

## An efficient quantum multiverse optimization algorithm for solving optimization problems

Samira Sarvari, Nor Fazlida Mohd. Sani, Zurina Mohd Hanapi, Mohd Taufik Abdullah

Department of Computer Science, Faculty of Computer Science and Information Technology,  
Universiti Putra Malaysia, Malaysia

---

### Article Info

#### Article history:

Received Jun 29, 2019

Revised Nov 2, 2019

Accepted Dec 1, 2019

---

#### Keywords:

Intrusion Detection System

Multiverse Optimization

Quantum Computing

Quantum Multiverse

Optimization

---

### ABSTRACT

Due to the recent trend of technologies to use the network-based systems, detecting them from threats become a crucial issue. Detecting unknown or modified attacks is one of the recent challenges in the field of intrusion detection system (IDS). In this research, a new algorithm called quantum multiverse optimization (QMVO) is investigated and combined with an artificial neural network (ANN) to develop advanced detection approaches for an IDS. QMVO algorithm depends on adopting a quantum representation of the quantum interference and operators in the multiverse optimization to obtain the optimal solution. The QMVO algorithm determining the neural network weights based on the kernel function, which can improve the accuracy and then optimize the training part of the artificial neural network. It is demonstrated 99.98% accuracy with experimental results that the proposed QMVO is significantly improved optimization compared with multiverse optimizer (MVO) algorithms.

*This is an open access article under the [CC BY-SA](#) license.*



---

### Corresponding Author:

Samira Sarvari,

Department of Computer Science, Faculty of Computer Science and Information Technology,

Universiti Putra Malaysia,

Serdang, 43400 Seri Kembangan, Selangor, Malaysia.

Email: samirasarvari82@yahoo.com

---

## 1. INTRODUCTION

The increase in the number of local networks has led to the continuous development of Internet data and the availability of massive amounts of network data has promoted the development of information technology, which requires careful attention. As a result, this evolution, in turn, has increased the system's vulnerability to various threats [1]. Any intrusion can have catastrophic consequences. For example, personal data may be destroyed, corrupted or illegally accessed as a result of breaches of confidentiality. In addition, infringements of integrity can lead to alteration of personal data. Computer network security has become a promising tool for secure channels. One of the promising tools for detecting attacks is the intrusion detection system (IDS). Cybersecurity infrastructures use IDS as an essential component and protect systems and infrastructures against various threats. An intrusion detection system consists of data collection, data clearing and pre-processing, intrusion detection, reporting and reasonable action, which is an essential part of these attack detection processes [2]. High classification accuracy and a low false alarm rate are the two main characteristics of well-developed IDS, so it is extremely important to develop mechanisms for intrusion detection in view of the conviction that suspicious activities can be detected by taking measures to prevent further breeding of computer networks or systems [3].

Data classification has been studied extensively in many computer fields and up to now, the development of classification has achieved great achievements and many types of classified technology

and theory will continue to emerge. In the face of a lot of noisy, cluttered, nonlinear data, artificial neural network (ANN) not only helps to make high - quality modeling and complete training in the process of using large amounts of data, but also has a test mode set to evaluate the performance of ANN.

ANNs are a form of machine learning algorithm inspired by the behavior of biological neurons in the brain and central nervous system, and use mathematical models to describe the architecture of biological neural networks to solve information processing problems. The ANN model comprises three layers: the input layer, the hidden layers and the output layer. The weights connecting the input layer to the hidden layer, as well as the bias values of the hidden layer, are randomly generated before the learning process. Only the weights connecting the hidden layer to the output layer are trained by the fast-linear regression. An example of a simple ANN with a single hidden layer is shown in Figure 1, where "I" is the input of the neural network and "W" is the weight given to each input.

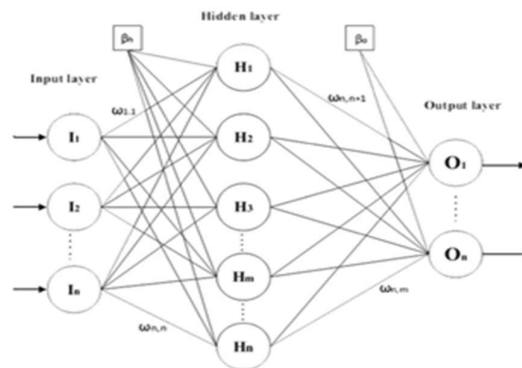


Figure 1. Simple ANN with a single hidden layer

Recently, researchers from all over the world have been improving the ANN according to different forecasting tasks and have obtained some satisfying results [4]. Nevertheless, the gradient-based learning algorithms are widely used to train traditional ANNs, which may result in some drawbacks such as the slow convergence speed, the local minimum, and the overfitting phenomenon. In order to solve the aforementioned problems, we focus our research on an improved machine learning algorithm based on neural networks with random weights and kernels (KNNRW). Recently, neural networks with random weights and kernels (KNNRW) [5] has been proposed by replacing the hidden nodes mapping with the kernel mapping. It does not need to determine the number of hidden nodes of KNNRW.

In order to solve the above problems, we focus our research on an improved neural network with random weights and kernels (KNNRW) and proposed quantum multiverse optimization (QMVO) algorithm. In this research, the neural network algorithm is first improved by determining the neural network weights based on the kernel function, and then optimize the training part of the artificial neural network to develop advanced detection approach for IDS.

## 2. BASICS AND BACKGROUND

In the last years, bio-inspired computing has witnessed advances, popularity, and interest in different areas of sciences and engineering. However, there are still some problems remaining unsolved. Some of these are classifications methodology, parameters tuning, the gap between theoretical and practical parts, large-scale real-world applications, and finally the selection of the appropriate algorithm for specific problem [6]. The motivation of the hybrid algorithms to solve the optimization problem is based on No Free Lunch (NFL) theorem [7]. According to NFL theorem, no algorithm is able to solve all the optimization problems. Therefore, this topic of research is open until now. Thus, researchers make many efforts to improve the current optimization algorithms to solve different complex problems. Some of them used the benefits of quantum computing (QC) to solve these problems such as the speed, efficiency, and performance of evolutionary algorithms. QC is a new emerging mechanism in computer science and engineering and other disciplines. QC is a branch of mathematics that uses the specificities of quantum mechanics for data transformation and information processing, which stored in a two-state quantum bits or qubit. It has gained the interest of researchers in the last years in the fields such as quantum algorithms and quantum computers.

QC depends on some principles of quantum mechanics in which the smallest information unit is called the quantum bit or qubit. QC uses 0 and 1 for representing the two basic states. The main difference between bit and qubit is that the qubit can be in a state between 0 and one not only in a state of 0 or 1 as in the classical bit. In QC, the processing of the enormous number of quantum states is implemented in a parallel way simultaneously [8]. Therefore, many types of research have studied the theoretical and practical studies to merge quantum computing and evolutionary computation [9]. Some of these are quantum genetic algorithms [10], quantum inspired scatter search [9, 11], quantum differential algorithm [12], mining large databases [13], 0–1 optimization problem [14], knapsack problem [15], traveling salesman problem [16], engineering inverse problem [17] and other areas of applications as in Gottfried and Yan [18].

Currently, quantum inspired algorithms are used to solve many combinatorial optimization problems as proposed [19]. There are many hybridized quantum evolutionary algorithms proposed in the literature such as quantum inspired evolutionary algorithms [10], quantum inspired immune algorithm [20], quantum PSO [21]. Sun et al. [21], quantum inspired PSO employed a probability searching technique, and the search space is transferred from classical space to quantum, where the particles' movement is similar to the ones with the quantum mechanics [22]. Quantum inspired evolutionary algorithms were introduced to solve the traveling salesman problem [23], where the crossover operation was performed based on the interference concept. Multiverse optimizer (MVO) is one of the bio-inspired algorithms [24]. The main inspiration of MVO is taken from multiverse theory in physics. It is based on three main concepts in cosmology. These concepts are a wormhole, white hole, and black hole. Like other evolutionary algorithms, MVO starts the optimization process by creating a population of solutions. In fact, this algorithm mimics the interaction between multiple universes through the wormhole, black hole, and white hole. The core idea of MVO came from the fact that larger universes tend to send objectives to smaller universes to reach a stable position. MVO has been used to solve many optimization problems. Nineteen unimodal/multimodal benchmark functions have been adopted to evaluate the performance of MVO [24]. Fariset et al. [25] used MVO to select the optimal feature subset. Moreover, they used it to optimize SVM parameters. Han and Kim [10], employed the MVO for training the multilayer perceptions neural network. Their approach was evaluated and benchmarked using nine different biomedical datasets selected from the UCI machine learning repository. To assess the performance of this algorithm, the obtained results are compared with five recent evolutionary meta-heuristic algorithms: particles warm optimization (PSO), genetic algorithm (GA), differential evolution (DE) algorithm, and cuckoo search (CS). These studies have been revealed that MVO algorithm is more efficient than the other algorithms, and it can obtain better results. However, MVO algorithm like most of the optimization algorithms suffers from low exploitation and convergence rate. Zouache and Moussaoui [15] proposed a hybrid algorithm based on using particles warm optimization (PSO) and multiverse optimization (MVO). They combine the exploitation capability of PSO and the exploration capabilities of MVO. The experimental results validate its effectiveness compared to standard PSO and MVO. There is always room for improvement and adaptation of an algorithm to solve a particular set of problems.

### 3. MATHEMATICAL MODELING

Neural networks with weights and kernels (KNNRW) have been proposed by introducing the kernel function mapping of SVM as the hidden node mapping of NNRW. The optimization problem of NNRW can be written as (1)

$$\min L_{P_{NNRW}} = \frac{1}{2} \|w\|^2 + \frac{C}{2} \sum_{i=1}^N \|\xi_i\|^2 \quad (1)$$

$$h(x_i) \cdot w = t_i - \xi_i, i = 1, \dots, N$$

Where  $\xi_i$  is the training error related to the  $i$ th training sample  $x_i$ ,  $C$  is the regularization coefficient, and  $h(x_i)$  denotes the  $i$ th row of  $H$ . The corresponding dual optimization problem of (1) can be formulated as (2)

$$L_{D_{NNRW}} = \frac{1}{2} \|w\|^2 + \frac{C}{2} \sum_{i=1}^N \|\xi_i\|^2 - \sum_{i=1}^N \alpha_i (h(x_i) \cdot w - t_i + \xi_i) \quad (2)$$

Where  $\alpha_i$  is the langrage multiplier with respect to the  $i$ th training sample  $x_i$ . The corresponding Karush-Kuhn-Tucker (KKT) conditions are as follows

$$\frac{\partial L_{D_{NNRW}}}{\partial w} = 0 \rightarrow w = \sum_{i=1}^N \alpha_i h(x_i)^T \rightarrow w = H^T \alpha \quad (3)$$

$$\frac{\partial L_{DNNRW}}{\partial \xi_i} = 0 \rightarrow \alpha_i = C \xi_i, i = 1, \dots, N \quad (4)$$

$$\frac{\partial L_{DNNRW}}{\partial x_i} = 0 \rightarrow h(x_i) \cdot w - t_i + \xi_i = 0, i = 1, \dots, N \quad (5)$$

Substituting (3) and (4) into (5), the following equation can be obtained

$$\left(\frac{I}{C} + HH^T\right) \alpha = T \quad (6)$$

Where  $I$  is an identity matrix.

Considering (3) and (6), the weight  $w$  can be calculated as

$$w = H^T \left(\frac{I}{C} + HH^T\right)^{-1} T \quad (7)$$

Thus, the output function of NNRW can be written as

$$f(x) = h(x)H^T \left(\frac{I}{C} + HH^T\right)^{-1} T \quad (8)$$

It can be seen from (8) that the specific form of  $h(x)$  is not important as long as the dot product of  $HH^T$  (or  $h(x)H^T$ ) is known. As a result, if the hidden node mapping  $h(x)$  is unknown, we can define the kernel matrix of KNNRW as follows

$$\begin{aligned} \Omega_{NNRW} &= HH^T : \\ \Omega_{NNRW_{ij}} &= h(x_i) \cdot h(x_j) = K(x_i, x_j) \end{aligned} \quad (9)$$

Consequently, the output function can be rewritten accordingly as

$$f(x) = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \left(\frac{I}{C} + \Omega_{NNRW}\right)^{-1} T \quad (10)$$

In order to further improvement of accuracy, we adjust and update the weights in ANN using the recently proposed Quantum multiverse optimization (QMVO) algorithm. In QMVO algorithm, each universe has a state depicted by wave function  $\psi(y, t)$  with probability density function of universe position  $|\psi(y, t)|^2$ . In quantum, the dynamic behavior of universe differs from the standard version of MVO. The updating position is mathematically formulated as follows

$$V(y) = -\lambda \delta(y) \quad (11)$$

Where  $\delta(y)$  is the Dirac delta function and  $k$  is a positive value.

The universe wave function in delta potential is defined as follows:

$$\psi(y) = \sqrt{\frac{m\lambda}{h^2}} \left( e^{-\frac{m\lambda|y|}{h^2}} \right) \quad (12)$$

Where  $m$  denotes the universe's mass and  $h$  denotes the reduced plank constant.

The particles (universes) are defined by the following (13)

$$y_i(t+1) = \begin{cases} \alpha + \beta \cdot |Mbest_i - y_i(t)| \cdot \ln(1/u) & \text{if } l \geq 0.5 \\ \alpha - \beta \cdot |Mbest_i - y_i(t)| \cdot \ln(1/u) & \text{if } l < 0.5 \end{cases} \quad (13)$$

Where  $u$  and  $l$  are randomly initialized in interval  $[0, 1]$ ,  $\beta$  denotes the contraction–expansion coefficient. It is linearly decreased over iteration. The mathematical formula is defined as follows

$$\beta = \frac{0.5 \times (Max_{iter} - iter)}{iter + 0.5} \quad (14)$$

Where  $Max_{iter}$  defined as the maximum number of iterations and  $iter$  denotes the current iteration number.  $Mbest$  is defined as the population's mean best or global point. It is mathematically defined as follows

$$Mbest = \frac{1}{M} \sum_{dim=1}^{MaxDim} \alpha_{g,dim}(t) \quad (15)$$

Where  $M$  is the number of universes (population size),  $g$  is the best universe  $\alpha$  index among all universes in the search space,  $dim$  denotes the universe dimension and  $MaxDim$  is the maximum number of dimensions. Through this, the local attractor in order to guarantee the convergence speed of the QMVO is defined as follows

$$\alpha = \frac{r_1 \alpha_{i,dim} + r_2 \alpha_{g,dim}}{r_1 + r_2} \quad (16)$$

Where  $r_1$  and  $r_2$  are the random numbers in range  $[0, 1]$ ,  $x_{i,dim}$  is the  $i$ th universe index in  $dim$ th dimensions of the hyperspace and  $x_{g,dim}$  denotes the  $g$ th best universe position index of  $dim$ th dimensions. The whole algorithm steps of QMVO are presented at algorithm.

#### Quantum Multi Verse Optimization Algorithm (QMVO)

- 1: Randomly set the initial values of the universes's positions (population size)  $M$ , dimensions  $MaxDim$ , lowerlbp and upper ubp boundaries, the maximum number of iterations  $Max_{iter}$  and the best universe  $\alpha_g$ .
- 2:  $SU$ =Sorted universes
- 3: Normalize the fitness value (the inflation rate) of the universes  $NI$ .
- 4: Set  $iter$ : = 1. {Counter initialization}.
- 5: **repeat**
- 6: Calculate the fitness value (the inflation rate) of the universes.
- 7: Check if the new universes positions go out of the search space boundaries and bring it back.
- 8: **for** ( $i=1$ :  $i \leq M$ ) **do**
- 9: Update  $Mbest$  and  $\beta$  using (14) and (15)
- 10: Set black hole index to  $i$ .
- 11: **for** ( $j=1$ :  $i \leq$  size of the universe position) **do**=
- 12:  $R_1$ = random ([0, 1])
- 13: **if**  $R_1 \leq NI (y_i)$  **then**
- 14: White hole index = Roulette Wheel Selection(-NI);
- 15:  $U$  (black hole index,  $j$ ) =SU (white hole index,  $j$ )
- 16: **end if**
- 17:  $R_2$ = random ([0, 1]);
- 18: **if**  $R_2 <$  Wormhole probability existence **then**
- 19:  $l$  = random ([0, 1]);
- 20:  $u$  = random ([0, 1]);
- 21: **if**  $l < 0.5$  **then**
- 22:  $y_i(t+1) = \alpha - \beta \cdot |Mbest_i - y_i(t)| \cdot \ln(1/u)$ ;
- 23: **else**
- 24:  $y_i(t+1) = \alpha + \beta \cdot |Mbest_i - y_i(t)| \cdot \ln(1/u)$ ;
- 25: **end if**
- 26: **end if**
- 27: **end for**
- 28: **end for**
- 29: Set  $iter = iter + 1$ . {Iteration counter increasing}
- 30: **until** ( $iter < Max_{iter}$ ) . {Termination criteria satisfied}.
- 31: Produce the best universe.

The training process is an important phase for the optimization of the ANN. In the ANN training, each individual represents all of the weights and biases of the ANN structure. The objectives of this process are to search for the synaptic weights of the ANN and to reduce the MSE, which represents the cost function of the ANN and achieve the highest classification and prediction accuracy. The training of the neural network is determined by the QMVO algorithm based on the weights and the error rate obtained is based on the following pseudo-code.

```

1: initialize all universe as input parameter
2: for i=1 until count of universe
1: calculate universei MSE
3: if universei MSE are the minimum vale in all MSE universe
4: neural network weight s is universei
5: best universe is universei
6: endif
7: endfor
8: output bet universe is weight neural network

```

Based on this pseudo-code, all worlds are considered as inputs of the system first, then the values of these worlds are considered as weights of the neural network. The MSE value or the error of each neural network is calculated according to the weights of each universe. Accordingly, the world with the lowest MSE rate is considered as the weights of the final neural network. In this way, the neural network is trained at each stage of the QMVO algorithm.

#### 4. RESULTS AND DISCUSSION

The proposed model was implemented and evaluated in MATLAB. In the QMVO+ANN training using the NSL-KDD dataset with number of iterations=100. The IDS was evaluated on several factors. The main factors included in this research are detection rate (DR), false alarm rate (FAR), and accuracy (ACC). In the literature, most of research works in the field of intrusion detection focused on the accuracy, the detection rate (DR), and false alarm rate (FAR). In this research, we have adopted the same metrics to evaluate the performance of our proposed approach. The experimental results in Figure 2 provides an evaluation of the performance of the ANN intrusion detection for the QMVO algorithm with 97.48 DR, 0.03 FAR, and 99.98 ACC. The results show the potential applicability of QMVO with ANN for developing practical IDSs.

Further-more, the comparisons of the performance results of the proposed ANN+QMVO and another model ANN+MVO dataset are shown in Figure 3. The proposed ANN+QMVO clearly performs the best in terms of ACC and DR. The data correctly classified by the proposed ANN+QMVO are more than correctly classified by ANN+MVO.

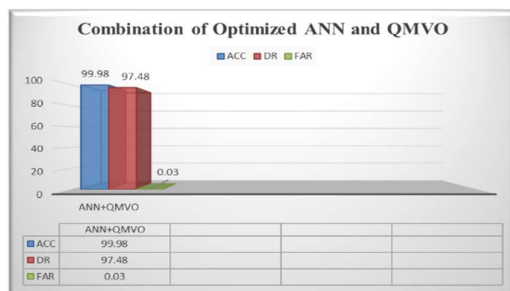


Figure 2. Combination of ANN and QMVO

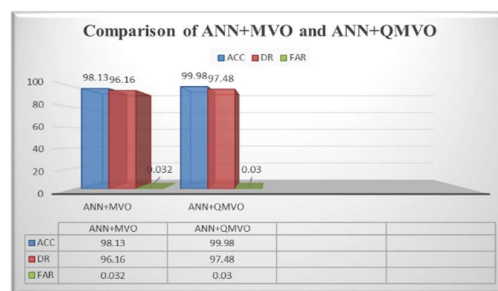


Figure 3. Comparison of ANN+MVO and ANN+QMVO

#### 5. CONCLUSION

There are various techniques of Artificial Neural Network, which can be applied to intrusion detection system. Each technique is suitable for some specific situation. QMVO is easy to implement, supervised learning artificial neural network. Number of the epochs required to train the network is high

as compare to the other ANN techniques. This research presents a new global optimization algorithm called quantum multiverse optimization (QMVO) with ANN and quantum behavior for solving the optimization problems. Therefore, the proposed method offers the best strong exploration and precise exploitation capabilities.

In this research, the proposed QMVO algorithm was only implemented for solving optimization problems. Thus, our future work will concentrate on implementing the QMVO in (i) solving more complex optimization problems with different properties and (ii) design conceptions for engineering, practical applications, and constrained problems.

## REFERENCES

- [1] I. Ahmad, A. B. Abdullah, and A. S. Alghamdi, "Application of artificial neural network in detection of probing attacks," in *Industrial Electronics & Applications, 2009. ISIEA 2009. IEEE Symposium*, pp. 557-562, 2009.
- [2] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Applied soft computing*, vol. 10, pp. 1-35, 2010.
- [3] S. M. H. Bamakan, B. Amiri, M. Mirzabagheri, and Y. Shi, "A new intrusion detection approach using PSO based multiple criteria linear programming," *Procedia Computer Science*, vol. 55, pp. 231-237, 2015.
- [4] Z. He, Q. Hu, Y. Zi, Z. Zhang, and X. Chen, "Hybrid intelligent forecasting model based on empirical mode decomposition, support vector regression and adaptive linear neural network," in *International Conference on Natural Computation*, pp. 324-327, 2005.
- [5] G.-B. Huang, "An insight into extreme learning machines: random neurons, random features and kernels," *Cognitive Computation*, vol. 6, pp. 376-390, 2014.
- [6] X.-S. Yang and M. Karamanoglu, "Swarm intelligence and bio-inspired computation: an overview," in *Swarm Intelligence and Bio-Inspired Computation*, ed: Elsevier, pp. 3-23, 2013.
- [7] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, pp. 67-82, 1997.
- [8] A. Montanaro, "Quantum algorithms: an overview," *npj Quantum Information*, vol. 2, p. 15023, 2016.
- [9] A. Layeb and D. E. Saidouni, "A new quantum evolutionary local search algorithm for MAX 3-SAT problem," in *International Workshop on Hybrid Artificial Intelligence Systems*, pp. 172-179, 2008.
- [10] K. H. Han and J. H. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, h-epsilon gate, and two-phase scheme," *IEEE transactions on evolutionary computation*, vol. 8, pp. 156-169, 2004.
- [11] A. Layeb, "Hybrid quantum scatter search algorithm for combinatorial optimization problems," *Journal of Annals. Computer Science Series*, vol. 8, pp. 227-244, 2010.
- [12] A. Draa, S. Meshoul, H. Talbi, and M. Batouche, "A quantum-inspired differential evolution algorithm for solving the N-queens problem," *Neural networks*, vol. 1, 2011.
- [13] M. Ykhlef, "A quantum swarm evolutionary algorithm for mining association rules in large databases," *Journal of King Saud University-Computer and Information Sciences*, vol. 23, pp. 1-6, 2011.
- [14] A. Layeb, "A hybrid quantum inspired harmony search algorithm for 0-1 optimization problems," *Journal of Computational and Applied Mathematics*, vol. 253, pp. 14-25, 2013.
- [15] D. Zouache and A. Moussaoui, "Quantum-Inspired Differential Evolution with Particle Swarm Optimization for Knapsack Problem," *J. Inf. Sci. Eng.*, vol. 31, pp. 1757-1773, 2015.
- [16] B. A. L. d. M. Herrera, L. d. S. Coelho, and M. T. A. Steiner, "Quantum inspired particle swarm combined with Lin-Kernighan-Helsgaun method to the traveling salesman problem," *Pesquisa Operacional*, vol. 35, pp. 465-488, 2015.
- [17] O. U. Rehman, S. Yang, and S. U. Khan, "A modified quantum-based particle swarm optimization for engineering inverse problem," *COMPEL-The International Journal For Computation And Mathematics In Electrical And Electronic Engineering*, vol. 36, pp. 168-187, 2017.
- [18] K. Gottfried and T.-M. Yan, *Quantum mechanics: fundamentals*: Springer Science & Business Media, 2013.
- [19] A. Layeb and D.-E. Saidouni, "A new quantum evolutionary algorithm with sifting strategy for binary decision diagram ordering problem," *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, vol. 4, pp. 47-61, 2010.
- [20] L. Jiao, Y. Li, M. Gong, and X. Zhang, "Quantum-inspired immune clonal algorithm for global optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, pp. 1234-1253, 2008.
- [21] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Evolutionary Computation, 2004. CEC2004. Congress*, pp. 325-331, 2004.
- [22] K.-L. Du and M. Swamy, "Search and optimization by metaheuristics," *Birkhäuser*, Jul 2016.
- [23] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Evolutionary Computation, 1996 Proceedings of IEEE International Conference*, pp. 61-66, 1996.
- [24] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, pp. 495-513, 2016.
- [25] H. Faris, M. A. Hassonah, A.-Z. Ala'M, S. Mirjalili, and I. Aljarah, "A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture," *Neural Computing and Applications*, vol. 30, pp. 2355-2369, 2018.