Geometric-inspired Particle Swarm Optimization (PSO) for Classification Tasks

Enoch Opanin Gyamfi
University of Electronic Science and Technology of China,
School of Information and Software Engineering,
Chengdu, P.R. China;
C.K. Tedam University of Technology and Applied Sciences,
Navrongo, Upper East Region, Ghana

Zhiguang Qin
University of Electronic Science and Technology of China,
School of Information and Software Engineering,
Chengdu, P.R. China

Juliana Mantebea Danso
University of Electronic Science and Technology of China,
School of Information and Software Engineering,
Chengdu, P.R. China

Daniel Adu-Gyamfi
C.K. Tedam University of Technology and Applied Sciences,
Navrongo, Upper East Region, Ghana

Nelson Opoku-Mensah
St. Monica's College of Education,
Mampong, Ashanti Region, Ghana

Noble Arden Elorm Kuadey
Ho Technical University, Ho, Volta Region, Ghana

Daniel Konin
Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development,
Kumasi, Ashanti Region, Ghana

ABSTRACT

The ultimate performance of particle swarm optimization is influenced by hyper-parameters like the inertia, cognitive and social coefficient values. These hyper-parameters have a significant effect on search capability of the particle swarm optimization. When looking at previous studies that are carried out to calculate these coefficients, none of these studies has been inspired by geometric techniques to illustrate for the influence of these components on best position realization. In this, article a geometric approach to how the allocation of social, cognitive and inertia regions on a search space enables particles to move to their best positions at every iteration time. In experiment and benchmark tests, the study validates the applicability of the proposed approach to classification problem using EMNIST dataset. The modified PSO approach gives successful results in separating data into appropriate classes which confirms that the proposed method is highly competitive in guiding the directional movement of the particles towards the best positions.

General Terms

Data classification, machine learning, pattern recognition

Keywords

Particle swarm optimization, geometric, classification, sub-swarm

1. INTRODUCTION

Several machine learning frameworks have been automated to yield some a considerable level of general accuracy. Some of these frameworks are data classification models, tasked to extract relevant features and describe these features as classes within datasets. Data classification models collects samples and learn similar standard features among them. Samples with identical measured features end up belonging to the same class. To mention a few of the most important areas, in which data classification is useful in but not limited to; dimensionality reduction [1-3], Noise reduction and denoising [1,4], Pattern recognition [1,5], Automatic knowledge discovery [1,6-7], Feature extraction (transformation [8]) techniques [9], Representative feature selection techniques (such as Information Gain [9], Relief, Fisher Score, Least Absolute Shrinkage and Selection Operator (LASSO), Matthews Correlation Coefficient (MCC), [1,10-13], Sparse learning [8,14-15].

Quite a lot of evolutionary approaches are designed to exploit such background features that interpret and represent classes within datasets. Among these approaches includes but not limited to; Principle Component Analysis (PCA) [9], Linear Discriminant Analysis (LDA) [3,16], Canonical Correlation Analysis (CCA) [17], t-distributed Stochastic Neighbor Embedding (tSNE) [18,19].This paper gives a novel dimension of understanding to Particle Swarm Optimization (PSO), one of the approaches inspired by [20] that mines features from a given dataset and classify. Modifications of the standard PSO algorithm, have shown equivalent outcomes to the other evolutionary data classification approaches [21]. PSO algorithm is easy to implement, computationally efficiency and robust in its control parameters. PSO is also efficient in exploiting non-linear features of huge search space problems. The purpose of this paper is to modify some relevant geometric methods that may improve the results of this simple yet standard and efficient PSO algorithm in classification task.

Since one of the strong characteristics of PSO is to explore the full potential search space for each data point using three main hyper-parameters, the inertia, cognitive and social velocities [22,23], the study proposes to tweak and modify these hyper-parameters using simple geometric strategies, so that the proposed technique could focused on only the optimized search region that is well suited for particle movement. As a benchmark for testing the accuracy of the proposed method, the study uses the EMNIST dataset which offers a unique image collection of digits/numbers captured from handwritings of individuals. Strategies are combined that obtains a high correlation between classes for the given dataset, using the PSO algorithm modified by geometric means.

2. THEORETICAL BASIS

Particle swarm algorithm is motivated by social standards that living organisms follow to collaborate among themselves as groups. This is noticeable in group scenarios. Such is a group of organisms of the same kind that incorporates intelligent behaviors with the aim of solving a given problem. It's observable in swarms of bees, schools of fish, flocks of bird and even human social behavior [23-25]. Such behavior is implemented within the particle swarm algorithm to solve various function optimization problems. In nature, the members of the swarm are assumed to represent the particles when such an algorithm is implemented. Movement of the organism, based on the frequently shared and communicated information, are implicitly implemented in an algorithm as movement of the particle also based some factors. Movement of the particles are controlled with regularly updating velocity [20-21].

Locally sharing of the detailed information that are propagated to globally within the swarm, can be likened to the strategies for updating positions of particles within the algorithm. Based upon this simulation, the standard PSO algorithm are summarized into three main steps of which the first two are trivial. First, the PSO algorithm evaluates the fitness of each particle, then secondly, updates the local and global fitness and lastly updates velocity and position of each particle using the following equation.

$$v_i(t+1) = wv_i(t) + c_1 r_1[\hat{x}_i(t) - x_i(t)] \\ + c_2 r_2[g_i(t) - x_i(t)] \quad (1)$$

The index of the particle is represented by i. Thus, $v_i(t)$ and $x_i(t)$ are the velocity and position of particle i at time t respectively. The parameters $w, c_1 and c_2$ are user-defined coefficients, while r_1 and r_2 are random values regenerated for each velocity update. The value $\hat{x}_i(t)$ is the individual best position and $g_i(t)$ is the group's global best candidate position. Each of the three terms, known as hyper-parameters of the equation, have diverse purpose. The first term $wv_i(t)$ is the inertia component, responsible for keeping the particle moving in the same direction it was originally heading. The second term $c_1 r_1[\hat{x}_i(t) - x_i(t)]$, called the cognitive component, acts as the particle's memory, causing it to return to the regions of the search space in which it has experienced high individual fitness. The third term of the equation $c_2 r_2[g_i(t) - x_i(t)]$, called the social component, causes the particle to move to the best region the swarm has found so far.

Since its introduction in 1995, Particle Swarm Optimization (PSO) [20] had been improved and widely used in many applications. Many PSO algorithms have been successfully applied mainly through exploiting combinations of diverse hyperparameters, to reduce redundant properties and extract relevant features from large datasets that may increase classification performance [26]. By doing so, the running time of classification can also be shortened and the intuitive structure of the data can be simplified. The PSO algorithm has been competitively used to solve classification problems mainly because, problems of feature extraction from large data are difficult [27]. From this point of view [26], argued that, in such difficulties, acquisition of optimal solution cannot be guaranteed until an exhaustive search technique like PSO is used, keeping in mind the amount of time to perform such a task.

To validate the application of PSO to classification task in research, the study conducts a simple search on two major scientific document repository, Google scholar and Scopus, for publications that contributed in the use of PSO for classification related task. Sousa, Neves and Silva [28]; Sousa, Silva and Neves [29]; Sousa, Silva and Neves [30]; Cervantes, Galván, and Isasi [31]; Zahiri and Seyedin [32]; De Falco, Della Cioppa and Tarantino [33], were some of the earliest literature that directly researched on particle swarm-based data mining algorithms for classification tasks. Within these same timeframe, cognate area of research to these earlier literature was the application of PSO in data clustering [34-35]. Synchronous to those timeframes, the trend of researching in/about association rule mining for classification using PSO took off [36]. This trend continued to fairly recent works like that of [37-38].

The PSO algorithm applied in classification tasks has many merits and because of that, many other new articles on the application of PSO to classification-related tasks have been published since then. But, its standard application result in a fast and premature convergence in addition to having weak local search ability [39].Several limitations were identified in the standard PSO and its application to classification task, out of which one gained attention. Because of its sensitivity to transformation, (rotation, translation, and scale) [39], it works poorly when additionally applied with geometric techniques.An approach to divide the entire particles into sub-swarms, is quite commonly and recently used as an alternative to increase the performance of standard PSO in a classification task [39], [40], A sub-swarm approach for PSO is time consuming when searching an entire high-dimensional problem space. The sub-swarm approach influenced the idea to partition of a 2D-subspace of particle swarms through geometric strategies, which has been widely known as a simple and easy steps to compute by digital devices, with less times and processing resources. This paper therefore uses this idea as a basis for constructing and partitioning a 2D subspace through geometric techniques for particles in a swarm to intelligently collaborate in finding the optimal solution.

3. METHODOLOGY

This section discusses the methodology for the proposed particle swarm optimization. The section also describes, initiates the main optimization procedure and reflects on how the experiential as well as social information are exchanged among data points. The proposed method first created a two-dimensional subspace for particles then taking each particles into consideration, it divides the subspace into regions. These regions becomes the search spaces for particles. The proposed approach reduces the geometric sizes of region (search space) for each particle and bound the particles with this reduced-size region. A search space which with worthy properties that positively correlate to the likelihood of getting nearer to destination is determines as the candidate search space. Thus, the position of the candidate particle can be determinedand its reduced-size and bounded region (search space) and use that to direct the mutual movement among other particles in finding the optimal solution (destination). This processes are explained individually in subsequent paragraphs.

3.1 Creating 2D Subspace for Particles

Firstly, the candidate data point achieves the best solution (position) through learning its coordinates as well as the coordinates of other data points which have the best fitness parameters. The elements of these best fitness parameters allows the candidate data point to diverge its direction, towards the target region but far from being pulled back, minimizing its unexpected movement towards the objective position. The velocity of the candidate data point assists in predicting the change in its movements to a desired position, pertaining to a reference data point. The velocity fundamentally specifies the direction of candidate data point. The computation for velocity is probable next to the procedure of forming regions through functional analysis. Thus, within a single quadrant of a Cartesian coordinate plane, regions were formed with respect to the relative vectors of the candidate data point, $\overrightarrow{w_i}$the reference data point, $\overrightarrow{u_i}$ and the origin vector $\vec{O}$.

In this logic, orthogonal data points for the candidate and reference data points were projected onto *x*- and *y*- axes. The base vectors $\overrightarrow{v_1}, \overrightarrow{v_2}, \overrightarrow{v_3}, \ldots, \overrightarrow{v_i}$were constructed with their initial point coinciding to vectors $\overrightarrow{w_i}$ and $\overrightarrow{u_i}$ at the origin vector $\vec{O}$ (0,0). This vector will run parallel to the origin vector and exactly on the axes. An orthogonal vector $\overrightarrow{z_1}, \overrightarrow{z_2}, \overrightarrow{z_3}, \ldots, \overrightarrow{z_i}$ was then dropped on $\overrightarrow{v_1}$ such that the terminal point of latteris the initial point of the current orthogonal vector, as well, the terminal point of $\overrightarrow{z_1}$is the terminal point of the candidate data point vector, $\overrightarrow{w_i}$. So, the vector of the candidate data point, $\overrightarrow{w_i}$, is computed as a addition of dual vectors that are respectively perpendicular to each other, thus $\overrightarrow{w_i} = \overrightarrow{v_1} + \overrightarrow{z_1}$. So since the properties $\overrightarrow{v_i} \| \vec{O}$, $\overrightarrow{w_i} \| \overrightarrow{u_i}$ and $\overrightarrow{w_i} = \overrightarrow{v_1} + \overrightarrow{z_1}$ holds, accurate results was obtained when similar idea was undertaken to correspondingly construct the vector of the desired data point, $\overrightarrow{u_i}$. By definition, If $\overrightarrow{w_i} = \overrightarrow{v_1} + \overrightarrow{z_1}$, $\overrightarrow{u_i} = \overrightarrow{v_2} + \overrightarrow{z_2}$, $\overrightarrow{v_i} \parallel \vec{O}$, $\overrightarrow{w_i} \parallel \overrightarrow{u_i}$, $(\overrightarrow{w_i} \vee \overrightarrow{u_i}) \perp \vec{O}$, then the task of $\overrightarrow{z_i}$ is orthogonal projection $p_{\vec{\vartheta}} \rightarrow$ of $\overrightarrow{w_i}$ and $\vec{u}$ along $\vartheta = \overrightarrow{v_i} \parallel \vec{O}$, and $\overrightarrow{z_i} \equiv p_{\vec{v}_i} \rightarrow (\overrightarrow{w_i} \vee \overrightarrow{u_i}) = \frac{\overrightarrow{w_i} \cdot \vec{v}}{\|\vec{v}\|^2} \vec{v}$ and $\overrightarrow{v_i}$ is the base vector component of $\overrightarrow{w_i} \vee \overrightarrow{u_i}$ parallel to $\vec{O}$ and $\vec{v}_i \equiv (\overrightarrow{w_i} \vee \overrightarrow{u_i}) - p_{\vec{v}_i}$

The initial and terminal components of $\overrightarrow{v_i}$, and$\overrightarrow{z_i}$ were derived from the Cartesian coordinates of the candidate and the referenced data points. Given coordinates (x_c, y_c) and (x_r, y_r) for the candidate and the referenced data points respectively, the initial points $\overrightarrow{v_i}$<x_c, y_c> and $\overrightarrow{z_i}$<x_r, y_c> in addition to the terminal points $\overrightarrow{v_i}$<x_r, y_c> and $\overrightarrow{z_i}$<x_r, y_r>could be derived. Notice that, the terminal point $\overrightarrow{v_i}$<x_r, y_c>is the initial point of $\overrightarrow{z_i}$. With just these two vectors, three points of the regular quadrilateral forming the region could be clearly deduced. Two diagonally-adjacent of these points were the candidate and the referenced data points. The third point coordinate (x_r, y_c) is the orthogonal point constructed earlier. However, four points are needed to form an enclosed region in a shape of regular quadrilateral. To complete the cyclic order so that a regular quadrilateral can be formed to represent the region, the slope vertices of two diagonally-adjacent points were used to deduce the last fourth point. Given the slope as $\frac{y_c - y_r}{x_c - x_r}$ the standard equation for computing the two diagonals of the regular quadrilateral is $\frac{y - y_c}{x - x_c} = \frac{y_c - y_r}{x_c - x_r}$. With the diagonals known, the resulting vector of the fourth point that completes the regular quadrilateral is (x_c, y_r). This point is a terminal tip of an upward vector from the initial tip of $\overrightarrow{v_i}$ that is parallel to z_i

3.2 Partitioning the 2D Subspace into Functional Regions as search spaces for particles

From here, and as shown in Fig. 1, the proposed approach form Functional Regions which permits an area to be organized as a search space for the particle. Centering on a focal area of this functional region, other region could be connected and formed. A Functional Region was drawn based on the orthogonally projected data points of the candidate and desired data points. The initially created Functional Region bordered the connected area of points within $\overrightarrow{v_i}$‖<x_r, y_r> in relation to its logical reference as (x_c, y_r) and $\overrightarrow{z_i}$‖<x_c, y_r> from the static point (x_c, y_c). The vector position <x_r, y_r> is a rightward displacement that maps the point (x_c, y_r) to a terminal point of $\overrightarrow{z_i}$. The vector position<x_c, y_r> is also a perpendicular translation that maps the initial point of $\overrightarrow{v_i}$ to point (x_c, y_r). The Functional Region was therefore in the form of a regular quadrilateral domain. This domain was referred to as the *social region* of the PSO algorithm. Ranges of the domain were therefore within this region. The domain of the *social region* are the ordered set of all its x-axis points and x∈ℝ. A range is the numerically ordered list of y-axis points that relates to the domain of the social region and satisfies y∈ℝ. Thus, the domains of regions are the entire *x*-values, then again, its ranges are all the corresponding *y*-values of the domains. The idea of the domain and range of the social region could be generated from a piece-wise function indicating the boundaries of the regular quadrilateral domain. For a regular quadrilateral with its bottom left corner point *x(t)*, being the position of thecandidate data point (x_c, y_c) and its sides being from the initials to terminals of ($\overrightarrow{v_i}$,$\overrightarrow{z_i}$), such a piece-wise function could be generated as:

$$f(x,y) = \begin{cases} (x, \overrightarrow{z_i}), (x, x_i) & x_i \leq x \leq \overrightarrow{z_i} \\ (y_i, y), (\overrightarrow{v_i}, y) & y_i \leq y \leq \overrightarrow{v_i} \end{cases} \tag{2}$$

By just swapping the domain and range from the function in a downward and inverse orders, two other regions were formed based from the social region. One region, termed as the *cognitive region*, reflects the social region using its base or bottom vector as the mirror line while the other region, termed as the *inertiaregion*, was created by reflecting the social region on its left vector. These two regions were diagonally related and they also share the candidate data point with the social region. This is illustrated this in Fig. 1.

Techniques were applied to find the parameters that minimizes the geometric size of these regions. During each iteration of the algorithm, each particle is evaluated in terms of its local and global best positions to reduce the size of the regions. The initial regions form the reduced regions which is further used to form bounded regions

3.3 Determining the New Candidate Position using Reduced and Bounded Regions

Considering the proposed PSO Algorithm, the explicit use of points, positions and velocities was necessary in expressing the mechanism of reducing the size the set of initial regions (inertia, cognitive and social) so that the particle can take its desired motion to a new best position. Let $x_c(t_i)$ be the user-supplied initial position of the candidate particle p_c,$\hat{x}_l(t_i)$ be the user-supplied expected best local position p_l in relative to its nearest particle of p_c within the group and $\hat{x}_g(t_i)$ be the user-supplied expected best position p_g in relative to the global best position p_g of the group. In terms of velocity, let $v_l(t_i)$, $v_g(t_i)$and $v_c(t_i)$ be computed inertia, cognitive and social velocities respectively. All these are relative to a particular iteration time t_i where l, c, g, i=0,1,2,3, …, n and n is the total number of iterations.

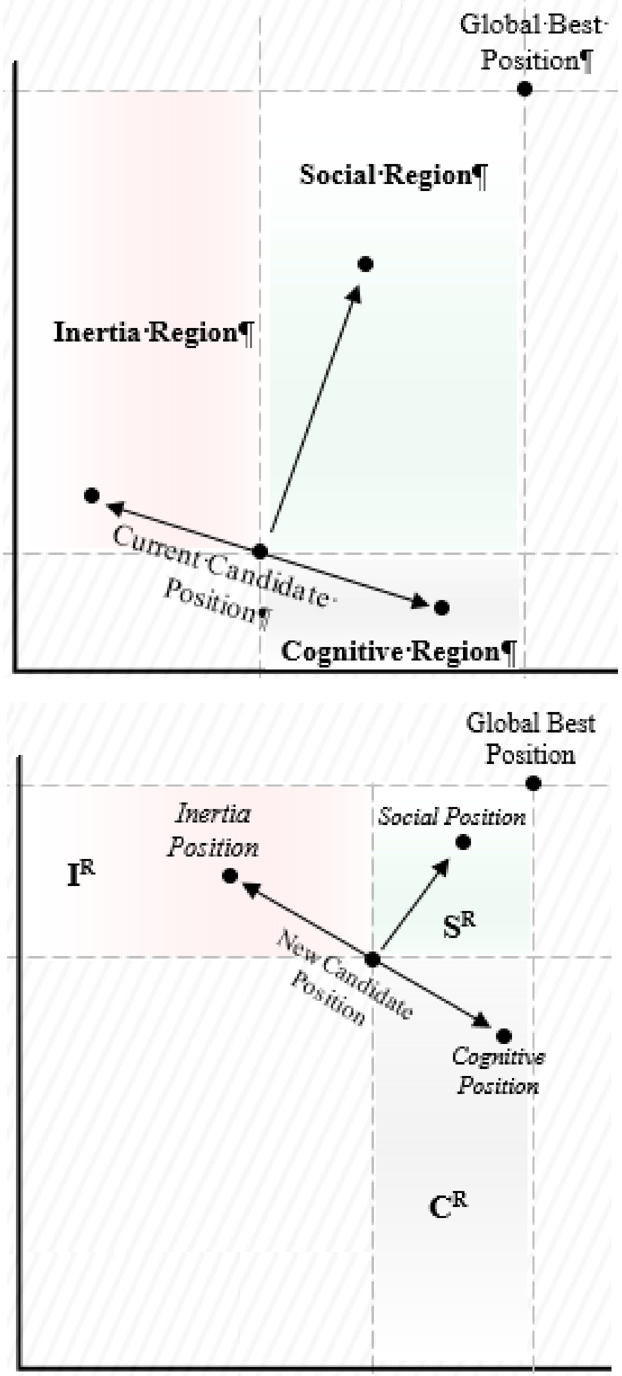


Fig. 1: Initial Inertial (IR), Cognitive (CR) and Social (SR) Regions

For the first iteration where i is indexed as '0' the default values are taken, $x_l(t_i)$ is the computed inertia position of $x_c(t_i)$ projected in relation to $\hat{x}_l(t_i)$, after an inertia velocity update $v_l(t_i)$, is applied to give $x_l(t_i) = x_c(t_i) + v_l(t_i)$. The same way, $x_g(t_i)$ is the calculated cognitive position of $x_c(t_i)$ that is predicted by the influence of $\hat{x}_g(t_i)$ after a cognitive velocity update $v_g(t_i)$, is computed and added using $x_g(t_i) = x_c(t_i) + v_g(t_i)$. Induced from the standard PSO algorithm, inertia velocity contributes significantly in the process of providing an appropriate balance for the updated social position of the previous candidate point. In this regard, the proposed PSO algorithm used the inertia velocity to determine the cognitive velocity, which when both combined, determine the contribution rate of a particle's previous social velocity to its next social velocity at the particular time step.

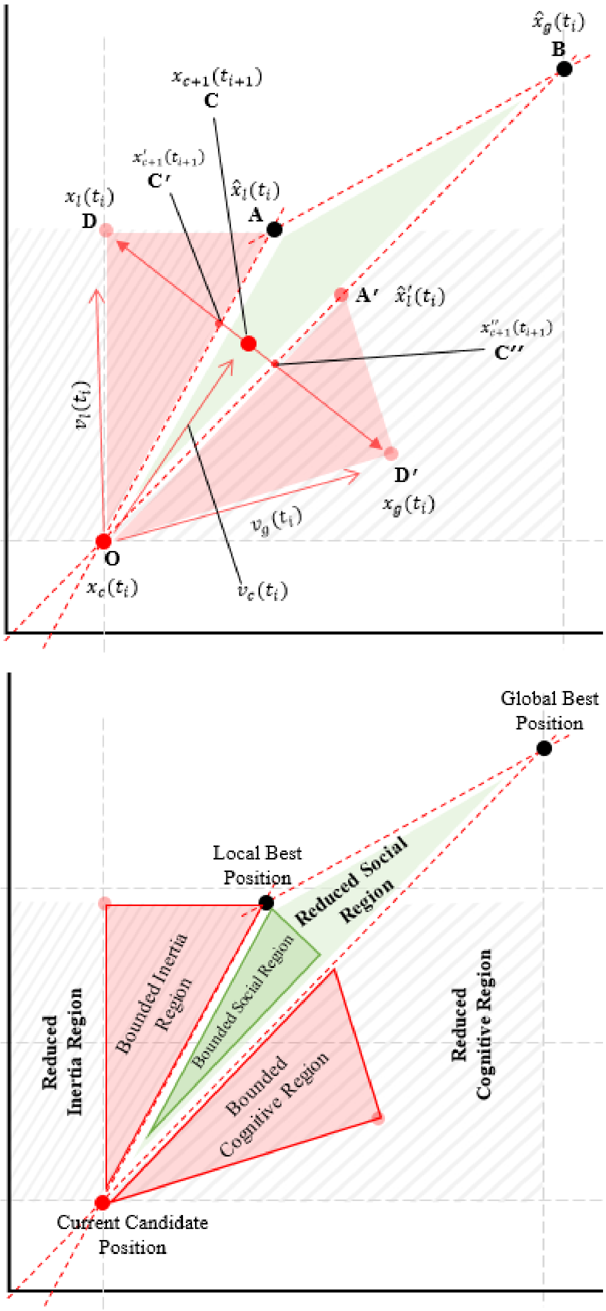


Fig. 2. Reduced and Bounded Inertial, Cognitive and Social Regions

Apart from joining points at positions $x_c(t_i)$, $\hat{x}_l(t_i)$and $\hat{x}_g(t_i)$, with a line to form a triangular reduced social region, the vertical and horizontal lines that have been generated through $x_c(t_i)\hat{x}_l(t_i)\hat{x}_l(t_i)$ perpendicular to the x- and y-axis, helps to form a reduced inertia region which is adjacent to the line joining the points at position $x_c(t_i)$ and $\hat{x}_l(t_i)$ and which also helps to form the reduced cognitive region, adjacent to the line joining the points at positions $x_c(t_i)$ and $\hat{x}_g(t_i)$. From Fig. 2. The reduced regions are condensed form of the initially created regions, out of which bounded regions are further formed. Also, form the same figure, bounded regions are the actual regions which has direct influence on the motion of the particle. Bounded inertia region when formed helped in forming the other two bounded regions. The vertical line within the reduced inertial region which passed through $x_c(t_i)$ and met the horizontal line through $\hat{x}_l(t_i)$ at the inertia position $x_l(t_i)$, with the inertia velocity $v_l(t_i)$ helped form a bounded cognitive region which is adjacent to line joining the $x_c(t_i)$ and $\hat{x}_l(t_i)$.

To achieve this, points at positions$x_c(t_i)$, $x_l(t_i)$ and $\hat{x}_l(t_i)$ forms a right-angle triangle. This triangle is the bounded inertia region. After the composition of reflection and rotation transformation of the right-angle bounded inertia region, over the two lines that intersects at $x_c(t_i)$, the cognitive region is formed. The triangular bounded inertia region is therefore reflected over adjacent line from $x_c(t_i)$ to $\hat{x}_l(t_i)$ and then rotated over adjacent line from $x_c(t_i)$ to $\hat{x}_g(t_i)$ to transform to the triangle of points at positions$x_c(t_i)$, $x_g(t_i)$and $\hat{x}_l^{'}(t_i)$.

Also, the inertia velocity $v_l(t_i)$ is also transformed as the cognitive velocity$v_g(t_i)$. The right-angle inertia position at $x_l(t_i)$ is compositely transformed to the right-angle social position at $x_g(t_i)$. A line connects $x_l(t_i)$to $x_g(t_i)$, which results in bisecting the reduced social region to form the bounded triangular social region closer to the point position $x_c(t_i)$. It is on this line, within the social region, that the new candidate point position$x_{c+1}(t_{i+1})$, is chosen for the second iteration. This is illustrated in Fig. 2

As *i* increases, the second iteration is index as *1* and initial points as well as velocities are computed consequently. Here social velocity and positions are updated as $v_{c+c}(t_{i+i})$ and $x_{c+c}(t_{i+i})$ iff *c*, *i* ≤ 1 ∀ c, $i \in \mathbb{N}_{\geq 0}$. If *c*, *i*> 1, let $v_{c+1}(t_{i+1})$ and $v_{c+2}(t_{i+2})$ be the updated velocity using equation (3), $x_{c+1}(t_{i+1}) = v_c(t_i) + x_c(t_i)$, and $x_{c+2}(t_{i+2}) = v_{c+1}(t_{i+1}) + x_{c+1}(t_{i+1})$ be the computed social positions of the candidate particle in the next successive two iterations up tot_{i+2}. In this logic, for the first iteration t_{i+1}, x_{c+1} is the social position of the $p_c, \hat{x}_l$ but a p_c for x_{c+2} in t_{i+2}. For t_{i+3}, x_{c+2} is the social position of x_{c+1} but becomes a p_c for x_{c+3}, and so forth. The equation of immediate next velocity update can be expressed in linear combination as follows

$$\begin{aligned} v_{c+1}(t_{i+1}) = \quad & weight\ (previous\ velocity) \\ & + \emptyset_1\ (\Delta(local\ best\ position, current\ candidate\ position) \\ & + \emptyset_1\ (\Delta(glocal\ best\ position, current\ candidate\ position) \qquad (3) \end{aligned}$$

Mathematically, the equation can be rewritten as:

$$\begin{aligned} x_{c+1}(t_{i+1}) = \quad & x_c(t_i) + v_c(t_i) \\ v_{c+1}(t_{i+1}) = \quad & \omega(u) + \emptyset_1\big(\hat{x}_l(t_i) - x_{c+1}(t_{i+1})\big) \\ & +\emptyset_2\left(\hat{x}_g(t_i) - x_{c+1}(t_{i+1})\right) \\ & u = v_c(t_i) \text{ for i} = 0, u = v_{c+1}(t_{i+1}) \\ & \text{for i} = 2\ and\ u = v_{c+n}(t_{i+n}) \qquad (4) \end{aligned}$$

Since the main feature of this equation is its linearity, $\omega, \emptyset_1$ and $\emptyset_2$ can be thought of as scaler coefficients if the takes the form of a convex combinations of positions (y, z) and velocity(u) vectors. The equation is again compressed into a form as:

$$v_{c+1}(t_{i+1}) = \omega(u) + \emptyset_1(y) + \emptyset_2(z) \qquad (5)$$

4. EXPERIMENTS

4.1 Dataset Preparation

The proposed alternate approach to PSO was experimented using a benchmark image dataset and validated by measuring its variant from usual PSO. The Extended MNIST (EMNIST) (Cohen, et. al., 2017) image dataset, a more comprehensive and current version, matches the basic structures and paradigms of the original nonlinear MNIST image dataset. For this work, only the number class labels (0 to 9) of EMNIST were used. Because of processing time constraints, the aim is to replicate a sub-dataset from the larger available set of the EMNIST. The test set consisted of 1000 handwritten digit images, which when classified, is dense. No pre-processing was done to the actual images within the dense dataset because its binary images were cleanly normalized to fit a center of a 28×28 field size and preserving aspect ratio. However, another set of data was derived from the dense dataset by lowering its density. Hence, another resultant datasets was created. The proposed PSO algorithm was however implemented on the resultant data while the propsed clustering approach was validated using the densed data. The resultant dataset ensured that some of the images corresponded to data points that were interpreted as distorted digits and were difficult to be identified by clustering technique. This resultant dataset was used to describe the classification performance of the proposed PSO algorithm. Consequently, the resultant dataset ended up with data points in the midst of different clusters, and it was the aim of the proposed PSO to output well segregated clusters of data points making it very difficult to even guessed if the data initially contained any distorted data points

4.2 Setup

The platform on which the proposed PSO algorithm was experimented on, is an Intel® core™ i3-2340M laptop PC of CPU speed @2.30GHz, with a 6GB RAM, running a Windows® 8.1 Operating System, and MATLAB® R2018a.

4.3 Accuracy Measurement and Validation

Notwithstanding these, carefully examining the output of the proposed PSO algorithm, discloses that considerable amount of the underlining local structure of the data is preserved. To statistically prove this, scoring matrices were generated to compute for the similarities between data points. Thus, since the approach still maintain the similarities within and across each class, it was tacitly assumed that, the proposed approach results in a higher symmetric similarity matrix which also implies less entropy. Accordingly, the generated similarity matrices for the results of the proposed PSO, although symmetric, should have an inherent information content that aligns with that of the original data as well as the results of the standard PSO algorithm. In determining the similarity of all the data points, a focal point is projected onto a line and a normal curve is centered on this focal point. The normal curve is used to measure the distance between that point of interest and all of the other data points within the dataset. In this logic, a mountain range is conveniently drawn on a focal data point. All other points are projected onto the mountain range using their individual corresponding distances from this focal point.

Close points to the focal point were observed to end up in high lands while distant points were projected on valleys of the mountain range. Thus, the measured distance is plotted using a normal distribution so that, distant points have very low similarity values and close points have high similarity values on the normal distribution curve.

Ultimately, the distance is measured from the points to the curve to determine the similarity scores with respect to the point of interest. This procedure was iterated for all the points. The width of the normal curve is correlated to the density of the cluster for the single focal data point such that denser regions have narrow curves while less dense regions have wider curves. Since the standard deviation of the normal curve is based on the relative tightness or looseness of surrounding data points for the focal data point, it was necessary to scale the similarities scores to add up to 1. Again, it was also necessary to compute the, average values for each two similarities scores from two directions. This ensured that two points had the same similarity scores when measured from opposite directions. However, similarity score from a point of interest to itself was set to 0. Eventually, a matrix of averaged and scaled similarity scores is constructed whereby each row and column represents the similarity scores calculated from a point of interest. An intuitive mathematical explanation on how pairwise similarity scores were computed is given below:

With an *N=1,2,3,…,* total number of *C* clusters of *M=1,2,3,…,* data points, a distance matrix *d*:$X_1 \rightarrow X_2$on a set X_1, ∀ points p, $C_{N,M}$, $C_{N,M+i}$, $C_{N+i,M}$ in X_1, is required to satisfy these conditions:$X_1 \in \mathbb{R}^m$ and $X_2 \in \mathbb{R}^n$ where m>n

$$d(C_{N,M},C_{N,M+i}) = \frac{d(C_{N,M},C_{N,M+i})+d(C_{N,M+i},C_{N,M})}{2}\ (avg)$$
$$d(C_{N,M},C_{N+i,M}) = 0 \text{ if } \mathrm{i} = 0$$
$$d(C_{N,M},C_{N+i,M}) \geq 0$$
$$d(C_{N,M},C_{N+i,M}) \geq 0 \qquad (6)$$

Given A_p as the event of a point being close to a focal point and B_p being the corresponding event of it being further from a focal point, any of the similarity distance scores between data points must satisfy one of the following conditional probabilities:

$$P(Ap|d(C_{N,M},C_{N,M+i}) \approx 0,$$
$$P(Bp|d(C_{N,M},C_{N+i,M}) \approx 1$$
$$P(Ap|d(C_{N,M},C_{N,M+i}) \approx d(C_{N,M},C_{N,M+(i+1)}))$$
$$P(Bp|d(C_{N,M},C_{N+i,M}) \ll d(C_{N,M},C_{N,M+i})) \qquad (7)$$

The matrix is visualized in Fig. 3. and colored cells represents high similarity and white colored cells match low similarity. The similarity from a point of interest to itself is blue colored which is actually defined as zero

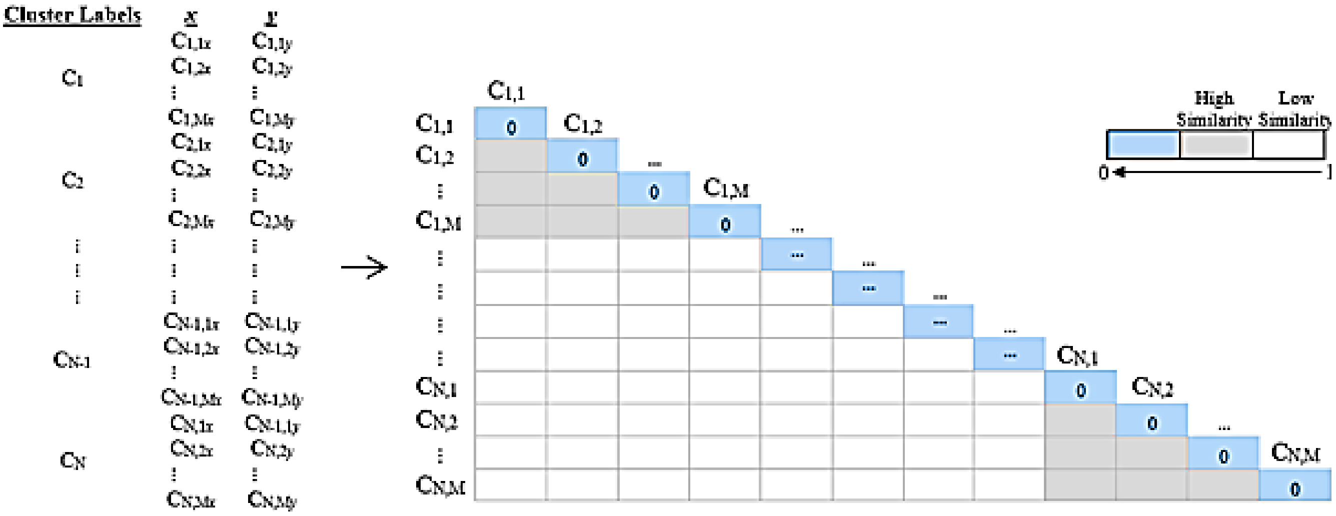


Fig. 3. Structure of Similarity Matrix

Comparativeinspection of the visual representation of the averaged and scaled similarity scores was based on a heat map, as shown in Fig. 4. The heat map orders the similarity scores with respect to the clusters, using color coding. The allocated intensity of the colors are dependent on the value, ranging from the lowest (0-highlighted green) through average (0.5- highlighted yellow) to the highest (1-highlighted red). Since the color hierarchy is from red through yellow to green, with high similarity scores attaining red colors and low scores attaining green color, this could be straightforwardly identify from the heat map, which data points forms a particular cluster, using the extent to which its similarity score is high (highlighted in red). So an estimate on how the proposed PSO affect the intrinsic local structure of the dataset and the pairwise similarity distances of data point, could be clearly deduced. From the heat map (a) visually presents some uneven spreads of data within some of the clusters whereas (b) shows clusters that have crisped data. Resemblances between the heat maps (b) and (c) in terms the diagonal positioning of clusters and high similarity scores, depicts how well a lot of the original information are retained after processing. This confirms that the proposed PSO clustering strategy, preserve the original clustering feature within the dataset.

Additional experiment is carried out to test the accuracy of the geometric-inspired PSO (g_PSO)approach against the traditional or standard PSO (s_PSO) on classification tasks. The task here was perform classification on a sub-dataset from the larger available EMNIST dataset.To ensureconsistency in a standard results, a comparison of g_PSOin classification, with s_PSO also in classification, wasdone by selecting the generation of g_PSOclassification with an equalnumber of clusters corresponding to the s_PSO classification. Table 1 presents and compare the results of g_PSO and s_PSO PSO in classification taskfor accuracy and standard deviation for 31 runs.

Table 1.Comparison of accuracy and Standard deviation (σ) of accuracy for Geometric-inspired PSO (g_PSO) and standard PSO (s_PSO) clustering using EMNIST Dataset.

Method	Avg	Accuracy	StdDev.(σ)
g_PSO	0.865	*88.6*	1.12
s_PSO	6.033	75.3	4.6

From Table 1, the accuracy of g_PSOis better thans_PSOin classification task on the subset dataset of EMNIST, with an accuracy of 88.6 and a lower average intra-cluster value of 0.865.

In our next experiment Table 2 is constructed. The table shows a calculated intra-cluster averages for each individual class to demonstrate the compactness of each class.It highlights the distribution of data within each class and throughout the entire dataset of 1000 data points/particles. The first columns represents the label of the classes, the second columnrepresents the number of data points per class and the thirdcolumn represents the average intra-cluster of an individual classwhereas the last and fourth column reprints the overall intra-cluster average

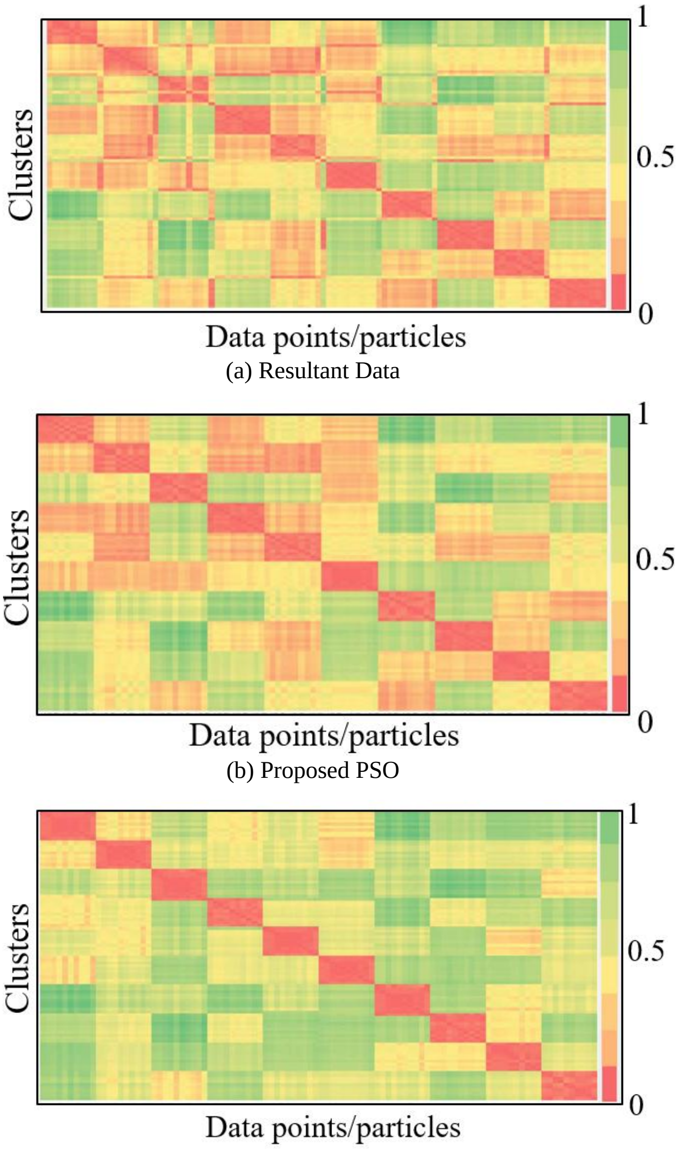
Table 2. Cluster distribution of Geometric-inspired PSO

Data		Per-class avgintra-cluster	Avg Intra-cluster
Label	Data per class		
0	109	0.79	0.865
1	106	0.72	
2	121	0.91	
3	89	1.07	
4	81	0.62	
5	112	0.39	
6	105	0.74	
7	92	0.68	
8	94	0.51	
9	91	2.22	

Table 2 presents detailed intra-cluster averages of Geometric-inspired PSO. These results show that in terms of intra-cluster averages theperformanceof the proposed technique is better and competitive.

5. CONCLUSION

Geometric application in PSO is a generalization of the classical PSO to the general Cartesian spaces. In particular, it applies the concepts of points, angles, composite transformations, line segments and convex combinations to these spaces. In this paper, a new geometric technique has been applied in parallel to PSO, to determine the effect of inertia components on social and cognitive regions. This is believe to be the first step-by-step basic strategy to directly apply the afore-mentioned concepts of geometry in PSO and further represented in mathematical expressions. In this algorithm, the three regions were explicitly created using geometric knowledge to prove how positions of particles are computed over a certain time of iterations.



(a) Resultant Data

(b) Proposed PSO

(c) Ground truth data

Fig. 4. Heat map for similarity scores between data points (a) before and (b) after the PSO was implemented and compared to (c) ground truth data

This study defines and compares how these three regions contribute to particles' movement. Additionally, the significant contribution of this paper was to clearly find out which of these three components is of much influence and on what sort of functions does it work best. In particular and without any doubts, the study confirms that the inertia component still remains the most influential component on the social and cognitive regions. The quality and reliability of the proposed PSO approach was tested and endorsed using classification problem. It was used to validate whether the classes formed from the dataset using the proposed PSO algorithm, was not by chance. This work has demonstrated that the proposed PSO on datasets results in an acceptable classification performance. The statistical estimate of correlation and similarity scoring, validated that the local structure of the data was maintained and was aligned with the classification process. So what the proposed PSO approach does is to guarantee the distinctiveness among classes and simultaneously preserve the similarities between these classes. Classes were formed in such a way that objects in the same class are very similar and objects in different classes are very distinct. The initial experiments show that it performs to an acceptable level on standard classification benchmarks, though competitive results may exist in literature

6. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

7. REFERENCES

[1] Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. Data classification: Algorithms and applications, 37-64, *Chapman & Hall/CRC, Boca Raton, FL*. ISBN:978-1-4665-8674-1.

[2] Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., et. al. (2020). Analysis of dimensionality reduction techniques on big data. IEEE Access, 8, 54776-54788.

[3] Sugiyama, M. (2007). Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. Journal of machine learning research, 8(5).

[4] Song, J., Dauphin, Y., Auli, M., & Ma, T. (2020). Robust and on-the-fly dataset denoising for image classification. In European Conference on Computer Vision (pp. 556-572). Springer, Cham.

[5] Sharma, P., & Kaur, M. (2013). Classification in pattern recognition: A review. International Journal of Advanced Research in Computer Science and Software Engineering, 3(4).

[6] Singh, B., Dubey, V., &Sheetlani, J. (2018). A review and analysis on knowledge discovery and data mining techniques. International Journal of Advanced Technology and Engineering Exploration, 5(41), 70-77.

[7] Kalanat, N. (2021). An overview of actionable knowledge discovery techniques. Journal of Intelligent Information Systems, 1-21.

[8] Masaeli, M., Dy, J. G. & Fung, G. (2010), From transformation-based dimensionality reduction to feature selection. In Proceedings of the 27th International Conference on Machine Learning (ICML), pages 751–758.

[9] Omuya, E. O., Okeyo, G. O., &Kimwele, M. W. (2021). Feature selection for classification using principal component analysis and information gain. Expert Systems with Applications, 174, 114765.

[10] Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S., & Moore, J. H. (2018). Relief-based feature selection: Introduction and review. Journal of biomedical informatics, 85, 189-203.

[11] Ghosh, P., Azam, S., Jonkman, M., et. al. (2021). Efficient Prediction of Cardiovascular Disease Using Machine Learning Algorithms with Relief and LASSO Feature Selection Techniques. IEEE Access, 9, 19304-19326.

[12] Aksu, D., Üstebay, S., Aydin, M. A., &Atmaca, T. (2018). Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm. In International symposium on computer and information sciences (pp. 141-149). Springer, Cham.

[13] Saqlain, S. M., Sher, M., Shah, F. A., Khan, I., Ashraf, M. U., Awais, M., & Ghani, A. (2019). Fisher score and Matthews correlation coefficient-based feature subset selection for heart disease diagnosis using support vector machines. Knowledge and Information Systems, 58(1), 139-167.

[14] Lei, C., & Zhu, X. (2018). Unsupervised feature selection via local structure learning and sparse learning. Multimedia Tools and Applications, 77(22), 29605-29622.

[15] Zhao, T., Liu, H., & Zhang, T. (2018). Pathwise coordinate optimization for sparse learning: Algorithm and theory. The Annals of Statistics, 46(1), 180-218.

[16] Ghosh, J., &Shuvo, S. B. (2019). Improving Classification Model's Performance Using Linear Discriminant Analysis on Linear Data. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[17] Guo, Y., Ding, X., Liu, C., &Xue, J. H. (2016). Sufficient canonical correlation analysis. IEEE Transactions on Image Processing, 25(6), 2610-2619.

[18] Agis, D., &Pozo, F. (2019). A frequency-based approach for the detection and classification of structural changes using t-SNE. Sensors, 19(23), 5097.

[19] Xu, X., Xie, Z., Yang, Z., Li, D., & Xu, X. (2020). A t-SNE based classification approach to compositional microbiome data. Frontiers in Genetics, 11, 1633.

[20] Eberhart, R., & Kennedy, J. (1995). Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks, . 4, 1942-1948, IEEE Service Center, Piscataway, New Jersey

[21] Eberhart R. C. & Shi Y. (2001). Tracking and optimizing dynamic systems with particle swarms, In *Evolutionary Computation. Proceedings of the 2001 Congress*. 1, 94–100, IEEE.

[22] Liang, J. J., Qin, A. K., Suganthan, P. N., &Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE transactions on evolutionary computation*, 10(3), 281-295.

[23] Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. Soft Computing, 22(2), 387-408.

[24] Gandomi, A. H., Yang, X. S., Talatahari, S., &Alavi, A. H. (Eds.). (2013). Metaheuristic applications in structures and infrastructures. Newnes.

[25] Sun, S., & Liu, H. (2013). Particle swarm algorithm: convergence and applications. In Swarm Intelligence and Bio-Inspired Computation (pp. 137-168). Elsevier.

[26] Tran, B., Xue, B., & Zhang, M. (2014). Overview of particle swarm optimisation for feature selection in classification. In Asia-Pacific conference on simulated evolution and learning (pp. 605-617). Springer, Cham.

[27] Yang, C. S., Chuang, L. Y., Ke, C. H., & Yang, C. H. (2008). Boolean binary particle swarm optimization for feature selection. In 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence) (pp. 2093-2098). IEEE.

[28] Sousa, T., Neves, A., & Silva, A. (2003). Swarm optimisation as a new tool for data mining. In *Proceedings international parallel and distributed processing symposium* (pp. 6-pp). IEEE.

[29] Sousa, T., Silva, A., &Neves, A. (2003). A particle

swarm data miner. In *Portuguese Conference on Artificial Intelligence* (pp. 43-53). Springer, Berlin, Heidelberg.

[30] Sousa, T., Silva, A., &Neves, A. (2004). Particle swarm based data mining algorithms for classification tasks. *Parallel computing*, 30(5-6), 767-783.

[31] Cervantes, A., Galván, I. M., &Isasi, P. (2005). Binary particle swarm optimization in classification, *Neural Network World*, Volume 15, Issue 3, Pages 229 – 241.

[32] Zahiri S. H., &Seyedin S.A. (2005). Intelligent particle swarm classifier, Iranian Journal of Electrical and Computer Engineering, Volume 4, Issue 1, Pages 63–70

[33] De Falco, I., Della Cioppa, A., & Tarantino, E. (2005). Evaluation of particle swarm optimization effectiveness in classification. In *International Workshop on Fuzzy Logic and Applications* (pp. 164-171). Springer, Berlin, Heidelberg.

[34] Van der Merwe, D. W., &Engelbrecht, A. P. (2003). Data clustering using particle swarm optimization. In The Congress on Evolutionary Computation, 2003. CEC'03. (Vol. 1, pp. 215-220). IEEE

[35] Omran, M., Engelbrecht, A. P., & Salman, A. (2005). Particle swarm optimization method for image clustering. International Journal of Pattern Recognition and Artificial Intelligence, 19(03), 297-321.

[36] Gao L., Gao H., Zhou C., & Yu D. (2004) Acquisition of pattern classification rule based on particle swarm optimization, HuazhongKejiDaxueXuebao (ZiranKexue Ban)/Journal of Huazhong University of Science and Technology (Natural Science Edition), Volume 32, Issue 11, Pages 24 - 26November 2004.

[37] Kuo, R. J., Chao, C. M., & Chiu, Y. T. (2011). Application of particle swarm optimization to association rule mining. *Applied soft computing*, 11(1), 326-336.

[38] Sarath, K. N. V. D., & Ravi, V. (2013). Association rule mining using binary particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 26(8), 1832-1840.

[39] Ab Wahab, M. N., Nefti-Meziani, S., &Atyabi, A. (2015). A comprehensive review of swarm optimization algorithms. *PloS one*, 10(5), e0122827.

[40] Chang Y & Yu G (2013). Multi-Sub-Swarm PSO Classifier Design and Rule Extraction. Int. Work. Cloud Computing Information Security. 2013: 104–107.

[41] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters.In *2017 International Joint Conference on Neural Networks (IJCNN)* 2921-2926). IEEE, *arXiv preprint arXiv:1702.05373*.