IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 11, 2020, accepted September 20, 2020, date of publication September 24, 2020,
date of current version October 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3026529

A Hybrid Metaheuristic Method in Training Artificial Neural Network for Bankruptcy Prediction

ABDOLLAH ANSARI[1,2], IBRAHIM SAID AHMAD[1,3], AZURALIZA ABU BAKAR[1], AND MOHD RIDZWAN YAAKUB[1]

[1]Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi 43600, Malaysia
[2]Institute of Intelligent Nano and Informatics (INI), Tehran 15336-66896, Iran
[3]Department of Information Technology, Faculty of Computer Science and Information Technology, Bayero University Kano, Kano 700241, Nigeria

Corresponding author: Ibrahim Said Ahmad (isahmad.it@buk.edu.ng)

This work was supported in part by the Universiti Kebangsaan Malaysia Research University Publication Incentive Grant GP-K012964-2020.

ABSTRACT Corporate bankruptcy prediction is an important task in the determination of corporate solvency, that is, whether a company can meet up to its financial obligations or not. It is widely studied as it has a significant effect on employees, customers, management, stockholders, bank lending assessments, and profitability. In recent years, machine learning techniques, particularly Artificial Neural Network (ANN), have widely been studied for bankruptcy prediction since they have proven to be a good predictor, especially in financial applications. A critical process in learning a network is weight training. Although the ANN is mathematically efficient, it has a complex weight training process, especially in computation time when involving a large training data. Many studies improved ANN's weight training using metaheuristic algorithms such as Evolutionary Algorithms (EA), and Swarm Intelligence (SI) approaches for bankruptcy prediction. In this study, two metaheuristics algorithms, Magnetic Optimization Algorithm (MOA) and Particle Swarm Optimization (PSO), have been enhanced through hybridization to propose a new method MOA-PSO. Hybrid algorithms have been proven to be capable of solving optimization problems faster, with better accuracy. The MOA-PSO was used in training ANN to improve the performance of the ANN in bankruptcy prediction. The performance of the hybrid MOA-PSO was compared with that of four existing algorithms. The proposed hybrid MOA-PSO algorithm exhibits promising results with a faster and more accurate prediction, with 99.7% accuracy.

INDEX TERMS Evolutionary optimization algorithms, bankruptcy prediction, artificial neural network, magnetic optimization algorithm, particle swarm optimization, metaheuristic.

I. INTRODUCTION

Bankruptcy is an unwanted phenomenon. It negatively affects business owners, managers, shareholders, employees, manufacturers, suppliers, customers, and the government [1]. Hence, the prediction of bankruptcy is of great importance in financial analysis and has widely been studied in recent decades. Studies on bankruptcy prediction fall under various disciplines such as finance, accounting, business management, and computer science. Traditionally, bankruptcy prediction used statistical approaches, and the one developed by Beaver in 1966 was among the first study on bankruptcy prediction [2]. However, machine learning techniques are now commonly used as studies have shown them to be more accurate [3]. One of the most widely used machine learning techniques in bankruptcy prediction is the Artificial Neural Network [4]. Studies on Artificial Neural Networks' (ANNs) application for bankruptcy prediction problems began in the 1990s, and they are used in today's widespread literature [3], [5], [6].

Furthermore, many studies have been done to compare statistical methods with ANNs approaches. The studies show that while ANNs suffer from a problem of interpretability, they are more efficient methods of forecasting

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

and outperform statistical methods because they provide more robust classifiers and handle complex underlying relationships [7]–[9]. However, ANNs still have some downsides, one of which is long training time. One of the most widely used algorithms in training ANNs is the Gradient Descent (GD) algorithm [10]. The GD algorithm has two main limitations. First, it can easily fall into local optimum, and secondly, it has a slow convergence rate [11]. Consequently, several metaheuristic optimization algorithms have been studied for improving the training of ANNs.

Magnetic Optimization Algorithm (MOA) is a relatively new heuristic optimization algorithm introduced by M. H. Tayarani-N & Akbarzadeh-T in 2008 inspired by magnetic field theory [12]. Studies have shown that the algorithm has a good performance in solving optimization problems. MOA was initially introduced to solve problems with continuous real search spaces [13], [14]. Particle swarm optimization (PSO) is a well-known metaheuristic optimization technique. PSO algorithm has widely been studied in solving optimization problems in many domains [15], [16]. Initially introduced by Eberhart & Kennedy in 1995, they proposed several modifications of the PSO algorithm to improve its performance [17].

In this paper, we designed an implementation of a hybrid algorithm for training ANN in bankruptcy prediction, based on MOA and PSO called MOA-PSO. MOA-PSO combines the local search capability of MOA with the social thinking capability of PSO. The experimental results show the superiority of our approach compared to existing approaches. The main contributions of this paper are:

1. Enhancing the Magnetic Optimization Algorithm (MOA) by hybridizing it with Particle Swarm Optimization (PSO).
2. Decreasing the time-complexity of weight training of ANN in Bankruptcy prediction.

The rest of the paper is organized as follows. Section 2 presents the related works; then, section 3 introduces the proposed method. Section 4 is dedicated to the results, while Section 5 presents the discussion and limitations, and finally, Section 6 concludes the paper.

II. RELATED WORKS

Metaheuristics approaches have been widely studied in training the Feedforward Neural Network (FFNN) to handle gradient-based algorithms' drawbacks, particularity backpropagation algorithms. For example, beginning in the 2000s, many works have concentrated on metaheuristics algorithms in training neural networks for binary classification problems such as bankruptcy prediction [18]–[20]. Metaheuristics approaches have been proven to be superior and more convenient to implement than gradient-based algorithms [21]. Pendharkar & Rodger proposed the usage of a Genetic Algorithm (GA) based artificial neural network (GA-ANN) to learn the weights of connection for classification problems [22]. Kiranyaz *et al.* proposed a multi-dimensional particle swarm optimization (MDPSO) approach for training an ANN to overcome the drawbacks of existing approaches [23].

However, recently many studies agree on the benefit of uniting mechanisms from different search methods. There is a widespread trend to design hybrid techniques in operations research and artificial intelligence [24], [25]. The primary inspiration for the hybridization of various algorithms is to use the complementary characteristics of different optimization approaches, as hybrids are supposed to take advantage of synergies. Hybridization is not only limited to the mixture of different metaheuristics but also comprises the use of hybrid algorithms, which syndicate local search or exact algorithms and metaheuristics algorithms [26].

Zhang *et al.* applied a hybrid algorithm by combining a PSO algorithm with a backpropagation algorithm called PSO–BP algorithm. PSO–BP trains the weights of FFNN. The hybrid algorithm not only can take advantage of the strong global searching capabilities of the PSOA, but also make use of the strong local searching capabilities of the BP algorithm [27]. Niu & Li presented a new hybrid global optimization algorithm called PSODE by combining PSO with Differential Evolution (DE). PSODE is a parallel algorithm in which PSO and DE are performed in parallel to improve the population with frequent information sharing [28]. Xinsheng Lai & Mingyi Zhang recommended a simple and effective joint model of GA and PSO. This collaborative model holds one population named public population on which GA and PSO run. After running this model on the public population, each section optimization results in an offspring population. Subsequently, the resulting new generation of the public population will be renewed by combining both offspring populations based on their best individuals' 'fitness'' [29].

In more recent studies, PratimSarangi *et al.* proposed a hybridized model of DE with a backpropagation algorithm called the DE-BP algorithm. The hybrid algorithm was applied to train the FFNN network's weights by utilizing the global searching characteristics of the DE evolutionary algorithm and strong local searching capabilities of the backpropagation algorithm [30]. Wu *et al.* proposed an effective hybrid PSO and GA called HPSOGA, which is used to determine the radial basis function neural network parameters. This hybrid algorithm is used to automatically build a radial basis function neural network (RBF-NN) [18]. Ghasemiyeh *et al.* proposed a new hybrid model based on Improved Cuckoo Search (ICS) and GA called ICSGA with ANN. They combined the advantages of GA and ICS to overcome the main disadvantage of GA, easily becoming trapped in the local minima through the ICS [19]. Yan *et al.* proposed a hybrid PSO and quasi-Newton (QN) algorithm on the CPU-GPU platform using OpenCL to accelerate ANN training. The PSO-QN implementation combines the PSO algorithm's strength in a global search and the advantage of the QN algorithm in a fast convergence rate [31].

On bankruptcy prediction, nowadays, machine learning models are commonly used. The most commonly used

TABLE 1. Dataset, techniques and result of some recent studies on bankruptcy prediction.

Paper	Dataset	Algorithms	Accuracy (%)	Observation	Period
[32]	Taiwan / Taiwan Economic Journal	SVM, KNN, CART, ANN, NB	82%	480	1999 - 2009
[33]	US and Canada NYU's Salomon Centre database	LDA, Logit, NN, SVM, bagging, boosting, and RF	87%	10,000+	1985 - 2013
[34]	Taiwan Economic Journal (TEJ) database	Fuzzy clustering, BPNN	95.25%	600	
[35]	France / DIANE	GP, SVM, Logit	94.2	540,000	2002 - 2006
[36]	Korea / Korean financial company	RF, DT, MLP, SVM	84.2% (AUC)	120, 335	2016 & 2017
[37]	Korea / Korean financial company	Cluster-based Boosting, GMBoost, DT, RF, MLP, AdaBoost	86.8% (AUC)	120, 335	2016 & 2017
[38]	France / Altares database	LDA, Logit, ANN, SVM, RF	81.1 (sensitivity)	1500	2013 & 2014
[39]	US / Compustat North America, Centre for Research in Security Prices (CRSP), Securities Exchange Commission (SEC).	Deep learning embeddings, CNN, SVM, Logit, RF	78.4% (AUC)	11,827	1994 - 2014
[40]	Korea / NICE Information Service Co.	Logit, RF, XGBoost, LightGBM, ANN	88% (AUC)	977,940	2011 - 2016
[41]	UCI & LIbSVM datasets	Bagged-pSVM, Boosted-pSVM,	85.42%	690 - 10503	-

algorithms include Support Vector Machines (SVM), ANN, Gaussian Process (GP), Classification and Regression Tree (CART), Logistic Regression (Logit), Decision Tree (DT), Random Forest (RF), Linear Discriminant Analysis (LDA), and ensemble learning techniques. Table 1 shows the datasets, techniques, and accuracy of some recent studies on Bankruptcy prediction.

III. THE PROPOSED METHOD

The proposed method is a hybrid of MOA and PSO algorithms. The main rationale for the hybridization is to utilize local search capability of MOA with social thinking capability of PSO. MOA was introduced by M. H. Tayarani-N & Akbarzadeh-T in 2008 [12] and was inspired by the electromagnetic force. The electromagnetic force is one of the four elements of force in nature that has a long-range effect; meaning the strength of the effect is dependent on the distance between the particles. The further two particles are apart, the weaker the effect, however, the effect only disappear when the distance between the particles is infinite. MOA was modelled on this principle. MOA contains 7 steps and the pseudocode presented in Algorithm 1.

[12] proposed seven main steps for MOA procedure which are written as follow:

1) Initial parameters: The first step is the solution initialization for $t = 0$. In this research, random initialization in used for assigning values to particles. In addition, constant parameters ρ and α are initialized in this phase.
2) Particle evaluation: The value of fitness in each particle is calculated in this step, and then stored in the magnetic field B_{ij}.
3) Normalization: The first B^t of all the particles gets normalized by using equation (1) in this step, and then the mass is calculated according to equation (2).

$$B_{ij} = \frac{B_{ij} - Min}{Max - Min} \tag{1}$$

where Min is the minimum value of B_{ij} for all the particles and Max is the maximum value of B_{ij} among all the particles.

Mass calculation based on the formula proposed by [12]

$$M_{ij}^t = \rho \times B_{ij}^t + \alpha \tag{2}$$

The movement of particles is controlled by two constant values of α and ρ for having a better balance between exploitation and exploration in different problems. Increasing the values of α and ρ makes the movement slower due to the heavy mass, that results in more exploitation. On the other hand, decreasing the values of ρ and α results in faster movement that causes more exploration.

4) Finding the neighbours: For finding the velocity of each particle in the overall population, forces accumulation from other particles to an appointed agent must be appraised. The first step to calculate the forces is finding particle neighbours. A lattice-like network was proposed by [12], which each particle is neighbouring with four other particles in the network.

Based on the connection network, the neighbours of each particle are found. In the lattice-like network, neighbours are acquired as follows:

$$N_{wk} = \{x_{w'k}, x_{wk'}, x_{w''k}, x_{wk''}\} \tag{3}$$

Algorithm 1 Magnetic Optimization Algorithm Pseudocode [42]

MOA Procedure

Step 1: Initial parameters
While Terminating criteria is not satisfied
Step 2: Evaluate particles in X^t and store their fitness values in the magnetic field B^t
Step 3: Normalize B^t and evaluate mass M^t according to formula (2)
for each particle **do**
$F_{ij} \leftarrow 0$
Step 4: Find neighbours based on network
for each neighbour of corresponding particle
Step 5: Calculate forces from neighbours
end
for each particle
Step 6: Update velocity based on neighbour forces
Step 7: Update the particle based on its corresponding velocity
end
end
end

where:

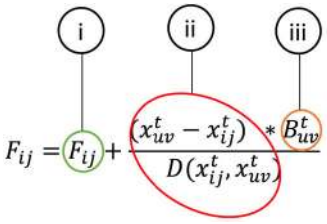
$$w' = \begin{cases} w-1, & w \neq 1 \\ S, & k = 1 \end{cases} \tag{4}$$

$$w'' = \begin{cases} w+1, & w \neq S \\ 1, & w = S \end{cases} \tag{5}$$

$$k' = \begin{cases} k-1, & k \neq 1 \\ S, & k = 1 \end{cases} \tag{6}$$

$$k'' = \begin{cases} k+1, & k \neq S \\ 1, & k = S \end{cases} \tag{7}$$

Calculating force: The force value F_{ij} is calculated using equation (8). This equation is a combination of similarity between two solutions and field of magnetic that illustrates the particle's fitness value.



(8)

Every particle in the lattice-like network (Fig. 1) connects to four neighbours. For estimating the summation of forces from its neighbours, the part (i) of equation (8) has been used. Part (ii) and (iii) of the equation calculate the mutual force between a particle and one of its neighbours. The part art (ii) is the difference and similarity between the particle and its neighbour. In this formula based on [12], the similarity is calculated by the difference between two solutions and divided by geometric distance. The third part (iii) of equation (8) is a magnetic part of the particle and calculated by normalization fitness value. The second and third sections, two concepts

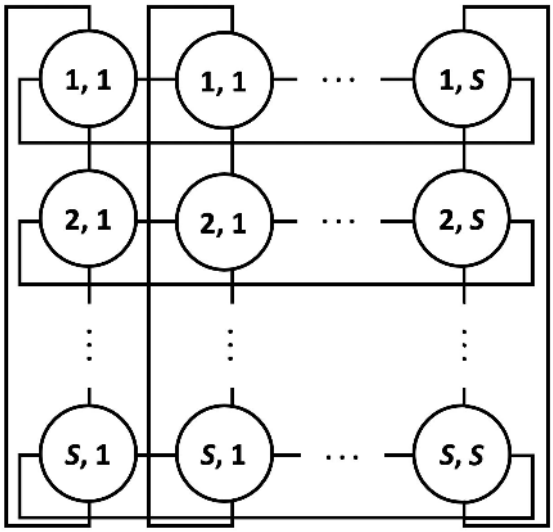


FIGURE 1. Lattice like network for finding neighbour particles [12].

in the optimization techniques (i.e. similarity between two particles and fitness value), are considered as contributing factors in finding a good solution for combinatorial problems.

To find the geometry distance of particles, D (…) is specified by $l^2 - norm$ of two solutions that are indicated in equation (9).

$$D\left(x_{ij}^t, x_{uv}^t\right) = \sqrt{\sum_{k=1}^{n} (x_{ij}^t - x_{uv}^t)^2} \tag{9}$$

The force in equation (8) is related to the distance between two particles and the fitness value of the one which we want to calculate its fitness.

Updating velocity: Particles in the PSO move to find better local and global solutions. Particles in MOA, like PSO, want to move in each iteration and it could be attained using

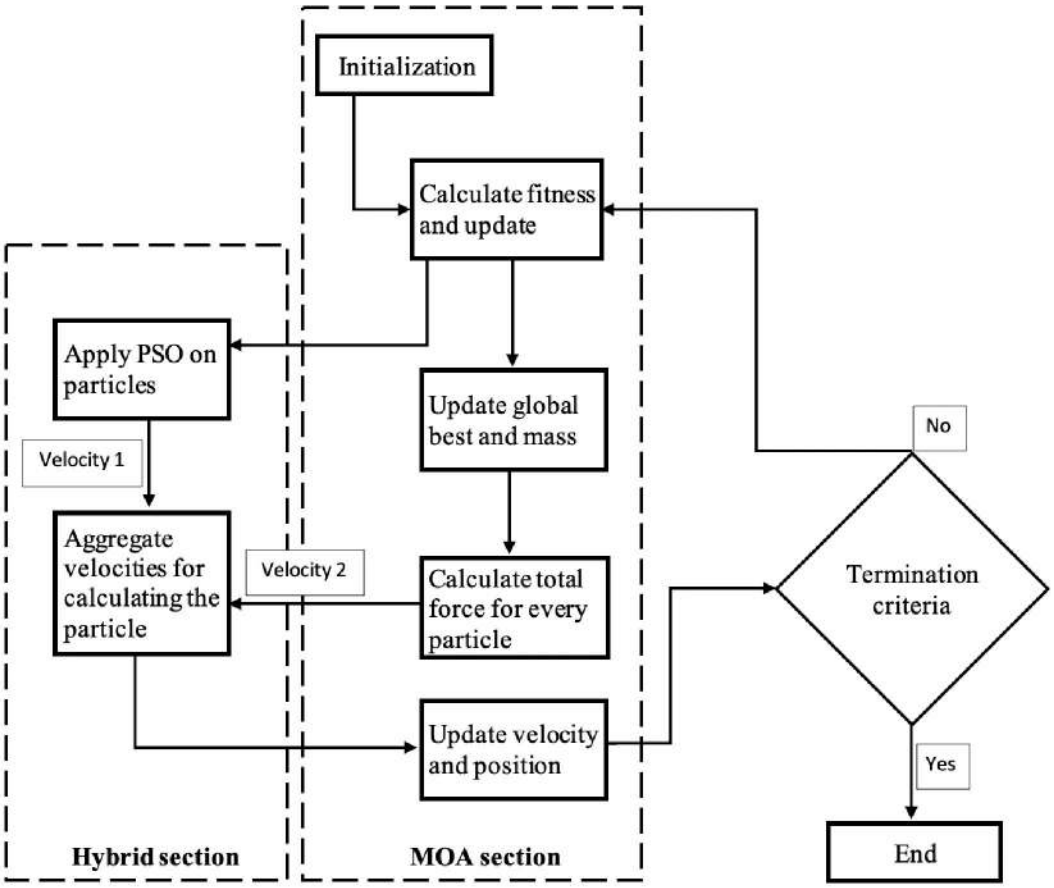


FIGURE 2. Comparison of MOA, PSO, and MOA-PSO in training ANN.

velocity that counts the movement amount of each particle based on its forces and masses. The velocity that is based on accumulated forces, mass and a random value can be calculated using equation (10).

$$v_{ij,k}^{t+1} = \frac{F_{ij}}{M_{ij}} \times R(l_k, u_k) \tag{10}$$

Updating particles: In this section, the position of the particle gets updated according to the value its velocity.

$$x_{ij}^{t+1} = x_{ij}^{t} + v_{ij}^{t+1} \tag{11}$$

A. HYBRIDIZATION MOA WITH PSO TO IMPROVE MOA PERFORMANCE

In this section, we present the method used in the hybridization of MOA and PSO. Fig. 2 presents the flowchart of the proposed method. In the flowchart, two main parts of MOA-PSO were highlighted. Part B is the basic MOA while part A is the hybrid method with PSO. Every part calculates a velocity and in the final part, these two velocities are aggregated.

Combining the capability of local search in the MOA with social thinking (*gbest)* ability of PSO is the fundamental purpose of hybridizing the MOA and PSO. Equation (12) has been proposed to combine these methods:

$$V_i\,(t+1) = w * V_i\,(t) + (P_{gb}\,(t) - P_i\,(t)) + V_{magnetic}\,(t) \tag{12}$$

$$X_i\,(t+1) = X_i\,(t) + V_i\,(t+1) \tag{13}$$

where $V_i(t)$ is the velocity of agent i at iteration t and is added to $(P_{gb}\,(t) - P_i\,(t))$ from PSO algorithm and $V_{magnetic}\,(t)$ which is calculated from MOA. The final velocity will be added to particles.

In hybrid MOA-PSO every agent is considered as one candidate solution and all the agents are initialized randomly in the first place. The gravitational constants (ρ and α), and resultant forces (F_{ij}) among agents are calculated using equation (2), (3) and (5) respectively after initialization. Then the particles accelerations are determined according to equation (6). The best solution in every iteration is then updated. Finally, all the agents' velocities are calculated using equation (12). The agents' positions are updated using equation (13). After finding the end criteria, the updating process of positions and velocities will be discontinued.

Mean Square Error (MSE) is considered as fitness value for the proposed algorithm, and the proposed algorithm attempts to minimize it. Moreover, particles with a higher quality of

solutions attract other particles. Another contributing feature of MOA-PSO is the distance between particles. It means that particles farther apart move faster towards other particles in comparison to particles near each other. In addition, unlike MOA, the proposed algorithm regards the global best solution in every iteration and this solution affects the final result. MOA-PSO algorithm can be adjusted between exploration and exploitation by changing the algorithm parameters.

B. DATASET DESCRIPTION

The dataset used in this paper was originally acquired by Atiya in 2001 [43]. It is obtained from the United States solvent and defaulted firms. It contains 65 attributes and 984 records. Every record from the second row to 984th row represents a firm. A firm could be displayed two or three times at various instances before default, for instance, 8 months before default and 20 months before default. The first record presents an index for every indicator. The entries are considered as indicator values except in the last three attributes. The last attribute is the target value (default (-1) or non-default (1)). The column next to the last column is ignored. Third to the last attributes presents the number of months before default (if non-defaulted, the entry would be 1000). Five or six indicators from all these supplied are used. The indicators we used finally in system 1 are the ones numbered: [10,103,50,47,102]. The dataset was used mainly because it is one of the few publicly available datasets on bankruptcy prediction acquired from official government source responsible for storing the records and it has been used in existing prominent studies. The full description of the dataset is shown in Table 8 in the appendix.

IV. RESULTS

In this section, extensive experiments were conducted to evaluate the efficiency of our proposed methods. We apply the proposed MOA-PSO, PSO, and MOA on FFNN optimization to address bankruptcy prediction problem. We compared the obtained results of MOA-PSO algorithm with the PSO acquired results and MOA attained results. We also compared the obtained results with that of [43]. All the implementations and experiments were done using Python programming language.

A. RESULTS OF PSO, MOA AND MOA-PSO ON FFNN

1) RESULT OF PARAMETER TUNING

The algorithm was tested with different values of parameters and the obtained results were observed. The obtained results include the average accuracy of ANN in 31 runs as well as the average time for each run of the algorithm on ANN. Another important acquired result is the standard deviation between different runs, which is a statistical attribute to measure the difference between the different runs values. Table 2 presents the results obtained from parameter tuning. There are three input parameters namely α, ρ, and w. The results based on the input parameters are presented under accuracy, time-per-run, and standard deviation of MSE.

In terms of the accuracy measure, the best accuracy was obtained on test 13 with 99.7% while the poorest accuracy value was obtained on test 8 with 96.7%. The rest of the test results fall between these two accuracies. It can be observed that the performance of the proposed algorithm is not significantly affected with the change in the input constant parameters (α, ρ and w) which this ability can be regarded as an advantage of MOA-PSO. This conclusion holds for prediction with ANN, however, there is no claim to expand the conclusion to the other NP-hard problems unless the proposed algorithm is tested on them.

The average of running time per run in the pre-set iteration number is termed the time-per-run. Test 8 provides the slowest performance with 155.56 seconds while test 3 is the fastest test with 154.31 seconds. Consequently, it can be observed that there is a small difference between these two values and therefore, a similar conclusion with that of accuracy measure can be drawn here, that is the change in the input constant parameters (α, ρ and w) have little effect on the running time.

Finally, in terms of standard deviation between different MSE measures in the training of ANN, the best standard deviation is obtained in test 13 with 0.001 and the worst is obtained in test 1 with 0.106. These standard deviation values are low and show that the results deviate less regarding different input parameter values.

2) PARAMETER SETTING FOR PSO, MOA AND MOA-PSO

First, PSO was applied to the FFNN. The parameters are set according to the ideal values acquired by Gudise & Venayagamoorthy [44] in their research. They applied PSO on FFNN and made a comparative study on the computational requirements of the PSO and backpropagation as neural networks training algorithms. In their paper, the obtained results proved that the weights of the FFNN converge faster using the PSO than using the BP algorithm. Their optimal parameters of *w, C1 and C2* were used in this experiment and the result is presented in Table 3.

Secondly, we applied MOA on the FFNN. The parameters are set according to the ideal parameter values acquired by [14]. They applied MOA on multi-layer perceptron NN as a novel training approach to cover the shortcomings of the previously used training algorithms. Their proposed algorithm was compared to other techniques such as PSO and GA-based learning method. The results showed that MOA has superiority over the GA and PSO for large numbers of training samples. For the test in this research, their optimal parameters of α and ρ were used and the result is presented in Table 4. On another hand, Table 5 shows the result of the default parameters for testing MOA on training weights of ANNs.

3) COMPARATIVE ANALYSIS OF PSO, MOA, AND MOA-PSO

In this section, we compared the performance of MOA-PSO algorithm with that of PSO and MOA algorithms in training

TABLE 2. Running 14 tests to get the best parameters value.

Test	ρ	α	w	Accuracy	Time-per- run	Standard deviation
1	0.01	0.01	0.01	94.769	154.54	0.106
2	1.00	1.00	0.40	99.39	154.42	0.016
3	0.01	0.50	2.00	96.85	154.31	0.072
4	0.50	0.01	1.00	97.472	154.31	0.060
5	0.10	0.10	0.80	97.025	154.37	0.095
6	0.50	1.00	2.00	97.962	154.32	0.047
7	0.10	0.50	0.40	99.51	154.39	0.004
8	0.01	0.10	0.01	96.785	155.56	0.077
9	1.00	0.10	1.00	98.54	154.46	0.042
10	0.10	0.01	0.40	99.161	154.44	0.015
11	1.00	1.00	2.00	98.812	154.45	0.037
12	0.50	0.01	1.00	97.254	154.54	0.051
13	**0.01**	**0.50**	**0.80**	**99.728**	**154.48**	**0.001**
14	0.10	0.10	2.00	98.66	154.48	0.041

TABLE 3. PSO parameters based on [44].

Parameter	Values
Run number	31
Iteration	200
Population	25
Training	70%
Test	30%
Initialization	Randomly
Search space range	[-1 , 1]
W	0.8
C1, C2	2

TABLE 4. MOA parameters according to [14].

Parameter	Values
Run number	31
Iteration	200
Population	25
Training	70%
Test	30%
Initialization	Randomly
Search space range	[-1 , 1]
ρ	0.1
α	0.1

ANN. From Table 2, the best accuracy in different tests with MOA-PSO algorithm is 99.728%. Also, the average time of each run in this algorithm is 154.48 seconds.

On the other hand, the performance of PSO and MOA algorithms are shown in Table 6. The best accuracy in PSO algorithm is 52.899% while that of MOA algorithm is 72.243% both of which are lower than the accuracy obtained using MOA-PSO algorithm. Similarly, the average time of each run in the PSO algorithm is 161.14 seconds while that of MOA algorithm is 155.7 seconds; meaning both PSO and MOA algorithms run slower than the MOA-PSO algorithm.

Consequently, the hybrid MOA-PSO algorithm has a better performance by providing a better balance between exploration and exploitation in searching for the problem space than MOA and PSO. Therefore, we can conclude that we improved the MOA through hybridizing with PSO for

TABLE 5. Run parameters of Hybrid MOA-PSO.

Parameter	Values
Run number	31
Iteration	200
Population	25
Training	70%
Test	30%
Initialization	Randomly
Search space range	[-1, 1]

TABLE 6. Comparisons between PSO, MOA, and Hybrid MOA-PSO.

Result	PSO	MOA	PSO-MOA
Accuracy (%)	52.899	72.243	**99.728**
Training Time/run (seconds)	161.14	155.7	**154.48**

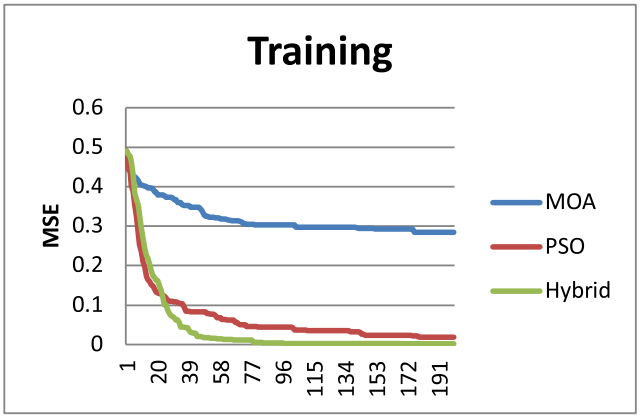


FIGURE 3. Comparison of MOA, PSO, and MOA-PSO in training ANN.

training ANN in terms of accuracy and running time. This improvement is because of implementing the concept of social behaviour of PSO in MOA.

Fig. 3 further illustrates the comparison between the three metaheuristics algorithms used for training ANN. As it can be seen, the proposed hybrid method outperforms the other two algorithms (i.e., PSO and MOA) in terms of MSE measure.

B. MOA-PSO PERFORMANCE IN BANKRUPTCY PREDICTION

Atiya [43] applied ANN by Financial Ration Method, and ANN with Financial Ratio and Equity-Based Model. In their research, they utilized the concept of out of sample (testing data) and in-sample (training data) for bankruptcy prediction. The comparative results are presented in Table 7.

As can be seen from Table 7, MOA-PSO attained the best result in terms of accuracy compared to the other approaches.

TABLE 7. MOA-PSO performance in bankruptcy properties.

	Training data	Testing data	Accuracy
PSO	688	296	52.899
MOA	688	296	72.243
MOA-PSO	688	296	**99.728**
ANN Financial Ratio	491	546	81.46
ANN Financial Ratio + Equity-Based	491	572	85.50

The superiority of MOA-PSO over the mentioned methods is mainly because of a better adjustment of weights and biases during the training section in prediction with ANN.

To test that the proposed approach has performed better than the baseline approach with statistically significantly different mean, an independent t-test was performed. Two pairs were taken, between the means of the runs of the proposed MOA-PSO and that of [43]. A confidence level of 95% was taken and the p-value in the three pairs is less than 0.05 and therefore the results can be accepted as significant. Thus, the proposed achieved statistically significantly better prediction accuracy and as such performed significantly better than the baseline models in bankruptcy prediction. The proposed method performed better than the baseline method because of the utilization of local search capability of MOA with social thinking of PSO algorithms that prove to be efficient.

V. DISCUSSION AND LIMITATIONS

In the experiments, the proposed MOA-PSO shows promising results in bankruptcy prediction. The results obtained are significantly better than the baseline approaches in terms of prediction accuracy and duration of time for execution. Two critical improvements led to this result. First, the problem of imbalance between exploration and exploitation of the MOA algorithm was targeted using the concept of social thing of the PSO algorithm. Secondly, the local search capability of MOA algorithm was utilized in the proposed MOA-PSO approach. In terms of complexity, the time complexity of the PSO algorithm is $O(n^2t)$ where n is the number for the inner loops, and t represents the outer loop iteration. On the other hand, MOA has a time complexity of $O(n)$ [12]. The total execution complexity of the MOA-PSO implementation is $O(n * I_{MOA-PSO})$, where $I_{MOA-PSO}$ is the total number of iterations of the MOA-PSO.

VI. SUMMARY AND CONCLUSION

In this paper, we first propose a method to improve the MOA algorithm by hybridizing it with the PSO algorithm called MOA-PSO and then used the hybrid method to train ANN for bankruptcy prediction. According to the obtained results, the developed hybrid MOA-PSO succeeds in improving the

TABLE 8. Dataset description.

Attribute	Indicator #	Attribute	Indicator #
Cash/tot assets	1	Rate of chg (ROC) of earnings per share (EPS)	17
Working capital/tot assets (TA)	4	ROC (gross operating income GOI)	19
Working capital/cur assets	7	ROC (net oper. Inc NOI)	20
Equity (EQ)/TA	10	ROC (sales)	21
1- (long term debt/TA)	13	ROC (gross profit margin)	22
Rate of chg of cash flow per share (CFPS)	16	ROC (net profit margin)	23
ROC (EPS from cont. operations)	18	A measure of share price chg	24
A measure of chg of gross oper mgn	25	Price book value ratio	65
One year chg in net profit mgn	26	Return on assets ROA	68
ROC (TA)	27	Return on equity	71
One year chg in EQ	28	Current ratio	74
Other ROC(CFPS) (other measure of chg)	29	Quick ratio	77
Other ROC(EPS)	30	Market capitalization/(long term debt LTD)	80
Other ROC (EPS cont oper)	31	Relative strength indicator	83
Other ROC(GOI)	32	Gross profit mgn	86
Other ROC(NOI)	33	Net profit mgn	89
Other ROC(sales)	34	One-year rel chg of CF	95
Gross profit mgn	35	One-year rel chg of GOI	96
Net profit mgn	36	One-year rel chg og NOI	97
A measure of dividend incr/decr	37	4 yr ROC (CF)	98
Cash flow (CF)/TA	38	4 yr ROC (GOI)	99
Earnings/TA	41	4 yr ROC (NOI)	100
Earnings cont oper/TA	44	3 yr ROC (CF)	101
GOI/TA	47	3 yr ROC (GOI)	102
NOI/TA	50	3 yr ROC (NOI)	103
Sales/TA	53	TA	104
PE ratio	56	Sector default prob	105
P/CF ratio	59	One year ROC (price)	106
Price sales ratio	62	4 yr ROC (price)	107
3 yr ROC (price)	108	3 yr ROC (EQ)	112
Price	109	Month till chapt 11 filing (for solvent co. entry = 1000)	113
A measure of ROC (price)	110	Indexing	114
Volatility	111	Target (1=solvent, -1=bankrupt)	115

efficiency of MOA and PSO algorithms. Using the hybrid MOA-PSO to train the FFNN not only increased the accuracy but also reduced the training time per run. In future, we plan to address some of the limitations of the proposed method. We plan to evaluate the approach with more recent but equally reputable datasets. Other extensions of MOA like Functional Sized Population MOA (FSMOA) could also be investigated for bankruptcy prediction.

APPENDIX

See Table 8 here.

REFERENCES

[1] F. Mokhatab Rafiei, S. M. Manzari, and S. Bostanian, "Financial health prediction models using artificial neural networks, genetic algorithm and multivariate discriminant analysis: Iranian evidence," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10210–10217, Aug. 2011, doi: 10.1016/j.eswa.2011.02.082.

[2] W. H. Beaver, "Financial ratios as predictors of failure," *J. Account. Res.*, vol. 4, pp. 71–111, 1966.

[3] S.-S. Park and M. Hancer, "A comparative study of logit and artificial neural networks in predicting bankruptcy in the hospitality industry," *Tourism Econ.*, vol. 18, no. 2, pp. 311–338, Apr. 2012, doi: 10.5367/te.2012.0113.

[4] M.-Y. Chen, "Bankruptcy prediction in firms with statistical and intelligent techniques and a comparison of evolutionary computation approaches," *Comput. Math. Appl.*, vol. 62, no. 12, pp. 4514–4524, Dec. 2011, doi: 10.1016/j.camwa.2011.10.030.

[5] M. Zavvar, S. Garavand, M. R. Nehi, A. Yanpi, M. Rezaei, and M. H. Zavvar, "Measuring reliability of aspect-oriented software using a combination of artificial neural network and imperialist competitive algorithm," *Asia-Pacific J. Inf. Technol. Multimedia*, vol. 5, no. 2, pp. 75–85, Dec. 2016, doi: 10.17576/apjitm-2016-0502-06.

[6] A. Ansari and A. A. Bakar, "A comparative study of three artificial intelligence techniques: Genetic algorithm, neural network, and fuzzy logic, on scheduling problem," in *Proc. 4th Int. Conf. Artif. Intell. Appl. Eng. Technol.*, Dec. 2014, pp. 31–36, doi: 10.1109/ICAIET.2014.15.

[7] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 2–17, Jan. 2009, doi: 10.1016/j.eswa.2007.10.005.

[8] A.-D. Almási, S. Wo niak, V. Cristea, Y. Leblebici, and T. Engbersen, "Review of advances in neural networks: Neural design technology stack," *Neurocomputing*, vol. 174, pp. 31–41, Jan. 2016, doi: 10.1016/j.neucom.2015.02.092.

[9] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 97–116, Apr. 2017, doi: 10.1016/j.engappai.2017.01.013.

[10] J. Ilonen, J.-K. Kamarainen, and J. Lampinen, "Differential evolution training algorithm for feed-forward neural networks," *Neural Process. Lett.*, vol. 17, no. 1, pp. 93–105, 2003, doi: 10.1023/A:1022995128597.

[11] G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel, and T. Goldstein, "Training neural networks without gradients: A scalable ADMM approach," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1–10. [Online]. Available: http://proceedings.mlr.press/v48/taylor16.pdf

[12] M. H. Tayarani-N and M. R. Akbarzadeh-T, "Magnetic optimization algorithms a new synthesis," in *Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 2659–2664, doi: 10.1109/CEC.2008.4631155.

[13] M.-H. Tayarani-N and M.-R. Akbarzadeh-T, "Magnetic-inspired optimization algorithms: Operators and structures," *Swarm Evol. Comput.*, vol. 19, pp. 82–101, Dec. 2014, doi: 10.1016/j.swevo.2014.06.004.

[14] S. Mirjalili and A. S. Sadiq, "Magnetic optimization algorithm for training multi layer perceptron," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw.*, May 2011, pp. 42–46, doi: 10.1109/ICCSN.2011.6014845.

[15] B. Nakisa, M. Z. A. Nazri, M. N. Rastgoo, and S. Abdullah, "A survey: Particle swarm optimization based algorithms to solve premature convergence problem," *J. Comput. Sci.*, vol. 10, no. 9, pp. 1758–1765, Sep. 2014, doi: 10.3844/jcssp.2014.1758.1765.

[16] V. Beiranvand, M. Mobasher-Kashani, and A. A. Bakar, "Multi-objective PSO algorithm for mining numerical association rules without *a priori* discretization," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4259–4273, Jul. 2014, doi: 10.1016/j.eswa.2013.12.043.

[17] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. MHS. Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, Oct. 1995, pp. 39–43, doi: 10.1109/mhs.1995.494215.

[18] J. Wu, J. Long, and M. Liu, "Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm," *Neurocomputing*, vol. 148, pp. 136–142, Jan. 2015, doi: 10.1016/j.neucom.2012.10.043.

[19] R. Ghasemiyeh, R. Moghdani, and S. S. Sana, "A hybrid artificial neural network with Metaheuristic algorithms for predicting stock price," *Cybern. Syst.*, vol. 48, no. 4, pp. 365–392, May 2017, doi: 10.1080/01969722.2017.1285162.

[20] A. A. Alnaqi, H. Moayedi, A. Shahsavar, and T. K. Nguyen, "Prediction of energetic performance of a building integrated photovoltaic/thermal system thorough artificial neural network and hybrid particle swarm optimization models," *Energy Convers. Manage.*, vol. 183, pp. 137–148, Mar. 2019, doi: 10.1016/j.enconman.2019.01.005.

[21] V. Georgescu, "Using nature-inspired metaheuristics to train predictive machines," *Econ. Comput. Econ. Cybern. Stud. Res.*, vol. 50, no. 2, pp. 1–20, 2016. [Online]. Available: http://www.ipe.ro/RePEc/cys/ecocyb_pdf/ecocyb2_2016p5-24.pdf

[22] P. C. Pendharkar and J. A. Rodger, "An empirical study of impact of crossover operators on the performance of non-binary genetic algorithm based neural approaches for classification," *Comput. Oper. Res.*, vol. 31, no. 4, pp. 481–498, Apr. 2004, doi: 10.1016/S0305-0548(02)00229-0.

[23] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization," *Neural Netw.*, vol. 22, no. 10, pp. 1448–1462, Dec. 2009, doi: 10.1016/j.neunet.2009.05.013.

[24] C. Blum and A. Roli, "Hybrid metaheuristics: An introduction," in *Hybrid Metaheuristics*, C. Blum, M. J. B. Aguilera, A. Roli, and M. Sampels, Eds. Berlin, Germany: Springer, 2008, pp. 1–30.

[25] J. Awwalu, A. A. Bakar, and M. R. Yaakub, "Hybrid N-gram model using Naïve Bayes for classification of political sentiments on Twitter," *Neural Comput. Appl.*, vol. 31, pp. 9207–9220, May 2019, doi: 10.1007/s00521-019-04248-z.

[26] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4135–4151, Sep. 2011, doi: 10.1016/j.asoc.2011.02.032.

[27] J.-R. Zhang, J. Zhang, T.-M. Lok, and M. R. Lyu, "A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training," *Appl. Math. Comput.*, vol. 185, no. 2, pp. 1026–1037, Feb. 2007, doi: 10.1016/j.amc.2006.07.025.

[28] B. Niu and L. Li, "A novel PSO-DE-based hybrid algorithm for global optimization," in *Advanced Intelligent Computing Theories and Applications: With Aspects of Contemporary Intelligent Computing Techniques*. Berlin, Germany: Springer, 2008, pp. 156–163.

[29] X. Lai and M. Zhang, "An efficient ensemble of GA and PSO for real function optimization," in *Proc. 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol.*, Aug. 2009, pp. 651–655, doi: 10.1109/ICCSIT.2009.5234780.

[30] P. PratimSarangi, A. Sahu, and M. Panda, "A hybrid differential evolution and back-propagation algorithm for feedforward neural network training," *Int. J. Comput. Appl.*, vol. 84, no. 14, pp. 1–9, Dec. 2013, doi: 10.5120/14641-2943.

[31] S. Yan, Q. Liu, J. Li, and L. Han, "Heterogeneous acceleration of hybrid PSO-QN algorithm for neural network training," *IEEE Access*, vol. 7, pp. 161499–161509, 2019, doi: 10.1109/ACCESS.2019.2951710.

[32] D. Liang, C.-C. Lu, C.-F. Tsai, and G.-A. Shih, "Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study," *Eur. J. Oper. Res.*, vol. 252, no. 2, pp. 561–572, Jul. 2016, doi: 10.1016/j.ejor.2016.01.012.

[33] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Syst. Appl.*, vol. 83, pp. 405–417, Oct. 2017, doi: 10.1016/j.eswa.2017.04.006.

[34] C.-H. Chou, S.-C. Hsieh, and C.-J. Qiu, "Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction," *Appl. Soft Comput.*, vol. 56, pp. 298–316, Jul. 2017, doi: 10.1016/j.asoc.2017.03.014.

[35] F. Antunes, B. Ribeiro, and F. Pereira, "Probabilistic modeling and visualization for bankruptcy prediction," *Appl. Soft Comput.*, vol. 60, pp. 831–843, Nov. 2017, doi: 10.1016/j.asoc.2017.06.043.

[36] T. Le, M. Lee, J. Park, and S. Baik, "Oversampling techniques for bankruptcy prediction: Novel features from a transaction dataset," *Symmetry*, vol. 10, no. 4, p. 79, Mar. 2018, doi: 10.3390/sym10040079.

[37] T. Le, L. H. Son, M. T. Vo, M. Y. Lee, and S. W. Baik, "A cluster-based boosting algorithm for bankruptcy prediction in a highly imbalanced dataset," *Symmetry*, vol. 10, no. 7, pp. 1–12, 2018, doi: 10.3390/sym10070250.

[38] D. Veganzones and E. Séverin, "An investigation of bankruptcy prediction in imbalanced datasets," *Decis. Support Syst.*, vol. 112, pp. 111–124, Aug. 2018, doi: 10.1016/j.dss.2018.06.011.

[39] F. Mai, S. Tian, C. Lee, and L. Ma, "Deep learning models for bankruptcy prediction using textual disclosures," *Eur. J. Oper. Res.*, vol. 274, no. 2, pp. 743–758, Apr. 2019, doi: 10.1016/j.ejor.2018.10.024.

[40] H. Son, C. Hyun, D. Phan, and H. J. Hwang, "Data analytic approach for bankruptcy prediction," *Expert Syst. Appl.*, vol. 138, Dec. 2019, Art. no. 112816, doi: 10.1016/j.eswa.2019.07.033.

[41] Z. Chen, W. Chen, and Y. Shi, "Ensemble learning with label proportions for bankruptcy prediction," *Expert Syst. Appl.*, vol. 146, May 2020, Art. no. 113155, doi: 10.1016/j.eswa.2019.113155.

[42] M. M. Kashani and M. Ayob, "Reduction operators for magnetic optimization algorithm," *J. Appl. Sci.*, vol. 14, no. 24, pp. 3446–3454, Dec. 2014, doi: 10.3923/jas.2014.3446.3454.

[43] A. F. Atiya, "Bankruptcy prediction for credit risk using neural networks: A survey and new results," *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 929–935, Jul. 2001, doi: 10.1109/72.935101.

[44] V. G. Gudise and G. K. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2003, pp. 110–117, doi: 10.1109/SIS.2003.1202255.

ABDOLLAH ANSARI received the M.Sc. degree in artificial intelligence from the Universiti Kebangsaan Malaysia, Malaysia, in 2013. He is currently the Chairman of the Board of the Institute of Intelligent Nano and Informatics (INI), and the Board of Gereh (node) Communications Development Company, Tehran, Iran. He has published many journal articles and attended many conferences.

IBRAHIM SAID AHMAD received the B.Sc. degree in computer science from Bayero University Kano, Nigeria, in 2011, the M.Sc. degree in information technology from the University of Nottingham, U.K., in 2014, and the Ph.D. degree from the Universiti Kebangsaan Malaysia, in 2020. He is currently a Lecturer with the Department of Information Technology, Bayero University Kano. He has published several journal articles and attended many conferences. His main research interests include data analytics and artificial intelligence specifically in business intelligence and computational intelligence.

AZURALIZA ABU BAKAR received the Ph.D. degree in artificial intelligence from the Universiti Putra Malaysia, in 2002. She was an Advisor of the Data Mining and Optimization Laboratory and a member of the Sentiment Analysis Laboratory. Since 2011, she has been a Professor of data mining with the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia. She has led 13 research projects, including three in progress and a member of 42 research projects. Her main research interests include data analytics and artificial intelligence specifically in rough set theory, feature selection algorithms, nature inspired computing, and sentiment analysis. She is a member of the IEEE Computational Intelligence Society. She has also served on roughly thirty conference and workshop program committees. She served as the Program Chair.

MOHD RIDZWAN YAAKUB received the Ph.D. degree from the Queensland University of Technology, Australia, in 2015. He is currently a Senior Lecturer with the Universiti Kebangsaan Malaysia. He is also the Head of the Sentiment Analysis Laboratory, Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia. He has led several research projects, supervised M.Sc. Students and co-supervised Ph.D. Students. He has published several journal articles and attended several referred conferences. His research interests include databases, data mining, and artificial intelligence.