applied sciences

MDPI

Article

An Improved Particle Swarm Optimization Algorithm for Data Classification

Waqas Haider Bangyal [1], Kashif Nisar [2], Tariq Rahim Soomro [3], Ag Asri Ag Ibrahim [4,*], Ghulam Ali Mallah [5], Nafees Ul Hassan [6] and Najeeb Ur Rehman [6]

1 Department of Computer Science, Kohsar University, Murree 47150, Pakistan
2 Victorian Institute of Technology, Adelaide 5000, Australia
3 College of Computer Science and Information Systems, Institute of Business Management, Korangi Creek Road, Karachi 75190, Pakistan
4 Faculty of Computing and Informatics, University Malaysia Sabah, Jalan UMS, Kota Kinabalu 88400, Malaysia
5 Department of Computer Science, Shah Abdul Latif University, Khairpur 66111, Pakistan
6 Department of Computer Science, University of Gujrat, Gujrat 50700, Pakistan
* Correspondence: awgasri@ums.edu.my

Abstract: Optimisation-based methods are enormously used in the field of data classification. Particle Swarm Optimization (PSO) is a metaheuristic algorithm based on swarm intelligence, widely used to solve global optimisation problems throughout the real world. The main problem PSO faces is premature convergence due to lack of diversity, and it is usually stuck in local minima when dealing with complex real-world problems. In meta-heuristic algorithms, population initialisation is an important factor affecting population diversity and convergence speed. In this study, we propose an improved PSO algorithm variant that enhances convergence speed and population diversity by applying pseudo-random sequences and opposite rank inertia weights instead of using random distributions for initialisation. This paper also presents a novel initialisation population method using a quasi-random sequence (Faure) to create the initialisation of the swarm, and through the opposition-based method, an opposite swarm is generated. We proposed an opposition rank-based inertia weight approach to adjust the inertia weights of particles to increase the performance of the standard PSO. The proposed algorithm (ORIW-PSO-F) has been tested to optimise the weight of the feed-forward neural network for fifteen data sets taken from UCI. The proposed techniques' experiment result depicts much better performance than other existing techniques.

Keywords: feed-forward neural network; quasi-random sequence; opposition rank-based inertia weight; particle swarm optimisation

check for updates

Citation: Bangyal, W.H.; Nisar, K.; Soomro, T.R.; Ag Ibrahim, A.A.; Mallah, G.A.; Hassan, N.U.; Rehman, N.U. An Improved Particle Swarm Optimization Algorithm for Data Classification. *Appl. Sci.* **2023**, *13*, 283. https://doi.org/10.3390/app13010283

Academic Editor: Giancarlo Mauri

Received: 12 September 2022
Revised: 19 November 2022
Accepted: 21 November 2022
Published: 26 December 2022

1. Introduction

Data classification is widely used in machine learning to solve problems such as spam email filtering, social network analysis, biological data analysis, diagnosing medical diseases, image and speech recognition [1]. The classification process contains two steps: the first step is training, which builds a model from the training samples, and the second step is the model predicting the labels of unlabeled test samples. In the machine learning field, an artificial neural network (ANN) is a classification problem-solving technique [2]. ANNs trained with Back-Propagation (BP) based on gradient descent technique generally slow convergence, are stuck in local optima, and require more training time [3]. The use of evolutionary algorithms in ANN teaching overcomes these shortcomings.

Swarm Intelligence is a field that consists of many individuals and deals with natural and artificial systems [4]. These systems are coordinated through self-organisation and decentralised control. In particular, this area focuses on collective behaviour that is displayed as a result of regional interactions between people and the environment [5]. Examples of checking systems through swarm intelligence are ants and termite swarms, fish swarms, bird swarms, and terrestrial animal swarms [6]. The swarm intelligence population consists of simple agents that are closer to the optimal outcome and interact with each other and environments. Emerging intelligence that communicates with its ecosystem is based on simple independent agents, is composed of other agents but behaves relatively individually from all other agents. Independent agents do not follow global plans or the instruction of other participants. Over the past two decades, swarm intelligence and nature-inspired computing have generated a lot of interest in almost all fields of science, industry, and engineering. Some human artifacts also belong to the field of swarm intelligence; in particular, some multi-robot systems. They are written to solve data analysis [7] and optimisation problems. Particle Swarm Optimization (PSO) [8], Cat Swarm Optimization (CSO) [9], Artificial Bee Colony (ABC) [10], Cuckoo Search (CS) [11] and Bat algorithm (BA) [12] belong to SI.

Particle Swarm Optimization (PSO) is a well-known example of swarm intelligence, introduced by Kennedy and Eberhart in 1995 [13] to solve global optimisation problems. Because of its simplicity and efficiency, it has been described in various engineering fields and has become the most effective method for solving optimisation problems. In PSO, different numbers of particles are known as a swarm, which search for the best possible solutions in search space [13–15]. In PSO, individuals are known as particles searching from one position to another; if any particle finds food without wasting time getting the food and sharing information of its position with another particle, all particles have to follow to reach that position [16]. Each particle follows the basic rule for determining its previous best position or neighbour. Each particle finds the optimal solution in search space. To enhance learning factors, minimise weights, and ANN architecture, the PSO algorithm has been used [17]. Consequently, it is usually used in engineering fields with practical applications. For example, PSO is commonly used for price forecasting, clustering, planning optimisation parameter optimisation, image processing and the medical field [18].

The PSO algorithm suffers from premature convergence and diversity problems. If PSO parameters are not properly set, then there are chances that it can get trapped in local optimum due to lack of local exploitation, global exploration and diversity issues in the search space [19]. To solve the combinatorial optimization problems, multiple modified PSO variants are proposed in [20,21], such as multi-objective optimization [22], constraint optimization [23], opposition-based variant [24], adaptive inertia weight [25] and mutation operator [26]. In addition, many modifications have been made to the PSO algorithm to improve its convergence. Inertia weight is introduced and gives more control over the particle capabilities for exploration (searching the entire space to gain a solution) and exploitation (searching the neighbourhood of the fittest solution).

Tizhoosh [27] introduced the concept of Opposition-Based Learning (OBL), which has so far been used to accelerate backpropagation learning on neural networks [28] and reinforcement learning. The purpose of OBL is to check the estimate (random guess) and its corresponding opposite estimate (opposite guess) at the same time to obtain a more approximate value for the current candidate solution [29]. It is used to generate opposite populations that inspire the real world's opposite. The probability of finding the best solution is increased through the opposite solution. It improves algorithms' performance and finds the best possible solution in search space [30]. OBL is used in many areas, such as agriculture for preserving water crops, soil purification, medical fields for disease diagnosis, scheduling of agricultural work [31]. It is applied in many well-known algorithms such as the Bat algorithm [32,33], PSO [34], Grey Wolf Optimizer (GWO) [35], Harmony Search (HS) [36], Differential Evolution (DE) [37] and Artificial Neural Networks (ANNs).

The distribution of random numbers is divided into three major categories: probability sequences, i.e., Exponential, Beta, Gamma, Lognormal, Quasi-random sequence, i.e., Halton, Van der corput, Hamersley, Sobol, Faure [38], and the pseudo-random sequences Multiply with-carry, Linear congruential generator, Mersenne twister, Philox, and Threefry [39]. A probability sequence is a sequence of continuous probabilities according to a particular distribution's scale factor and shape [40]. Pseudo-random sequences and quasi-random sequences outperform for a globally optimal solution due to cover all search spaces. We used QRS (Sobol sequence (S), Halton sequence (H), Faure sequence (F), Gaussian (G), and Lognormal (LN)) for the initialisation of the population.

This behavior is more persevering and intolerable for multimodal problems, as it contains many local and global optimums. The most considerable factor for this deprived performance can be the insufficient distribution of the population in the search area, i.e., to conclude that if the initial population does not search the complete search space adeptly, it is difficult to locate the robust solution points, and thus the results omit the global best solution. This problem can be overcome by adding the most structured and organized random distribution for population initialization. Random number sequences vary with respect to the nature of their morphological design, i.e., quasi-random sequences, pseudo-random sequences, and probability distribution. Due to this fact, a novel initialisation population method used a Quasi-random sequence (Faure) to create the initialisation of the swarm, and through the opposition-based method, an opposite swarm was generated. We proposed an Opposition rank-based inertia weight approach to adjust inertia weights of particles to increase the performance of the standard PSO. The proposed algorithm (ORIW-PSO-F) has been tested to optimise the weight of the feed-forward neural network for fifteen data sets taken from UCI.

This paper proposes new variants of the PSO algorithm, Faure, with opposition-based PSO-ranked inertia weight (ORIW-PSO-F). The proposed algorithms improve global search ability and solve real-world classification problems. The proposed algorithms have two main effective improvements in initialisation strategies (Faure) with Opposition based learning and rank-based inertia weight. We present a new Quasi-random initialisation strategy (Faure) to initialise the search space particles, and opposite particles generated using Opposition-based learning. Particle inertia weight was updated with opposition rank-based inertia weight-balancing exploration and exploitation. We trained artificial neural networks using pseudo-random sequences on real-world classification problems. From the well-known repository UCI, fifteen data sets were taken in order to compare the performance of classifiers.

The simulation results show that the proposed variant ORIW-PSO-F provided better results as compared to ORIW-PSO, ORIW-PSO-S, ORIW-PSO-H, ORIW-PSO-LN and ORIW-PSO-G. The proposed study is useful in a wide range of computer domains, including neural network training, classification problems, data mining, image processing, min-max problems, game, single processing, multi-objective optimisation, and complex real-world optimisation. It is also applicable to solving most numerical optimisation problems and problems converted to optimisation problems.

The contributions of this work are summarized as follows:

- To propose a novel initialisation population method using a Quasi-random sequence (Faure) to create the initialisation of the swarm, and through the opposition-based method, an opposite swarm is generated and the proposed Opposition rank-based inertia weight approach adjusts the inertia weights of particles;
- To find the best accuracy and compare its result with the previous state-of-the art approaches.

The rest of the article is structured as follows: Section 2 discusses Materials and Methods, including an initialisation strategy, opposition rank inertia weight and the training, and basic PSO working. Section 3 provides the results and discussion. Section 4 describes conclusions and possible work, followed by the references section.

2. Materials and Methods

2.1. Related Work

In [39], the authors proposed a multi-mean PSO algorithm known as the MMPSO algorithm for training multi-layer feed-forward neural networks (MLFNN). MMPSO finds better solutions than PSO and MMPSO has multiple swarms to find the best solution that is better than PSO.

The authors [40] introduced a new, modified PSO algorithm with two main modifications. The first, known as a self-adaptive parameter, and the second, a strategy-based method known as the SPS-PSO algorithm, optimise feed-forward neural networks by feature selection. The SPS-PSO algorithm was applied to deal with the large-scale FFNN optimisation problem and reduce the computational complexity. The authors present a new variant of PSO (NMPSO) to solve nonlinear pattern classification problems [41]. The proposed methodology structure is an ANN that offers optimal precision for a particular problem. In addition, this study introduces a new method for selecting the maximum number of neurons (MNN). The architecture simultaneously develops transfer function types and synaptic weights. The proposed method was tested to accuracy by solving synthetic pattern recognition problems. Furthermore, the artificial neural network designed using the proposed method was compared with ANN designed manually using backpropagation and learning algorithms of Levenberg–Marquardt.

Experiments were performed using 10 datasets from UCI to check the performance of the MMPSO algorithm [42]. As a result of the experiment, it was shown that the proposed algorithm was executed more efficiently than other algorithms. An improved variant of the PSO algorithm called LPSONS was presented in [43] to increase the optimisation speed of the standard PSO algorithm to train ANNs. The proposed algorithm implemented the PSO velocity operator with the Mantegna–Lévy distribution to improve the diversity of the population and increase accuracy. Bottom of Form The proposed LPSONS algorithm was used to optimise the feed-forward multi-layer perceptron (MLP) ANN training.

In [44], the authors presented the partial opposition-based learning with PSO, or POPSO, algorithm to increase the performance of basic PSO. Partial opposition-based learning (POBL) generates the opposite swarm of the original swarm. POPSO is used to train the MLFNN for mining medical data classification problems. Compared to all other algorithms, POPSO provides a better compromise between sensitivity and specificity when classifying medical datasets. An ANN trained with the PSO algorithm has been used to distinguish dengue hemorrhagic fever (DHF) and dengue fever (DF) patients from patients recovering or not who have Parkinson's disease [45]. Finally, NNPSO was tested with a multi-layer neural network feed-forward network (MLPFFN) classifier to classify dengue fever patients from recovered or non-recovered patients.

In [17], the authors presented the WELL sequence, also known as the Well Equidistributed Long-period Linear with Particle swarm optimisation (WELL-PSO) to overcome the limitation of the basic PSO algorithm. A novel quasi-random sequence initialisation scheme, the WELL sequence is used to generate the initial population. The proposed method also trains NN and offers better results than existing training algorithms (with basic PSO methods and improved variants). The experimental results show that WELL-PSO performs better on real-world classification problems than improved variant and standard method PSO. PSO algorithms were combined with the forward feedback of the neural network in the Cleveland Clinic database to reduce the 13 effective attributes to 8 factors and optimise the accuracy and cost [46]. The researchers used the four different research classification methods, and the results show that the feature selection in the neural network FFBP and PSO algorithm is more effective. A modified PSO algorithm combined with a new training algorithm was proposed by [47] for the time series problem. The training algorithm has no exploded or vanished gradient problem because it does not require gradients. They compared the accuracy of the proposed learning algorithm using a deep recurrent neural network with LSTM and PSGM ANNs on ten-time series. The forecasting performance of the proposed algorithm was superior to the other methods.

Khan et al. [48] proposed Advance Particle Swarm Optimization (APSO) using NN to reduce training time and improve classification accuracy. In advance, Particle Swarm Optimization (APSO) inertia weight is updated with constriction factor to avoid the local optimal problem. The accuracy of the proposed algorithm was also checked with the different numbers of a neuron [49]. The proposed algorithms perform faster convergence than backpropagation neural networks. For short term price forecasting (STPF) and classification purposes, the authors presented Fuzzy adaptive particle swarm optimisation (FA-PSO) with Feed Forward Neural Networks (FFNN) in [50]. To prevent local optima, the proposed algorithm uses dynamic inertia weight. Weights and biases were constructed using FAPSO for FFNNs with fixed architecture. To predict the price of power, the proposed method is used.

In [51], the authors present centripetal accelerated particle swarm optimisation (CAPSO) combined with advanced PSO and Newton motion laws to evolve accuracy and ANN learning. The authors trained a feed-forward multi-layer neural network (FFNN) with the CAPSO algorithm to solve classification problems for diagnosing nine medical diseases. The CAPSO algorithm exhibits superior classification accuracy compared to most of the well-known algorithms used to diagnose nine medical disorders. Wahab et al. [52] presented a method to train an ANN using the PSO algorithm to identify damage to structures. The proposed algorithm was used to overcome ANN limitations by reducing the computation time by applying the gradient descent method used in neural network training. Numerical and experimental models using various damage conditions were used to evaluate the effectiveness of the proposed algorithm. The proposed algorithm ANN-PSO easily found damaged locations. Furthermore, PSO and its various improved variants have been successfully applied in energy saving domains for multiple reasons, including the appropriate size adjustment for energy systems with the use of PSO.

2.2. Research Methodology

This paper proposes new variants of Faure with opposition-based PSO ranked inertia weight (ORIW-PSO-F) to balance exploration and exploitation and prevent stuck local optima. We proposed two modifications to PSO algorithms: initialisation strategy is Faure initialisation techniques with OBL, and opposition rank inertia weight is adjusting inertia weight of the particle. The initialisation strategy uses the QRS approach and OBL to initialise the initial population in PSO algorithms.

In this paper, there are three main contributions: initialisation strategies, Opposition-based learning, and Opposition Rank-based inertia weight. OBL is used to generate the opposite swarm of the current swarm and enhance the performance of algorithms. In OBL, we used jumping probability 0.3 for opposite population generation. It must be pointed out here that all variables were dynamically generated while accumulating the opposite population due to generational jumping. In the current set, each variable used maximum and minimum values to calculate the opposite points instead of the defined boundary spacing. Opposition Rank-based inertia weight adjusted for each particle according to its fitness rank. The lowest fittest particle had maximum inertia weight that moved fast compared to the fittest particle.

2.3. Random Number Generator

A set of numbers that occur in an order in which values cannot be predicted based on the past and present and the value cannot be predicted based on a specific uniformly distributed set of numbers is called a random number. Random numbers are generated in a uniform distribution by the built-in library function $Rand(x_{min}, x_{max})$. The continuous uniform probability density function determines the effect of uniformity for all sequences [53]. It generates a sequence based on the probability density function. The probability density function is defined as:

$$f(t) = \begin{cases} \frac{1}{p-q} & for\ p < t < q \\ 0 & for\ t < p\ or\ t > q \end{cases} \tag{1}$$

where p and q are the maximum likelihood parameters. The $f(t)$ value is useless at the boundary between p and q, as it has no effect on the integral of $f(t)$ dt in any range. The score likelihood function calculates the maximum likelihood parameter estimate. It is given as follows:

$$l(p,q|t) = n\ log(q-p) \tag{2}$$

Flowchart chart show below in Figure 1.

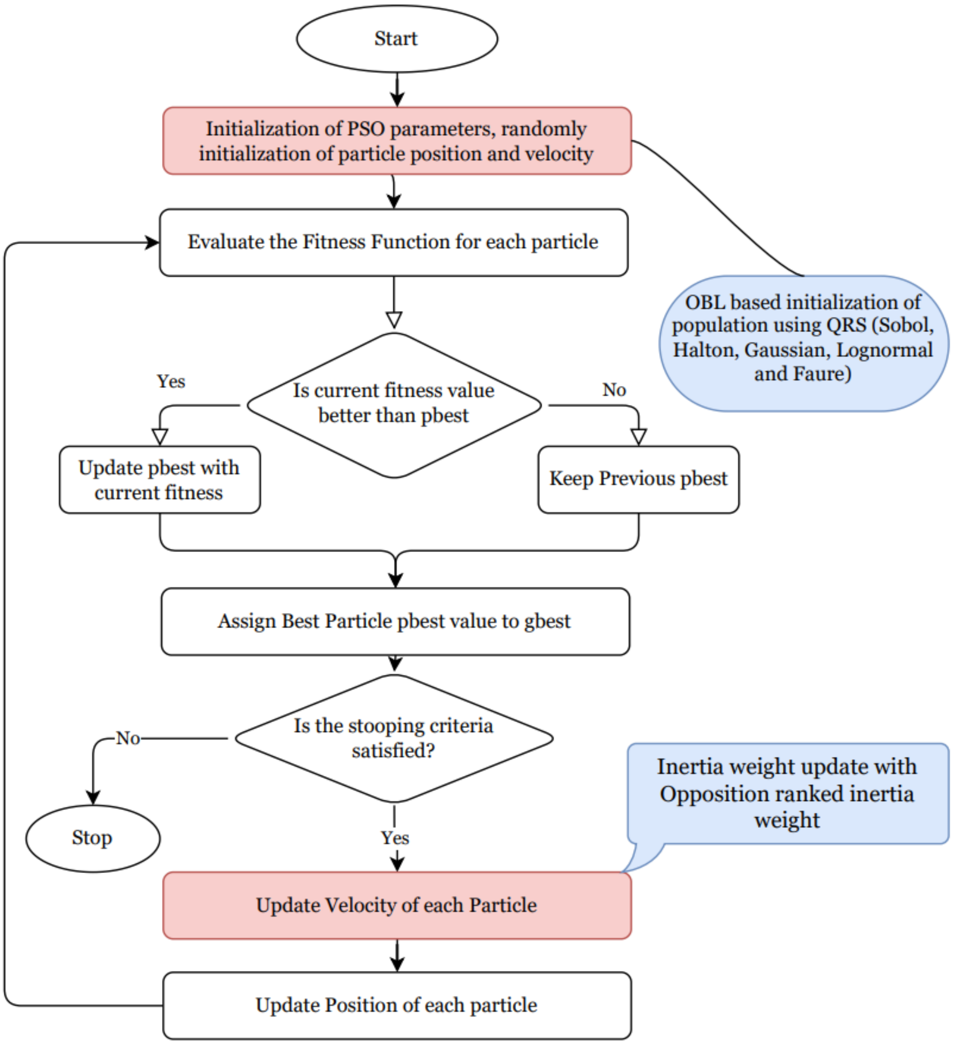


Figure 1. Proposed methodology.

2.4. Quasi-Random Sequence

Low-discrepancy techniques are also deterministic point generators that mean the point set with a high level of uniformity. The quasi-random sequence (QRS) approach was used to generate a low discrepancy set, and it was neither random nor pseudo-random. Quasi-random sequence generators reduce the discrepancy (non-uniformity) from the distribution with an equal component of points in each sub-cube of a uniform partition of the hypercube and fill the “holes” in any initial segment of the generated QRS. This technique avoids clustering and can accelerate convergence; however, quasi-random numbers are normally too uniform to pass randomness tests. QRS was used as an initial value for the global optimisation problem. QRS explores more space than a random sequence used in applied mathematics. QRS is used to initialise the population in optimisation algorithms. Famous quasi-random sequences such as Sobol, Halton, Faure, Gaussian, and Lognormal are used for the initialisation of population.

2.4.1. Sobol

An example of quasi-random sequences is the Sobol sequence. A set of direction numbers has to be required to generate a Sobol sequence. Sobol provides the liberty while selecting the initial direction numbers. By using Sobol sequences, we can generate results for selected dimensions. Base 2 generates a finer uniform partition of unit intervals for these sequences. Sobol sequence is generated through the following equation and Figure 2:

$$c = c_1 2^0 + c_2 2^1 + c_3 2^2 + \ldots + c_n 2^{n-1} \quad (3)$$

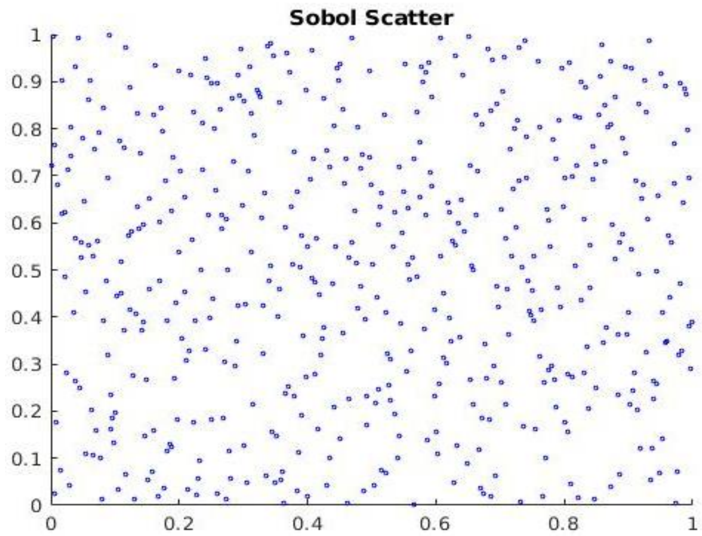


Figure 2. Sample points generated using Sobol distribution.

2.4.2. Halton

Each dimension of the Halton sequence uses coprime bases, also known as van der Corupt sequences. Halton sequences are of low discrepancy and deterministic, used in the Monte Carlo simulation. It is an improved variant of the van der Corput sequence as shown below in Figure 3.

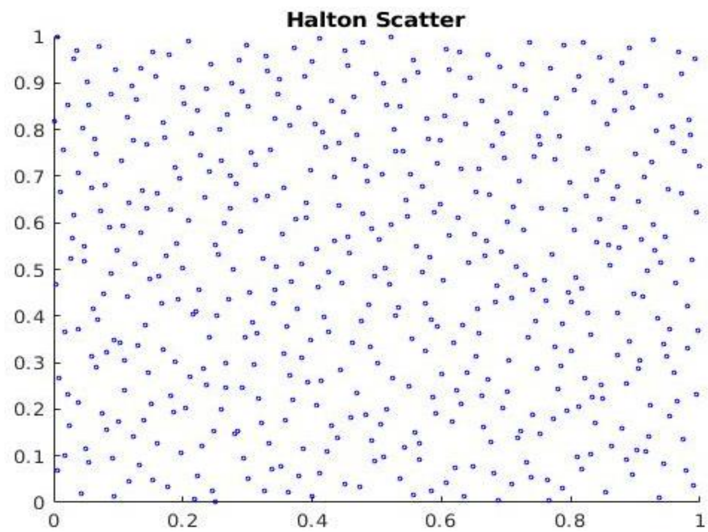


Figure 3. Sample points generated using Halton distribution.

2.4.3. Gaussian

Another name of the normal distribution is the Gaussian distribution; mostly, data occur near the mean, and hence it has a bell curve. For the representation of real-valued random variables we used continuous-distribution-type Gaussian distribution, as shown in Figure 4.

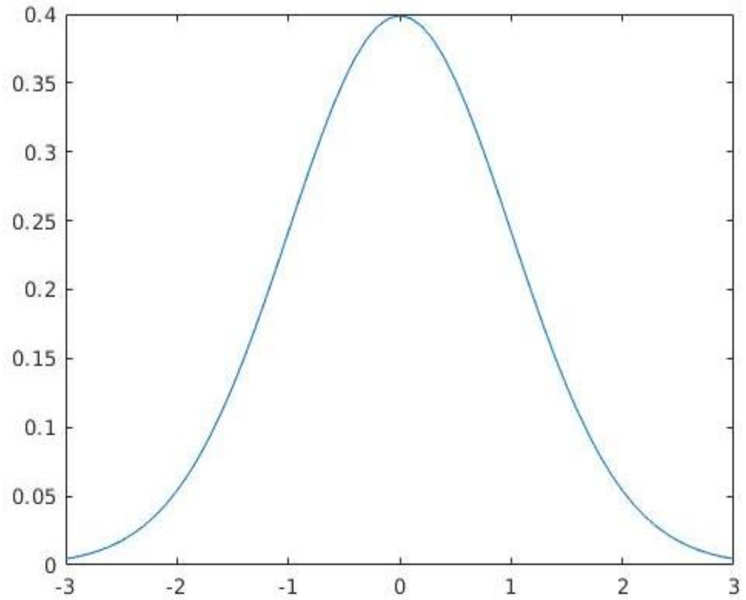


Figure 4. Density plot for Gaussian distribution.

2.4.4. Lognormal

The logarithm of the log normal distribution follows a normal distribution and type of probability distribution. The log normal distribution is skewed to the right and applicable when the growth rate is positive because the logarithm of value exists only when positive. As shown in Figure 5.

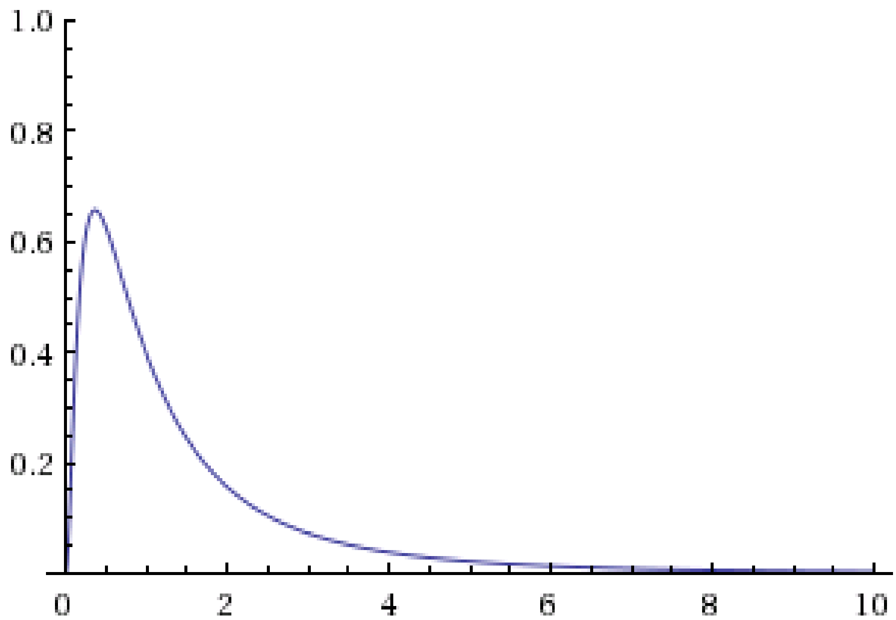


Figure 5. Density plot for lognormal distribution.

2.4.5. Faure

The Faure sequence is an approach to generating a Low Discrimination Sequence; it enhances the most basic idea of the van der Corput sequence for higher dimension. The basic method is to generate the sequence of the van der Corput methodology. Distribution of 100 particles in the search space [0, 1] is shown in Figure 6 for Quasi sequence uniform distribution while pseudo sequence is shown in Figure 7.

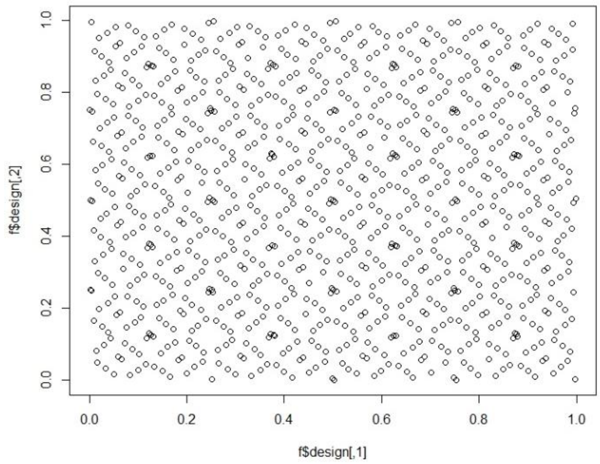


Figure 6. Quasi-random distribution.

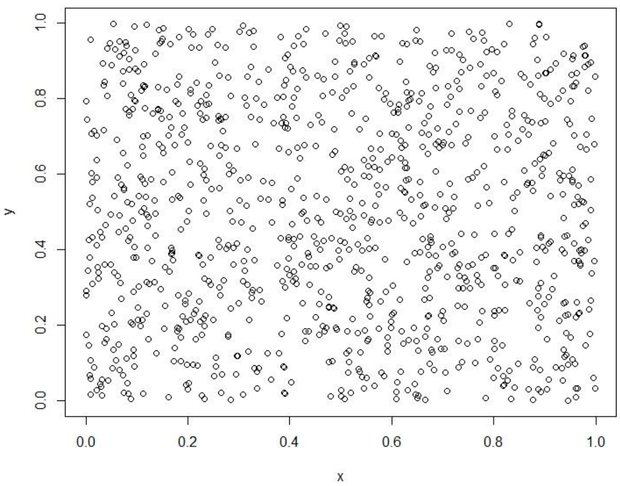


Figure 7. Pseudo-random distribution.

2.5. Opposition Based Learning

The optimisation process will end when the optimal solution is near the random guess and fast convergence. The optimisation process will take a lot of time when the optimal solution is far from random guesses, and convergence will be slower. Random guesswork and its counter-guessing can be generated simultaneously to increase the availability of the optimal solution [54], and whereas a metaheuristic algorithm enhances a problem with the best solution, the initial solution is usually randomly generated [55]. However, using OBL achieves a better candidate start date and has a better chance of finding a better area, even without prior knowledge. The main idea behind OBL is to make the current estimate x' that is randomly generated for each solution x of a given problem. To find the opposite value x' of current value x, we calculate it as follows:

$$x' = a + b - x \tag{4}$$

where $[a,b]$ are the intervals of real number x.

Let us assume $x_i = (x_1, x_2, x_3, \ldots, x_d)$ are points in search space d dimension with interval x_i belonging to $\{a_i, b_i\}$, so the opposite point is as follows:

$$x'_i = a_i + b_i - x_i \tag{5}$$

Let us assume, given $g(x)$ is the evolution function and the unknown function is $f(x)$; if $f(x)$ is better than $f(x')$, then learning continues with x; otherwise, we continue with x'.

The OBL method used in the proposed algorithm differs from existing OBL-based algorithms. The existing OBL method first randomly initialises the population and then calculates the opposite population. The first population is initialised through Quasi-random sequence (Faure) in proposed algorithms and calculates its opposite.

2.6. Opposition Rank Base Inertia Weight

All the inertia weight has a significant influence on the performance of the standard PSO algorithm [56]. Inertia weight is the effective parameter that maintains the velocity of particles. Inertia weight is vital for balancing local search, known as exploitation (for lower values), and global search, also known as exploration (for higher values). Researchers have proposed many variants of it. This paper adopted a rank-based strategy to solve the problems. Particle inertia weight was updated with the rank-based inertia weight strategy. Opposition Rank-based strategy adjusted inertia weight according to particle fitness value and assigned a fitness rank to each particle. The most suitable particles near the best position will move slowly, while the fast-moving particles far away from the best position will continue to explore. The suitable particle is selected from either current or their opposite. It has improved local and global search at the same time. To find the rank base inertia weight of each particle, we calculated it as follows:

$$W_{(i)}(t) = W_{max} + (((W_{min} - W_{max}))/n) * R_{(i)}(t) \tag{6}$$

where n is the size of the population and $R_{(i)}(t)$ is the fitness rank of i particle. The slowly moving particles are the fittest particles that have the lowest inertia weight and fitness rank 1. The fast-moving particle has the highest fitness value rank, and the highest inertia weight is assigned. Maximum inertia weight (W_{max}) was set at 0.9, and 0.4 was the minimum inertia weight. Below Figure 8 show the feed forward neural network.

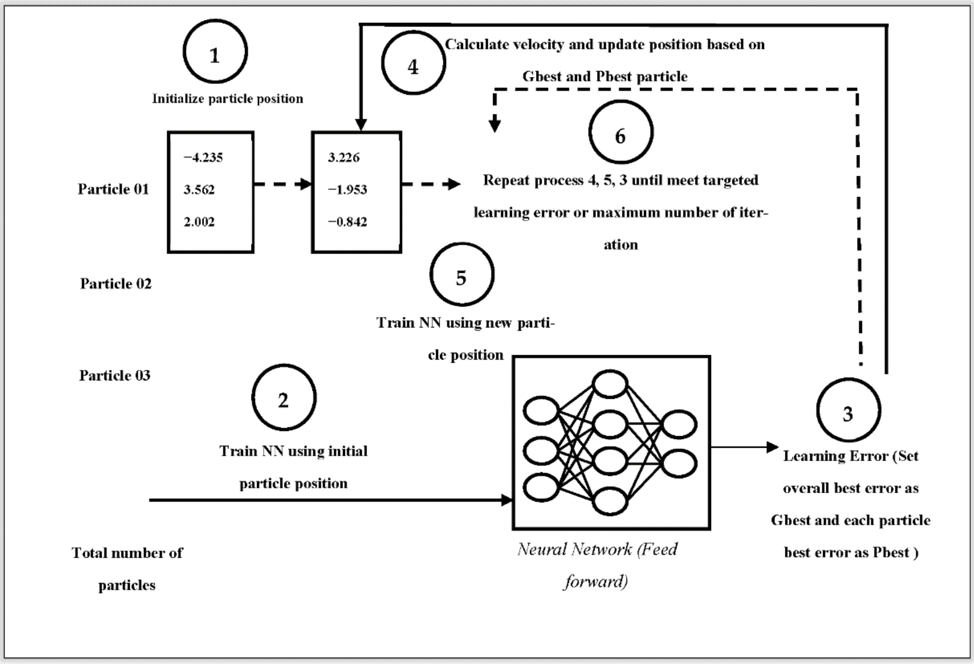


Figure 8. Feed-forward neural network.

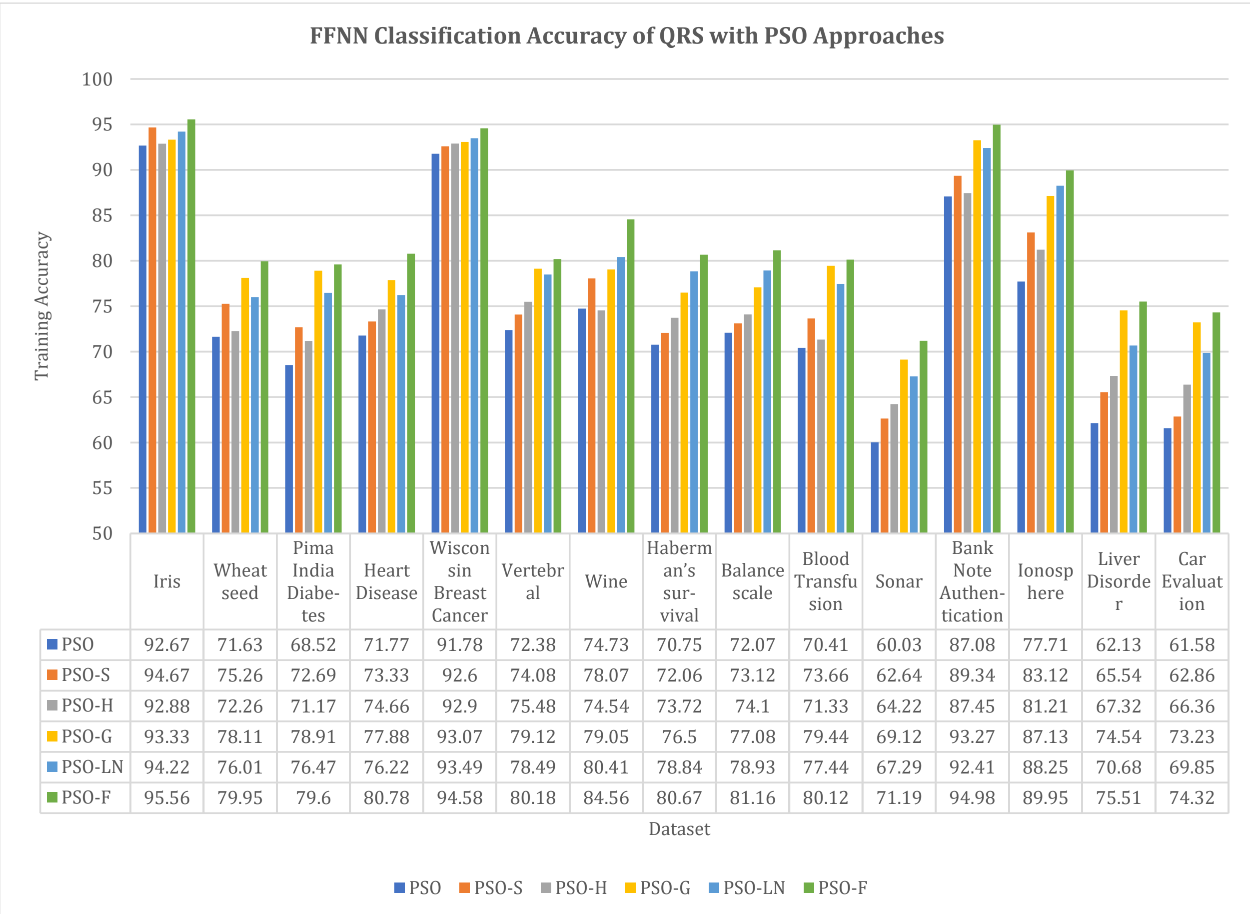
3. Results and Discussion

The proposed approach of ORIW-PSO-F was implemented in Matlab 2016 and a computer with 2.00 GHz along with the 8 GB RAM, Core™ i3-5005U CPU processor specification. We conducted experiments using fifteen datasets that have been taken from the UCI repository.

We separated the datasets into two parts: training and testing. The size of the training portion is 70% of the dataset and the testing portion is 30% of the dataset. On the range [50, 50], the initialization of the training weights is random. The dataset's features are shown in Table 1. These characteristics include the total number of instances that participated in each dataset, the total number of features, and the number of classes in each data set (such as a binary class problem or multiclass problem).

The proposition of this study continues to observe whereby the unique characteristics of experimental results rely on dimensions of these standard data sets. In the experiments, three simulation experiments were performed, where the following features of ORIW-PSO-F were observed: the effect of using different Initializing PSO approaches, the effect of using different Dimensions for problems, and a comparative analysis.

Feed-forward neural network weights are trained on a particle swarm optimisation algorithm (PSO), Sobol with PSO (PSO-S), Halton with PSO (PSO-H), Log normal with PSO (PSO-LN), Gaussian with PSO (PSO-G), and Faure with PSO (PSO-F). PSO-F performs well on these datasets and shows good results. To evaluate the performance, the PSO-F algorithm was compared with a variant of PSO such as standard PSO, PSO-S, PSO-H, PSO-G, and PSO-LN on fifteen real data sets extracted from UCI. Simulation results show that neural network training using the PSO-F algorithm performed well and provided better accuracy than other PSO approaches. After simulation, the result was excellent in training the NN using the PSO-F algorithm, and shows better precision and accuracy than traditional approaches. The accuracy results of classification problems are depicted in Figure 9, and an accuracy graph represents the same figure for fifteen data sets.



	Iris	Wheat seed	Pima India Diabe-tes	Heart Disease	Wisconsin Breast Cancer	Vertebral	Wine	Haberman's sur-vival	Balance scale	Blood Transfusion	Sonar	Bank Note Authen-tication	Ionosphere	Liver Disorder	Car Evaluation
PSO	92.67	71.63	68.52	71.77	91.78	72.38	74.73	70.75	72.07	70.41	60.03	87.08	77.71	62.13	61.58
PSO-S	94.67	75.26	72.69	73.33	92.6	74.08	78.07	72.06	73.12	73.66	62.64	89.34	83.12	65.54	62.86
PSO-H	92.88	72.26	71.17	74.66	92.9	75.48	74.54	73.72	74.1	71.33	64.22	87.45	81.21	67.32	66.36
PSO-G	93.33	78.11	78.91	77.88	93.07	79.12	79.05	76.5	77.08	79.44	69.12	93.27	87.13	74.54	73.23
PSO-LN	94.22	76.01	76.47	76.22	93.49	78.49	80.41	78.84	78.93	77.44	67.29	92.41	88.25	70.68	69.85
PSO-F	95.56	79.95	79.6	80.78	94.58	80.18	84.56	80.67	81.16	80.12	71.19	94.98	89.95	75.51	74.32

Figure 9. Testing accuracy result of QRS with PSO.

Table 1. Datasets detail.

Sr. No	Dataset	No of Attributes	Number of Labels	Number of Records
1	Iris	4	3	150
2	Wheat seed	7	3	210
3	Pima India Diabetes	8	2	768
4	Heart Disease	13	2	270
5	Wisconsin Breast Cancer	10	2	699
6	Vertebral	6	2	310
7	Wine	13	3	178
8	Haberman's survival	3	2	306
9	Balance scale	4	3	625
10	Blood Transfusion	4	2	748
11	Sonar	60	2	208
12	Bank Note Authentication	4	2	1372
13	Ionosphere	34	2	351
14	Liver Disorder	6	2	345
15	Car Evaluation	6	4	1728

A one-way ANOVA test with a significance level of 0.05 was implemented on testing the accuracy of six PSONN approaches. The result of the one-way ANOVA test is depicted in Table 2. In Table 2, the value of significance is 0.04902, which is less than 0.05, indicating a significant difference between all PSONN variants with a 95% confidence level. Therefore, PSONN variants were significantly different from each other. The one-way ANOVA test results are represented in Figure 10, which shows PSO-F has significantly better results than other PSO approaches. The graph of the ANOVA test shows the PSO-F is significantly distinct from all other PSO approaches.

Table 2. One-way ANOVA result test of QRS with PSO approaches.

Parameter	Relation	Sum of Squares	df	Mean Square	F	Significance
Testing Accuracy	Between groups	903.2158	5	180.6432	2.334319	0.049042

The weights of the feed-forward neural network were trained on opposition-based PSO (OPSO), Sobol with opposition-based PSO (OPSO-S), Halton with opposition-based PSO (OPSO-H), log normal with opposition-based PSO (OPSO-LN), Gaussian with opposition-based PSO (OPSO-G), and Faure with opposition-based PSO (OPSO-F). We prepared a feed-forward neural network using the weight optimisation process. The performance of OPSO, OPSO-S, OPSO-H, OPSO-G, OPSO-LN OPSO-F and state-of-the-art NN algorithms were tested on 15 well-known datasets. To evaluate the performance of the OPSO-F algorithm, it was compared with variants of PSO such as standard OPSO, OPSO-S, OPSO-H, OPSO-G, and OPSO-LN on fifteen real data sets extracted from UCI. The detail of the fifteen data sets are present in Table 1. OPSO-F is well-performing on these datasets and shows good result. Simulation results in Figure 11 show that neural network training using the OPSO-F algorithm performed well and provided better accuracy than other PSO approaches. Testing accuracy graph of fifteen datasets are represented in Figure 11.

A one-way ANOVA test with a significance level of 0.05 was implemented on the testing accuracy of six PSONN approaches. Table 3 depicts the results of the one-way ANOVA test. The significance value in Table 3 is 0.0494, which is smaller than 0.05, showing that there is a significant difference between all PSONN variants with a 95% confidence level. Therefore, PSONN variants are significantly different from each other. Figure 12 depicts the one-way ANOVA test results, which shows OPSO-F significantly has better results than other PSO approaches. The graph of ANOVA test show the OPSO-F is significantly distinct from all other PSO approaches.

The weights of the feed-forward neural network were trained using opposition-based PSO ranked inertia weight (ORIW-PSO), Sobol with opposition-based PSO ranked inertia weight (ORIW-PSO-S), Halton with opposition-based PSO ranked inertia weight (ORIW-PSO-H), log normal with opposition-based PSO ranked inertia weight (ORIW-PSO-LN), Gaussian with opposition-based PSO ranked inertia weight (ORIW-PSO-G), and Faure with opposition-based PSO ranked inertia weight (ORIW-PSO-F). We prepared a feed-forward neural network using the weight optimisation process. The performance of ORIW-PSO, ORIW-PSO-S, ORIW-PSO-H, ORIW-PSO-G, ORIW-PSO-LN, ORIW-PSO-F and state-of-the-art NN algorithms have been tested on 15 well-known datasets. To evaluate the performance of the proposed ORIW-PSO-F technique, it was compared with variants of PSO such as ORIW-PSO, ORIW-PSO-S, ORIW-PSO-H, ORIW-PSO-G, and ORIW-PSO-LN on fifteen real data sets extracted from UCI. The detail of these fifteen data sets is presented in Figure 13, where ORIW-PSO-F is well-performing on these datasets and shows good results. Simulation results in Figure 13 show that neural network training using the ORIW-PSO-F algorithm performed well and provided better accuracy than other PSO approaches. After simulation, the result is excellent in training the NN using the ORIW-PSO-F algorithm, and shows better precision and accuracy than traditional approaches. The ORIW-PSO-F algorithm can be effectively used for real-world complex statistical problems and data classification problems in the future. The accuracy results of classification problems are depicted in Figure 14, and the accuracy graph is represented in the same figure for fifteen data sets.

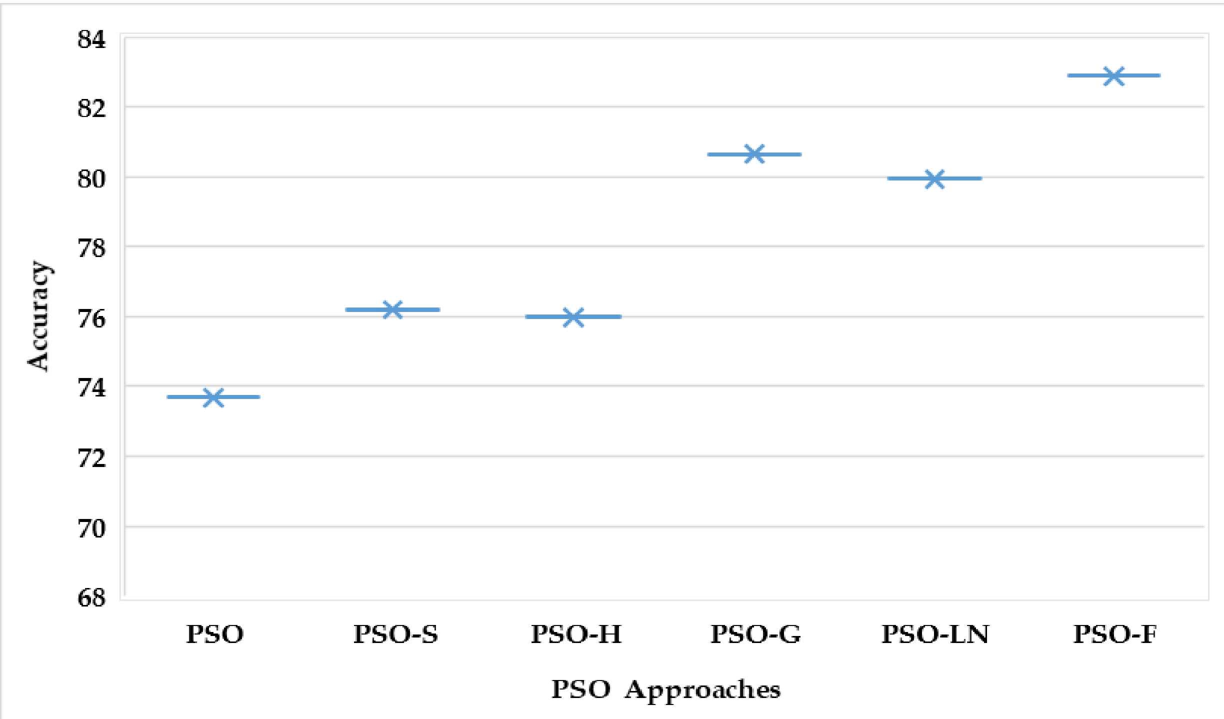


Figure 10. Result of one-way ANOVA test on testing accuracy (QRS with PSO approaches).

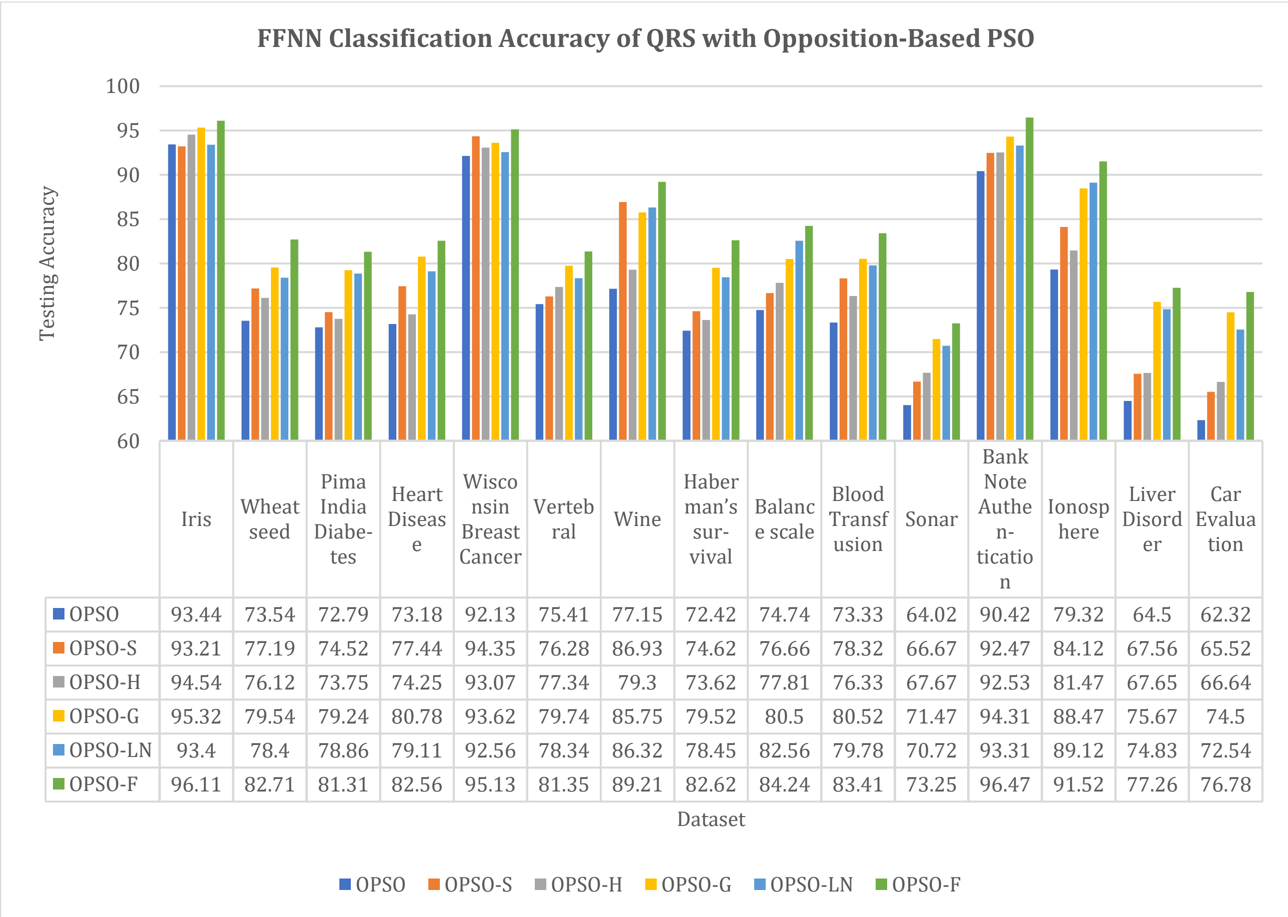


Figure 11. Testing accuracy result of QRS with opposition-based PSO approaches.

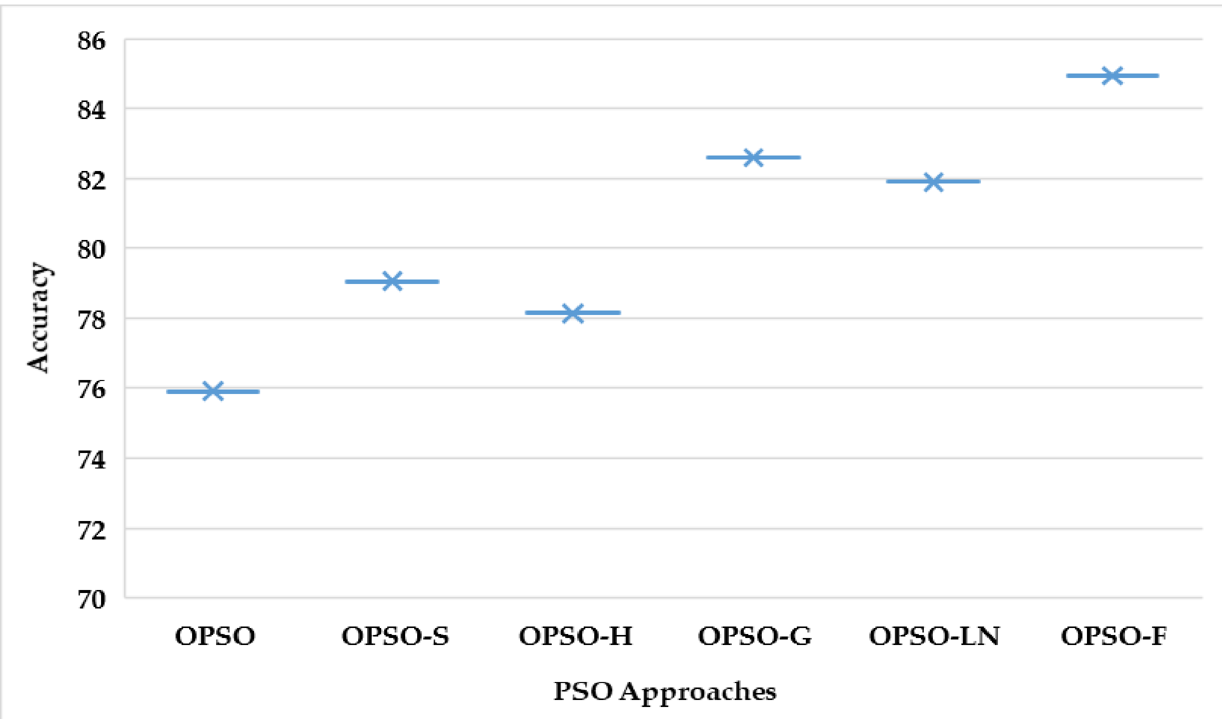


Figure 12. One-way ANOVA test on testing accuracy.

The one-way ANOVA test with a significance level of 0.05 was implemented on testing accuracy of six PSONN approaches. Table 4 shows the results of the one-way ANOVA test. The significance level in Table 4 is 0.04902, which is less than 0.05, indicating that there is a significant difference with a 95% confidence level in all PSONN variants. Therefore, PSONN variants are significantly different from each other. Figure 10 depicts the one-way ANOVA test results, which shows ORIW-PSO-F has significantly better results than other PSO approaches. The graph of the ANOVA test shows that the ORIW-PSO-F is significantly distinct from all other PSO approaches.

Table 3. One-way ANOVA result test of QRS with Opposition-Based PSO approaches.

Parameter	Relation	Sum of Squares	df	Mean Square	F	Significance
Testing Accuracy	Between groups	818.691	5	163.738	2.334012	0.0494

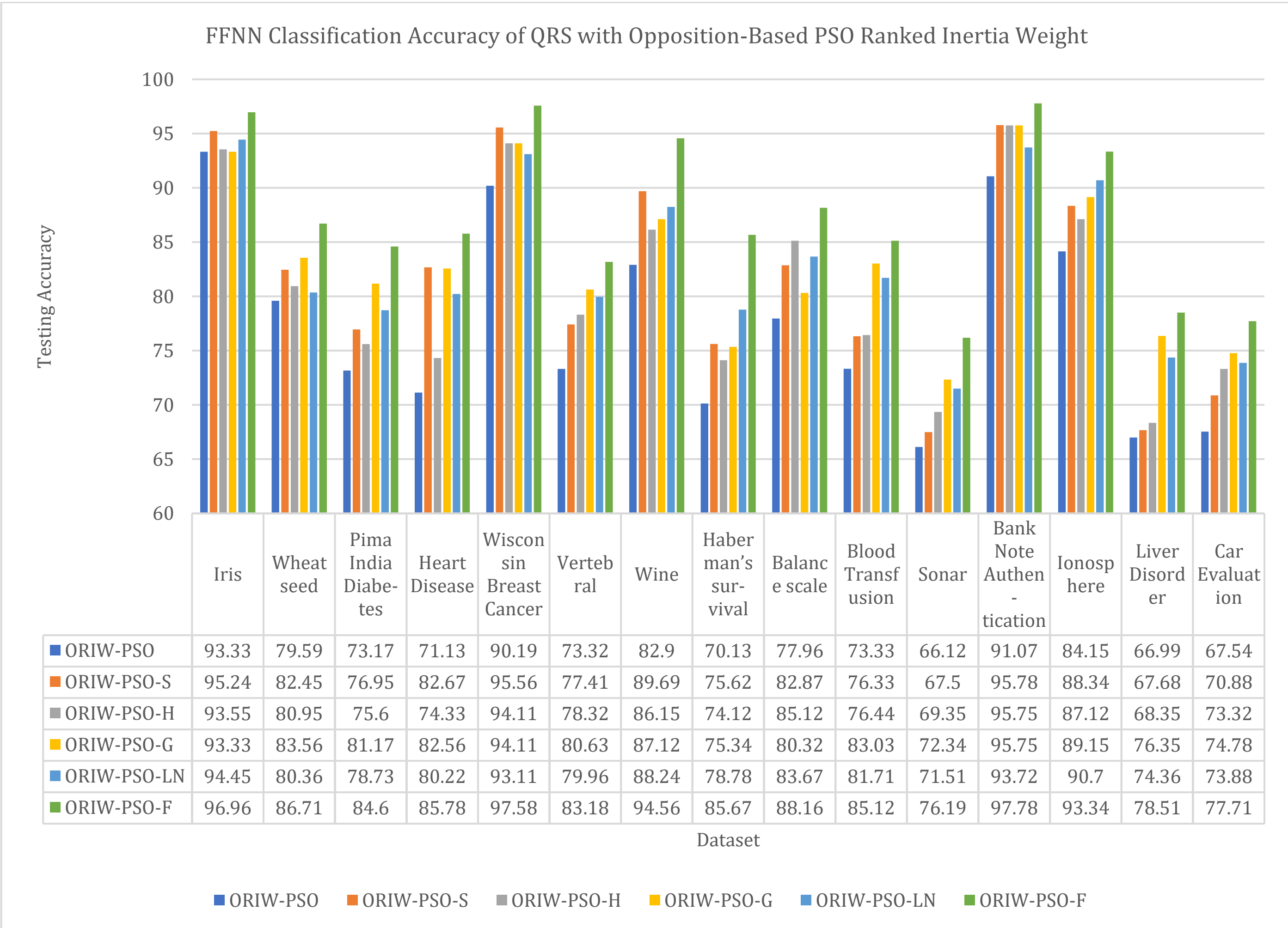


Figure 13. Testing accuracy result of QRS with opposition-based PSO ranked based inertia weight approaches.

Table 4. One-way ANOVA result test of QRS with opposition-based PSO rank based inertia weight approaches.

Parameter	Relation	Sum of Squares	df	Mean Square	F	Significance
Testing Accuracy	Between groups	818.095	5	163.619	2.334622	0.04804

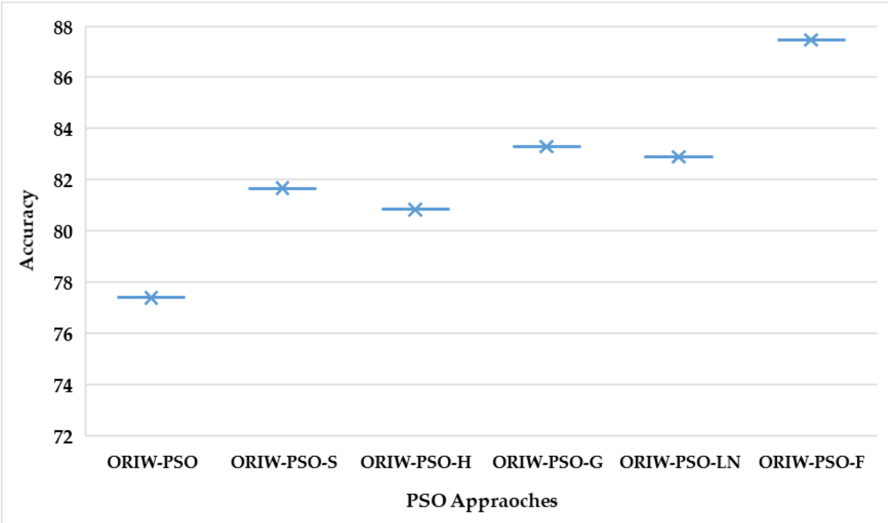


Figure 14. Result of one-way ANOVA test on testing accuracy.

4. Conclusions

PSO has been widely used in various fields to solve real nonlinear complex optimisation problems. It still requires extensive testing to improve its performance, and researchers have proposed several variants of PSO. This paper gives exhaustive detail for the training of feed-forward neural network (FFNN) utilised for different PSO approaches with QRS (Faure) and opposition rank-based inertia to solve premature convergence and local optima problems for best results. In the proposed technique, the initialisation scheme of QRS was used with opposition base method to generate the initial population. Opposition rank-based inertia is used to balance exploitation and exploration searchability. The accuracy results show that the proposed technique ORIW-PSO-F is better than other improved variants. The results illustrate how the proposed techniques affect convergence speed and diversity. Although the primary purpose of this research is to develop a future direction of our work, it could be applied to other stochastic-based meta-heuristic algorithms using mutation operators with this initialisation method. Similarly, this approach and its future variants can also be observed in the energy-saving application area.

Author Contributions: Methodology, K.N.; Software, W.H.B.; Investigation, G.A.M.; Data curation, N.U.R.; Project administration, T.R.S., A.A.A.I. and N.U.H. All authors have read and agreed to the published version of the manuscript.

Funding: The manuscript APC is supported by Universiti Malaysia Sabah, Jalan UMS, 88400, Kota 599 Kinabalu, Malaysia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jiao, Z.; Hu, P.; Xu, H.; Wang, Q. Machine learning and deep learning in chemical health and safety: A systematic review of techniques and applications. *J. Chem. Health Saf.* **2020**, *27*, 316–334. [CrossRef]
2. Thanga Selvi, R.; Muthulakshmi, I. An optimal artificial neural network based big data application for heart disease diagnosis and classification model. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 6129–6139. [CrossRef]
3. Bangyal, W.H.; Ahmad, J.; Shafi, I.; Abbas, Q. A forward only counter propagation network-based approach for contraceptive method choice classification task. *J. Exp. Theor. Artif. Intell.* **2012**, *24*, 211–218. [CrossRef]
4. Sethi, I.K.; Li, X.; Cheng, L.; Yadavalli, S.; Zhang, L. *Nature-Inspired Computing and Optimization—Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2017; p. 494.

5. Slowik, A.; Kwasnicka, H. Nature inspired methods and their industry applications-swarm intelligence algorithms. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1004–1015. [CrossRef]
6. Selvaraj, S.; Choi, E. Survey of swarm intelligence algorithms. In Proceedings of the 3rd International Conference on Software Engineering and Information Management, Sydney, NSW, Australia, 12–15 January 2020.
7. Rostami, M.; Berahmand, K.; Nasiri, E.; Forouzande, S. Review of Swarm Intelligence-Based Feature Selection Methods. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104210. [CrossRef]
8. Eberhart, R.; Kennedy, J. New Optimizer Using Particle Swarm Theory. In Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
9. Chu, S.C.; Tsai, P.W.; Pan, J.S. Cat swarm optimization. In *Pacific Rim International Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4099, pp. 584–588.
10. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
11. Yang, X.S.; Deb, S. Cuckoo Search via Lévy Flights. In Proceedings of the 2009 World Congress on Nature and Biologically Inspired Computing (NaBIC), New Delhi, India, 9–11 December 2009; pp. 210–214.
12. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO)*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
13. Salaria, U.A.; Menhas, M.I.; Manzoor, S. Quasi Oppositional population based global particle swarm optimizer with inertial weights (Qpgpso-w) for solving economic load dispatch problem. *IEEE Access* **2021**, *9*, 134081–134095. [CrossRef]
14. Zhang, J.; Sheng, J.; Lu, J.; Shen, L. UCPSO: A uniform initialized particle swarm optimization algorithm with cosine inertia weight. *Comput. Intell. Neurosci.* **2021**, *2021*, 1–18. [CrossRef]
15. Biswas, A.; Majumdar, A.; Das, S.; Baishnab, K.L. OCSO-CA: Opposition based competitive swarm optimizer in energy efficient IoT clustering. *Front. Comput. Sci.* **2022**, *16*, 1–11. [CrossRef]
16. Bangyal, W.H.; Nisar, K.; Ibrahim, A.A.B.A.; Haque, M.R.; Rodrigues, J.J.P.C.; Rawat, D.B. Comparative analysis of low discrepancy sequence-based initialization approaches using population-based algorithms for solving the global optimization problems. *Appl. Sci.* **2021**, *11*, 7591. [CrossRef]
17. Bangyal, W.H.; Hameed, A.; Alosaimi, W.; Alyami, H. A new initialization approach in particle swarm optimization for global optimization problems. *Comput. Intell. Neurosci.* **2021**, *2021*, 1–17. [CrossRef]
18. Pervaiz, S.; Ul-Qayyum, Z.; Bangyal, W.H.; Gao, L.; Ahmad, J. A systematic literature review on particle swarm optimization techniques for medical diseases detection. *Comput. Math. Methods Med.* **2021**, *2021*, 10. [CrossRef]
19. Ashraf, A.; Almazroi, A.A.; Bangyal, W.H.; Alqarni, M.A. Particle swarm optimization with new initializing technique to solve global optimization problems. *Intell. Autom. Soft Comput.* **2022**, *31*, 191–206. [CrossRef]
20. Shayeghi, H.; Mahdavi, M.; Bagheri, A. An improved DPSO with mutation based on similarity algorithm for optimization of transmission lines loading. *Energy Convers. Manag.* **2010**, *51*, 2715–2723. [CrossRef]
21. Jarboui, B.; Cheikh, M.; Siarry, P.; Rebai, A. Combinatorial particle swarm optimization (CPSO) for partitional clustering problem. *Appl. Math. Comput.* **2007**, *192*, 337–345. [CrossRef]
22. Mahmoud, A.; Yuan, X.; Kheimi, M.; Almadani, M.A.; Hajilounezhad, T.; Yuan, Y. An improved multi-objective particle swarm optimization with TOPSIS and fuzzy logic for optimizing trapezoidal labyrinth weir. *IEEE Access* **2021**, *9*, 25458–25472. [CrossRef]
23. Nshimirimana, R.; Abraham, A.; Nothnagel, G. *A Multi-Objective Particle Swarm for Constraint and Unconstrained Problems*; Springer: London, UK, 2021; Volume 33, ISBN 0123456789.
24. Chen, X.; Li, K. Collective information-based particle swarm optimization for multi-fuel CHP economic dispatch problem. *Knowl. Based Syst.* **2022**, *248*, 108902. [CrossRef]
25. Zdiri, S.; Chrouta, J.; Zaafouri, A. An expanded heterogeneous particle swarm optimization based on adaptive inertia weight. *Math. Probl. Eng.* **2021**, *2021*. [CrossRef]
26. Khan, R.A.; Yang, S.; Khan, S.; Fahad, S. A multimodal improved particle swarm optimization for high dimensional problems in electromagnetic devices. *Energies* **2021**, *14*, 8575. [CrossRef]
27. Tizhoosh, H.R. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, Vienna, Austria, 28–30 November 2005; pp. 695–701.
28. Waseem, Q.; Alshamrani, S.S.; Nisar, K.; Isni, W.; Wan, S. Future technology: Software-defined network (SDN) forensic. *Futur. Technol. Softw. Defin. Netw. Forensic. Symmetry* **2021**, *13*, 767. [CrossRef]
29. Bangyal, W.H.; Ahmad, J.; Abbas, Q. Recognition of off-line isolated handwritten character using counter propagation network. *Int. J. Eng. Technol.* **2013**, *5*, 227–230. [CrossRef]
30. Bangyal, W.H.; Qasim, R.; Rehman, N.; Ahmad, Z.; Dar, H.; Rukhsar, L.; Aman, Z.; Ahmad, J. Detection of fake news text classification on COVID-19 using deep learning approaches. *Comput. Math. Methods Med.* **2021**, *2021*, 1–14. [CrossRef] [PubMed]
31. Nisar, K.; Sabir, Z.; Zahoor Raja, M.A.; Ibrahim, A.A.A.; Rodrigues, J.J.P.C.; Khan, A.S.; Gupta, M.; Kamal, A.; Rawat, D.B. Evolutionary integrated heuristic with gudermannian neural networks for second kind of lane–emden nonlinear singular models. *Appl. Sci.* **2021**, *11*, 4725. [CrossRef]
32. Damaševičius, R.; Maskeliūnas, R. Agent state flipping based hybridization of heuristic optimization algorithms: A case of bat algorithm and krill herd hybrid algorithm. *Algorithms* **2021**, *14*, 358. [CrossRef]

33. Bangyal, W.H.; Hameed, A.; Ahmad, J.; Nisar, K.; Haque, M.R.; Ibrahim, A.A.A.; Rodrigues, J.J.P.C.; Khan, M.A.; Rawat, D.B.; Etengu, R. New modified controlled bat algorithm for numerical optimization problem. *Comput. Mater. Contin.* **2022**, *70*, 2241–2259.
34. Hassan, N.U.; Bangyal, W.H.; Khan, M.S.A.; Nisar, K.; Asri, A.; Ibrahim, A.; Rawat, D.B. Improved opposition-based particle swarm optimization algorithm for global optimization. *Symmetry* **2021**, *13*, 2280. [CrossRef]
35. Dubey, S.M.; Dubey, H.M.; Salkuti, S.R. Modified quasi-opposition-based grey wolf optimization for mathematical and electrical benchmark problems. *Energies* **2022**, *15*, 5704. [CrossRef]
36. Cui, Y.; He, H.; Dong, W.; Liu, L.; Liu, H. The opposition-based learning parameter adjusting harmony search algorithm research on radars optimal deployment. *Math. Probl. Eng.* **2022**, *2022*, 1–30. [CrossRef]
37. Wang, W.; Wang, H.; Sun, H.; Rahnamayan, S. Using opposition-based learning to enhance differential evolution: A comparative study. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 71–77.
38. Ashraf, A.; Pervaiz, S.; Bangyal, W.H.; Nisar, K.; Ibrahim, A.A.A.; Rodrigues, J.J.P.C.; Rawat, D.B. Studying the impact of initialization for population-based algorithms with low-discrepancy sequences. *Appl. Sci.* **2021**, *11*, 8190. [CrossRef]
39. James, F. A review of pseudorandom number generators. *Comput. Phys. Commun.* **1990**, *60*, 329–344. [CrossRef]
40. Bangyal, W.H.; Ahmad, J.; Abbas, Q. Analysis of learning rate using CPN algorithm for hand written character recognition application. *Int. J. Eng. Technol.* **2013**, *5*, 187–190. [CrossRef]
41. Garro, B.A.; Vázquez, R.A. Designing artificial neural networks using particle swarm optimization algorithms. *Comput. Intell. Neurosci.* **2015**, *2015*, 1–20. [CrossRef]
42. Surono, S.; Goh, K.W.; Onn, C.W.; Nurraihan, A.; Siregar, N.S.; Borumand Saeid, A.; Wijaya, T.T. Optimization of Markov weighted fuzzy time series forecasting using genetic algorithm (GA) and particle swarm optimization (PSO). *Emerg. Sci. J.* **2022**, *6*, 1375–1393. [CrossRef]
43. Tarkhaneh, O.; Shen, H. Training of feedforward neural networks for data classification using hybrid particle swarm optimization, mantegna lévy flight and neighborhood search. *Heliyon* **2019**, *5*, e01275. [CrossRef]
44. Si, T.; Dutta, R. Partial opposition based particle swarm optimizer in artificial neural network training for medical data classification. *Int. J. Inf. Technol. Decis. Mak.* **2019**, *18*, 1717–1750. [CrossRef]
45. Chatterjee, S.; Hore, S.; Dey, N. Dengue fever classification using gene expression data: A PSO Based artificial neural network approach. In *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*; Springer: Singapore, 2017; pp. 331–341.
46. Feshki, M.G.; Shijani, O.S. Improving the heart disease diagnosis by evolutionary algorithm of PSO and feed forward neural network. In Proceedings of the 2016 Artificial Intelligence and Robotics (IRANOPEN), Qazvin, Iran, 9 April 2016; pp. 48–53.
47. Bas, E.; Egrioglu, E.; Kolemen, E. Training simple recurrent deep artificial neural network for forecasting using particle swarm optimization. *Granul. Comput.* **2021**, *2*, 1–10. [CrossRef]
48. Khan, T.A.; Zain-Ul-Abideen, K.; Ling, S.H. A hybrid advanced PSO-neural network system. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; pp. 1626–1630.
49. Duong, H.Q.; Nguyen, Q.H.; Nguyen, D.T.; Van Nguyen, L. PSO based hybrid PID-FLC sugeno control for excitation system of large synchronous motor. *Emerg. Sci. J.* **2022**, *6*, 201–216. [CrossRef]
50. Yadav, A.; Peesapati, R.; Kumar, N. Electricity price forecasting and classification through wavelet—Dynamic weighted. *IEEE Syst. J.* **2017**, *12*, 3075–3084.
51. Beheshti, Z.; Mariyam, S.; Shamsuddin, H.; Beheshti, E.; Sophiayati, S. Enhancement of artificial neural network learning using centripetal accelerated particle swarm optimization for medical diseases diagnosis. *Soft Comput.* **2014**, *18*, 2253–2270. [CrossRef]
52. Wahab, M.A.; Nguyen, H.X.; Roeck, G. Damage detection in structures using particle swarm optimization combined with artificial neural network. *Smart Struct. Syst.* **2021**, *1*, 1–12.
53. Matsumoto, M. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* **1998**, *8*, 3–30. [CrossRef]
54. Mahdavi, S.; Rahnamayan, S.; Deb, K. Opposition based learning: A literature review. *Swarm Evol. Comput.* **2017**, *39*, 1–23. [CrossRef]
55. Xu, Q.; Wang, L.; Wang, N.; Hei, X.; Zhao, L. A review of opposition-based learning from 2005 to 2012. *Eng. Appl. Artif. Intell.* **2014**, *29*, 1–12. [CrossRef]
56. Bansal, J.C.; Singh, P.K.; Saraswat, M.; Verma, A.; Jadon, S.S.; Abraham, A. Inertia weight strategies in particle swarm optimization. In Proceedings of the 2011 Third World Congress on Nature and Biologically Inspired Computing, Salamanca, Spain, 19–21 October 2011; pp. 633–640.