C. Ratanavilisagul, "Modified Feature Selection with BPSO to Apply PSO for Solving Handwritten Digits," 2024 12th International Electrical Engineering Congress (iEECON), Pattaya, Thailand, 2024, pp. 1-6, doi: 10.1109/iEECON60677.2024.10537900.

Modified Feature Selection with BPSO to apply PSO for solving Handwritten Digits

Chiabwoot Ratanavilisagul
Department of Computer and Information Sciences, Faculty of Applied Science
King Mongkut's University of Technology North Bangkok (KMUTNB)
Bangkok, Thailand
chaibwoot@gmail.com, chaibwoot.r@sci.kmutnb.ac.th

***Abstract*—The task of Handwriting Digits Recognition represents a classification challenge involving the interpretation of handwritten digits sourced from diverse mediums such as paper, photos, touch screens, and other devices. This challenge is very intricate due to the distinctiveness of each individual's handwriting, and several essential traits that influence the interpretation process. Numerous researchers have devoted efforts to solve this problem. Recent research has improved preprocessing and feature extraction techniques by using particle swarm optimization to address the handwritten digit recognition problem. The outcomes of these experiments have shown obtain good results. Nevertheless, outcomes from this technique can be further improved. The feature selection technique popularly applies pattern recognition to improve outcomes. Thus, this study proposes a modified feature selection technique that developed from binary particle swarm optimization to optimize particle swarm optimization for handwritten digit recognition. This proposed feature selection technique has demonstrated the potential to yield higher Recognition Rates significantly in both training and testing phases. Comparative analysis against the original technique, devoid of the feature selection technique, reveals that the proposed method exhibits superior Recognition Rates when tested on both the original MNIST datasets and the individual's each person handwritten digit datasets.**

Keywords— Handwritten Digits Recognition, Particle Swarm Optimization, K-Means Binary Particle Swarm Optimization, Feature Selection

I. Introduction

Handwriting recognition (HWR) is the computer's capacity to comprehend and interpret human handwriting, often found in sources like paper documents, photographs, touch screens, and more. Handwritten Digits Recognition (HDR) is a specific aspect of HWR, focusing solely on interpreting handwritten numbers, such as digits from 0 to 9 [1]. This problem poses a significant challenge in the field of pattern recognition due to the inherent variability in people's handwriting. Each individual's handwriting carries unique traits influenced by personal style and conditions at the time of writing. Consequently, even if the same individual writes the same number twice, the results might not match [2]. Notably, there are numerous defining characteristics in handwriting [3], including aspects like line quality, spacing, height, width, font size, stroke variations, stroke connections, slant, pen angle, and other writing conditions. These intricacies make the task of HDR particularly demanding. The Modified National Institute of Standards and Technology database (MNIST) [4], [5] is a well-known handwritten datasets and it is used to compare classification performance of algorithms. Over the years, the MNIST dataset has spurred extensive research, contributing to the development of more efficient HDR methods. Recently, many researchers have suggested employing heuristic algorithms to address the challenges in HDR and related problems. Heuristic and Meta-heuristic algorithms like Particle Swarm Optimization (PSO) [6-9] and Ant Colony Optimization (ACO) [10], [11] have proven to be effective in addressing highly complex problems. PSO, in particular, has shown promise and is well-suited for solving HDR [7], [8].

PSO draws its inspiration from the behaviors of flying birds and their methods of communication [12]. It is a stochastic global optimization technique employed for solving optimization problems with the primary objective of finding the global optimum of a fitness function. PSO boasts several advantages, including rapid convergence, simplicity, and minimal parameters to adjust [6].

Notably, technologies like Pytesseract [13], Google Lens [14], and Apple Live Text [15] have evolved from Deep Learning (DP). In their study, J. Samleepant and C. Ratanavilisagul [8] experimented with these technologies using datasets of individual handwritten digits contributed by thirty-five volunteers. These datasets had never been subjected to prior training. Their experimental results revealed that, without training, DP exhibited subpar predictive performance. In comparison, when pitted against algorithms emphasizing swift learning and minimal resource consumption like PSO, the algorithm proposed in their paper, which employed PSO for training, yielded superior results, including higher Recognition Rates.

As previously mentioned, the advantages of PSO include swift convergence and simplicity. When applied to the task of HDR, PSO facilitates rapid model training. Therefore, PSO is a suitable choice for practical applications in HDR. Consequently, this paper centers its attention on the utilization of PSO for HDR problem-solving. In recent times, many researchers use of PSO for HDR problem-solving, including: N. O. S. Ba-Karait and S. M. Shamsuddin [7] employed PSO to tackle the HDR. They encoded HDR as a multi-dimensional feature space and employed PSO to seek the best positions for each feature space's centroid. Their experiments using the MNIST dataset demonstrate that PSO is effective in addressing HDR challenges. J. Samleepant and C. Ratanavilisagul [8] made adjustments to the feature extraction process to facilitate the use of PSO in solving the HDR. Pre-processing methods were applied prior to PSO. This pre-processing involved steps such as noise removal,

image segmentation, thinning processing, and checking the pixel chain code (TCCC) to extract features from the images. Subsequently, K-means clustering was used to categorize the data. PSO was then employed to search for the optimal positions of each feature space's centroids. This searching process served as training to create a new model. The proposed method is referred to as PSO with thinning processing and checked pixel chain code processing (PSOTCCC). PSOTCCC was evaluated using both MNIST datasets and datasets containing individual's handwritten digits, achieving a high recognition rate with reduced runtime.

Feature Selection (FS) functions as an intermediary process that diminishes the size of feature sets, limiting them to only the vital and pertinent features [16]. This aspect holds significant importance within the domain of pattern recognition as it discerns informative features, thereby mitigating the challenge of dimensionality and ultimately enhancing the overall accuracy of classification [17]. Several researchers have introduced feature selection methods developed through Metaheuristics to address Handwritten Digits Recognition (HDR) and similar problems, aiming to achieve higher Recognition Rates: L. M. Seijas et al. [18] presented feature selection techniques based on Binary Fish School Search (BFSS), Advanced Binary Ant Colony Optimization (ABACO), and Binary Particle Swarm Optimization (BPSO) for HDR. Their experiments demonstrate that feature selection plays a crucial role in significantly enhancing classification performance. Victor L. F. Souza et al. [19] proposed feature selection using BPSO for handwritten signature verification. Their experiments indicated that BPSO-based feature selection improves model performance within SVM, yielding superior results. Guha, R., and colleagues [20] integrated PSO into the feature selection process for Handwritten Arabic Character recognition. Their experiments showcased that the proposed feature selection method enhances the accuracy of recognizing isolated handwritten Arabic characters. Guha et al. [17] introduced a novel approach called Memory-Based Histogram-Oriented Multi-objective Genetic Algorithm for feature selection in handwritten numeral recognition. Their experiments illustrated that employing feature selection in this context enhances accuracy and reduces computation time, signifying a contemporary advancement in the field. Sarra Rouabhi et al. [16] proposed feature selection techniques utilizing BPSO and Binary Gravitational Search Algorithm. Their experiments demonstrated that this method notably improves recognition accuracy. As previously noted, Feature Selection plays a pivotal role in reducing feature dimensionality, thereby augmenting recognition accuracy. This study proposes the integration of Feature Selection techniques with models generated by PSO to further enhance recognition accuracy. PSOTCCC specifically operates with a limited number of feature dimensions during training and testing. Therefore, employing Feature Selection alongside PSOTCCC becomes crucial for improving recognition accuracy. Moreover, the heuristic algorithms used for FS need to exhibit rapid convergence, simplicity, minimal parameter adjustment, and utilize resources from the PSO process. This ensures easy implementation while achieving optimal performance. Accordingly, this research suggests the implementation of Feature Selection using BPSO to enhance recognition accuracy. Consequently, this paper recommends enhancing PSOTCCC by incorporating Feature Selection techniques post-execution, aiming to achieve higher Recognition Rates using models derived from PSOTCCC.

The remainder of this document is structured as outlined below: Section 2 explains the background (Particle Swarm Optimization, Binary Particle Swarm Optimization, Euclidean Distance, K-means Clustering, and PSO with Data Clustering). Section 3 explains the related works (PSOTCCC). Section 4 explains the proposed method. Section 5 explains the experiment results. Section 6 and concludes this research and directions for development this research in future work.

II. BACKGROUNDS

A. Particle Swarm Optimization

PSO finds its roots in emulating the collective behavior observed in nature, such as the synchronized movement observed in bird flocks or fish schools. Falling under the realm of stochastic optimization algorithms, PSO aims to pinpoint optimal solutions. At its core, PSO is inspired by the collaborative nature of animal groups seeking sustenance. These groups communicate and collaborate to locate food sources, directing the entire group toward these essential resources. In the context of PSO, these food sources symbolize the target solutions that particles strive to uncover.

Each entity within the particle population is denoted as a "particle" or "X," and each particle holds distinct position and velocity attributes. These particles navigate through the search space guided by their velocities, the best position discovered among all particles known as GBEST, and their individual best positions referred to as PBEST. PSO initiates by randomly setting particle positions and velocities, and the evaluation of each particle's position is established by the objective function of the optimization problem. Throughout each iteration, every particle revises both its position and velocity following the formula provided below:

$$V'_{i,d} = \varpi V_{i,d} + \eta_1 \text{rand}()(P_{i,d} - X_{i,d}) + \eta_2 \text{rand}()(P_{g,d} - X_{i,d}) \quad (1)$$

$$X'_{i,d} = X_{i,d} + V'_{i,d} \quad (2)$$

Where $X_{i,d}$ represents the previous positions of the i-th particle in the d-th dimension, $X'_{i,d}$ denotes the current positions of the i-th particle in the d-th dimension, $V_{i,d}$ is indicative of the previous velocity of the i-th particle in the d-th dimension, $V'_{i,d}$ represents the current velocity of the i-th particle in the d-th dimension, $P_{i,d}$ stands for the individual best (PBEST) position of the i-th particle in the d-th dimension, and $G_{i,d}$ signifies the global best (GBEST) position in the d-th dimension. ϖ corresponds to the inertia weight, η_1 and η_2 are acceleration constants, and rand() is a uniformly generated random number within the range of [0, 1]. There's a defined maximum velocity (V_{max}), and if the computed velocity of a particle exceeds this limit, it is replaced with the value of V_{max}.

B. Binary Particle Swarm Optimization

Kennedy and Eberhart [21] pioneered the binary particle swarm optimization (BPSO) algorithm, enabling the standard PSO to function within binary problem spaces. BPSO has since become widely embraced as a standard practice. In this algorithm, each dimension of a particle consists solely of 0 or 1 values. During the initialization phase, the initial

population randomly assigns each dimension of the particle to either 0 or 1, with the probability of selecting 1 governed by a value between 0 and 1 referred to as POP. BPSO updates the velocity according to (3), where the velocity is represented by a set of real numbers while the particle position is comprised of bits. Therefore, transforming the velocity into a set of probabilities is achieved using the sigmoid function, as illustrated in (4). The position update principle is guided by the velocity, dictating the probability for each position (bit) to select either zero or one. The position update procedure is detailed in (5), where $\text{sig}(v'_{i,d})$ represents the sigmoid function responsible for converting the velocity into probabilities.

$$v'_{i,d} = v_{i,d} + \eta_1 rand()(P_{i,d} - X_{i,d}) + \eta_2 rand()(P_{g,d} - X_{i,d}) \quad (3)$$

$$\text{sig}(v'_{id}) = \frac{1}{1 + e^{-v'_{id}}}, e \approx 2.7182818 \quad (4)$$

$$X'_{i,d} = \begin{cases} 1 & if\ rand() < \text{sig}(v'_{id}) \\ 0 & Otherwise \end{cases} \quad (5)$$

C. Euclidean Distance

The Euclidean Distance calculates the spatial separation between two points within a coordinate system. It relies on the principles of the Pythagorean Theorem. The K-Nearest Neighbors (KNN) algorithm extensively utilizes this method as a fundamental component, primarily because it leverages all pertinent data to classify items based on their proximity in a feature space [22], [23]. Its definition is as follows:

$$D = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \quad (6)$$

In the context of Euclidean Distance, D represents the spatial separation between two points. When dealing with two-dimensional space, this distance is essentially the measure between two sets of values (x_i, y_i), where i range from 1 to k, and k denotes the data points involved.

D. K-Means Clustering

The K-means algorithm is a widely used unsupervised learning technique employed to cluster unlabeled data by grouping features. Its primary objective is to minimize the average squared distance between points within the same cluster. In conventional K-means, individual data points are referred to as "objects." A collection of objects forms a "class" or "centroid vector," and both classes and objects possess specific attributes. The classification is achieved by measuring the Euclidean distance between the attributes of objects and classes. The fundamental idea behind K-means can be summarized as follows: an object is assigned to a particular class based on the smallest Euclidean distance between its attributes and those of the class. Standard K-means initiates by randomizing attributes of classes, and the evaluation of each class is determined using the following expression:

$$d(z_p, m_j) = \sqrt{\sum_{k=1}^{N_d} (z_{p,k} - m_{j,k})^2} \quad (7)$$

$$J_e = \frac{\sum_{j=1}^{N_c} [\sum_{\forall Z \in C_{i,j}} d(z_p, m_j) / |C_{i,j}|]}{N_c} \quad (8)$$

In this context, N_d represents the total number of attributes, N_c signifies the total number of classes, Z_p is an object denoted as p, $Z_{p,k}$ corresponds to attribute k within object p, m_j represents class j, m_{jk} refers to attribute k within class j, $|C_{i,j}|$ is the count of objects belonging to class j, and $d(z_p, m_j)$ indicates the distance between object p and class j. The algorithm can be terminated when the maximum allowable number of iterations is reached. Its practical appeal lies in its simplicity and speed. The algorithm operates as follows: Step 1, Determine the number of clusters by designating them as k reference points (representing the number of neighboring points). Step 2, allocate each data point to the nearest k reference point. Step 3, update the positions of these k reference points. Step 4, Continue to iterate through steps 2 and 3 using the updated cluster centers until there is no further movement.

E. Particle Swarm Optimization with Data Clustering

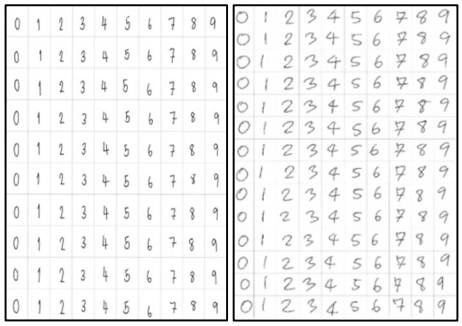
The classic PSO algorithm was originally designed to address benchmark function problems. To adapt PSO for data clustering tasks, modifications are made to the objective function and particle positions. In this context, an individual particle corresponds to a specific class [9]. The fitness of a particle is evaluated using (4) and (5). The PSO procedure for data clustering is identical to the standard PSO process. Except, every particle consists of the centroids of all the classes, resulting in a two-dimensional array for X_i, where $x_{i,d}$ can be expressed as $x_{i,j,k}$. Here, 'j' represents the class of each centroid, and 'k' denotes the attributes of each class. Likewise, the velocity of each particle is composed of the centroids of all classes, yielding a two-dimensional array for v_i, with $v_{i,d}$ being rewritten as $v_{i,j,k}$.

III. Relate Works

J. Samleepant and C. Ratanavilisagul modified the feature extraction for applying PSO to create model for solving HDR. The proposed feature extraction used a novel black pixel-by-pixel counting methods. Their proposal method is called PSOTCCC. The processes of PSOTCCC have a total of 4 steps and can summarize as follows: Step 1, Fig. 1 displays a sample of handwritten digits from an individual. Initially, the original image is cropped to isolate the handwritten numbers within a single frame. Subsequently, the Discrete Wavelet Transform (DWT) technique is employed to extract crucial features, effectively eliminating unwanted artifacts that can interfere with image processing, such as specks and scanning-related dust. Furthermore, image segmentation is used to trim the surrounding white space, ensuring that the numbers are centered. This is important because some input datasets may include images with off-center numbers, located in various positions (top-left, top-right, bottom-left, bottom-right, etc.), which could lead to recognition errors. Finally, the image is resized to 10x10 pixels, allowing for straightforward analysis of the black pixel count, representing the digital number. Step 2, Thinning is a technique that simplifies the input data by removing any dots or the outer layer of a number until it becomes a single pixel in width. In other words, it transforms all the strokes or arcs of each input digit into thin objects. The outcome is an array composed of binary values, where 1 indicates a black pixel and 0 signifies a white pixel. The subsequent step involves tallying the black pixels in the resulting array, a method referred to as TCCC, which draws inspiration from the Freeman chain code approach [24]. The process of counting the black pixels is categorized into four types: First, counting 10x2 pixels in each row yields 5 attributes. Second, counting 5x5 pixels in the up-left, up-right, bottom-left, and bottom-right regions yields 4 attributes. Third, counting 10x5 pixels in the upward and downward directions yields 2 attributes. Finally, counting

5x10 pixels in the left and right directions yields 2 attributes. In total, there are 13 attributes in this approach. Step 3, The outcomes from earlier stages were incorporated into the PSO method for the training phase. This led to the generation of the output model, which represents the optimal position of each individual feature. Step 4, The model derived from PSO is utilized during the testing phase to assess the recognition accuracy of the model.

Fig. 1. The individual's each person handwritten digit datasets example from volunteers.



IV. Proposal Works

The PSOTCCC method surpasses other PSO-based training techniques in achieving superior recognition rates. Nevertheless, models come from using PSOTCCC can be further developed to achieve heightened recognition rates by incorporating Feature Selection technique. In case of PSOTCC, the feature is to attribute or dimension. Feature Selection involves the selection of crucial attributes from the entire set to reduce dimensionality. This reduction in the number of dimensions leads to faster training and testing processes, requiring fewer resources. By isolating essential attributes, this process eliminates numerous noisy attributes. These noisy attributes, if included, can lead to errors in data classification. Consequently, Feature Selection significantly amplifies model performance, resulting in elevated Recognition Rates during the training phase. Notably, heightened Recognition Rates during training often correspond to similarly improved rates during testing or practical usage phases. As a result, Feature Selection contributes to accelerated processes and enhanced overall classification accuracy, underscoring its crucial role in the realm of pattern recognition.

As previously mentioned, this research proposes the utilization of Feature Selection in conjunction with PSOTCCC to enhance overall classification accuracy. Typically, Feature Selection is employed before the model creation process to reduce dimensions and expedite training. However, executing Feature Selection before the creation of models with PSOTCC is not suitable due to the small number of dimensions in PSOTCCC (13 attributes). Implementing Feature Selection prior to the PSOTCCC process disrupts and diminishes the performance of the model creation within PSOTCCC. Models created from PSOTCCC with Feature Selection perform worse than those without it. Therefore, for PSOTCCC, I suggest employing Feature Selection after the model is created by PSOTCCC. This approach enables Feature Selection to refine the model and enhance classification accuracy, ensuring that Feature Selection is applied post the model creation by PSOTCCC. Consequently, this method yields an improved model and achieves higher recognition rates in both the training and testing phases.

Moreover, heuristic algorithms utilized for Feature Selection should exhibit swift convergence, simplicity, require minimal parameter adjustments, and seamlessly integrate resources with the PSO process. This study introduces Feature Selection implemented through BPSO to augment recognition accuracy, as BPSO effectively utilizes resources alongside PSO. BPSO implementation enables it to use all resources from PSO such as particles, velocities, PBEST, and GBEST.

The outcome of the PSOTCCC is a classification model, employed for data classification purposes. This model undergoes enhancement through the feature selection developed using BPSO. BPSO collaborates with the classification model to generate a feature selection model, comprised of binary numbers. If a dimension is designated as 1, it is considered during the classification phase, whereas if set as 0, that dimension is excluded from the classification. Executing BPSO produces the feature selection model. During testing or practical use, both the classification model and the feature selection model work in tandem. Integrated feature selection aids in eliminating noise attributes. The results from the feature selection model alongside the classification model can elevate Recognition Rates in both training and testing phases. This paper introduces the application of Feature Selection derived from Binary Particle Swarm Optimization with Particle Swarm Optimization, specifically within the context of thinned processing and checked pixel Chain Code processing (FPSOTCCC).

The main concept of FPSOTCCC can be explained as follows: initiating feature extraction in line with the PSOTCCC process. Subsequently, the PSO process runs until completion, culminating in the creation of the classification model. Following this, the BPSO process is implemented with the classification model to produce the feature selection model. Ultimately, both models are employed for data classification during the test or usage phases. The pseudo code for FPSOTCCC is provided as follows.

```
Initial PBEST, GBEST, RBEST
For gen = 0 to MAXGEN
    For i = 0 to N
        Evaluate particle fitness according to (8)
        If F(Xi) < F(PBESTi)
            PBEST = Xi
        End If
        If F(Xi) < F(GBEST)
            GBEST = Xi
        End If
        If F(Xi) < F(RBEST)
            RBEST = Xi
        End If
        Update particle position according to (1) and (2)
    End For
Next gen
Initial position and velocity of each particle (dimensions of particles are changed to binary form)
Initial PBEST, GBEST
For gen = 0 to BMAXGEN
    For i = 0 to N
        Evaluate fitness of RBEST according to (8) if Xi, d = 1
        If F(Xi) < F(PBESTi)
            PBEST = Xi
        End If
        If F(Xi) < F(GBEST)
            GBEST = Xi
        End If
        Update particle position according to (3) and (5)
    End For
Next gen
Save RBEST for the classification model created by PSO from Training phase
Save GBEST for the feature selection model created by BPSO from Training phase
```

Use both models to classify data in Test phase.

In the context of these parameters: N_a stands for the number of attributes, N_c represents the number of classes, N_{ch} signifies the count of input data or characters used for training, N denotes the total number of particles, RBEST refers to the best answer that hasn't been reset, $F(X_i)$ corresponds to the fitness of the 'i' particle, $F(PBEST_i)$ is the fitness derived from PBEST for the 'i' particle, $PBEST_i$ designates the PBEST of the 'i' particle, F(GBEST) signifies the fitness of the global best (GBEST). MAXGEN is the maximum number of generations of PSO. BMAXGEN is the maximum number of generations of BPSO, and $a_{i,j,k}$ represents the value of attribute 'k' for class 'j' and character 'k'.

$$x_{i,j,k} = \frac{\sum_{p}^{N_{ch}} a_{p.j.k}}{N_{ch}} \quad (9)$$

$$V_{max,d} = 0.1 \times (UL_d - LL_d) \quad (10)$$

$$V_{i,d} = -\mathrm{V}_{max,d} + \mathrm{rand}() \times 2 \times V_{max,d} \quad (11)$$

Equations (9), (10), (11) are used with the initial position and velocity of each particle, in both FPSOTCCC and PSOTCCC. Equation (9) is employed to determine the initial position of each particle, and this is computed based on the average value of a characteristic of class 'j' and attribute 'k'. Equation (10) is utilized to calculate V_{max} for attribute 'd', with UL_d representing the maximum value for attribute 'd', LL_d representing the minimum value for attribute 'd', and $V_{max,d}$ indicating the maximum velocity for attribute 'd'. Equation (11) is employed to establish the initial v Initial position and velocity of each particle according to (9) and (11)

V. Experiments and Results

A. Datasets

FPSOTCCC and PSOTCCC were subjected to testing using the MNIST datasets, and datasets of handwritten digits contributed by thirty-five volunteers. The MNIST dataset comprises 60,000 training samples and 10,000 test samples, and it can be downloaded from [25]. In the preprocessing stage, the images within the MNIST dataset are converted into binary numeral bitmaps, centered, and size-normalized to 28x28 pixels. The dataset containing handwritten digits from individuals consists of 35 datasets, with each dataset corresponding to the work of a different volunteer who filled out an A4-sized paper using a 0.7 mm black gel pen, resulting in numbers ranging from 0 to 9, as shown in Figure 1. The scanned papers were processed using a clean, dust-free scanner that ensured minimal interference with the model's recognition. The scanned files are in ".BMP" format, and the paper size is 1600x2240 pixels, with each paper containing 100 characters. For training, 90 characters from each row, spanning rows 1 to 9, were utilized, while the remaining 10 characters in row 10 were reserved for testing. The MNIST dataset is employed as a standard benchmark for assessing algorithm performance, whereas the dataset of handwritten digits from individuals is utilized to demonstrate the practical applicability of the results.

B. The Measures of Algorithm Performance

For evaluating the algorithm's performance, I rely on the average results obtained from multiple runs, as both FPSOTCCC and PSOTCCC are stochastic algorithms. The performance metrics used in the experiments are described as follows: The Average Recognition Rate (ARR) Percentage is calculated as the average percentage of correctly identified predictions divided by the total number of data across all runs. ARR serves as an indicator of the algorithm's classification performance. Higher ARR values correspond to better classification performance. SD, which stands for "Standard Deviation," serves as an indicator of the reliability of a clustering algorithm. A smaller SD value signifies a higher level of reliability in the clustering results.

C. Parameters Setting

The parameters used in all experiments for PSO are as follows: both acceleration constants, η_1 and η_2, are set to 1.45, ϖ is set to 0.75, MAXGEN is set to 100 and the population size is 5000. The parameters used in all experiments for BPSO are as follows: η_1 and η_2 are set to 2. POP is set to 0.8. BMAXGEN is set to 100. When working with the MNIST datasets, 10 runs of experiments are conducted. In the case of the dataset comprising individual's handwritten digits, 10 runs are performed for each paper. The parameters for the compared algorithms and the proposed algorithm are configured based on the recommendations outlined in the original paper. This research is conducted by a personal computer of AMD FX-8320 with 3.5 GHz CPU, 8 GB RAM and Visual C++ 2010 as the programming language.

D. Experiment of Proposed Algorithm

The experimental in Table 1 and Table 2 show FPSOTCCC outperforms PSOTCCC in terms of classification accuracy, evident in its higher ARR across all datasets, encompassing both the MNIST datasets and the individual's handwritten digit dataset. Additionally, FPSOTCCC showcases superior reliability, denoted by lowest SD values across all datasets. Thus, FPSOTCCC excels over PSOTCCC in both classification accuracy and reliability. The experimental illustrates FPSOTCCC can improve results from PSOTCCC about 5% both two datasets. These experiments show that BPSO-derived feature selection substantially enhances the classification performance of PSOTCCC in handling HDR, signifying the effectiveness of feature selection in augmenting classification accuracy. The data presented in two tables indicate that achieving a high classification performance during the training stage has a significant influence on obtaining high classification performance in the testing stage as well. Consequently, FPSOTCCC generates an good model with a high recognition rate during the training stage, consequently leading to good classification accuracy during the testing phase.

TABLE I. Comparative SD and ARR of PSOTCCC and FPSOTCCC on MNIST datasets

Technique		PSOTCCC	FPSOTCCC
Train	ARR	78.98	83.28
	SD	2.94	2.03
Test	ARR	70.73	75.59
	SD	8.49	6.20

TABLE II. Comparative SD and ARR of PSOTCCC and FPSOTCCC on the individual's handwritten digits datasets

Technique		PSOTCCC	FPSOTCCC
Train	ARR	87.74	92.17
	SD	0.58	0.22
Test	ARR	80.37	85.26
	SD	8.96	7.18

VI. Conclusion

PSOTCCC effectively tackles HDR, yielding promising outcomes. The experimental results underscore PSOTCCC's superiority over other algorithms. While PSOTCCC can create a good classification model, these models can improve using feature selection techniques. This study introduces a feature selection technique developed from binary particle swarm optimization applied alongside the classification model to enhance its performance. The enhancement results in a feature selection model, and when both models are used for data classification, it significantly improves classification performance, leading to higher Recognition Rates. The experimental outcomes clearly demonstrate that FPSOTCCC outperforms PSOTCCC in terms of reliability and classification performance across diverse datasets, including standard datasets and those representing practical datasets. In the future, there are plans to advance FPSOTCCC further to achieve even higher Recognition Rates in both training and testing stages. Future efforts will involve integrating novel Feature Extraction techniques into FPSOTCCC. ~~Additionally, addressing the vulnerability of PSO to easily trap in local optima is a priority by novel strategies like mutation operations, resets, or initializations will be explored to mitigate or resolve these issues, aiming to further elevate recognition rates.~~

Acknowledgment

This research was funded by Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Thailand contract no. 671155.

References

[1] Muhammad Arif Mohamad, Haswadi Hassan, Dewi Nasien and Habibollah Haron, "A Review on Feature Extraction and Feature Selection for Handwritten Character Recognition," International Journal of Advanced Computer Science and Applications(ijacsa), 6(2), 2015.

[2] Harralson et. al. Handwriting Examination: Theory, Proficiency, and Methods. A Survey of Forensic Handwriting Examination Research in Response to the Nas Report. Retrieved 2013.

[3] M. H. Alkawaz, C. C. Seong and H. Razalli, "Handwriting Detection and Recognition Improvements Based on Hidden Markov Model and Deep Learning," 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), 2020, pp. 106-110.

[4] Y. LeCun, C. Cortes e J. C. B. Christopher. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[5] Deng, L, "The mnist database of handwritten digit images for machine learning research," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141-142, Nov 2012.

[6] C. Ratanavilisagul and B. Kruatrachue, " A modified particle swarm optimization with mutation and reposition," in International Journal of Innovative Computing, Information and Control Volume 10, Number 6, December 2014, pp. 2127-2142.

[7] N. O. S. Ba-Karait and S. M. Shamsuddin, "Handwritten Digits Recognition Using Particle Swarm Optimization," 2008 Second Asia International Conference on Modelling & Simulation (AMS), 2008, pp. 615-619.

[8] J. Samleepant and C. Ratanavilisagul, "Modified Feature Extraction to apply PSO for solving Handwritten Digits," 2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE), Phitsanulok, Thailand, 2023, pp. 356-361.

[9] C. Ratanavilisagul, "A novel modified particle swarm optimization algorithm with mutation for data clustering problem," 2020 5th International Conference on Computational Intelligence and Applications (ICCIA), 2020, pp. 55-59.

[10] C. Ratanavilisagul, "Modified Ant Colony Optimization for Vehicle Routing Problem with Time Windows using Limited Search Space and Novel Updating Pheromone and Re-initialization Pheromone Techniques," ICIC Express Letters, Part B: Applications (ICIC-ELB), Vol. 1, pp. 571-580, 2022.

[11] C. Ratanavilisagul, "Modified Ant Colony Optimization with route elimination and pheromone reset for Multiple Pickup and Multiple Delivery Vehicle Routing Problem with Time Window", Journal of Advanced Computational Intelligent and Intelligent Informatics Vol. 26, No. 6, 2022, pp. 959 - 964.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, Vol. 4, pp. 1942-1948.

[13] How to OCR with Tesseract, OpenCV and Python, Nanonets [Online]. Available: https://nanonets.com/blog/ocr-withtesseract

[14] Google Lens, Wikipedia [Online]. Available: https://en.wikipedia.org /wiki/Google_Lens

[15] Handwriting Recognition: More than Just MNIST, [Online]. Available: https://medium.com/nerd-for-tech/handwriting-recognition -more-than-just-mnist-36e68832de73

[16] Sarra Rouabhi, Khadidja El-Kobra Belbachir, Redouane Tlemsani et al. Optimizing Handwritten Arabic Character Recognition: Feature Extraction, Concatenation, and PSO-Based Feature Selection., 07 September 2023, PREPRINT (Version 1) available at Research Square [https://doi.org/10.21203/rs.3.rs-3310505/v1]

[17] Guha, Ritam, Ghosh, Manosij, Singh, Pawan Kumar, Sarkar, Ram and Nasipuri, Mita. "M-HMOGA: A New Multi-Objective Feature Selection Algorithm for Handwritten Numeral Classification" Journal of Intelligent Systems, vol. 29, no. 1, 2020, pp. 1453-1467. https://doi.org/10.1515/jisys-2019-0064

[18] L. M. Seijas, R. F. Carneiro, C. J. Santana, L. S. L. Soares, S. G. T. A. Bezerra and C. J. A. Bastos-Filho, "Metaheuristics for feature selection in handwritten digit recognition," 2015 Latin America Congress on Computational Intelligence (LA-CCI), Curitiba, Brazil, 2015, pp. 1-6, doi: 10.1109/LA-CCI.2015.7435975.

[19] Victor L. F. Souza, Adriano L. I. Oliveira, Rafael M. O. Cruz, Robert Sabourin, "Improving BPSO-based feature selection applied to offline WI handwritten signature verification through overfitting control," Computer Vision and Pattern Recognition, May 2020, pp. 69-70

[20] Guha, R., Ghosh, M., Singh, P.K. et al. A Hybrid Swarm and Gravitation-based feature selection algorithm for handwritten Indic script classification problem. Complex Intell. Syst. 7, 823–839 (2021). https://doi.org/10.1007/s40747-020-00237-1

[21] J. Kennedy, R. Eberhart, A discrete binary version of the particle swarm algorithm, Proceedings of the world Multiconference on Systemics, Cybernetics, Informatics, pp.4104-4109, 1997.

[22] U. R. Babu, Y. Venkateswarlu and A. K. Chintha, "Handwritten Digit Recognition Using K-Nearest Neighbour Classifier," 2014 World Congress on Computing and Communication Technologies, 2014, pp. 60-65.

[23] T. Makkar, Y. Kumar, A. K. Dubey, Á. Rocha and A. Goyal, "Analogizing time complexity of KNN and CNN in recognizing handwritten digits," 2017 Fourth International Conference on Image Information Processing (ICIIP), 2017, pp. 1-6.

[24] D. Nasien, H. Haron and S. S. Yuhaniz, "The Heuristic extraction algorithms for freeman chain code of handwritten character," 2011 International Journal of Experimental Algorithms (IJEA), 1 (1). pp. 1-20. ISSN 2180-1.

[25] MNIST Dataset, [Online]. Available: https://git-disl.github.io/ GTDLBench/datasets/mnist_datasets