A Quantum Beta Distributed Multi-Objective Particle Swarm Optimization Algorithm for Twitter Fake Accounts Detection

Ahlem Aboud [a, b, *], Nizar Rokbani [b,c], Seyedali Mirjalili [d,i], Abdulrahman M. Qahtani [e], Fahd S. Alharithi [e], Omar Almutiryh [f], Habib Dhahri [f], Amir Hussain [g] and Adel M. Alimi [b, h]

[a] University of Sousse, ISITCom, 4011, Sousse, Tunisia.

[b] REGIM Lab: REsearch Groups in Intelligent Machines, University of Sfax, National Engineering School of Sfax (ENIS), BP 1173, Sfax, 3038, Tunisia.

[c] High Institute of Applied Science and technolgy of Sousse, University of Sousse, Tunisia.

[d] Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane, Australia.

[e] Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box. 11099, Taif 21944, Saudi Arabia.

[f] College of Applied Computer Science, King Saud University, Riyadh, Saudi Arabia.

[g] Edinburgh Napier University, School of Computing, Edinburgh EH10 5DT, Scotland, U.K.

[h] Department of Electrical and Electronic Engineering Science, Faculty of Engineering and the Built Environment, University of Johannesburg, South Africa.

[i] Yonsei Frontier Lab, Yonsei University, Seoul, South Korea.

----

[*] Corresponding Author

E-mail Address: ahlem.aboud@regim.usf.tn (A. Aboud)

Contributing Authors: E-mail addresses: nizar.rokbani@ieee.org (N. Rokbani), ali.mirjalili@gmail.com (S. Mirjalili), amqahtani@tu.edu.sa (A. M. Qahtani), f.alshalawi@tu.edu.sa (F. S. Alharithi), oalmutiry@ksu.edu.sa (O. Almutiry), hdhahri@ksu.edu.sa (H. Dhahri), a.hussain@napier.ac.uk (A. Hussain), adel.alimi@ieee.org (A. M. Alimi).

Abstract

Fake account detection is a topical issue when many Online Social Networks (OSNs) encounter problems caused by a growing number of unethical online social activities. This study presents a new Quantum Beta-Distributed Multi-Objective Particle Swarm Optimization (QBD-MOPSO) system to detect fake accounts on Twitter. The proposed system aims to minimize two objective functions simultaneously: specifically features dimensionality and classification error rate. The QBD-MOPSO has two optimization profiles: the first uses a quantum behaved equation for improving the exploratory behaviour of PSO, while the second uses a beta function to enhance PSO's exploitation. Six variants of the QBD-MOPSO approach are proposed to account for various data distribution types. The QBD-MOPSO system provides a feature selection technique based on the sigmoid function for position binary encoding. Each particle has a binary vector as a potential solution for feature subset selection, and a bit with the value of "1" indicating selection of a feature and "0" otherwise. Machine learning based classification models are trained and tested using a subset of selected features. An extensive experimental study is carried using two benchmark Twitter datasets with 1982 and 928 accounts. From 46 original features, QBD-MOPSO has selected 32 and 25 pertinent features and accurately classified 99.19% and 97.52% account on the datasets.

Keywords: Beta Function, Feature Selection, Fake Account Detection, Quantum Beta Multi-Objective Particle Swarm Optimization, Machine Learning, Distributed System, Quantum Computing.

1. Introduction

Online Social Networks have become a crucial part of daily life. Social media and mobile devices are driving the growth of the World Wide Web as well. According to the digital report[1] published in January 2020, out of 7.75 billion people worldwide, there are: 5.19 billion phone users, 4.54 billion Internet users, 3.8 billion active social media users, and 3.75 billion mobile social media users. The world's internet users spending an average of six hours online each day. In the last decade, a large number of users have become addicted to the use of well-known online social networks like Facebook, Twitter, and Instagram. There is not only an addiction to good habits such as communication and sharing information, but a substantial part of the users in OSNs are not humans but fake or bot accounts controlled by a computer to gain popularity and promote business activities for financial gain.

At the starting period of the last decade several techniques were developed to manage the user profiling problem. First, user profiling has been presented as the process of capturing information about users and their interests which is called the User Data Discovery (UDD) model refereed to the Knowledge Data Discovery (KDD) model. In this context, several approaches have been developed, and regrouped into three categories: explicit, implicit and hybrid user profiling techniques while the main issues covered the process of information retrieval and collection of the user's information [1]. Implicit user profiling approaches are referred to static or factual profiling that provides the static process to analyse and collect static and predictable characteristics about users by filing some online forms.

However, explicit approaches are referred to behavioral, adaptive and ontological profiling that leads to the dynamic process of collecting future behaviors and learn about users. This is done using several filtering techniques [2] such as

[1] https://wearesocial.com/fr/blog/2020/01/digital-report-2020

Rule based filtering, Collaborative filtering and content-based filtering. Furthermore, hybrid approaches have combined the advantages of both explicit and implicit methods taking into consideration the static and dynamic characteristics of the user profile to maintain the accuracy of temporal information.

During the last five years, a variety of approaches have been developed to manage user profiling problem not only with regards to data discovery but also for unhealthy activities detection for spam/ non-spam accounts [3], fake or bot accounts [4], fake followers [5], fake news [6], and fake engagement [7] using different Machine Learning Algorithms (MLAs) for classification purposes. Generally speaking, the classification task involves five main steps: data collection, feature extraction, feature selection, classification and prediction. The feature selection step has been considered as a challenging problem for classification, clustering, time series prediction and regression tasks. This study focuses mainly on the feature selection problem in the classification task. Feature Selection (FS), is also known as dimensionality reduction technique [8], which is defined as the process to select a small subset of features to enhance the performance of machine learning models with best accuracy, interpretability, and to minimise the computational time [9]. Figure 1, presents a classification of the existing feature selection approaches. According to data labels, there are three categories including; supervised, unsupervised and semi-supervised techniques. The main difference between categories has been caused respectively by the presence, the absence and the existence of a small portion of labelled data [8]. The input labelled data makes supervised methods more specific for the decision making. Compared with unsupervised methods, the supervised approaches have produced a high accuracy, but the human intervention for the supervised learning need a high computational cost, and cannot be useful for the real-time data. Furthermore, the input unlabelled data makes a low complexity for unsupervised methods, and the labels are determined automatically by the machine which are very useful for real-time data.

However, three categories are considered according to the search strategy including; filter, wrapper and embedded methods. There have been both advantages and disadvantages to feature selection methods, depending on factors such as computational cost, speed, the dimension of the data, criteria for selecting features, and machine learning algorithms. Filter methods have been characterised by a high speed of treatment, a low computational cost, and well designed for a high dimensional data, however the use of statistical criterion does not guarantee the best subset of selected features [10]. The wrapper method has included a learning algorithm to determine the accuracy of the selected features [11], and to guarantee a better result compared with filter methods, but it was not performed with high dimensional data. Embedded techniques were the hybridization of both filter and wrapper methods. As filter methods, a statistical criterion is used for features dimensionality reduction, and as the wrapper method, the learning algorithm is used to determine the best subset of features leading to the high classification accuracy [11].

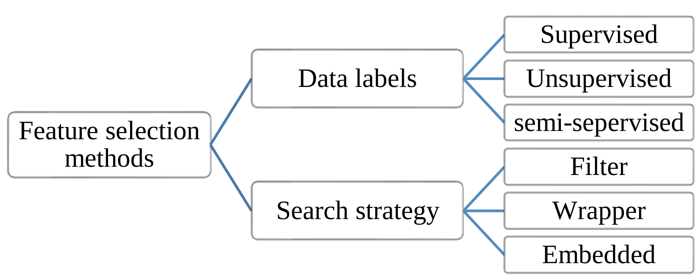


Figure 1. Classification of feature selection methods

User profiles on OSN include personal information, shared content, links, and social interaction relationships. Many online contents are shared to attract a large number of users. Credible content is one of the most basic criteria for trusting web users. However, due to the widespread availability and usability of several features, many automatic programs (aka. bot accounts) are developed simulating human behavior. Such accounts are considered fake and impose malicious activities on social media platforms [4]. They are automatically controlled by a computer system to spread harmful activities [12]. The absence of a picture profile or online activities is not a very reliable method for detecting fake accounts. Therefore, it is very difficult to distinguish between bots and human accounts on OSNs. This has made fake account detection a challenging problem that has attracted several researchers.

Thus, Twitter online social networks have attracted many researchers due to the severity of Twitter social spambots problem [13] and the public availability of datasets which were easier to find than those of other OSNs. In this study, we aim to concentrate on Twitter fake accounts detection using machine learning algorithms for the classification task. The proposed QBD-MOPSO algorithm aims to examine the process of selecting relevant features for obtaining a high classification rate. The optimization process of QBD-MOPSO is done based on the Revised Quantum-behaved Particle Swarm Optimization (RQPSO) [14], and the Gaussian Quantum-behaved Particle Swarm Optimization algorithm (GAQPSO) [15] as well as the use of beta function provided by Alimi [16] in 2003. QBD-MOPSO algorithm presents two optimization profiles. In the first profile, all particles are subjected to quantum PSO approaches (RQPSO and GAQPSO) with random uniform and Gaussian distributions to better explore the search pace respectively. The second profile is for exploitation enhancement using a beta function with three data distribution shapes namely; Gaussian, linear decrease and exponential. In more details, QBD-MOPSO starts with a random initialisation of N particles. Each particle is a potential solution optimized in the search space. The dynamic switching phases are assumed by the two optimization profiles which are symmetric about the mean personal best position ($mbest$). A particle P is optimized for exploration phase, if the current position (X) is less than the mean best ($mbest$) position. Otherwise, it was considered for the exploitation phase. At each iteration, the particle position is updated as follows:

- Exploration phase: particle position is updated using the quantum equation in (RQPSO and GAQPSO).
- Exploitation phase: particle position is updated using the beta function.

The application of the QBD-MOPSO method for identifying fake accounts is denoted by the Neuro-QBD-MOPSO method. To select features, a primordial step is added to the QBD-MOPSO algorithm and named position binary encoding based on the sigmoid function. Only bits with the value of “1” are considered as selected features and used to train and test the classification model. Last but not least, one compromise solution is chosen to determine the subset of pertinent features, and determined using the nearest non-dominated solution to the utopian point.

The rest of this paper is resumed as follows: Section 2 presents an overview of the existing fake accounts detection approaches-based on features selection. Section 3, details the proposed Quantum Beta Distributed Multi-Objective Particle Swarm Optimization (QBD-MOPSO) system. Section 4, details the Neuro-QBD-MOPSO architecture for fake account detection. The preliminary of the experimental study and the comparative results are discussed in Section 5. Finally, Section 6 concludes the paper and suggests future work.

2. Overviews of Fake Account Detection Approaches-based Feature Selection Techniques

On OSNs, several users aim to gain popularity not only by sharing healthy information about a specific domain of interest but also by introducing malicious activities, such as posting fake links and news. In 2016, the annual web traffic report[2] stated that more than 16.7 billion web visits to 100,000 randomly-selected web sites had been analysed and detected more than 51.8% of bot users. In 2018, the industry report[3] announced 42.2% of all internet traffic wasn't human. Nevertheless, the increase of fake accounts generation has been attributed to several extreme situations, including elections, Black Friday, the COVID-19 pandemic and many other national or international events, activities, and diseases. A high numbers of user profiling techniques have been developed, and were aimed to address different issues on OSN like user interest detection, sentiment analysis, spam detection and fake account detection. This study addresses the problem of identifying fake Twitter accounts as a step toward fake news detection.

As can be seen in Table 1, there are a wide range of techniques in the literature developed for feature selection to determine the most effective characteristic of a fake user. In 2020, Rostami and Karbasi [17] used the Minimum Redundancy –Maximum Relevance algorithm (mRMR) [18] to identify the relevant subset of features with less redundancy. However, the previous feature selection techniques in [4], and [19] examined the best feature set based on the highest relation to the target class without taking into consideration the issue of independence and redundancy between the selected features [20]. Ahmed and Abulaish [19], developed a generic statistical approach for spam detection-based Twitter and Facebook datasets. Azab *et al.* [4], have used the GAIN univariate algorithm for feature selection to determine the most effective subset of features that enhance the classification performance instead of using all features. In the most of cases, the use of statistical criterion does not guarantee the best subset of selected features [10].

Davis *et al*.[21], developed the BotOrNot platform using the Random Forest classifier as a black box approach for feature dimensionality reduction, and aims to evaluate whether a Twitter account is controlled by a human or machine. 1K features are extracted from the interaction patterns and the content. All collected features are regrouped into six classes of network features, user, friends, temporal, content and sentiment features. Cresci *et al.* [22], proposed a Digital DNA model to predict online user behaviors such as new content, following or replying to other users. Yang *et al*. [23], have presented an empirical analysis of profile-based feature evasion tactics and content-based feature evasion tactics. Miller *et al.* [24], introduced a clustering model for anomaly detection. Moreover, different approaches have been proposed to examine the stability of selected features by computing the similarity of the subset [25] or the use of machine learning algorithms to calculated the accuracy of the model using only the selected feature set [17].

Nevertheless, a variety of population-based approaches have been designed for linear static and dynamic multi-objective optimization problems as well as for solving a set of complex problems involving at least three objective functions in [26] to [31]. A set of evolutionary-based approaches like Genetic Algorithm (GA) [32], Particle Swarm Optimization (PSO) [33], Genetic Programming (GP) [34], and Ant Colony Optimization (ACO) [35]are used to solve the problem of feature selection. Figure 2, details the iterative steps of the feature selection process which are; initialization, feature subset discovery, feature subset evaluation and results validation. For feature selection methods,

[2] https://www.imperva.com/blog/bot-traffic-report-2016/
[3] https://www.globaldots.com/bad-bot-report-2018

the key factors are the search techniques and evaluation criteria.

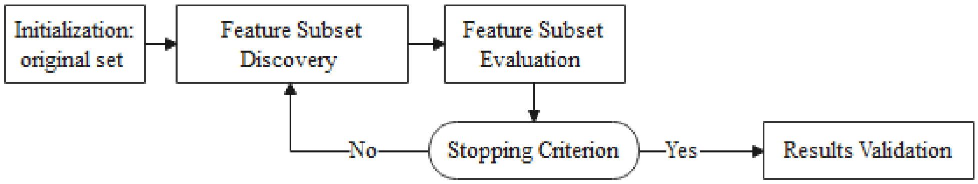


Figure 2. The iterative steps of the feature selection process

In 2015, Xue *et al.* [36] have published a survey paper to review the existing contributions based on population-based algorithms for solving single and multi-objective feature selection problem to minimise two objective functions (i) number of features and (ii) classification error rate. Search techniques, evaluation criteria, and the number of fitness functions are the three main concepts of evolutionary approaches. First, greedy search algorithms such as sequential forward selection (SFS) [37], and sequential backward selection (SBS) [38] are the well-known heuristic search techniques for feature selection. The main disadvantage of these methods is the 'nesting effects', so removed or selected features cannot be used for later testing. In addition, evolutionary techniques-based GA [39], GP [40], PSO [41], [42], and ACO [43] have been considered to determine which non-dominated solution provides the best trade-off between the number of features and the classification accuracy. Most of the existing feature selection methods suffer from the issues of high computational time/cost and stagnation in local optimum. In contrast, few works use PSO for fake account detection, where the POS algorithm is only used to optimize the parameters of both the logistic regression model [44], and the Q-learning method [45].

To sum up, the most existing contributions for fake account detection are done for minimising the number of the selected features taken into consideration the stability issue of the subset. The main goal was to enhance the computational time and maximise the accuracy rate of machine learning algorithms. The proposal consists in strongly exploring by a swarm having a quantum behavior before switching to a more stable behavior. This contribution is devoted to detect fake accounts on Twitter based on a new PSO-based approach whose role is to improve the self-learning of the deep neural detection system. Besides, several existing quantum-behaved PSO methods are reported in the next section.

Table 1. Existing methods for fake accounts detection on OSNs.

References	FS techniques	Tested Classifiers	Nb. Selected features	Accuracy	Datasets
Rostami and Karbasi [17], 2020	Minimum Redundancy Maximum Relevance algorithm [18].	10-fold cross validation using: (Random Forest, Naïve Bayes, SVM)	Test set 1: 8 Test set 2: 7 numerical	Best classifier: SVM Test set 1: 98% Test set 2: 97.1%	-Two Twitter datasets of Cresci *et al*. [13]
Cresci *et al*. [22], 2016	Digital DNA inspired by the biological DNA to model online user behaviors	ten-fold cross validation using: Bayes Net classifier	14 generic statistical features	Test set 1: 97.6% Test set 2: 92.9%	- <u>Dataset 1</u>: political -<u>Dataset 2:</u> Amazon
Davis *et al*. [21], 2016	Compute the bot-likelihood score using MLAs.	Ten-fold cross-validation using: Random Forest	1000 numerical feature values	95% AUC (Area Under ROC Curve).	-Dataset of 15k manually verified social bots and 16k legitimate accounts.
Azab *et al.* [4], 2016	GAIN univariate algorithm	5-fold cross validation using: (Random Forest, Decision Tree, Naïve Bayes, Neural Network, SVM)	7 numerical feature values	F-Measure (%) using: -RF:82.7, DT: 85.03, NB: 85.36, NN: 84.87, and SVM: 85.06	-Dataset of Twitter accounts collected by "the Fake project"
Miller *et al.* [24], 2014	Anomaly detection approach based on clustering model is built on normal twitter users with all outliers being treated as spam	Clustering algorithms (StreamKM++, DenStream, Combined)	126 numerical feature values	Accuracy using: - StreamKM++: 93.93% - DenStream: 97.11% - Combined: 98%	Dataset with 3239 user accounts including sample tweet (training set: 1587, test set: 1652)
Yang *et al*. [23], 2013	empirical analysis profile-based feature evasion tactics and content-based feature evasion tactics	10-fold cross validation using: (Random Forest, Decision Tree, Bayes Net, and Decorate)	25 numerical feature values	Best F1 Measure using Dataset I: RF :90%, Dataset II: RF :94.7%	-<u>Dataset I:</u> 20,000 accounts spam tweets, -<u>Dataset II:</u> 35,000 Twitter accounts
Ahmed and Abulaish [19], 2013	Generic statistical approach	Naïve Bayes, Jrip, and J48	14 generic statistical features	<u>Combined datasets:</u> detection rate (DR): 95.7%, false positive (FPR): 4.8% <u>Facebook dataset:</u> <u>DR:</u>96.4 %, FPR: 8.9%, <u>Twitter dataset</u> DR: 97.6%, FPR: 7.5%	Facebook and Twitter datasets

3. The existing Quantum-behaved PSO Methods

In 2004, Sun *et al.* [46] introduced quantum computing into the standard PSO algorithm. Quantum behaved PSO (QPSO) outperforms traditional PSO [33] with fewer control parameters and assumes a high level of convergence during the optimization process. So, instead of using a uniform stochastic distribution of particles' positions and velocity as in the original PSO algorithm. The quantum state of each particle is depicted by the wave function $\Psi(x,t), \forall \lim_{x\to\pm\infty} \Psi(x) = 0$. In quantum 3-dimensional time-space, the particle position in a point (x, y, z) is measured based on the probability density function $|\Psi(x)|^2$ satisfying the normalization condition in Equation (1).

$$\int_{-\infty}^{+\infty} |\Psi(x)|^2\, dxdydz = \int_{-\infty}^{+\infty} Q\, dxdydz = 1 \tag{1}$$

$$\text{Subject to: } \Psi(\mathrm{x}) = \frac{1}{\sqrt{L}} exp(-\|p - x\|/L)$$

The time dependent state of each particle at the time t is determined using the time-dependent Schrodinger equation $|\Psi(x,t)|^2 = 1/L\, exp(-2\|p - x\|/L)$ in [46].

3.1 Standard Quantum PSO (QPSO)

The first quantum-behaved PSO (QPSO) [46] is obtained through stochastic simulation of Monte Carlo measurement, when the particle position $X(t)$ is given by: $X(t) = p \pm \frac{L}{2} ln(1/u)$, with $L_{(t+1)} = 2 \times \beta \times |p - X(t)|$ and the update equation $x_i(t+1)$ of the particle i is presented in Equation (2).

$$x_i(t+1) = pbest_i^t \mp \beta \times |pbest_i^t - x_i^t| \times ln\left(\frac{1}{u}\right) \tag{2}$$

The β parameter of QPSO is the contraction expansion factor of the algorithm (positive real number) is also called "Creativity" or "Imagination" of the particle. The update rules are dependant to the personal best position ($pbest$) affected by a random uniform distribution of the parameter u uniformly distributed between 0 and 1.

To prevent the premature convergence, which a phenomenon that prevents an algorithm from finding an accurate estimation of the global optimum in meta-heuristics, a few QPSO improvements have been developed in the literature, including the Quantum Delta-Potential-Well-based Particle Swarm Optimization (QDPSO) algorithm [46], the Revised QPSO (RQPSO) [14], and the Gaussian Quantum-behaved PSO (GAQPSO) [15]. More details are presented as follows:

3.2 Quantum Delta-Potential-Well-based Particle Swarm Optimization (QDPSO)

In [46], Sun *et al.* have assumed that a quantum particle moves through a Delta potential well with a probability $Z > 0.5$. The particle is moved in a limited search space with respect to Z, otherwise it would appear out with a probability of $1 - Z$.

QDPSO benefits from the local attractor (La) instead of the simple use of $pbest$ position to guarantee convergence. In addition, the β coefficient is a positive real number set to: $\beta = \frac{1}{g}, \forall\, g > ln\sqrt{2}$, which balances local and global searching ability. The update of the particle position in QDPSO algorithm is done using the Monte Carlo method in Equation (3).

$$x_i(t+1) = \begin{cases} If\ rand(0,1) \geq 0.5\ then: \\ \quad La_i + L * (ln(1/u)) \\ Else: \\ \quad La_i - L * (ln(1/u)) \end{cases} \tag{3}$$

where,

- $La_i = (\varphi_1 * pbest_i(t) + \varphi_2 * gbest(t)) / (\varphi_1 + \varphi_2)$
- $L = \beta\ * abs\ (La_i - x_i(t))$
- $u, \varphi_1,\ and\ \varphi_2 =\ rand\ (0,1)$
- $\beta = {}^{1}/_{g}, \forall\, g = ln\sqrt{2}$
- T= shows the maximum number of iterations, and
- $gbest$=is the global best position

3.3 Revised Quantum PSO (RQPSO)

The global search ability of the QPSO system is denoted by the Revised Quantum PSO (RQPSO) [14]. The main difference between QDPSO and RQPSO is in the use of Mean Best Position ($mbest$), which is denoted by the Mainstream Thought Point presenting the center-of-gravity global best ($gbest$) position as presented in Equation (4).

$$mbest = \frac{1}{N}\sum_{i=1}^{N} pbest_i = \frac{1}{N}\sum_{i=1}^{N} pbest_{i1}, \ldots, \frac{1}{N}\sum_{i=1}^{N} pbest_{in} \tag{4}$$

where,

- N : indicates the size of the swarm, and.
- $mbest$: represents mean global $pbest$ position among all particles.

However, the equation to update particle position is modified by the parameter L which is equal to $\beta\ * abs\ (\text{mbest} - x_i(t))$, φ_1 and φ_2 are two random parameters uniformly distributed between 0 and 1. The modified equation is presented in Equation (5).

$$x_i(t+1) = \begin{cases} If\ rand(0,1) \geq 0.5\ then: \\ \quad La_i + L * (ln(1/u)) \\ Else: \\ \quad La_i - L * (ln(1/u)) \end{cases} \tag{5}$$

where,

- $La_i = (\varphi_1 * pbest_i(t) + (1 - \varphi_2) * gbest(t)) / (\varphi_1 + \varphi_2)$,
- $L = \beta\ * abs\ (\text{mbest} - x_i(t))$,
- $u, \varphi_1,\ and\ \varphi_2 =\ rand\ (0,1)$, and
- $\beta = 0.5 + 0.5 * (T - t)/T$.

3.4 The Gaussian QPSO algorithm (GAQPSO)

The improved variant of QPSO system is denoted by the Gaussian QPSO algorithm (GAQPSO) [15], where the position is updated through a Gaussian distribution. The GAQPSO algorithm is developed to deal with the issue of trapping in the local optimum. In this case, the main modification between RQPSO and GAQPSO is in the random parameters u, φ_1, and φ_2 which are modified to follow a Gaussian probability distribution with zero mean and unit variance. The update of particle position in GAQPSO is done using Equation (6).

$$x_i(t+1) = \begin{cases} If\ rand(0,1) \geq 0.5\ then: \\ \quad La_i + L * (ln(1/u)) \\ Else: \\ \quad La_i - L * (ln(1/u)) \end{cases} \tag{6}$$

where,

- $La_i = (\varphi_1 * pbest_i(\mathrm{t}) + (1 - \varphi_2) * gbest(t)) / (\varphi_1 + \varphi_2)$,
- $L = \beta\ * abs\ (\mathrm{mbest} - x_i(t))$,
- $u, \varphi_1, and\ \varphi_2 =\ abs\ (N(0,1))$,
- $\beta = 0.5 + 0.5 * (T - t)/T$,
- T= is the maximum number of iterations, and
- $gbest$=indicates the global best position.

4. The Proposed Quantum Beta Distributed Multi-Objective Particle Swarm Optimization Algorithm

The purpose of this section is to present the general description of the proposed Quantum Beta Multi-Objective Particle Swarm Optimization (QBD-MOPSO) algorithm. The proposed QBD-MOPSO algorithm in this work benefits from two optimization profiles for exploration and exploitation phases. The dynamic switching profiles are done according to the two variants of the quantum-behaved PSO approaches (GAQPSO, and RQPSO) for the first exploration phase, and the three types of parameters configuration of beta function using Gaussian, linear decrease, and exponential data distribution. The two profiles of the QBD-MOPSO system are presented in Equation (7). A particle P_i is performed in a quantum exploration profile, if it's current position ($\mathrm{X_i}$) is greater or equal to the mean personal best position ($mbest$), otherwise it was considered for the second beta exploitation profile.

$$X_i(t+1) = \begin{cases} \rightarrow \textbf{Exploration Profile} \\ \text{IF} \quad \mathrm{X_i(t)} \geq \text{mbest then:} \\ \quad \mathrm{X_{t+1}} = \text{update position using quantum} \\ \qquad \text{equation in RQPSO or GAQPSO} \\ \text{Else} \\ \rightarrow \textbf{Exploitation Profile} \\ \mathrm{X_{t+1}} = \text{update position using beta function} \end{cases} \tag{7}$$

The two dynamic switched profiles are detailed as follows:

- **Profile 1: exploration phase using quantum-behaved PSO**

For the exploration phase, all particles' positions are updated using the same equation in RQPSO and GAQPSO as explained in Equations (5) and (6) respectively.

- **Profile 2: exploitation phase using beta-behaved PSO**

For the exploitation phase, the particle positions are updated using Equation (8) including the use of the beta function.

$$X_i(t+1) = X_t + V_{t+1} \tag{8}$$

where, V_{t+1} is the velocity of the particle has followed different distribution shapes according to the beta function, as presented in Equation (9).

$$V_{t+1} = V_t + \beta(x)(P_{best}(t) - X_t) + \beta(x)(g_{best}(t) - X_t) \tag{9}$$

where; $gbest$ is the global best solution for all neighbours in swarm and $pbest$ is the best local experience of each particle. Both $gbest$ and $pbest$ are used to affect the updated position of each particle at each iteration (t). $\beta(x)$ is the Beta function proposed by Alimi [16], and presented in Equation (10), and $mbest$ is computed using the Equation (4).

The beta function was first introduced as a neural activation function [16] and demonstrated its ability to generate rich and flexible shapes (asymmetry, linearity, etc.). Also, the beta function has been adapted for different data distributions for feed-forward Neural Network (NN) [16] and investigated for Dynamic MOP and Many-Objective Optimization Problem [29]. According to the different configuration of both properties of p and q, three different shapes are considered in this study: the beta function with Gaussian, linear decrease, and exponential distributions as presented in the following Figure 3.

$$\beta(x) = \beta(x_0, x_1, p, q)(x) = \begin{cases} \text{if } x \in [x_0, x_1] \text{ then:} \\ \left(\frac{x - x_0}{x_c - x_0}\right)^p \left(\frac{x_1 - x}{x_1 - x_c}\right)^q \\ 0 \qquad \text{otherwise} \end{cases} \tag{10}$$

where, p, q, x_0, x_1 are real numbers and x_c is the beta center defined in Equation (11).

$$x_c = \frac{px_1 + qx_0}{p+q} \tag{11}$$

where,

- x_c: is the beta center point,
- x_0, and x_1 : are real numbers of the beta function in Equation (10), and
- p and q: are the control properties of the beta function in Equation (10).

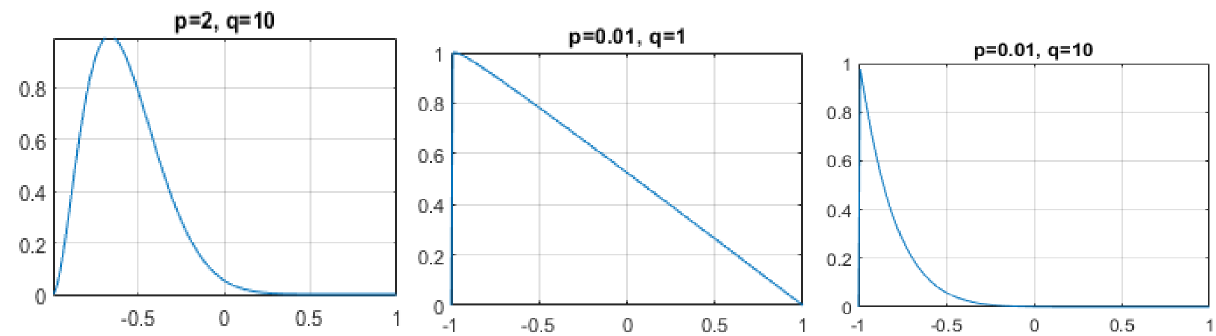
The multi-dimensional Beta function is defined in Equation (12) presenting the product of m one-dimensional Beta function.

$$\beta(x) = \prod_{k=1}^{m} \beta(x_k, p_k, q_k, x_{0,k}, x_{1,k}) \tag{12}$$

where,

- $\prod_{k=1}^{m} \beta(x_k, p_k, q_k, x_{0,k}, x_{1,k})$ is the product of m one-dimensional Beta function in Equation (10), and
- m: is the dimension of the search space,

Based on different configurations of the parameters p and q, Figure 3 illustrates three examples of shapes that can be generated from the beta-function in Equation (10). According to Figure 3 (a), the beta-function has been assumed to have a Gaussian distribution by setting p and q to 2 and 10 respectively. In Figure 3 (b), the linear decrease shape of the beta-function is obtained by assigning a value of 0.01 to the parameter p and 1 to the parameter q. Figure 3 (c) also shows an exponential data distribution curve with p equal to 0.01 and q fixed to 1.



(a) (b) (c)

Figure 3. The Data Distribution Curve of Beta function with (a) Gaussian, (b) linear decrease, and (c) exponential distribution according to the different proprieties' configuration of p and q.

Taken into consideration the previous example that presents the diversification of the shapes that can be obtained by the beta function, six variants of QBD-MOPSO approach are proposed based on GAQPSO and RQPSO for the exploration phase, and the three different distribution shapes of beta function for the exploitation enhancement. The six variants are developed to study the diversification of the data distribution of the new proposed QBD-MOPSO algorithm, and all variants are illustrated in Figure 4 and detailed as follows:

- **In Revised Quantum Beta Distributed Multi-Objective Particle Swarm Optimization (RQBD-MOPSO) system** the exploration profile is done using the update position of RQPSO algorithm as presented in Equation (5). However, three variants of beta exploitation profiles are done according to the parameter's configurations of the beta function in Equation (8). The three variants of RQBD-MOPSO approach are as follows:
  - **RQBD-MOPSO-V1**: exploration profile is done using RQPSO, and the exploitation profile is with a gaussian beta-behaved PSO.
  - **RQBD-MOPSO-V2**: exploration profile is with RQPSO, and the exploitation phase is with linear decreased beta-behaved PSO.
  - **RQBD-MOPSO-V3**: exploration phase is with RQPSO, and exploitation with exponential beta-behaved PSO.

- **In the Gaussian Quantum Beta Distributed Multi-Objective Particle Swarm Optimization (GAQBD-MOPSO) system,** the particles positions are updated using Equation (6) of GAQPSO for the exploration step. According to the three-beta configurations in Equation (8) for the exploitation profile, the three variants of the GAQBD-MOPSO system are as follows:
    - **GAQBD-MOPSO-V1**: exploration is with GAQPSO, and exploitation is with Gaussian beta-behaved PSO.
    - **GAQBD-MOPSO-V2**: exploration is with GAQPSO, and exploitation is with a linear decreased beta function.
    - **GAQBD-MOPSO-V3**: exploration with GAQPSO, and exploitation with exponential beta-behaved PSO.

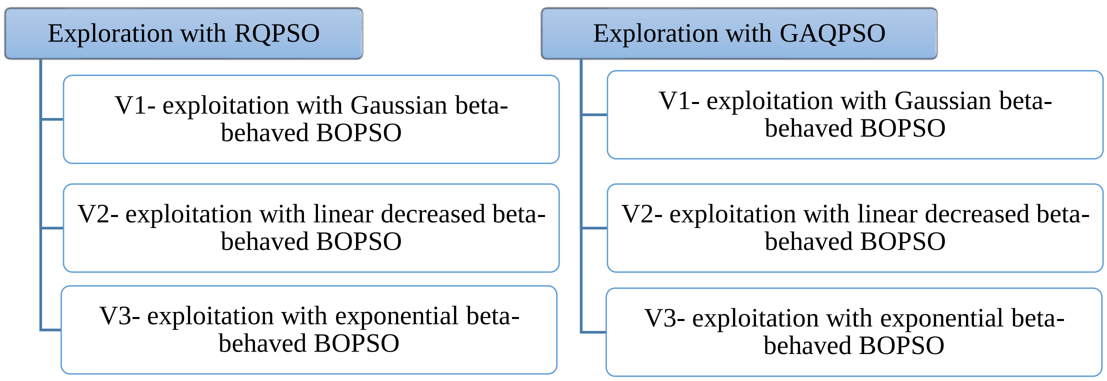


Figure 4. Six Variants of the Proposed QBD-MOPSO Algorithm with (V1) Gaussian, (V2) linear decreased, and (V3) exponential beta function

Figure 5 illustrates the flowchart of the proposed QBD-MOPSO approach. The details of the steps are as follows:

✓ **Step 1: initialization**

The first step aims to create a swarm of N particles with random positions $\mathrm{X}_i(\mathrm{t})$ and zero velocity $\mathrm{V}_i(\mathrm{t})$ vectors. Each particle p_i , $\forall\ i = 1 \ldots N$ has defined in m-dimensional search space. The iterative optimization process is considered to evaluate the fitness function and to update the particle positions. At each iteration, all non-dominated solutions are stored in the leader's archive.

✓ **Step 2: fitness function evaluation**

At each iteration t, a predefined fitness function $F(x,t)$ was evaluated.

✓ **Step 3: select pbest, gbest and mbest**

The global best solution ($gbest$) is selected randomly from the leader's archive. The personal best solution ($pbest$), is the best historical experience. The mean personal best solution ($mbest$) is determined using Equation 4.

✓ **Step 4: update the particles positions**

Compared to the existing PSO approaches, the proposed QBD-MOPSO algorithm benefits from a new equation to update the particles positions. As presented in Equation (7), a new optimization equation is used with two optimization profiles, where the position X of each particle p_i is distributed symmetrically about the mean personal best position

($mbest$). The first optimization profile is for exploration phase using the Quantum-behaved equation of RQPSO and GAQPSO. However, the second optimization profile is based on Beta-behaved function for exploitation enhancement.

✓ **Step 5: Update the leader's archive**

At each iteration, all non-dominated solutions in the leader's archive (A) are re-evaluated, and the dominated solutions are removed from the archive [47].

✓ **Step 6: Stopping criterion**

The QBD-MOPSO system is stopped when the maximum number of iterations is met.

✓ **Step 7: Output of QBD-MOPSO: determine the non-dominated solutions**

At the maximum number of iterations, a set of compromise solutions which are stored in the leader's archive (A) are considered as the output of the proposal and denoted by Pareto Optimal Front (POF).

✓ **Step 8: Decision Making: determine the best compromise solution**

For decision making, one best compromise solution is selected. The most known standard criterion in the optimization field is the use of the Utopian point mechanism [48], which is defined as an ideal infeasible solution that minimises the objective functions. After determining the utopian point, the Euclidian distance between this point and all non-dominated solutions in POF is computed. Then, the optimal particle with the smallest distance to the utopian point is selected as a compromise solution.

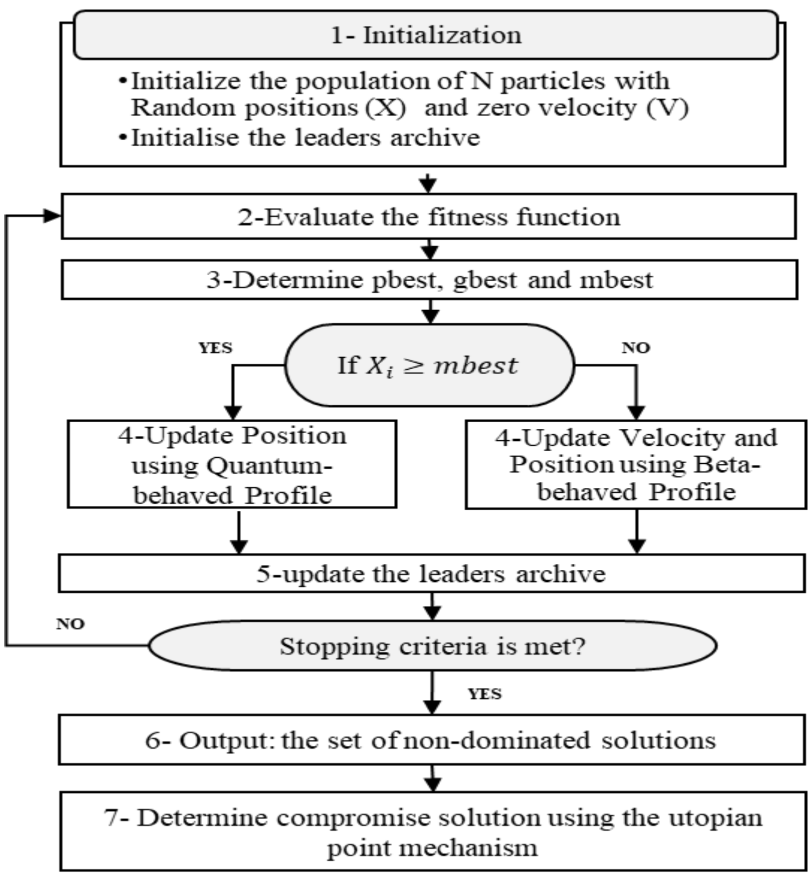


Figure 5. The Flowchart of the Proposed Quantum Beta Distributed Multi-Objective Particle Swarm Optimization (QBD-MOPSO) System.

Let's determine the complexity of QBD-MOPSO algorithm. The proposed QBD-MOPSO algorithm aims to optimize a swarm of N particles, each particle is a candidate solution performed until a maximum number of iterations T_{max} is reached. First, at the iteration $t = 1$ the initialization procedure is started including the following steps:

- initialize random positions X_i , $\forall\ i = 1 \dots N$ with m-dimensional search space, and zero velocities,
- evaluate the fitness function for n particles,
- apply the dominance operator to determine non-dominated solutions and stored in the leader's archive (A).

So, the determination of the initialization procedure takes $O(N)$ times.

Second, the main loop is executed until the maximum number of iterations T_{max} is reached. The running time of the QBD-MOPSO algorithm consists of K iterative loops performing logarithmic statements and takes $(K\ log(N))$ times. At each iteration, the above steps 2 to 6 are executed. The update of particle positions for the exploration or the exploitation profile is being preceded by determining the global best solution $(gbest)$ from the leader's archive (A), the personal best position for each particle $(pbest)$, and the mean best particle $(mbest)$. Furthermore, the fitness function is evaluated. Assuming that the particle is pre-sorted, the determination of each loop takes $O(\ log\ (N)$ times. At each time t, the leaders archive (A) is updated and all dominated solutions are removed, so this step takes $O(N)$ times. Finally, the best compromise solution is determined using the Utopian point mechanism and the determination takes $O(N)$ times. To sum up, the overall complexity of the proposed QBD-MOPSO algorithm is equal to $O(K\ Log(N))$.

Based on previous work, Sun *et al.* [46] have concluded the high performance of Quantum behaved PSO (QPSO) compared to the traditional PSO algorithm [33], characterised with fewer control parameters and assumes a high level of convergence during the optimization process. However, the main advantage of the proposed QBD-MOPSO algorithm is proved over their simplicity in terms of complexity which is equal to $O(K \times \log(N))$. The QBD-MOPSO algorithm in this work benefits from two optimization profiles for exploration and exploitation phases. When, the dynamic switching profiles are the main properties of the proposed algorithm investigating a high flexibility to produce several types of data distributions. The quantum and beta behaved rules provide a higher level of convergence toward the global best solutions

5. The Neuro-QBD-MOPSO Architecture for Fake Account Detection

In this section, the problem statement is presented, and the proposed QBD-MOPSO algorithm is applied to identify fake Twitter accounts and denoted by the Neuro-QBD-MOPSO Architecture.

5.1 Problem Statement

In this study, Feature Selection has been presented as a minimization Multi-Objective Optimization Problem (MOP) [9], and presented in Equation (13). Let's consider n data points $X = \{x_i\}_1^n$ that presents the input dataset. Each sample x_i has d-dimensional features $\{f_1, f_2, \dots, f_d\}$. Two objective functions are considered which are the features dimensionality $f_1(x)$ in Equation (14) and the classification error rate $f_2(x)$ in Equation (15) [9].

$$\text{Minimize } F(x) = \begin{cases} \mathrm{f}_1(\mathrm{x}): \text{features dimensionality} \\ \mathrm{f}_2(\mathrm{x}): \text{classification error} \end{cases} \tag{13}$$

$$f_1(x) = \alpha * \frac{\#features}{\#All\ features} + (1-\alpha) * \frac{Error\_Rate}{Error_{All}} \tag{14}$$

where; α is a constant value $\alpha \in [0,1]$, $\#features$: are the dimensionality of selected features, $\#All\ Features$: is the total number of original features. $Error\_Rate$: is the classification error rate of selected features. $Error_{All}$: is the classification error rate using all features.

$$f_2(x) = \frac{FP+FN}{TP+TN+FP+FN} \tag{15}$$

where; FP is the False Positive, FN is the False Negative, TP is the true positive, and TN is the true negative.

5.2 General Description: Neuro-QBD-MOPSO Architecture for Fake Account Detection

The QBD-MOPSO algorithm is proposed for pertinent features selection to detect fake accounts on Twitter. Figure 6 shows the Neuro-QBD-MOPSO architecture. The Neuro-QBD-MOPSO system takes a labelled Twitter dataset as input and performs the iterative process of the QBD-MOPSO algorithm to determine the features to be selected. Each particle has a subset of selected features that are used in training and validating the machine learning model. At the maximum number of iterations, the model is tested with the best feature set that has the lowest error rate. Last but not least, the list of fake accounts is determined as the output of the proposed system. Four steps are involved in achieving the feature selection step, namely; dataset collection, data pre-processing, feature extraction, and feature analysis. Figure 7 illustrates the overall steps.

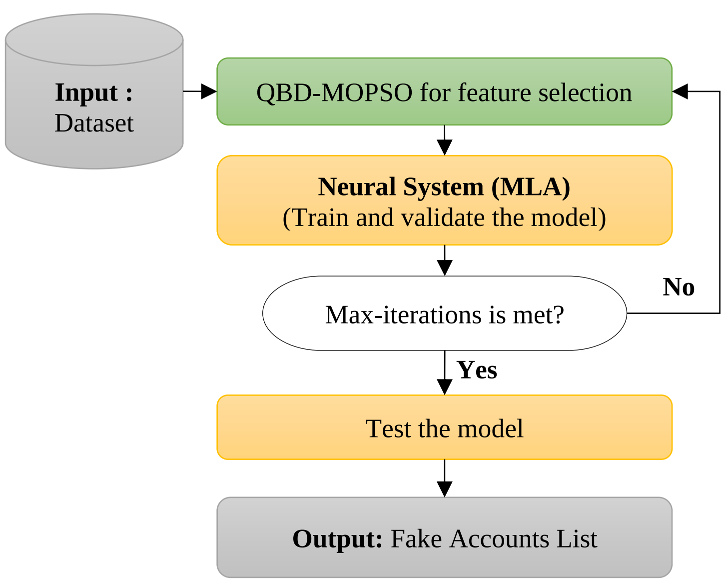


Figure 6. The Neuro-QBD-MOPSO Architecture for Fake Account Detection.

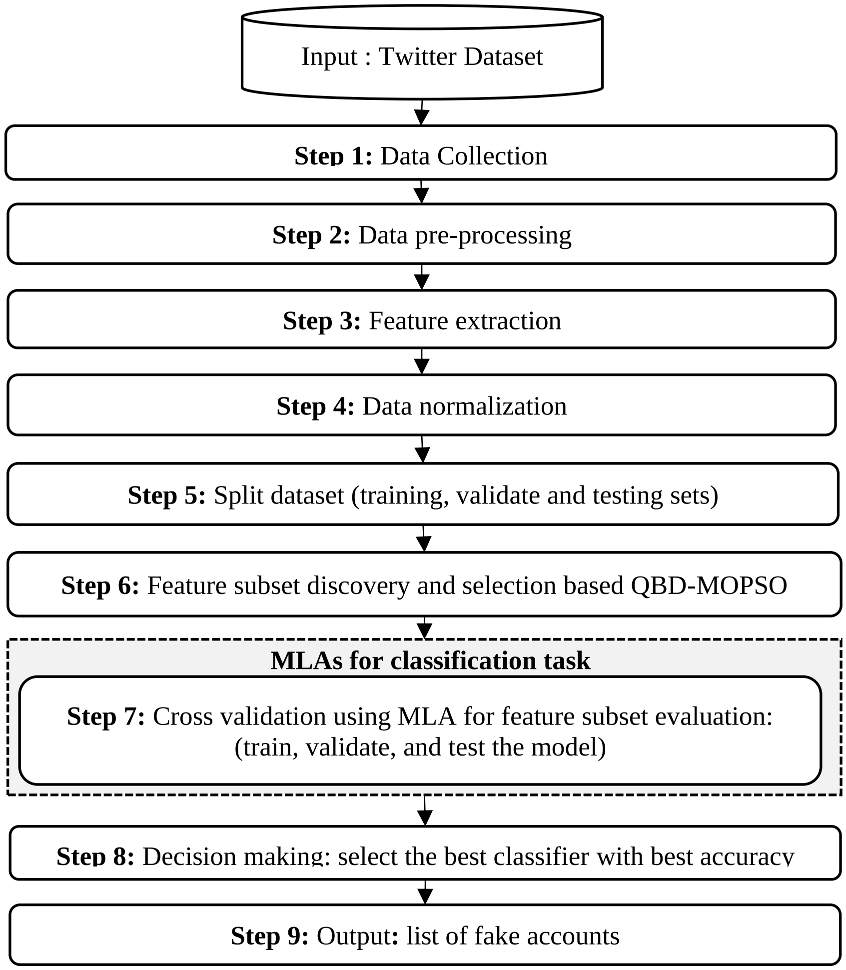


Figure 7. The Overall Process of QBD-MOPSO Algorithm for Fake Account Detection.

Figure 7 shows the overall process for detecting fake accounts using the QBD-MOPSO algorithm. The steps are as follows:

5.2.1 Dataset Collection

This step provides a starting point before diving into data exploration, and aims to select the main important columns from the original datasets. As shown in Table 2, two Twitter datasets proposed by Cresci *et al*. [31] are considered. A total number of 2910 Twitter accounts stored in both datasets. Dataset 1 contains 1982 Twitter accounts, while dataset 2 contains 928 accounts equally divided between human and social spam bot accounts.

Table 2. Cresci Datasets Properties

Properties	**Dataset 1**	**Dataset 2**	**Total**
Nb. accounts	1982	928	2910
Nb. tweets	4 061 598	2 628 181	6 689 779
Description	Genuine accounts + Social Spam Bot 1 (retweets of an Italian political candidate)	Genuine accounts + Social Spam Bot 3 (spammers of products on sale at Amazon.com)	-

5.2.2 Data Pre-processing

Several tasks are considered for text cleaning and presented as follows:

- Convert the corpus of tweets to lowercase.
- Removing the numbers from tweets using regular expressions to reduce the irrelevant features.
- Remove the set of symbols or punctuations.
- Remove white spaces from the tweet.
- Stop word removal: remove the common words in the language that do not carry a relevant meaning using natural language processing mechanisms.
- Stemming step to reduce the word to its stem forms using the Porter stemming algorithm that aims to remove the common morphological and inflexional endings from words (examples: users ➔ user, profiling ➔ profile).
- Lemmatization aims to reduce the word to the correct base forms using the lemmatization tool denoted by WordNet Lemmatize presented in Python Natural Language Toolkit (NLTK) library [49].

5.2.3 Features Extraction

In this study, the user profile properties and the content of tweets are the main information sources for feature extraction. As mentioned by Rostami and Karbasi [17], 46 original features are collected from Cresci datasets [13] using a set of standard statistical criteria such as entropy, and standard deviation. Table 3, has detailed 22 features extracted from the user profile information's. Table 4 presents 24 extracted features based on the tweets content.

Table 3. Extracted Features based on User profile properties

ID	Features based on User profile properties
F1	Follower count
F2	Follower count/Account Age
F3	Following count
F4	Following count/Account Age
F5	Follower count/ Following count
F6	Follower count/ Following count
F7	(2* Follower count) -Following count
F8	Follower count/ Follower + Following
F9	Favorite's count
F10	Favorite's count/ Account Age
F11	Tweet count
F12	Tweet count/Account Age
F13	List count
F14	List count/ Account Age
F15	Favorites count/Tweet count
F16	List count/ Follower count
F17	GEO Tag
F18	Retweet count
F19	Retweet count/Tweet count
F20	The consecutive Retweets interval mean
F21	The consecutive Retweets interval Standard deviation
F22	Number of times the tweets sent by the user are retweeted by other users

Table 4. Extracted Features based on Tweets Content

ID	Features based on tweets content
F23	Hashtag count
F24	Hashtag count/ Tweet count
F25	Hashtag-per-tweet Standard deviation
F26	Hashtag-per-tweet Entropy
F27	Tweets-with-Hashtags proportion
F28	The consecutive tweets interval means
F29	The consecutive tweets interval Standard deviation
F30	Link count
F31	Link count/ Tweet count
F32	Link-per-tweet standard deviation
F33	Link-per-tweet Entropy
F34	Tweets-with-Links proportion
F35	Mention count
F36	Mention count/Tweet count
F37	Mention-per-tweet Standard deviation
F38	Mention-per-tweet Entropy
F39	Tweets-with-Mentions proportion
F40	Reply count
F41	The consecutive Replies interval mean
F42	The consecutive Replies interval standard deviation
F43	Reply count/ Mention count
F44	The total number of the received likes
F45	Received likes count/ Tweet count
F46	Received like-per-tweet Standard deviation

5.2.4 Behavior Analysis of Fake and Human Web Users

The behavior analysis is considered to understand the attitude and the ethics of fake and human users on Twitter. Many features can be extracted from the tweets and denoted by content-based features extraction aiming to extract several features by parsing the content of each tweet such as the number of hashtags per tweet, the number of mentions per tweet, the length of the tweet and many others. In this sub-section, the sentiment analysis [50] process was first done. Sentiment analysis is an important topic in the field of Natural Language Processing (NLP) and presents a very high subject over many studies to detect negative, positive and neutral sentiment presenting a subjective opinion based on text analysis. In this contribution, the corpus of tweets is used to produce the following labels (1: positive, 0: neutral and -1: negative).

Before starting the sentiment analysis task, the text pre-processing step is considered to clean the corpus of tweets and detailed in Step 2. Furthermore, the Text-Blob python library is used for tweet processing and sentiment analysis by computing two properties; polarity and subjectivity for each tweet. The two properties are presented as a float value in the range of $[-1; 1]$ for the polarity property and $[0; 1]$ for the subjectivity property. The two Cresci datasets [13] have 2910 online user accounts regrouped equally; 1455 for fake and 1455 for human accounts. Based on Figures 8 and 9, it is remarkable that the most important number of tweets are with a neutral sentiment. However, we can conclude that the human accounts have the ability to express their opinion and feelings in the corpus of the tweet compared with fake users which have a large number of neutral opinions.

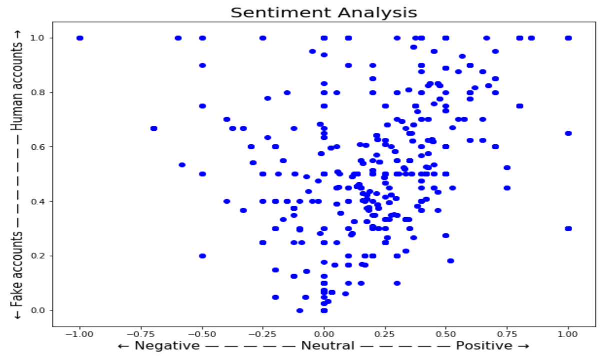


Figure 8: Sentiment analysis using polarity and subjectivity for fake and human accounts.

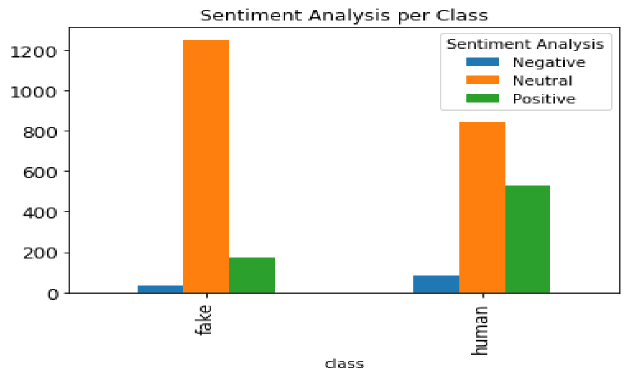
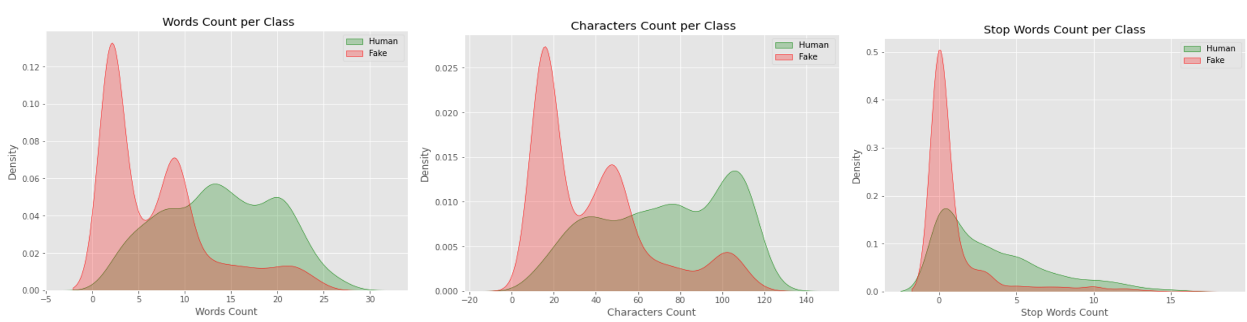


Figure 9. Classification of sentiment analysis for human and fake profiles.

Table 5 presents the number (Nb.) and the proportion in percentage of the users accounts according to the results of the sentiment analysis process. For fake accounts, there are 2.41%, 85.84% and 11.75% for negative, neutral and positive opinions respectively compared to the human users which have 5.49%, 58.08% and 36.43%.

Table 5. Comparative results of sentiment analysis for bot and human accounts on Twitter.

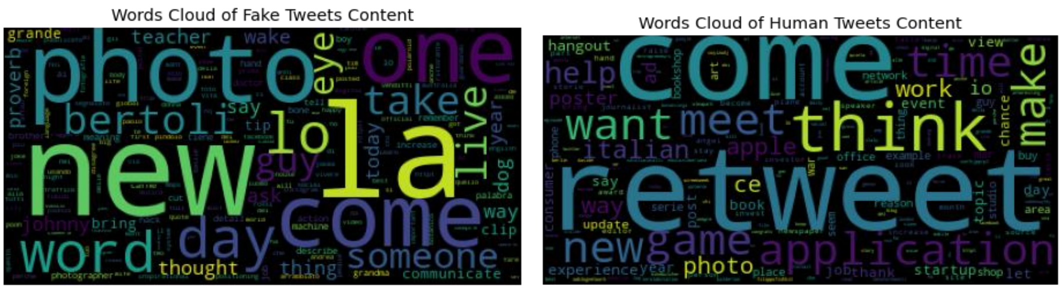
Sentiment	Fake		Human	
	Nb.	%	Nb.	%
Negative	35	**2.41**	80	**5.49**
Neutral	1249	**85.84**	845	**58.08**
Positive	171	**11.75**	530	**36.43**
Total	1455	100	1455	100



(a) (b) (c)

Figure 10. (a) Words Count, (b) Characters Count, and (c) Stop Words Count of Fake and Human Tweets Content.

The plots in Figure 10 illustrates the words count, the number of characters, and the number of stop words per class using the Tweet content. It can be remarkable that the majority of fake users have the habit of writing a Tweet corps of 5 to 10 words composed of 10 to 50 characters. However, the tweet corpus of the majority of human users is composed of 15 to 25 words and 80 to 110 characters. A spam-bot or fake user does not have the habit of using a lot of stop words compared with human user.



(a) (b)

Figure 11. Words Cloud of (a) Fake and (b) Human Tweets Content.

Figure 11 presents the words cloud to visualize the representation of the word frequency of fake and human tweets content, and highlight the words which are frequently used.

5.2.5 Feature selection based QBD-MOPSO algorithm

In this sub-section, the Neuro-QBD-MOPSO system is applied to the feature selection problem to minimize simultaneously the features dimensionality using Equation (14) and the classification error rate using Equation (15). The main steps of the Neuro-QBD-MOPSO system are detailed as follows:

In the QBD-MOPSO algorithm, the position of each particle is n-dimensional equal to the number of features in the dataset. In addition, the position Binary Encoding ($XBE\,(t)$) is added and initialized to present the bits of selected features that monitor the evaluation of the classifier.

- ✓ **Feature subset discovery**

At each iteration (t), the feature subset discovery step generates the candidate feature subset. It is a search procedure started with all available features in the dataset. The search step aims to find the best subset of features investigated to evaluate the fitness function of each particle.

- ✓ **Feature subset selection**

The binary position ($XBE_i(t)$) of each particle p_i is computed using the Sigmoid Transfer Function (STF) in Equation (16).

$$STF_i(t) = \frac{1}{1+e^{-X_i(t)}} \tag{16}$$

where: $X_i(t)$ is the position of the particle i.

The binary bits ($XBE_i\,(t)$) is determined according to Equation (17):

$$XBE_i(t) = \begin{cases} 1 & if\ \ STF_i\,(t) \geq rand \\ 0 & otherwise \end{cases} \tag{17}$$

In the vector of binary position ($XBE_i(t)$), "1" represents the feature is selected and "0" otherwise.

- ✓ **Subset evaluation**

For each particle, the subset of selected features is evaluated using the two objective functions provided in Equations (14) and (15) to train the classification model. This step is optimized until a maximum number of iterations. At each iteration, all the non-dominated solutions are stored in the leader's archive (A).

- ✓ **Decision making for best features subset selection**

At the maximum number of iterations, the Pareto Optimal Front (POF) is used to determine the Utopian Point and the compromise solution. For decision making, one best compromise solution is selected as presented in Figure 12. Finally, the subset of selected features by the compromise solution is considered to test and determine the performance of MLAs.

- ✓ **Stopping criterion**

The optimization process of QBD-MOPSO system is stopped when the maximum number of iterations is met.

- ✓ **The output of QBD-MOPSO for fake account detection**

Determine the compromise solution, the subset of the selected features and best classifier with best accuracy.

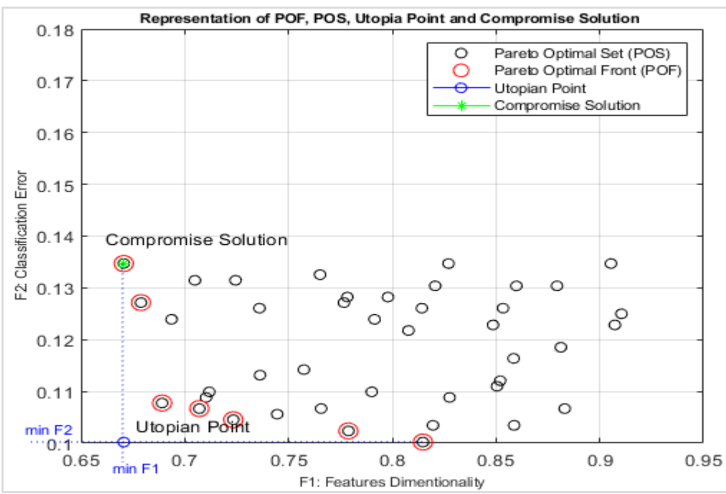


Figure 12. Example of Graphical representation of Pareto Optimal Front (POF), Pareto Optimal Set (POS), Utopian Point and Compromise Solution using QBD-MOPSO system.

As shown above, the application of QBD-MOPSO to detect fake accounts involves nine steps, which are illustrated in Figure 7: Data collection, data pre-processing, feature extraction, data normalisation and splitting the dataset into training, validation and testing sets. Then, the QBD-MOPSO algorithm is used to determine the best feature set when all particles are considered for training and validating the machine learning model. However, the best feature set selected by the compromise solution is considered for testing the model and has the highest accuracy rate. Finally, the classifier with the best accuracy is selected for decision making and the list of fake accounts is generated.

6. Experimental Study

This section presents the experimental study. Subsection 6.1 outline the state-of-the-art methods and explains the performance metrics used. Subsection 6.2 describes the parameter settings. The quantitative results and discussion are presented in subsection 6.3.

6.1 Preliminary and Performance Metrics

In this experimental study, the proposed QBD-MOPSO algorithm is compared with the state-of-the-art methods proposed by the following authors: Rostami and Karbasi [17] , Ahmed and Abulaish [19], Davis *et al.* [21], Cresci *et al.* [22], Yang *et al.* [23], Miller *et al.* [24] using the two Twitter datasets provided by Cresci *et al*. [13] detailed in Table 2. The experimental results of this study are compared to the methods discussed by Rostami *et al.* [17] using three machine learning algorithms (MLAs) namely: Random Forest (RF), Naïve Bayes (NB), and Support Vector Machine (SVM). 10-fold cross-validation technique is used to compute the performance criteria of RF, NB, and SVM algorithms. However, hidden layers are considered for the feedforward Neural Network (NN) algorithm, and the split of the datasets is as follows; 70% for training, 15% for validation and 15% for testing. For each MLA, we aim to compute the confusion matrix for performance measurement. In this case of study, three performance criteria are considered:

- ✓ **Accuracy (Acc.)** or classification rate presents the percent of the correct classified samples computed using Equation (18).

$$Accuracy = \frac{TP+TN}{TN+FN+TP+FP} \tag{18}$$

where, TP, TN, FN, and FP have been easily determined from the confusion matrix in Table 6.

Table 6. the confusion matrix

	Predicted class		
True class		**True**	**False**
	True	True Positive (TP)	False Positive (FN)
	False	True Negative (TN)	False Negative (FP)

- ✓ **F-Measure** is computed using Equation (19) based on the precision and recall criteria.

$$\mathrm{F-measure} = \frac{2\times\mathrm{Recall}\times\mathrm{Precision}}{\mathrm{Recall}+\mathrm{Precision}} \tag{19}$$

where; precision and recall are respectively equal to TP / (TP + FP) and TP / (TP + FN).

- ✓ **Matthew's correlation coefficient (MCC)** is the most important criteria for classification performance, and calculated using Equation (20).

$$\mathrm{MCC} = \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{20}$$

6.2 Parameters Settings

All variants of the QBD-MOPSO approach are implemented on a personal computer with 8 GB of RAM, 1 To and i7 Intel processors using MATLAB programming platform. The parameters setting of all variants are resumed in Table 7. In this study, the swarm size is set to 50, and the maximum number of iterations is 50.

Table 7. Parameters Setting for RQBD-MOPSO and RQBD-MOPSO variants with (V1) Gaussian, (V2) linear decreased, and (V3) exponential beta function.

Profiles of QBD-MOPSO	Parameters	RQBD-MOPSO			GAQBD-MOPSO		
		V1	V2	V3	V1	V2	V3
Quantum-behaved PSO	φ_1 φ_2 u	$rand\ (0,1)$			$N\ (0,1)$		
Beta-behaved PSO	p	2	0.01	0.01	2	0.01	0.01
	q	10	1	10	10	1	10

6.3 Results and Discussion

This section presents experimental results with all the available features, as well as comparative results using a subset of selected features based on QBD-MOPSO variants.

- ✓ **Quantitative results using all features**

Four MLAs are first tested using all the available features in both datasets. The purpose of this study is to illustrate the importance of using a small subset of features and their impact on the classification accuracy. In Table 8, the quantitative results are shown using 46 original features. I can be evidently seen that NN is the best classifier, with an accuracy rate of 98.89% using the first dataset. Nevertheless, random forest is the best classifier on the second dataset, with an accuracy rate of 96.12%.

Table 8. Performance criterions (%) of MLAs using all features.

	MLAs	Performance criterions (%) with all features		
		Acc.	**F-Measure**	**MCC**
Dataset 1	Random Forest	97.78	97.78	95.56
	Naïve Bayes	84.66	83.37	70.18
	SVM	97.83	97.87	95.74
	Neural Network	**98.89**	**98.89**	**97.78**
Dataset 2	Random Forest	**96.12**	**96.12**	**92.24**
	Naïve Bayes	84.91	84.81	69.83
	SVM	93.32	93.11	86.79
	Neural Network	94.82	94.71	89.73

- ✓ **Quantitative results of QBD-MOPSO variants using a subset of selected features on datasets 1 and 2**

The six variants of the QBD-MOPSO system are tested using both datasets 1 and 2. Based on data distribution types for exploration and exploitation profiles, there are three RQBD-MOPSO variants (V1, V2 and V3) and three GAQBD-MOPSO variants (V1, V2 and V3). In all variants, particles "fly" symmetrically to the center-of-gravity, which is the mean $pbest$ presenting the global attractor particle. Particles whose positions are greater or equal to the mean solution position are considered in the exploration profile, otherwise they are considered in the exploitation profile. Within an exploration profile, a RQBD-MOPSO algorithm updates particle positions according to a uniform random distribution between 0 and 1, but a GAQBD-MOPSO algorithm updates them according to a gaussian distribution with zero mean and unit variance. In addition, the particles positions in the exploitation profile are updated by using beta functions for gaussian (V1), linear decrease (V2), and exponential (V3) distributions. The classification performance of NB, RF, SVM, and NN classifiers is detailed in Table 9, along with the dimensionality subset of selected features determined by the different variants of the QBD-MOPSO system. It is remarkable that QBD-MOPSO variants are more competitive than the state-of-the-art methods using dataset 1.

The reported performance criteria of the NN classifier have demonstrated that GAQBD-MOPSO (V2) is the best approach for fake account detection using dataset 1 with the highest accuracy rate of 99.19% using 32 selected features. Moreover, the comparative results in Table 10 have shown the superiority of GAQBD-MOPSO (V2) using dataset 2 with an accuracy rate of 97.52 % with 25 selected features. The first 32 features are divided equally as follows; 16 features are selected based on the user profile properties which are presented in Table 3 (ID: F2, F4, F5, F6, F7, F9, F12, F14, F15, F16, F17, F18, F19, F20, F21, F22) presenting 73% of all user profile properties, and 16 features are selected based on tweet content as presented in Table 4 (ID: F27, F28, F29, F30, F31, F32, F35, F36, F38, F40, F41, F42, F43, F44, F45, F46) presenting 67% of all features based on the tweet content. For the second 25 selected features, 10 features based on the profile's properties (45%) are selected and their ID in Table 3 are as follows (ID: F1, F2, F4, F5, F7, F9, F15, F16, F18, F21), and 15 features are based on the tweet content from Table 4 (ID: F23, F24, F26, F27, F29, F31, F32, F34, F35, F36, F37, F38, F43, F44, F46) presenting 63% of the features-based content.

- ✓ **Comparative Results of the best variant "GAQBD-MOPSO (V2)" versus State-of-the-art Methods**

Table 9 and Table 10 provide the classification performance of the state-of-the-art methods compared with all variants of the proposed QBD-MOPSO algorithm. For both datasets, QBD-MOPSO has the ability to assume a high accuracy rate compared with the supervised methods proposed by Rostami and Karbasi [17], Davis *et al.* [21] and Yang *et al.* [23] as well as the unsupervised approaches proposed by Ahmed and Abulaish [19], Cresci *et al.* [22], and Miller *et al.* [24]. The proposed QBD-MOPSO presents high accuracy when using a supervised NN classifier, and it is able to identify 32 pertinent features from 46 original features when using dataset 1. Furthermore, only 25 features are selected with the GAQBD-MOPSO (V2) system in RF classifier based on the dataset 2. In [17] Rostami and Karbasi, used the Minimum Redundancy –Maximum Relevance algorithm (mRMR) as feature selection technique, and had the ability to select 8 and 7 pertinent features using dataset 1 and dataset 2 respectively.

The selected features are based on tweets content, and SVM classifier is the top performer with an MCC equal to 96.06% for dataset 1, and 94.19% for dataset 2. Despite the minimum number of selected features (8 and 7) in [17], QBD-MOPSO can achieve a high MCC (98.39% and 95.06%) with 32 and 25 optimal features using both datasets for the NN and RF respectively. In [22], Cresci et al. proposed a DNA fingerprinting method and achieve an MCC 95.20% and 86.70% regarding 14 features [19]. Davis et al. [21], proposed the BotOrNot system and achieved a classification rate of 17.4% on dataset 1 and 37.8% on dataset 2, using a random forest classifier with 1000 Twitter feature account. In [24], Miller et al. have considered 126 features to test their model, and does not assume a good result compared to all methods. Also, Ahmed and Abulaish [19] have proposed Graph clustering and Community Detection methods and 14 generic statistical features are selected to test the unsupervised model. Last but not least, Yang et al. [23], have proposed an empirical analysis of profile-based feature evasion tactics and content-based feature evasion tactics using 25 features. However, it fails to obtain a good classification accuracy (MCC=4.3% with dataset 1and MCC=28.7% with dataset 2). Compared with all methods, QBD-MOPSO can achieve a good performance using a dynamic feature selection according to quantum weights and the diversification of beta profiles, which are encoded using the sigmoid function.

Table 11 shown a comparative result of the proposed GAQBD-MOPSO (V2) versus the three standard approaches to quantum-behaved PSO namely; QPSO, RQPSO and GAQPSO. To study the performance of the novel proposal compared with other original systems (QPSO, RQPSO and GAQPSO). All algorithms are executed in the same condition with the NN for dataset 1 and the RF for dataset 2. Of course, it is remarkable that GAQBD-MOPSO (V2) is the winner and has achieved the highest accuracy rates with NN classifier using dataset 1 and RF using dataset 2. Figures 13 and 14 have shown that GAQBD-MOPSO (V2) has the highest accuracy rate for detecting fake accounts compared to the existing methods.

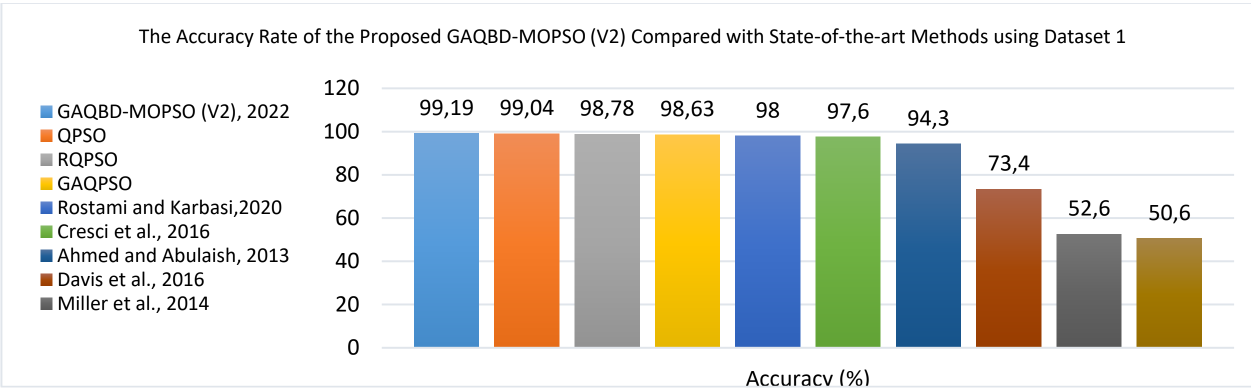


Figure 13. The Accuracy Rate of the GAQBD-MOPSO (V2) Compared with state-of-the-art Methods using Dataset 1.

Table 9. Comparative Results of **QBD-MOPSO** variants with (V1) Gaussian, (V2) linear decreased, and (V3) exponential beta function compared with state-of-the-art methods using **dataset 1**.

	Compared Approaches		Selected features	Performance criterions (%) using the best features subset									
				TP	**TN**	**FP**	**FN**	**Precision**	**Recall**	**Specificity**	**Acc.**	**F-Measure**	**MCC**
NB	RQBD-MOPSO	V1	14	812	929	179	62	81.94	92.91	83.84	87.41	87.08	76.21
		V2	30	949	936	42	55	95.76	94.52	95.71	95.11	95.14	90.22
		V3	7	790	884	201	107	79.71	88.07	81.47	84.46	83.69	69.23
	GAQBD-MOPSO	V1	31	950	936	41	55	95.86	94.52	95.80	95.15	95.19	90.32
		V2	21	931	934	60	57	93.94	94.23	93.96	94.09	94.08	88.19
		V3	19	922	935	69	56	93.03	94.27	93.12	93.69	93.65	87.39
SVM	RQBD-MOPSO	V1	32	967	990	24	1	97.57	99.89	97.63	98.73	98.72	97.50
		V2	26	969	991	22	0	97.78	100	97.82	98.89	98.87	97.80
		V3	35	966	990	25	1	97.47	99.89	97.53	98.68	98.67	97.40
	GAQBD-MOPSO	V1	26	970	990	21	1	97.88	99.89	97.92	98.89	98.87	97.79
		V2	25	967	991	24	0	97.57	100	97.63	98.78	98.77	97.60
		V3	30	968	990	23	1	99.89	99.89	95.83	98.78	99.89	97.60
RF	RQBD-MOPSO	V1	14	975	985	16	6	98.38	99.38	98.40	98.89	98.88	97.78
		V2	12	970	991	21	0	97.88	100	97.92	98.94	98.92	97.90
		V3	13	975	986	16	5	98.38	99.48	98.40	98.94	98.93	97.88
	GAQBD-MOPSO	V1	23	980	978	11	13	98.89	98.69	98.88	98.78	98.79	97.57
		V2	18	975	990	16	1	98.38	99.89	98.40	99.14	99.13	98.29
		V3	17	974	991	17	0	98.28	100	98.31	99.14	99.13	98.29
NN	RQBD-MOPSO	V1	25	972	991	19	0	98.08	100	98.11	99.04	99.03	98.10
		V2	24	964	991	27	0	97.27	100	97.34	98.63	98.61	97.31
		V3	18	955	990	36	1	96.36	99.89	96.49	98.13	98.09	96.32
	GAQBD-MOPSO	V1	27	959	990	32	1	96.77	99.89	96.86	98.33	98.30	96.71
		V2	**32**	976	990	15	1	98.48	99.89	98.50	**99.19**	99.18	**98.39**
		V3	17	974	991	17	0	98.28	100	98.31	99.14	99.13	98.29
State-of-the art methods	Rostami and Karbasi [17], 2020		-	-	-	-	-	98.00	98.10	98.00	98.00	98.00	96.06
	Cresci *et al*. [22], 2016		-	-	-	-	-	98.20	97.20	98.10	97.60	97.70	95.20
	Davis *et al.* [21], 2016		>1000	-	-	-	-	47.10	20.80	91.80	73.40	28.80	17.40
	Miller *et al*. [24], 2014		126	-	-	-	-	55.50	35.80	69.80	52.60	43.50	5.90
	Ahmed and Abulaish [19], 2013		-	-	-	-	-	94.50	94.40	94.50	94.30	94.40	88.60
	Yang *et al*. [23], 2013		25	-	-	-	-	56.30	17.00	86.00	50.60	26.10	4.30

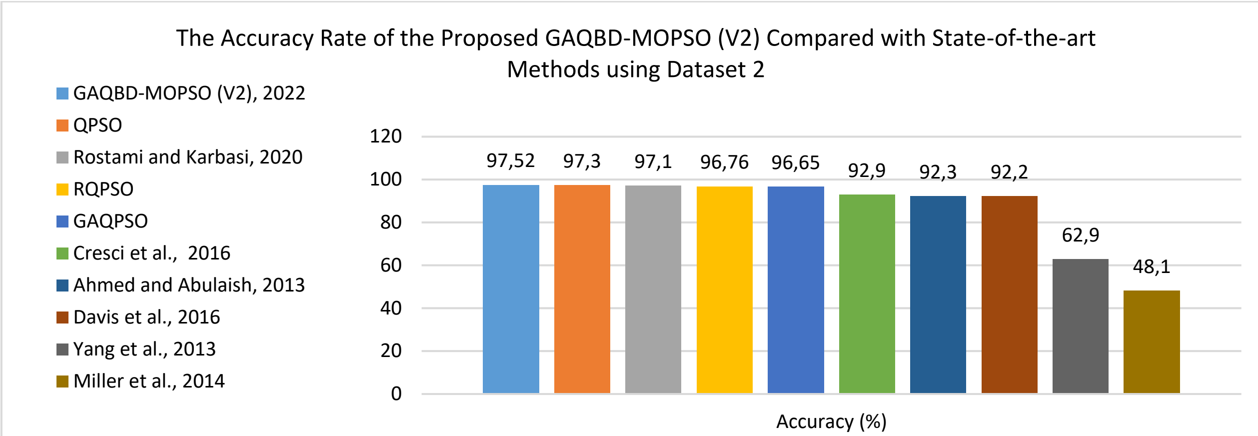


Figure 14. The Accuracy Rate of GAQBD-MOPSO (V2) Compared with state-of-the-art Methods using Dataset 2.

Table 10. Comparative Results of **QBD-MOPSO** variants with (V1) Gaussian, (V2) linear decreased, and (V3) exponential beta function compared with state-of-the-art methods using **dataset 2.**

	Compared Approaches		Selected features	Performance criterions (%) using the best features subset									
				TP	**TN**	**FP**	**FN**	**Precision**	**Recall**	**Specificity**	**Acc.**	**F-Measure**	**MCC**
NB	RQBD-MOPSO	V1	19	443	347	21	117	95.47	79.11	94.29	85.13	86.52	71.81
		V2	9	436	353	28	111	93.97	79.71	92.65	85.02	86.25	71.19
		V3	17	444	354	20	110	95.69	80.14	94.65	85.99	87.22	73.37
	GAQBD-MOPSO	V1	15	446	351	18	113	96.12	79.78	95.12	85.88	87.19	73.32
		V2	29	453	149	11	315	97.62	58.98	93.12	64.87	73.53	39.36
		V3	21	444	336	20	128	95.68	77.62	94.38	84.05	85.71	70.02
SVM	RQBD-MOPSO	V1	8	437	404	27	60	94.18	87.92	93.73	90.62	90.94	81.45
		V2	26	445	421	19	43	95.90	91.18	95.68	93.31	93.48	86.75
		V3	24	447	418	17	46	96.33	90.66	96.09	93.21	93.41	86.59
	GAQBD-MOPSO	V1	21	445	417	19	47	95.90	90.44	95.64	92.88	93.09	85.93
		V2	20	444	419	20	45	95.68	90.79	95.44	92.99	93.17	86.11
		V3	24	445	421	19	43	95.90	91.18	95.68	93.31	93.48	86.75
RF	RQBD-MOPSO	V1	20	443	460	21	4	95.47	99.10	95.63	97.30	97.25	94.67
		V2	22	449	451	15	13	96.76	97.18	96.78	96.98	96.97	93.96
		V3	14	455	448	9	16	98.06	96.60	98.03	97.30	97.32	94.62
	GAQBD-MOPSO	V1	19	430	462	34	2	92.67	99.53	93.14	96.12	95.98	92.46
		V2	**25**	445	460	19	4	95.90	99.10	96.03	**97.52**	97.48	**95.09**
		V3	33	441	460	23	4	95.04	99.10	95.23	97.09	97.02	94.26
NN	RQBD-MOPSO	V1	16	440	384	24	80	94.82	84.61	94.11	88.79	89.43	78.15
		V2	16	443	409	21	55	95.47	88.95	95.11	91.81	92.09	83.84
		V3	21	453	383	11	81	97.62	84.83	97.20	90.08	90.78	81.10
	GAQBD-MOPSO	V1	27	445	420	19	44	95.90	91.00	95.67	93.21	93.38	86.54
		V2	22	443	402	21	62	95.47	87.72	95.03	91.05	91.43	82.43
		V3	31	454	438	26	10	94.58	97.84	94.39	96.12	96.18	92.29
State-of-the art methods	Rostami and Karbasi [17], 2020		-	-	-	-	-	96.50	97.90	96.40	97.10	97.10	94.19
	Cresci *et al*. [22], 2016		-	-	-	-	-	100	85.80	100	92.90	92.30	86.70
	Davis *et al*. [21], 2016		>1000	-	-	-	-	63.50	95.00	98.10	92.20	76.10	73.80
	Miller *et al*. [24], 2014		126	-	-	-	-	46.70	30.60	65.40	48.10	37.00	-4.30
	Ahmed and Abulaish [19], 2013		-	-	-	-	-	91.30	93.50	91.20	92.30	92.30	84.70
	Yang *et al*. [23], 2013		25	-	-	-	-	72.70	40.90	84.80	62.90	52.40	28.70

Table 11. Performance criterions (%) of the best classifiers (Random Forest and Neural Network) for the proposed GAQBD-MOPSO (V2) versus the standard QPSO, RQPSO, and GAQPSO.

Datasets	**Compared Approaches**	**Selected features**	**Performance criterions (%) using the best features subset**									
			TP	TN	FP	FN	Precision	Recall	Specificity	Acc.	F-Measure	MCC
Dataset 1	GAQBD-MOPSO-V2 (NN)	**32**	976	990	15	1	98.48	99.89	98.50	**99.19**	99.18	**98.39**
	QPSO (NN)	33	974	989	17	2	98.28	99.79	98.31	99.04	99.03	98.09
	RQPSO (NN)	22	967	991	24	0	97.57	100	97.63	98.78	98.77	97.60
	GAQPSO (NN)	20	966	989	25	2	97.47	99.79	97.53	98.63	98.62	97.30
Dataset 2	GAQBD-MOPSO-V2 (RF)	**25**	445	460	19	4	95.90	99.10	96.03	**97.52**	97.48	**95.09**
	QPSO (RF)	25	443	460	21	4	95.47	99.10	95.63	97.30	97.25	94.67
	RQPSO (RF)	19	448	450	16	14	96.55	96.96	95.56	96.76	96.76	93.53
	GAQPSO (RF)	29	437	460	27	4	94.18	99.09	94.45	96.65	96.57	93.43

The state-of-the-art methods compared with the proposed QBD-MOPSO algorithm have been divided into supervised and unsupervised methods for classifying spambots. In supervised methods, Yang *et al*. [23] proposed a spambot detection system based on machine learning algorithms to predict human and spambot accounts. In addition, Davis *et al*. [21] proposed the BotOrNot Blackbox platform. The analysis of the results in Figure 15 shows that both the methods in [21] and [23] failed in classification and most bot accounts were classified as human with a recall rate less than 50. For dataset 2, Figure 16 shows that the precision and recall values of the system proposed by Davis *et al.* [21] are unbalanced, resulting in lower accuracy of the classification model, and that the system of Yang *et al*. [23] fails with a recall rate below 50. In unsupervised methods, Miller *et al*. [24] proposed a stream clustering model based on DenStream [51] and StreamKM++ [52] clustering algorithms for spambot detection to determine the cluster of the feature vector for a set of unlabelled samples in the dataset. Figures 15 and 16 show that the clustering algorithm proposed in [24] doesn't achieve good recall in datasets 1 and 2. So it's very difficult to detect spambots from data streams. Ahmed and Abulaish [19], proposed a graph clustering algorithm based on Markov Clustering Algorithm (MCL) [53]. However, Cresci *et al.* [13] have replaced the MCL algorithm with the Fastgreedy community detection algorithm [54], thus avoiding the problem of identifying two distinct clusters.

Inspired by the biological DNA sequence, Cresci *et al*. [22] proposed a bio-inspired model called digital DNA system aimed at recognising the behaviour of online users. The digital DNA was expressed by a string encoding each user's behaviour. Then, the Longest Common Substring (LCS) measurement was used to determine the anomalous similarities between the sequences, and the longest DNA sequences were labelled as spambot accounts. From Figures 15 and 16, it can be seen that the recall rate for datasets 1 and 2 is unbalanced, leading to a reduction in the performance criteria. Furthermore, Rostami and Karbasi [17] proposed a multi-objective feature selection approach to select a stable subset of features based on the highest relation to the target class and the least redundancy among the features using the Minimum Redundancy – Maximum Relevance algorithm (mRMR) [18]. In this study, the QBD-MOPSO algorithm for detecting fake accounts on Twitter is presented and denoted the Neuro- QBD-MOPSO system. Figures 15 and 16 illustrate the superiority of the algorithm GAQBD-MOPSO -V2 compared to other methods based on the precision and recall criteria. Neuro-QBD-MOPSO is a supervised method that is inspired by the standard quantum PSO algorithm. It is capable of detecting human and fake accounts on Twitter much more accurately than other methods.

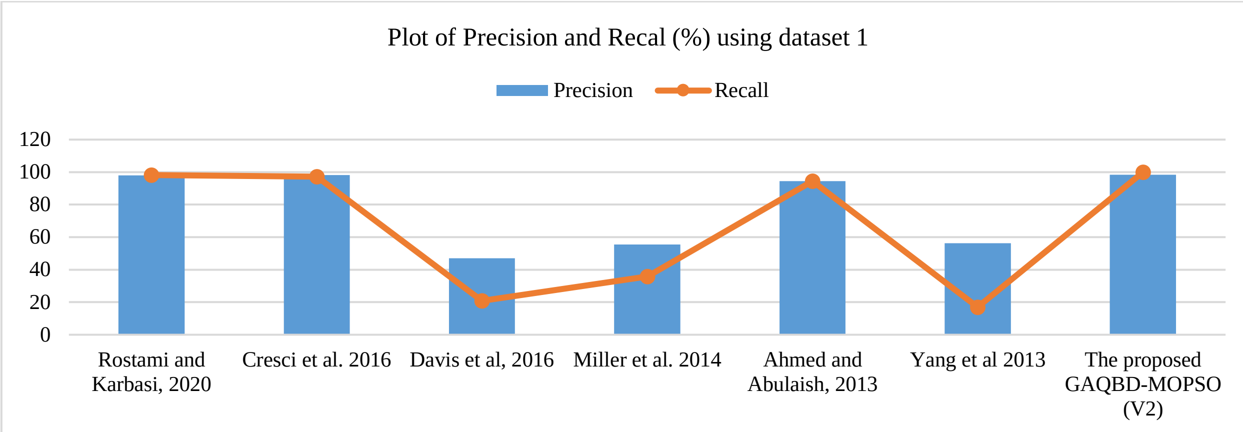


Figure 15. The Precision and Recall Rate of GAQBD-MOPSO (V2) Compared with state-of-the-art Methods using Dataset 1.

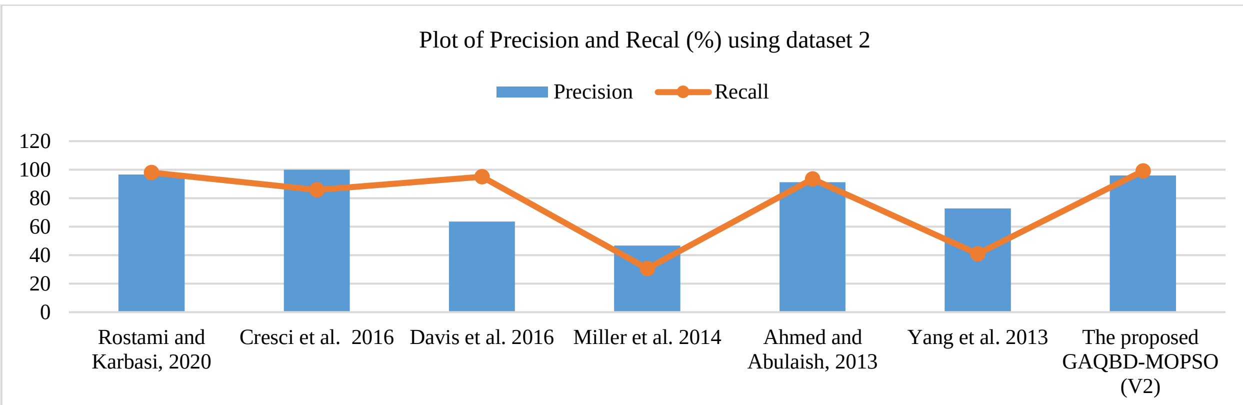


Figure 16. The Precision and Recall Rate of GAQBD-MOPSO (V2) Compared with state-of-the-art Methods using Dataset 2.

7. Conclusion

This paper proposed a new Quantum Beta Distributed Multi-Objective Particle Swarm Optimization (QBD-MOPSO) algorithm, comprising six quantum variants with different beta-profiles. The QBD-MOPSO system was used for pertinent feature selection to detect fake accounts on Twitter. The main goal was to minimize both the features' dimensionality and the classification error rate. The six variants of the QBD-MOPSO approach were proposed with two optimization profiles, the first was for exploration using a quantum-behaved MOPSO, and the second was for exploitation phase using a beta-behaved MOPSO. Both profiles were assumed over new mathematical rules to optimize and update the velocities and the positions of particles in the search space. At each iteration, binary encoding is fixed using the sigmoid function. Therefore, the bit '1' indicates a selected feature and '0' otherwise. The proposed system was tested on the two benchmark Twitter datasets and achieved excellent results compared with state-of-the-art methods. The GAQBD-MOPSO (V2) system was found to achieve an accuracy rate of 99.19% on dataset 1 and 97.52% for dataset 2. For future work, we will address the challenge of online feature selection to predict online fake accounts on OSNs, taking into account the stability of the feature subset. Also, a new investigation will be proposed for fake news detection on Twitter.

Acknowledgment

We deeply acknowledge Taif University for Supporting this study through Taif University Researchers Supporting Project number (TURSP-2020/347), Taif University, Taif, Saudi Arabia. The research leading to these results has received funding from the Ministry of Higher Education and Scientific Research of Tunisia under the grant agreement number LR11ES48.

References

[1] S. Kanoje, S. Girase, and D. Mukhopadhyay, "User Profiling Trends, Techniques and Applications," *undefined*, vol. 1, no. 1, 2015.

[2] S. B.-T. I. Encyclopedia and undefined 2003, "Personalization and customization technologies," *books.google.com*, Accessed: Feb. 01, 2022. [Online]. Available: https://books.google.com/books?hl=fr&lr=&id=wshm3f0hyI8C&oi=fnd&pg=PA51&dq=Personalization+and+Customization+Technologies&ots=-j-m2hYRy2&sig=fQuECXOHrziApigtn9CWKdiRMro

[3] Z. Alom, B. Carminati, and E. Ferrari, "Detecting spam accounts on Twitter," *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, pp. 1191–1198, Oct. 2018, doi: 10.1109/ASONAM.2018.8508495.

[4] A. Azab, A. M. Idrees, M. A. Mahmoud, and H. Hefny, "Fake Account Detection in Twitter Based on Minimum Weighted Feature set," *undefined*, 2015.

[5] J. Castellini, V. Poggioni, and G. Sorbi, "Fake twitter followers detection by denoising autoencoder," *Proceedings - 2017 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2017*, pp. 195–202, Aug. 2017, doi: 10.1145/3106426.3106489.

[6] S. R. Sahoo and B. B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Applied Soft Computing*, vol. 100, p. 106983, Mar. 2021, doi: 10.1016/J.ASOC.2020.106983.

[7] F. C. Akyon and M. Esat Kalfaoglu, "Instagram Fake and Automated Account Detection," *Proceedings - 2019 Innovations in Intelligent Systems and Applications Conference, ASYU 2019*, Oct. 2019, doi: 10.1109/ASYU48272.2019.8946437.

[8] J. Miao and L. Niu, "A Survey on Feature Selection," *Procedia Computer Science*, vol. 91, pp. 919–926, Jan. 2016, doi: 10.1016/J.PROCS.2016.07.111.

[9] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013, doi: 10.1109/TSMCB.2012.2227469.

[10] K. Sutha and J. Tamilselvi, "A Review of Feature Selection Algorithms for Data Mining Techniques," *undefined*, 2015.

[11] J. Tang, S. Alelyani, H. L.-D. classification: A. and, and undefined 2014, "Feature selection for classification: A review," *cvs.edu.in*, Accessed: Feb. 02, 2022. [Online]. Available: http://www.cvs.edu.in/upload/feature_selection_for_classification.pdf

[12] F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley, and H. Liu, "A new approach to bot detection: Striking the balance between precision and recall," *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*, pp. 533–540, Nov. 2016, doi: 10.1109/ASONAM.2016.7752287.

[13] S. Cresci, R. di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," *26th International World Wide Web Conference 2017, WWW 2017 Companion*, pp. 963–972, Jan. 2017, doi: 10.1145/3041021.3055135.

[14] J. Sun, W. Xu, and B. Feng, "A global search strategy of Quantum-behaved Particle Swarm Optimization," *2004 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 111–116, 2004, doi: 10.1109/ICCIS.2004.1460396.

[15] J. Sun, W. Fang, V. Palade, X. Wu, and W. Xu, "Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point," *Applied Mathematics and Computation*, vol. 218, no. 7, pp. 3763–3775, Dec. 2011, doi: 10.1016/J.AMC.2011.09.021.

[16] A. M. Alimi, "BETA NEURO-FUZZY SYSTEMS," vol. 1, no. 1, pp. 23–41, 2003.

[17] R. R. Rostami and S. Karbasi, "Detecting fake accounts on twitter social network using multi-objective hybrid feature selection approach," *Webology*, vol. 17, no. 1, Jun. 2020, doi: 10.14704/WEB/V17I1/A204.

[18] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Proceedings of the 2003 IEEE Bioinformatics Conference, CSB 2003*, pp. 523–528, 2003, doi: 10.1109/CSB.2003.1227396.

[19] F. Ahmed and M. Abulaish, "A generic statistical approach for spam detection in Online Social Networks," *Computer Communications*, vol. 36, no. 10–11, pp. 1120–1129, Jun. 2013, doi: 10.1016/J.COMCOM.2013.04.004.

[20] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005, doi: 10.1109/TPAMI.2005.159.

[21] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "BotOrNot," pp. 273–274, 2016, doi: 10.1145/2872518.2889302.

[22] S. Cresci, R. di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “DNA-Inspired Online Behavioral Modeling and Its Application to Spambot Detection,” *IEEE Intelligent Systems*, vol. 31, no. 5, pp. 58–64, Sep. 2016, doi: 10.1109/MIS.2016.29.

[23] C. Yang, R. Harkreader, and G. Gu, “Empirical evaluation and new design for fighting evolving twitter spammers,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1280–1293, 2013, doi: 10.1109/TIFS.2013.2267732.

[24] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang, “Twitter spammer detection using data stream clustering,” *Information Sciences*, vol. 260, pp. 64–73, Mar. 2014, doi: 10.1016/J.INS.2013.11.016.

[25] S. Lee, B. Schowe, and V. Sivakumar, “Feature Selection for High-Dimensional Data with RapidMiner,” *undefined*, 2012, doi: 10.17877/DE290R-14289.

[26] A. Aboud, R. Fdhila, and A. M. Alimi, “Dynamic Multi Objective Particle Swarm Optimization Based on a New Environment Change Detection Strategy,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Nov. 2017, vol. 10637 LNCS, pp. 258–268. doi: 10.1007/978-3-319-70093-9_27.

[27] A. Aboud, R. Fdhila, and A. M. Alimi, “MOPSO for dynamic feature selection problem based big data fusion,” in *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*, Feb. 2017, pp. 3918–3923. doi: 10.1109/SMC.2016.7844846.

[28] A. Aboud, R. Fdhila, A. Hussain, and A. M. Alimi, “A novel Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization (DPb-MOPSO) algorithm,” TechRxiv, Dec. 2020. doi: 10.36227/TECHRXIV.13325354.V1.

[29] A. Aboud, N. Rokbani, S. Mirjalili, and A. Alimi, “A Distributed Bi-behaviors Crow Search Algorithm for Dynamic Multi-Objective Optimization and Many-Objective Optimization,” Sep. 2021, doi: 10.36227/TECHRXIV.16607858.V2.

[30] A. Aboud *et al.*, “A Distributed Multifactorial Particle Swarm Optimization Approach,” Dec. 2021, doi: 10.36227/TECHRXIV.17260040.V1.

[31] A. Aboud *et al.*, “DPb-MOPSO: A Novel Dynamic Pareto bi-level Multi-Objective Particle Swarm Optimization Algorithm,” Dec. 2021, doi: 10.36227/TECHRXIV.17207576.V1.

[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[33] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[34] D. A. Savic, G. A. Walters, and J. W. Davidson, “A Genetic Programming Approach to Rainfall-Runoff Modelling,” *Water Resources Management 1999 13:3*, vol. 13, no. 3, pp. 219–231, 1999, doi: 10.1023/A:1008132509589.

[35] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[36] B. Xue, M. Zhang, … W. B.-I. T. on, and undefined 2015, “A survey on evolutionary computation approaches to feature selection,” *ieeexplore.ieee.org*, Accessed: Feb. 02, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7339682/

[37] A. W. Whitney, “A Direct Method of Nonparametric Measurement Selection,” *IEEE Transactions on Computers*, vol. C–20, no. 9, pp. 1100–1103, 1971, doi: 10.1109/T-C.1971.223410.

[38] T. Marill and D. M. Green, “On the Effectiveness of Receptors in Recognition Systems,” *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 11–17, 1963, doi: 10.1109/TIT.1963.1057810.

[39] Y.-S. Jeong, K. S. Shin, and M. K. Jeong, “An evolutionary algorithm with the partial sequential forward floating search mutation for large-scale feature selection problems,” *Journal of the Operational Research Society 2014 66:4*, vol. 66, no. 4, pp. 529–538, Mar. 2014, doi: 10.1057/JORS.2013.72.

[40] S. M. Winkler, M. Affenzeller, W. Jacak, and H. Stekel, “Identification of cancer diagnosis estimation models using evolutionary algorithms: A case study for breast cancer, melanoma, and cancer in the respiratory system,” *Genetic and Evolutionary Computation Conference, GECCO’11 - Companion Publication*, pp. 503–510, 2011, doi: 10.1145/2001858.2002040.

[41] Y. Zhang, D. Gong, Y. Hu, and W. Zhang, “Feature selection algorithm based on bare bones particle swarm optimization,” *Neurocomputing*, vol. 148, pp. 150–157, Jan. 2015, doi: 10.1016/J.NEUCOM.2012.09.049.

[42] B. Xue, M. Zhang, and W. N. Browne, “Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms,” *Applied Soft Computing*, vol. 18, pp. 261–276, May 2014, doi: 10.1016/J.ASOC.2013.09.018.

[43] S. M. Vieira, J. M. C. Sousa, and T. A. Runkler, “Two cooperative ant colonies for feature selection using fuzzy models,” *Expert Systems with Applications*, vol. 37, no. 4, pp. 2714–2723, Apr. 2010, doi: 10.1016/J.ESWA.2009.08.026.

[44] K. K. Bharti and S. Pandey, “Fake account detection in twitter using logistic regression with particle swarm optimization,” *Soft Computing*, vol. 25, no. 16, pp. 11333–11345, Aug. 2021, doi: 10.1007/S00500-021-05930-Y/FIGURES/5.

[45] G. Lingam, R. R. Rout, and D. V. L. N. Somayajulu, “Deep Q-Learning and Particle Swarm Optimization for Bot Detection in Online Social Networks,” *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*, Jul. 2019, doi: 10.1109/ICCCNT45670.2019.8944493.

[46] J. Sun, B. Feng, and W. Xu, “Particle swarm optimization with particles having quantum behavior,” *Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004*, vol. 1, pp. 325–331, 2004, doi: 10.1109/CEC.2004.1330875.

[47] C. A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, vol. 2, pp. 1051–1056, 2002, doi: 10.1109/CEC.2002.1004388.
[48] Y. Qi, Q. Zhang, X. Ma, Y. Quan, and Q. Miao, "Utopian point based decomposition for multi-objective optimization problems with complicated Pareto fronts," *Applied Soft Computing*, vol. 61, pp. 844–859, Dec. 2017, doi: 10.1016/J.ASOC.2017.08.036.
[49] A. Farkiya, P. Saini, S. Sinha, and S. Desai, "Natural Language Processing using NLTK and WordNet," 2015, Accessed: Feb. 02, 2022. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.735.2951&rep=rep1&type=pdf
[50] H. Kaur, V. Mangat, and Nidhi, "A survey of sentiment analysis techniques," *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, pp. 921–925, Oct. 2017, doi: 10.1109/I-SMAC.2017.8058315.
[51] L. Jinxian and L. Hui, "A density-based clustering over evolving heterogeneous data stream," *2009 Second ISECS International Colloquium on Computing, Communication, Control, and Management, CCCM 2009*, vol. 4, pp. 275–277, 2009, doi: 10.1109/CCCM.2009.5267735.
[52] M. R. Ackermann, M. Märtens, C. Raupach, K. Swierkot, C. Lammersen, and C. Sohler, "StreamKM++," *Journal of Experimental Algorithmics (JEA)*, vol. 17, May 2012, doi: 10.1145/2133803.2184450.
[53] S. van Dongen, "Graph Clustering Via a Discrete Uncoupling Process," *http://dx.doi.org/10.1137/040608635*, vol. 30, no. 1, pp. 121–141, Feb. 2008, doi: 10.1137/040608635.
[54] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 70, no. 6, p. 6, Dec. 2004, doi: 10.1103/PHYSREVE.70.066111/FIGURES/3/MEDIUM.

Ahlem Aboud she is currently a PhD student in computer science in ISITCom, University of Sousse and REGIM-lab: REsearch Group on Intelligent Machines at ENIS Sfax, University of Sfax. She received his engineering degree in 2015 from the ISIMS Sfax. She is currently IEEE student member since October 2015. His research interests include, dynamic multi-objective optimization problem, evolutionary computation, collective intelligence methods, and machine learning algorithms.

Nizar Rokbani He is an Assit Prof of Industrial Computing with Institute of Applied Science and technology of Sousse, Tunisia Since 2014. He graduated in Electrical Engineering from the National Engineering School of Tunis, ENIT in 1995. He obtained a Master degree in industrial computing in 2003 from the National Engineering School of Sfax, ENIS, and a PHD in Electrical Engineering in 2013. His research interests include applications of intelligent techniques such as Swarm intelligence, computational intelligence, fuzzy logic, evolutionary algorithms to robotic systems and industrial processing. He funded several society chapters, including RAS and SMC 2009, and OES 2016, he is also the founding counselor of ISSAT Sousse Sb 2015, and Polytech Sousse Sb, 2016. He is The IEEE RAS Tunisia Chair 2015-2018, the chapter was awarded: R8 Outstanding Society Chapter 2016, Best RAS Society Chapter award for 2017, and Outstanding Society chapter award of Tunisia Section for 2017.

Seyedalii Mirjalili is a professor and the director of the Centre for Artificial Intelligence Research and Optimization at Torrens University Australia. He is internationally recognized for his advances in Swarm Intelligence and Optimization, including the first set of algorithms from a synthetic intelligence standpoint - a radical departure from how natural systems is typically understood - and a systematic design framework to reliably benchmark, evaluate, and propose computationally cheap robust optimization algorithms. Seyedali has published over 300 publications with over 35,000 citations and an H-index of 65. As the most cited researcher in Robust Optimization, he is in the list of 1% highly-cited researchers and named as one of the most influential researchers in the world by Web of Science. Seyedali is a senior member of IEEE and an associate editor of several journals including Neurocomputing, Applied Soft Computing, Computers in Biology and Medicine, Advances in Engineering Software, Applied Intelligence, and IEEE Access. His research interests include Robust Optimization, Engineering Optimization, Multi-objective Optimization, Swarm Intelligence, Evolutionary Algorithms, and Artificial Neural Networks. He is working on the application of multi-objective and robust meta-heuristic optimization techniques as well.

Dr Abdulrahman Qahtani is currently Assistant professor in computer science department at Taif University. Dr. Qahtani received his PhD from Southampton University in 2015. He has extensive experience in software engineering and development process in a distributed domain. His research focused on customization process across organizational boundaries. Recently, he applied machine learning algorithms on software engineering data to predict time and cost estimation for multi-client's projects.

Fahd Alharithi received bachelor's degree in computer science from Taif University (TU), Saudi Arabia, in 2008, and the master's degree in computer science from University of New Haven (UNH), United States in 2012 and the Ph.D. degree in computer science from Florida Institute of Technology, United States in 2019. He is currently an Assistance Professor with the College of Computers and Information Technology, Taif University, Saudi Arabia. His research interests include human–computer interaction, cloud computing, the Internet of Thing, artificial intelligent, and machine learning.

Omar Almutiry received the PhD degree from the University of Southampton, UK, in 2017. Since 2018, he has been with the Applied Computer Science College, King Saud University, Saudi Arabia, where he is currently working as an Assistant Professor with the Department of Applied Computer Sciences. He is also assistant to the General Director of the Almuzahmiyah branch for academic affairs and development. His research areas include data science, health informatics, deep learning applications in healthcare and medical fields.

Habib Dhahri was born in Sidi Bouzid, Tunisia in 1975. He graduated in Computer Science in 2001, obtained his PhD in Computer Science in 2013 from the National Engineering School of Sfax (ENIS). He is now an assistant professor in Computer Science at the King Saud University. His area of interest includes computational intelligence, soft computing techniques and Machine Learning for Healthcare Data. He has authored and co-authored more than 30 publications in journals and conferences. He also serves as a reviewer for several international scientific journals.

Amir Hussain he received his B.Eng (highest 1st Class Honours with distinction) and Ph.D degrees, from the University of Strathclyde, Glasgow, U.K., in 1992 and 1997, respectively. Following postdoctoral and academic positions at the Universities of West of Scotland (1996-98), Dundee (1998-2000) and Stirling (2000-18) respectively, he joined Edinburgh Napier University (in Scotland, UK) in 2018 as Professor and founding Head of the Data Science and Cyber Analytics (DSCA) Research Group (managing 20 academics and research staff). As part of the latter, he is also founding Head of the Cognitive Big Data and Cybersecurity (CogBiD) Research Lab

Adel M. Alimi (S'91–M'96–SM'00) was born in Sfax, Tunisia, in 1966. He received the Degree in electrical engineering in 1990, the Ph.D. degree in electrical and computer engineering from the Polytechnic University of Montreal, Montréal, QC', Canada, in 1995, and the HDR degree in electrical and computer engineering from the National Engineering School of Sfax, Sfax, in 2000. He is currently a professor of electrical and computer engineering with the University of Sfax, Sfax. His current research interests include applications of intelligent methods (neural networks, fuzzy logic, and evolutionary algorithms) to pattern recognition, robotic systems, vision systems, industrial processes, intelligent pattern recognition, learning, analysis, and intelligent control of large-scale complex systems. Prof. Alimi is a member of IAPR, INNS, and PRS. He is a member and an Associate Editor of the Editorial Board of many international scientific journals, including Pattern Recognition Letters, Neurocomputing, Neural Processing Letters, International Journal of Image and Graphics, Neural Computing and Applications, International Journal of Robotics and Automation, and International Journal of Systems Science. He was a Guest Editor of several special issues of international journals, including Fuzzy Sets and Systems, Soft Computing, the Journal of Decision Systems, Integrated Computer-Aided Engineering, and Systems Analysis Modelling and Simulations.