithm 2 with DS-k-means, a density-sensitive distance is incorporated into k-means to replace the Euclidean distance. The justification for this step is borne out of the fact that poor assignment of particles to clusters is inevitable especially where the particle has equal minimum euclidean distance to a number of clusters. Consequently, the centroids are forced to converge to local minimal and as such would be unable to typify data groups as desired (Olugbara *et al.*, 2015). However, employing a density-based objective function is capable of converging to global optimum even with arbitrary and non-convex shaped clusters (Joshi and Kaur, 2013). Clusters can easily be formed by data points located in dense regions while the low density regions separate data points from different clusters.

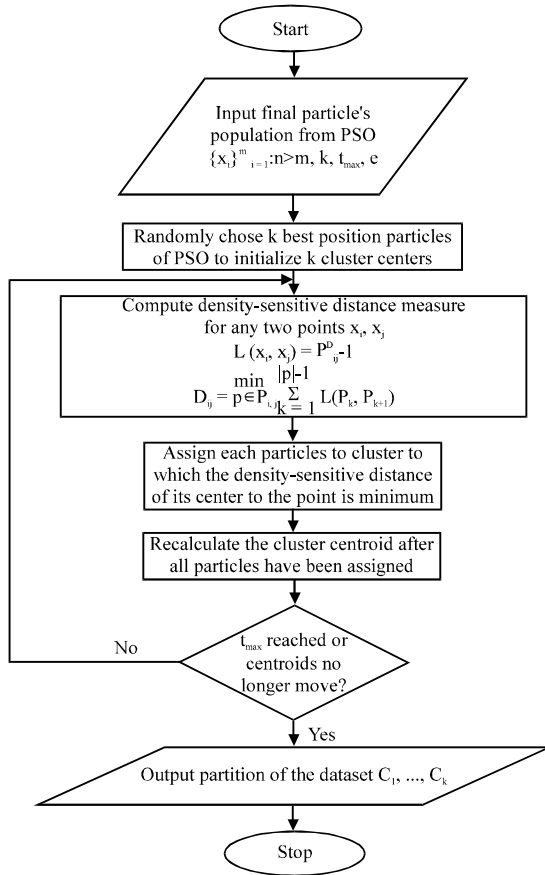


Fig. 6: The flow diagram of the HYBRID NPSO-DS k-means algorithm

Algorithm 2; The hybrid NPSO-DS k-means algorithm:

Input: m data points $\{x_i\}_{i=1}^m$, cluster number k , maximum iteration number, t_{max} stop threshold e

Output: Partition of the dataset C_1, \dots, C_k

- (1) randomly choose k data points using the K best position particles of PSO to initialize k cluster centers
- (2) for any two data points x_i and x_j do
- (3) compute the density-sensitive distance using Eq. 5 and 6
- (4) assign each particle to the closest centroid calculated by the minimum density-sensitive distance
- (5) if all particles have not been assigned, then go to (4) else go to (6)
- (6) recalculate new centroid for each cluster
- (7) end for
- (8) if centroids move or the maximum number of iterations, t_{max} , not reached, then go to (2) else go to (9)
- (9) stop

Performance evaluation metrics: The developed NPSO-DS k-means algorithm was evaluated using the following metrics:

Clustering time: This represents the time requirement to cluster all data points. This parameter depends on the platform where the clustering is implemented and will dictate if real-time functionality is available or not.

Sum-of-Squared Error (SSE): This is the sum of squares of the departure from the average for each calculated value of data (Peng and Xia, 2005):

$$SSE = \sum_{i=1}^n (x_i - \bar{x})^2 \quad (14)$$

where, n denotes the number of particles and x_i represents the actual value of the i th particle.

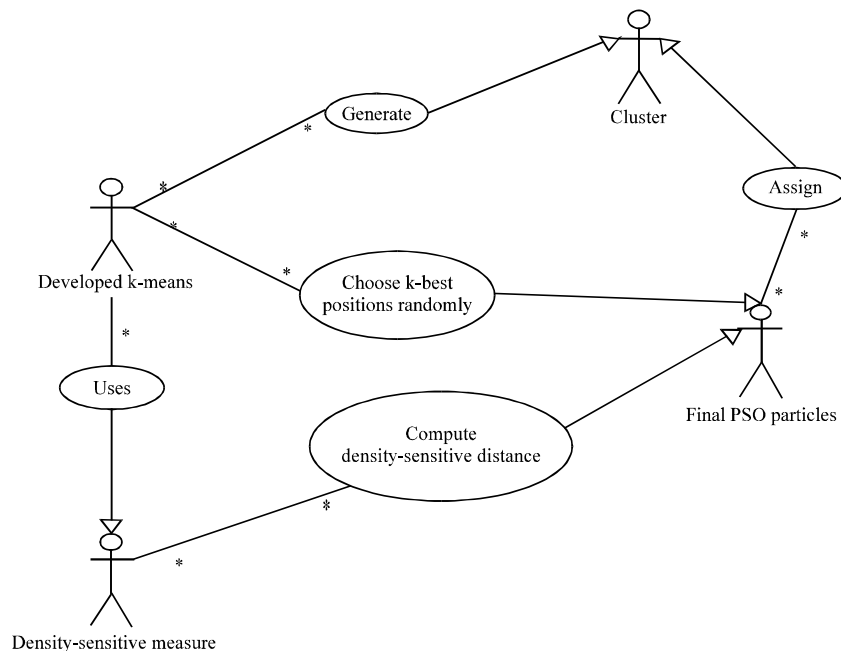


Fig. 7: Use cases Diagram of the HYBRID NPSO-DS k-means algorithm

Clustering accuracy: This is the Rand Index (RI), a measure that describes the actual percentage of documents that are correctly mapped to their corresponding clusters. It is depicted as Rand (1971):

$$\text{Clustering accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (15)$$

where, TP, TN, FP and FN represent the true positive, true negative, the false positive and the false negative values, respectively. In this study, TP defines two close particles that are correctly allocated to same cluster, a TN correctly assigns two contrasting particles in different clusters. Similarly, FP defines two contrasting particles that are wrongly assigned to the same cluster while the FN wrongly assigns two close particles in different clusters.

RESULTS AND DISCUSSION

In this study, a hybrid NPSO-DS k-means algorithm was developed and benchmarked with (Eq. 3) variants which are k-means, PCA-based HYBRID (K-PSO) and UFT-k-means. All the algorithms were implemented in MATLAB 7.7.0 (R2008b) environment on a Windows 7 Ultimate 32-bit operating system amd Athlon (tm) X2 DualCore QL-66 central processing unit with a speed of 2.2 GHZ, 2 GB RAM and 320 GB hard disk drive. We tested for values of K = 2, 3, 4 and the outputs obtained for Educational Process Mining (EPM) and wine datasets

are shown in Table 2 and 3, respectively. In all the evaluations, results of the (Eq. 3) effective metrics for evaluating an optimal clustering algorithm which include clustering accuracy clustering time and SSE were recorded (Shen *et al.*, 2010). In Fig. 8 and 9, the sample visual outputs of NPSO-DS and PCA-based HYBRID (K-PSO) k-means with EPM dataset are respectively, shown for K = 2, 3 and 4.

Clustering accuracy (rand index): As shown in Fig. 10, the clustering accuracies produced by the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means for 2 clusters (K = 2) using EPM dataset are 64.8, 72.1, 77.7 and 80.2%, respectively. For 3 clusters (K = 3) using EPM dataset, the accuracies obtained by the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means are 67.3, 76.4, 79.1 and 83.6%, respectively. When cluster number was increased to 4 (K = 4), the original k-means, PCA-based HYBRID (K-PSO), UFT-K-means and the developed NPSO-DS k-means yielded accuracies of 69.2, 83.9, 87.3 and 92.4%, respectively on EPM dataset. However, in Fig. 11, the clustering accuracies produced by the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS K-means for 2 clusters (K = 2) using wine dataset are 88.5, 89.4, 91.1 and 93.6%, respectively. The accuracies produced by the original k-means, PCA-based HYBRID (K-PSO), UFT-K-means and the

Table 2: Evaluation results of the NPSO-DS K-means using the EPM dataset

No. of clusters	Algorithm	Clustering accuracy (%)	Clustering time (sec)	Sum of squared error
2	k-means	64.8	83.2	0.48
	PCA-based HYBRID (K-PSO)	72.1	99.4	0.39
	UFT-k-means	77.7	87.2	0.33
	Developed NPSO-DS k-means	80.2	85.7	0.28
3	K-means	67.3	78.7	0.42
	PCA-based HYBRID (K-PSO)	76.4	91.8	0.32
	UFT-k-means	79.1	84.7	0.27
	Developed NPSO-DS k-means	83.6	80.5	0.21
4	k-means	69.2	74.4	0.36
	PCA-based HYBRID (K-PSO)	83.9	87.2	0.28
	UFT-k-means	87.3	80.4	0.22
	Developed NPSO-DS k-means	92.4	74.8	0.13

Table 3: Evaluation results of NPSO-DS K-means using the wine dataset

No. of clusters	Algorithm	Clustering accuracy (%)	Clustering time (sec)	Sum of squared error
2	k-means	88.5	16.7	0.164
	PCA-based HYBRID (K-PSO)	89.4	24.3	0.160
	UFT-k-means	91.1	21.7	0.156
	Developed NPSO-DS k-means	93.6	17.2	0.133
3	k-means	91.3	14.3	0.148
	PCA-based HYBRID (K-PSO)	92.2	23.8	0.126
	UFT-k-means	92.9	19.9	0.113
	Developed NPSO-DS k-means	94.8	15.1	0.098
4	k-means	92.8	12.1	0.119
	PCA-based HYBRID (K-PSO)	94.1	21.7	0.106
	UFT-k-means	95.6	18.4	0.097
	Developed NPSO-DS k-means	96.3	13.9	0.082

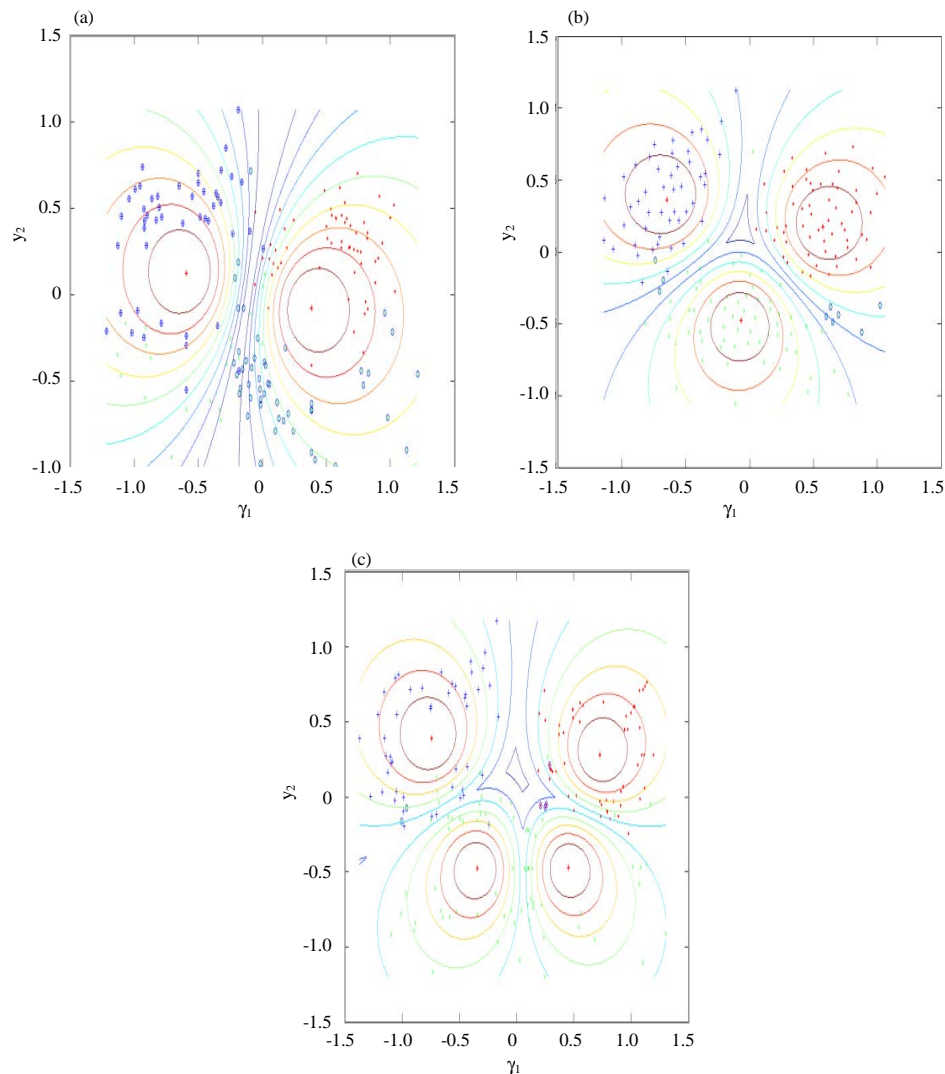


Fig. 8: Sample output of NPSO-DS k-means with EPM dataset: a) ($K = 2$); b) ($K = 3$) and c) ($K = 4$)

developed NPSO-DS k-means are 91.3, 92.2, 92.9 and 94.8%, respectively, for 3 clusters ($K = 3$) with wine dataset.

When cluster number was increased to 4 ($K = 4$), the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means yielded accuracies of 92.8, 94.1, 95.6 and 96.3% , respectively, on wine dataset.

Clustering time: The execution time of the NPSO-DS k-means obtained on EPM dataset is presented in Fig. 12. The original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means converged approximately in 83.2, 99.4, 87.2 and 85.7s, respectively with 2 clusters. Similarly, the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means converged in,

PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means converged in approximately 78.7, 91.8, 84.7 and 80.5, respectively, for 3 clusters. When cluster number was increased to 4 ($K = 4$), the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means converged at approximate clustering time of 74.4, 87.2, 80.4 and 74.8, respectively Furthermore, the execution times used by the NPSO-DS k-means on wine dataset is conceptually represented in Fig. 13.

The original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means converged approximately in 16.7, 24.3, 21.7 and 17.2s, respectively with 2 clusters. Similarly, the original k-means ately 14.3, 23.8, 19.9 and 15.1s, respectively, for 3 clusters.

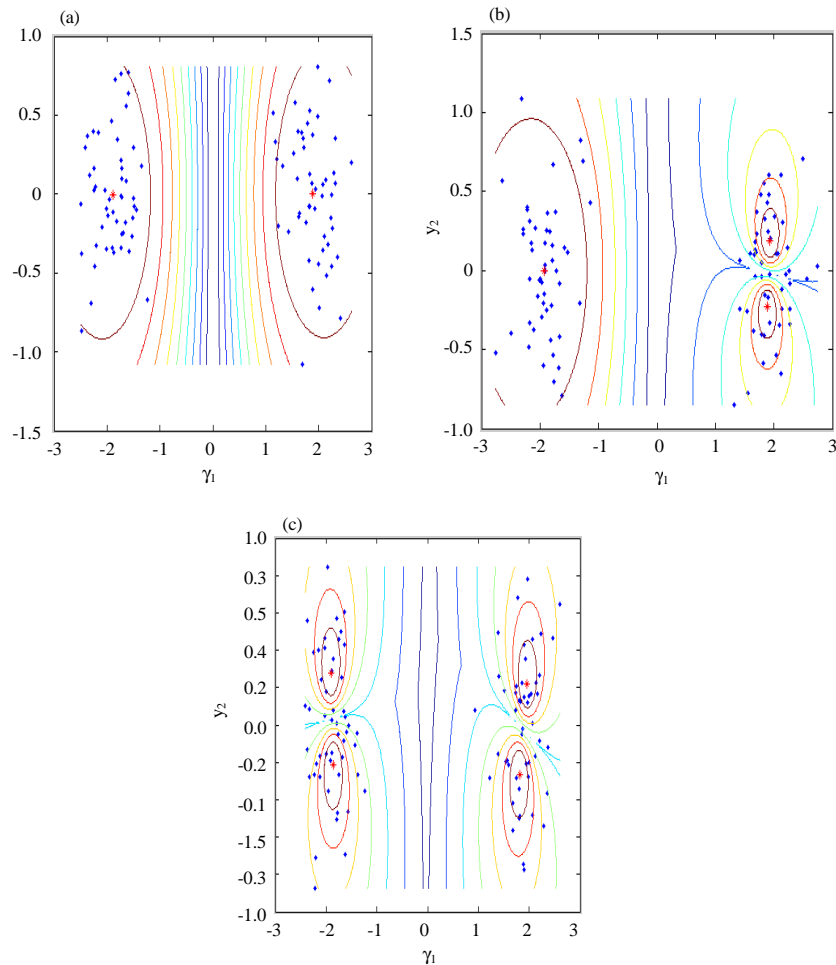


Fig. 9: Sample output of PCA-based HYBRID (K-PSO) with EPM dataset: a) ($K = 2$); b) ($K = 3$); c) ($K = 4$)

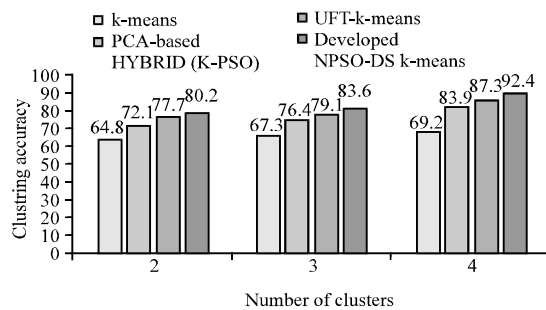


Fig. 10: Accuracy of the NPSO-DS k-means on EPM dataset; Clustering accuracy on EPM dataset of 230318 instances

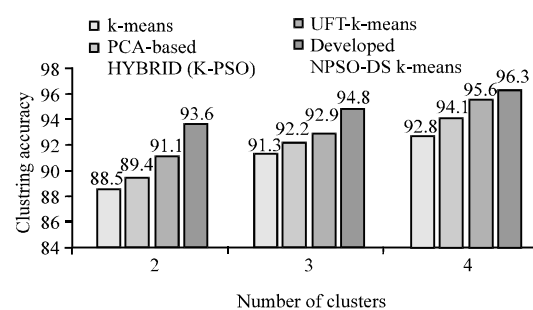


Fig. 11: Accuracy of the NPSO-DS k-means on wine dataset; Clustering accuracy on EPM dataset of 178 instances

When cluster number was increased to 4 ($K = 4$), the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means converged at approximate clustering time of 12.1, 21.7, 18.4 and 13.9, respectively.

Sum of Squared Error (SSE): The SSE incurred by the clustering algorithms over EPM dataset is presented in Fig. 14. The original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means incurred error of 0.48, 0.39, 0.33 and 0.28,

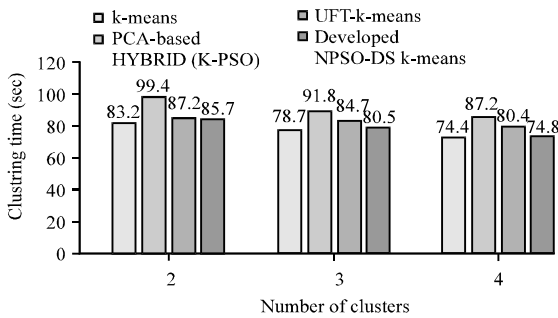


Fig. 12: Execution time of the NPSO-DS k-means on EPM dataset; Clustering accuracy on EPM dataset of 230318 instances

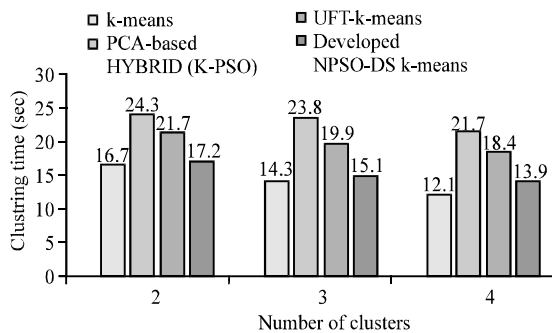


Fig. 13: Execution time of the NPSO-DS k-means on Wine dataset; Clustering accuracy on EPM dataset of 178 instances

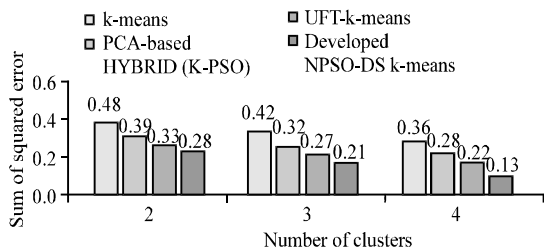


Fig. 14: Error obtained from the NPSO-DS k-means on EPM dataset; Clustering accuracy on EPM dataset of 230318 instances

respectively with 2 clusters. Similarly, the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means yielded error of 0.42, 0.32, 0.27 and 0.21, respectively, for 3 clusters. When the cluster number was increased to 4 ($K = 4$), the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means had errors of 0.36, 0.28, 0.22 and 0.13, respectively. In Fig. 15, the errors obtained by NPSO-DS k-means and benchmark clustering algorithms over wine dataset are presented.

The original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means

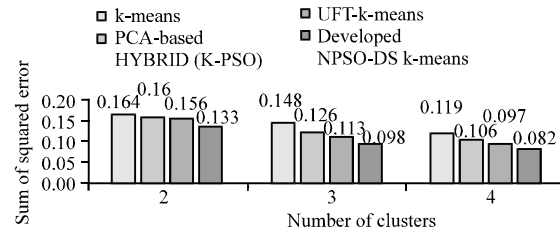


Fig. 15: SSE obtained from the algorithms on wine dataset; Clustering accuracy on EPM dataset of 178 instances

incurred error of 0.164, 0.16, 0.156 and 0.133, respectively with 2 clusters. In addition, the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means yielded error of 0.148, 0.126, 0.113 and 0.098, respectively, for 3 clusters. However, when the cluster number was increased to 4 ($K = 4$), the original k-means, PCA-based HYBRID (K-PSO), UFT-k-means and the developed NPSO-DS k-means had errors of 0.119, 0.106, 0.097 and 0.082, respectively.

The developed NPSO-DS k-mean algorithm has a dominant performance with both the EPM and the wine datasets compared with the baseline k-means, UFT-k-means and PCA-based HYBRID (K-PSO) clustering algorithms using clustering accuracy and SSE. The least accuracies produced by k-means in all the evaluations conducted using EPM dataset indicated that k-means is not a good candidate for clustering large real world dataset such as EPM which contains 230318 instances. However, as cluster number increases, k-means shows more improvements in accuracy but nevertheless, its accuracy was the least among other algorithms considered. With cluster numbers (2-4), accuracies (64.8, 67.3 and 69.2%) were obtained, respectively, for k-means. This reveals that the higher the number of clusters, the better the clustering accuracy of k-means algorithm. This was also a general behaviour of other algorithms evaluated.

k-means results obtained in this study corroborates with the assertion of Zheng *et al.* (2014) that k-means can fail with large and noisy dataset because it only converges to local minima and suffers the limitation imposed on it by Euclidean distance similarity metric by default. However, while tested with wine dataset which contains only 178 instances, k-means drastically improved as accuracies (88.5, 91.3 and 92.8%) were obtained for cluster numbers (2, 3 and 4), respectively. This implies that k-means is a very good algorithm for small datasets as stated by Twinkle *et al.* (2014). It is worthy of mentioning that k-means is the most efficient algorithm as it produces the least clustering time in all the evaluations conducted on EPM and wine

datasets, followed by the developed NPSO-DS k-means, UFT-k-means and the PCA-based HYBRID (K-PSO) algorithm in that order. In all the evaluations conducted on wine and EPM datasets, the developed NPSO-DS k-means algorithm is the most accurate and with the least SSE followed by UFT-k-means, PCA-based HYBRID (K-PSO) and the original k-means in that order. This challenging performance by the developed NPSO-DS k-means is associated with optimal data normalization and relevant particle selection procedures as well as a globally converging density-sensitive distance measure incorporated into the developed NPSO-DS k-means algorithm. Oludare *et al.* (2014) and Temitayo *et al.* (2012) stated that improvements obtained for feature selection and data normalization procedures invariably impacts on the performance of data mining algorithms which justifies the results obtained for the NPSO-DS k-means algorithm.

CONCLUSION

This study presents a NPSO-DS k-means with relevant optimal particle selection and density-sensitive distance measure. The results reveal that the developed NPSO-DS k-means method has a more dominant performance over the conventional k-means, UFT-k-means and PCA-based HYBRID (K-PSO) algorithms especially, whilst considering clustering accuracy. This challenging performance by the developed NPSO-DS k-means is borne out of relevant particle selection procedure as well as the globally converging density-sensitive distance measure incorporated into the developed NPSO-DS k-means algorithm. Oludare *et al.* (2014) and Temitayo *et al.* (2012) stated that improvements obtained for efficient and effective feature selection procedures invariably enhance the effectiveness of clustering algorithms which justifies the results obtained for the NPSO-DS k-means algorithm. The least accuracies produced by k-means in all the evaluations corroborated with the assertion of Zheng *et al.* (2014) that k-means is not a good candidate for clustering large real world datasets. The developed NPSO-DS k-means can identify non-convex clustering structures, thus, generalizing the application area of the conventional k-means algorithm. The hypothetical results on EPM world dataset which contains 230318 instances validate the effectiveness of the developed algorithm. The developed NPSO-DS k-means algorithm can be applied in situations where the distributions of data points are not compact super-spheres. However, the near-optimal clustering time produced by the developed NPSO-DS k-means can be further investigated for possible improvements. The developed

NPSO-DS k-means clustering performs best in all the evaluation conducted on EPM and wine datasets using clustering accuracy and SSE as evaluation metrics. However, it yielded higher clustering time than the baseline k-means only. This could be as a result of the time required to normalize and select relevant features at each generation of NPSO technique before final clustering of resultant particles by DS-K-means. Though, it is more computationally efficient than UFT-k-means and PCA-based HYBRID (K-PSO), nevertheless, further, research can be directed along this direction.

REFERENCES

- Adebisi, A.A., O.E. Olusayo, O. Stephen, A.A.B. Olatunde and A.O. Titilayo, 2012. Development of a hybrid k-means-expectation maximization clustering algorithm. *J. Comput. Modell.*, 2: 1-23.
- Adigun, A.A., T.M. Fagbola and A. Adegun, 2014. Swarmdroid: Swarm optimized intrusion detection system for the android mobile enterprise. *IJCSI Int. J. Comput. Sci.*, 11: 62-69.
- Arthur, D. and S. Vassilvitskii, 2007. K-means++: The advantages of careful seeding. *Proceedings of the 18th Annual ACM-SIAM Symposium of Discrete Analysis*, January 7-9, 2007, New Orleans, LA., USA., pp: 1027-1035.
- Ayodele, O., F. Temitayo, S.O. Olabiyisi, E.O. Omidiora and J. Oladosu, 2016. Development of a modified local binary pattern-gabor wavelet transform aging invariant face recognition system. *Proceedings of the International Conference on Computing Research and Innovations (OcRI'16)*, September 7-9, 2016, University of Ibadan, Ibadan, Nigeria, pp: 108-114.
- Bai, Q., 2010. Analysis of particle swarm optimization algorithm. *Comput. Inf. Sci.*, 3: 180-180.
- Bolelli, L., S. Ertekin, D. Zhou and C.L. Giles, 2007. A clustering method for web data with multi-type interrelated components. *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, May 08-12, 2007, ACM, Banff, Alberta, Canada, ISBN:978-1-59593-654-7, pp: 1121-1122.
- Chen, J. and H. Zhang, 2007. Research on application of clustering algorithm based on PSO for the web usage pattern. *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, September 21-25, 2007, IEEE, Shanghai, China, ISBN:978-1-4244-1311-9, pp: 3705-3708.
- Eberhart, R.C. and Y. Shi, 2001. Particle swarm optimization: Developments, applications and resources. *Proceedings of the Congress on Evolutionary Computation*, Volume 1, May 27-30, 2001, Seoul, South Korea, pp: 81-86.

- Eberhart, R.C., P.K. Simpson and R. Dobbins, 1996. Computational Intelligence PC Tools. Academic Press Professional, San Diego, CA., USA., ISBN: 0-12-228630-8, pp: 212-223.
- Elbatta, M.T. and W.M. Ashour, 2013. A dynamic method for discovering density varied clusters. *Intl. J. Signal Process. Image Patt. Recognit.*, 6: 123-134.
- Gupta, N. and R.L. Ujjwal, 2013. An efficient incremental clustering algorithm. *World Comput. Sci. Inf. Technol. J.*, 3: 97-99.
- Guyon, I., 2008. Practical Feature Selection: From Correlation to Causality. In: *Mining Massive Data Sets for Security: Advances in Data Mining, Search, Social Networks and Text Mining and their Applications to Security*, Fogelman-Soulie, F., P. Domenico, P. Jakub and S. Ralf (Eds.). IOS Press, Amsterdam, Netherlands, ISBN:9781586038984, pp: 27-43.
- Han, J. and M. Kamber, 2006. *Data Mining Concepts and Techniques*. 2nd Edn., Morgan Kaufmann, San Francisco, CA, USA.
- Hung, M.C., J. Wu, J.H. Chang and D.L. Yang, 2005. An efficient k-means clustering algorithm using simple partitioning. *J. Inf. Sci. Eng.*, 21: 1157-1177.
- Jeong, H. and B. Gautam, 2012. Mining student behavior models in learning-by-teaching environments. *Proceedings of the 1st International Conference on Educational Data Mining*, June 20-21, 2008, Vanderbilt University, Nashville, USA., pp: 127-136.
- Joshi, A. and R. Kaur, 2013. A review: Comparative study of various clustering techniques in data mining. *Intl. J. Adv. Res. Comput. Sci. Software Eng.*, 3: 55-57.
- Karami, A. and M. Guerrero-Zapata, 2015. A fuzzy anomaly detection system based on hybrid PSO-kmeans algorithm in content-centric networks. *Neurocomputing*, 149: 1253-1269.
- Kaur, J. and J.S. Bal, 2017. A study of particle swarm optimization based k-means clustering for detection of dental caries in dental X-ray images. *Intl. J. Adv. Res. Comput. Sci.*, 8: 289-294.
- Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. *Proc. IEEE Int. Conf. Neural Networks*, 4: 1942-1948.
- Keshtkar, F. and W. Gueaieb, 2006. Segmentation of dental radiographs using a swarm intelligence approach. *Proceedings of the 2006 Canadian Conference on Electrical and Computer Engineering*, May 7-10, 2006, IEEE, Ottawa, Canada, pp: 328-331.
- Koay, C.A. and D. Srinivasan, 2003. Particle swarm optimization-based approach for generator maintenance scheduling. *Proceedings of the 2003 IEEE International Symposium on Swarm Intelligence SIS'03 (Cat. No.03EX706)*, April 26, 2003, IEEE, Indianapolis, USA., pp: 167-173.
- Mahmood, S., M.S. Rahaman, D. Nandi and M. Rahman, 2015. A proposed modification of k-means algorithm. *Intl. J. Mod. Educ. Comput. Sci.*, 7: 37-42.
- Mary, C.I., M. MCA and S.K. Raja, 2010. A modified Ant-based clustering for medical data. *Intl. J. Comput. Sci. Eng.*, 2: 2253-2257.
- Matthew, F.T., B.R. Seyi and O. Akinwale, 2013. Image clustering using a hybrid GA-FCM algorithm. *Int. J. Eng. Technol.*, 3: 99-107.
- Momin, B.F. and P.M. Yelmar, 2012. Modifications in K-means clustering algorithm. *Intl. J. Soft Comput. Eng.*, 2: 334-354.
- Niu, Q. and X. Huang, 2011. An improved fuzzy C-means clustering algorithm based on PSO. *J. Software*, 6: 873-879.
- Oludare, O., O. Stephen, O. Ayodele and F. Temitayo, 2014. An optimized feature selection technique for email classification. *Int. J. Sci. Technol. Res.*, 3: 286-293.
- Olugbara, O.O., E. Adetiba and S.A. Oyewole, 2015. Pixel intensity clustering algorithm for multilevel image segmentation. *Math. Prob. Eng.*, 2015: 1-19.
- Padhy, N., P. Mishra and R. Panigrahi, 2012. The survey of data mining applications and feature scope. *Int. J. Comput. Sci., Eng. Inform. Technol.*, 2: 43-58.
- Patel, V.R. and R.G. Mehta, 2011. Impact of outlier removal and normalization approach in modified K-means clustering algorithm. *Intl. J. Comput. Sci. Issues*, 8: 354-359.
- Pavlik Jr., P.I., H. Cen, L. Wu and K.R. Koedinger, 2008. Using item-type performance covariance to improve the skill model of an existing tutor. *Proceedings of the 1st International Conference on Educational Data Mining*, June 20-21, 2008, Universite du Quebec a Montreal, Montreal, Canada, pp: 77-86.
- Peng, J. and Y. Xia, 2005. A cutting algorithm for the minimum sum-of-squared error clustering. *Proceedings of the 2005 SIAM International Conference on Data Mining*, April 21-23, 2005, Society Industrial and Applied Mathematics, Newport Beach, California, USA., ISBN:978-0-89871-593-4, pp: 150-160.
- Pudil, P., F.J. Ferri, J. Novovicova and J. Kittler, 1994. Floating search methods for feature selection with nonmonotonic criterion functions. *Proceedings of the 12th IAPR International Conference on Pattern Recognition and Signal Processing (Cat. No.94CH3440-5)* Vol. 3, October 9-13, 1994, IEEE, Jerusalem, Israel, pp: 279-283.
- Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.*, 66: 846-850.
- Romero, C. and S. Ventura, 2007. Educational data mining: A survey from 1995-2005. *Exp. Syst. Appl.*, 33: 135-146.

- Saini, G. and H. Kaur, 2014. A novel approach towards k-mean clustering algorithm with PSO. *Intl. J. Comput. Sci. Inf. Technol.*, 5: 5978-5986.
- Sethi, C. and G. Mishra, 2013. A linear PCA based hybrid K-means PSO algorithm for clustering large dataset. *Intl. J. Sci. Eng. Res.*, 4: 1559-1566.
- Shanmugapriya, B. and M. Punithavalli, 2012. A modified projected K-means clustering algorithm with effective distance measure. *Int. J. Comput. Applic.*, 44: 32-36.
- Shen, H., L. Jin, Y. Zhu and Z. Zhu, 2010. Hybridization of particle swarm optimization with the k-means algorithm for clustering analysis. *Proceedings of the 2010 IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, September 23-26, 2010, IEEE, Changsha, China, pp: 531-535.
- Shinde, P.V. and B.L. Gunjal, 2012. Particle swarm optimization-best feature selection method for face images. *Intl. J. Sci. Eng. Res.*, 3: 1-5.
- Siedlecki, W. and J. Sklansky, 1988. On automatic feature selection. *Intl. J. Pattern Recognit. Artif. Intell.*, 2: 197-220.
- Su, M.C. and C.H. Chou, 2001. A modified version of the k-means algorithm with a distance based on cluster symmetry. *Trans. Pattern Anal. Machine Intel.*, 23: 674-680.
- Sun, J., W.B. Xu and B. Ye, 2006. Quantum-behaved particle swarm optimization clustering algorithm. *Lecture Notes Comput. Sci.*, 4093: 340-347.
- Temitayo, F., O. Stephen and A. Abimbola, 2012. Hybrid GA-SVM for efficient feature selection in E-mail classification. *Comput. Eng. Intell. Syst.*, 3: 17-28.
- Twinkle, G., F. Lofter and M. Arun, 2014. Survey on various enhanced k-means algorithms. *Intl. J. Adv. Res. Comput. Commun. Eng.*, 3: 43-61.
- Verma, A. and A. Kuma, 2014. Performance enhancement of k-means clustering algorithms for high dimensional data sets. *Intl. J. Adv. Res. Comput. Sci. Software Eng.*, 4: 5-9.
- Wang, L., B. Liefeng and J. Licheng, 2012. A Modified k-means clustering with a density-sensitive distance metric. *Masters Thesis, Department of Information and Computer Science, University of California, Irvine, California.*
- Wang, X. and Y. Bai, 2016. A modified minmax-means algorithm based on PSO. *Comput. Intell. Neurosci.*, 2016: 1-13.
- Wei, M., T.W. Chow and R.H. Chan, 2015. Clustering heterogeneous data with K-means by mutual information-based unsupervised feature transformation. *Entropy*, 17: 1535-1548.
- Weijun, X., W. Zhiming, Z.H. Wei and Y. Genke, 2004. A new hybrid optimization algorithm for the job-shop scheduling problem. *Proceeding of the 2004 International Conference on American Control Vol. 6*, June 30-July 02, 2004, IEEE, Boston, Massachusetts, USA., pp: 5552-5557.
- Zhang, C. and Z. Fang, 2013. An improved k-means clustering algorithm. *J. Inform. Comput. Sci.*, 10: 193-199.
- Zhang, H. and G. Sun, 2002. Feature selection using TABU search method. *Pattern Recogn.*, 35: 701-711.
- Zheng, L., T. Li and C. Ding, 2014. A framework for hierarchical ensemble clustering. *ACM. Trans. Knowl. Discovery Data*, 9: 1-23.
- Zhou, D., O. Bousquet, T.N. Lal, J. Weston and B. Scholkopf, 2004. Learning with Local and Global Consistency. In: *Advances in Neural Information Processing Systems*, Thrun, S., L. Saul and B. Scholkopf (Eds.). Cambridge, Massachusetts, USA., pp: 321-328.
- Zhu, X., 2006. Semi-supervised learning literature survey. *MCS Thesis, Department of Computer Sciences, University of Wisconsin-Madison, Madison, Wisconsin.*