Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 3873-3889
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6186

Quantum inspired genetic algorithm model based thirteen types automatic modulation classification

Hussain A. Rasool[a,*], Bayan M. Sabbar[a]

[a] *Department of Information and Communication Engineering, Al-Nahrain University, Iraq*

(Communicated by Madjid Eshaghi Gordji)

Abstract

The popularity of automatic modulation categorization (AMC) is high in recent years owing to the many advantages. When it comes to communication, reliability in an AMC is very critical. Increasing the number of signals exponentially increases the cost of using the AMC. Precise classification methods, such as neural networks, in which either the parameters of the neural network or the dimensions of the input or output variables are modified dynamically, are not successful in obtaining high accuracy results. To improve the accuracy of the modulation categorization, this study employs a "QIGA" feature selection model based on a Quantum (inspired) Genetic Algorithm (QIGA). QIGA is used to choose the correct functionality and to limit the number of examples that must be learned so that the overall system time is shortened and the cost of computing is reduced. Selecting excellent characteristics is enhanced via quantum computing and this is done to lower the complexity of the solutions. The internal validation results demonstrated that the QIGA model significantly improved the statistical match quality and significantly outperformed the other models.

Keywords: Quantum computing, Genetic Algorithm, Automatic Modulation Classification.

1. Introduction

A localization algorithm's primary goal is to determine where a node is located. However, it must meet the necessary criteria to be helpful. Mostly, the type of application for which the localization method is designed defines the parameters. This section describes some of the basic architectural characteristics that any localization algorithm should have [1].

*Corresponding author

Email addresses: hussienmscit2@gmail.com (Hussain A. Rasool), bayan.mahdi@coie-nahrain.edu.iq (Bayan M. Sabbar)

Received: June 2021 *Accepted:* October 2021

- Because RF-based localization algorithms are so useful, they are quite popular. An RF transmitter is placed in the sensor nodes for a limited range. This radio capacity is effective in addition to its main purpose of data transmission, thus it's helpful for localization as well.
- The fundamental nature of a wireless sensor network is self-configuring. Careful consideration should be given to the ad hoc nature of the network by the localization method.
- To enable the localization process to respond fast, the nodes should be able to determine their position as quickly as feasible. Deploying sensor nodes will be done quickly using this technology.
- Accuracy for the application for which it is being utilized should be adequate for the positioning of the sensor node.
- It must be stable in order to function when circumstances are less than ideal.
- In order to be adaptable, the algorithm should be able to readjust even if sensor nodes are installed or removed. Additionally, the method is useful for sensor networks that include from a small number of nodes to many nodes.
- In order to function while self-sufficient, sensor nodes, being autonomous, are likely to need their own power supply. The localization algorithm should therefore be energy-effective and, ideally, energy-conscious.
- To make sure the translation algorithm can react to changes in the number of beacon nodes, it should be able to. When the number of accessible beacon nodes grows, the system should also be able to anticipate the approximate location of nodes. But as the number of nodes grows, node predictions may vary in accuracy.
- widely accessible beacon nodes provide for better localization calculations; a localization algorithm can get a more accurate estimate of node positions using a greater number of widely accessible beacon nodes.

The categorization of prediction methods takes into consideration three main factors: detection accuracy, applicability and configurability. In order for this to work, there must be consistency. Approval is just a requirement but is not sufficient on its own. When an object is designed specifically for a certain person, the reasons for utilizing it may have no explanatory value. Regression analysis is an established and often used method that just needs the support of a well-intentioned tool. The level of information and dedication required to construct the neural network is significant. There are several kinds of experiments that seem to favor trial and error [2].

GA is an adaptive search method that is used in feature selection classification modulations, where it utilizes heuristic search. When it comes to the degree of difficulty, GA was shown to perform better than NN. GA is also very reliable when selecting the most relevant characteristic for the classification process, which inevitably results in significant problems. In order to attain both convergence and efficiency, the NN structure optimization was observed. Various kinds of design optimization results may be obtained in neural network design optimization. Neural network development may not use an optimum method. A new "teaching" algorithm may be developed using a genetic algorithm [4]. In Figure 1, this method is shown.

The range of variables utilized will affect the performance of a binary GA. This gene selection was designed to identify those who would be imitated or changed. But having fewer children confers

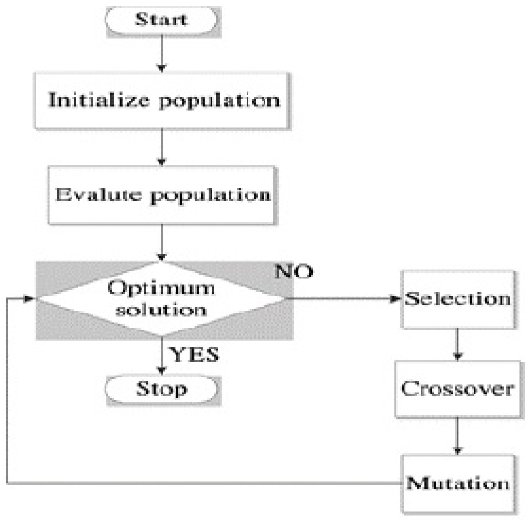

Figure 1: Genetic Algorithm Flowchart

a smaller chance of being chosen, which is necessary for the discovery process to be worldwide and not concentrated in a certain area. The three main techniques used in selection are the wheel of fortune, a tournament and the wheel of fortune. The authors of the book entitled [5] provide more information about the book. In the research done by O'Neill, the GA-dependent tournament selection was reported to achieve a minimum total distance while running quicker than the other two selection strategies. However, for smaller problem sizes, it is only true that these patterns hold true. It is suggested that when the problem of expansion grows, the tournament is also at danger of early convergence with the use of the proportional roulette wheel.

Because of its superior capability in handling new challenges, quantum computing is particularly effective [9]. Researchers believe that quantum computers' strengths are due to their use of micro- and macro-space searches, which is associated with increased efficiency and better outcomes [11]. A superposition state is a technique of addressing combinational problems that change component variables drawn from the principles of quantum-mechanics in scanning methods such as quantum circuits [12]. In quantum computing, the quantum gadget will be found in multiple locations (states) simultaneously as it waits for processing. When two or more states exist in exact superposition, it is exactly when quantum computing power is being exerted. Until quite recently, computers have usually been in a single state. A superposition of states may be used to house quantum computers. This is the final parallel processing of these experiments [13].

It is targeted at the investigation of how Quantum-inspired Genetic Algorithm (QGAGA) modulates (QIGA). Fitness superposition is used to enhance the selection method and to keep the calculation costs down. Furthermore, it has the capacity to foster variety and to manage population crossover (divergence) and convergence (convergence) in the course of mutations (divergence) and cross-cutting (convergence) interactions [13]. The paper's main contributions are: (1) this article is concerned with speeding up the use of conventional classification algorithms by replacing them with QIGA, while trying to select the features; and (2) it talks about possible problems and concerns associated with using traditional optimization algorithms and is committed to devising a new algorithm. It is shown through a series of tests that the QIGA method is faster and more accurate than any previous wider prototypes.

The remainder of the paper is organized accordingly. The latest related work is seen in section 2. The detailed process of the proposed QIGA algorithm is given in Section 3. Section 4 presents the findings of the experiments and their discussion. We end this paper in Section 5.

2. Related Work

AMC can be done in two ways: probability-based (LB) approach [?]. [14] and a feature-based (FB) method [15]. The LB technique verifies signal reception for different probability solutions with indeterminate modulation parameters that are dependent on the spectral density of power (PSD). While the LB solution achieves the greatest outcomes, it comes with a significant computational cost. When a frequency offset is present, one disadvantage is the impact of residual channel effects, as well as phase and timing problems caused by model mismatch [16]. On the other hand, the FB approach makes certain trade-offs in terms of accuracy but gains in terms of smoother execution owing to reduced computing complexity [17].

Three characteristics of the FB technique are frequently mentioned in the text: instantaneous information [18], wavelet coefficients [15, 21] and higher-order statistics [22, 25]. Following feature extraction, the AMC moves on to categorization. Previously published study found that artificial neural networks (ANNs) and genetic algorithms (GAs) were predominantly utilized for classification, whereas KNNs and genetic programming (GPs) were used on a case-by-case basis [2, 26]). The authors of [27] conducted a multi-class classification using specified thresholds and obtained findings with the assistance of general practitioners. The fundamental flaw is that threshold values are inconvenient. Not only do we have to manually establish these threshold numbers, but it's also inconvenient and time-consuming. In, we discussed the weighted fitness function for data categorization [28]. The workout function has been modified online to give more weight to difficult-to-identify data. The authors of [29] proposed the idea of dividing n-class classification into multi-class classification using GP. This approach has inherited the two-class dilemma's simplicity. In [30], the authors examined the performance of AMC machine learning algorithms utilizing Rayleigh and AWGN fading networks.

3. Motivation

A major topic in the digital communication area is AMC (automatic modulation categorization). While using traditional techniques, however, it is not possible to get a very accurate categorization of modulation. Additional issues include: expensive processing costs, non-convergence to optimum global convergence and premature convergence with current GA-based prediction techniques. The systematic QIGA model is adequate for eliminating potential mistakes in the GA-based modulation categorization. This paradigm, which is capable of tackling grouping, is crucial if we want to maintain variety in the population.

4. Proposed Algorithm

This section explains the automatic modulation categorization quantum-inspired prediction technique. Each of the modulation types (or symbols) used in the input signal comprise thirteen of the following modulation methods: BPSK, QPSK, 8PSK, 16QAM, 64QAM, 256QAM, 2PAM, 4PAM, 8PAM, 16PAM,2Fsk,4Fsk and 8Fsk. To obtain the optimal modulation classification functions, the system uses GA. In order to discern the variation in population sizes, quantum computing is used inside the proposed framework to control the randomness supplied by probabilistic models of quantum chromosomes, which are known as qubits. As each new generation differs greatly from the last, it takes less generations to get to the optimal answer. In the next sections, the measurements and features are explained in more depth.

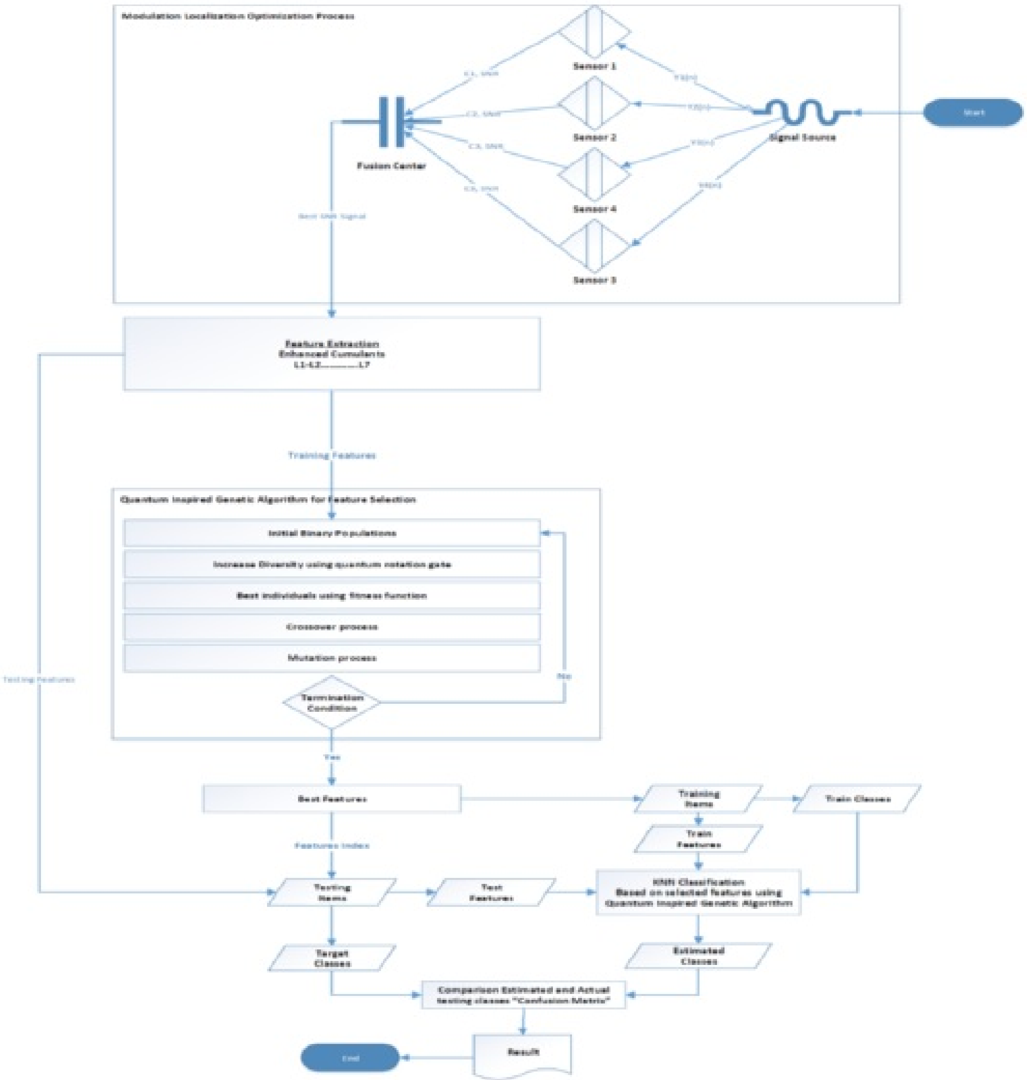

Figure 2: Quantum inspired Genetic Algorithm Modulation Classification model

5. Building Database

In this randomly generated dataset, each sensor will repeatedly broadcast its random signal before it is received by the fusion center. Fusion center will analyze the signal-to-noise ratio and then pick the strongest signal for its use. The dataset's characteristics are known as "Enhanced Cumulants." These descriptive words are kept with the categorization of each modulation in a database essential to the classification of each modulation.

6. Quantum Genetic Algorithm Model

Recently, the Q-bit format was used for minimization issues in QIGA based on the idea and principle of quantum computing. The representation has the property of being capable of representing any linear superposition. A Q-bit is the smallest unit of information that can be stored in a two-state quantum computer. It may be in either the "1" or "0" state, or in any superposition "two states". A bit of one Q-state may be stated as:

$$|\psi_s\rangle = \gamma |\, 0\rangle + \beta \, |1\rangle \tag{6.1}$$

The amplitudes of the given conditions are specified by the values of α and β, which are difficult integers. $|\, \gamma \,|^2$ and $|\beta|^2$, If the Q-bit is discovered in the "1" state or the "0" state, then the chance is 100 percent. It is assured that by normalizing the state to the unity, it would be ensured that:

$$|\gamma_1|^2 + |\beta_2|^2 = 1 \tag{6.2}$$

To maintain and keep track of one gene in QIGA, single-qubit is used. In each qubit, there exists the possibility of the qubit being in a "1" state, a "0" state, or any kind of superposition of the two. More accurately, this gene's data cannot be considered completely reliable, but there is still the possibility that it will complete to all available information. Genes here each have one-bit. The multi-qubit system is used to encapsulate the multi-state operator node, for the general case.

$$q_j^t = \left[\begin{array}{c|c|c|c|} \gamma_1^t & \gamma_2^t & \cdots & \gamma_m^t \\ \beta_1^t & \beta_2^t & \cdots & \beta_m^t \end{array} \right] \tag{6.3}$$

chromosome of the t^{th}*generation and the* j^{th}

individual is represented by q_j^t and m is the gen index number. Because of qubit encoding, the superposition of many states may be instantly embodied by one person, leading to greater variety with the QIGA than with the standard GA method. A derivation of convergence can be found in [1] which asserts that it may be attained via the qubit statement. As $|\gamma|^2$ or $|\beta|^2$ attitudes to 0 or l, the qubit-chromosome shares to one-state.

For each qubit is prepared to$(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$. In other words, this authorizes that the possible state of a single-qubit gene may definite the superposition of all possible states with equal chance. Use Quantum-rotation gate for the software update implementation method [43,45].

$$\bigcup(\theta_i) = \begin{bmatrix} \cos\theta_{\ i} & -\sin\theta_{\ i} \\ \sin\theta_{\ i} & \cos\theta_{\ i} \end{bmatrix} \tag{6.4}$$

where $\bigcup$ is a unitary operation on any single qubit and θ_i is the angle of qubit-rotation, defined as:

$$\theta_i = S\left(\gamma_m, \beta_m\right) * \Delta\theta_i \tag{6.5}$$

the symbol of θ_i *is* $S\left(\gamma_m, \beta_m\right)$ that controls the direction, and $\Delta\theta_i$ is the rotation-gate magnitude which is illustrated in Fig. 3 and So, γ_m^* and β_m^* are designed as:

$$\begin{bmatrix} \gamma_m^* \\ \beta_m^* \end{bmatrix} = \bigcup(\theta) \begin{bmatrix} \gamma_m \\ \beta_m \end{bmatrix} \tag{6.6}$$

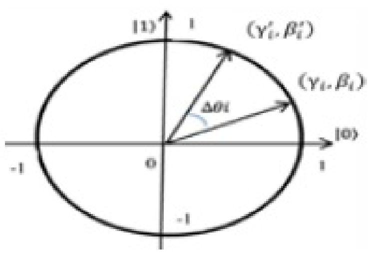

Figure 3: Rotation Gate for Qubit

Table 1: Rotation angle selection strategy

x_i	b_i	$f(x_i) > f(b_i)$	$\Delta\theta_i$ δ	$S(\gamma_m, \beta_m) * \Delta\theta_i$			
				$\gamma_m * \beta_m > 0$	$\gamma_m * \beta_m < 0$	$\gamma_m = 0$	$\beta_m = 0$
0	0	False	0	0	0	0	0
0	0	True	0	0	0	0	0
0	1	False	0	0	0	0	0
0	1	True	δ	1	-1	0	±1
1	0	False	0	0	0	0	0
1	0	True	δ	-1	1	±1	0
1	1	False	0	0	0	0	0
1	1	True	0	0	0	0	0

Once all of the updated populations have been measured, the last step is to get a set of determined explanations. The method for doing measurement as: create a random number τ in [0, 1]. If $\tau > |\gamma_{ij}^t|^2$ if the evaluating result is 1, else it is 0. Then consider the set of solutions together with their fitness, the best tree and its fitness, as well as the binary solution's set of solutions. *P(t)* is a previously nominated and deposited function that will be used by future generations. $f(x_i)$ of the most recent quantified rate of an item, as shown in Table 1, is used to determine whether or not to renew the item q_j^t with the current evolutionary aim's fitness$f(b_i)$.

As a result, control the qubit of the corresponding bit in order to achieve the probability scale and continue on the path of assisting the existence of b_i. In addition, the angle step of the update is represented by the symbol δ. The value of δ has an effect on the pace of convergence; if the value is high, the resolution may shift away from the local optimum or may arrive at the local optimum too soon. As a result of this, the dynamic tune δ of is accepted and it receives a value flanked by the values 0.2 and 0.8 as determined by the variance of the genetic generations.

Table 2: Genetic Algorithm Parameters

Parameter	Default Value
Population size	50
Generation Number	10
Crossover Ratio	0.7
Mutation Ratio	0.3

7. Simulation Result and Analysis

This portion validates the performance of the proposed method by conducting a series of experiments. Furthermore, the output is contrasted with the standard genetic algorithm in order to determine the precision of the classification of the proposed solution. Many separate modulation databases may be used to assess the feasibility of the proposed model. Genetic Function Approximation (GFA) algorithm provides a new solution to the AMC problem. Unlike most other research algorithms, GFA offers multiple models where model populations are generated by the evolution of random initial models using a genetic algorithm.

The framework is introduced in the form of a MATLAB library built to be easy to use in custom applications. Tests was carried out on a computer with Intel(R) Zeon(R) CPU E5430@ 2.66GHz (2 processor), 16GB RAM PC operating Microsoft Windows 10-64 bit. The findings of the simulation approve the potential of the proposed technique to obtain a detailed classification of modulation.

We concluded that our best approximation with 13 modulations and 700 generated signal samples was over 96.79% correct. It is confirmed that the proposed QIGA displayed superior modulation classification performance in comparison to the 90.83% GA algorithm as seen in fig. 4 and 5 in order. By increasing the dataset samples to 19950 sample, the accuracy increases. The increase in accuracy illustrates that the sufficient samples lead to better performance.

Overall, a quantum computing concept that inspires GA leads to a plurality of populations rather than a classical GA in most cases. This variation contributes to the achievement of optimal solutions for the appropriate fitness functions. Furthermore, when it comes to quantum chromosomes, the linear superposition of all conceivable binary states provides a significant deal more diversity than the traditional classical depiction. Implementation of the quantum rotation gate is required in order to get the chromosomal individuals closer to optimum solutions. The results of both kinds of classifiers are shown in Table 3 of this report.

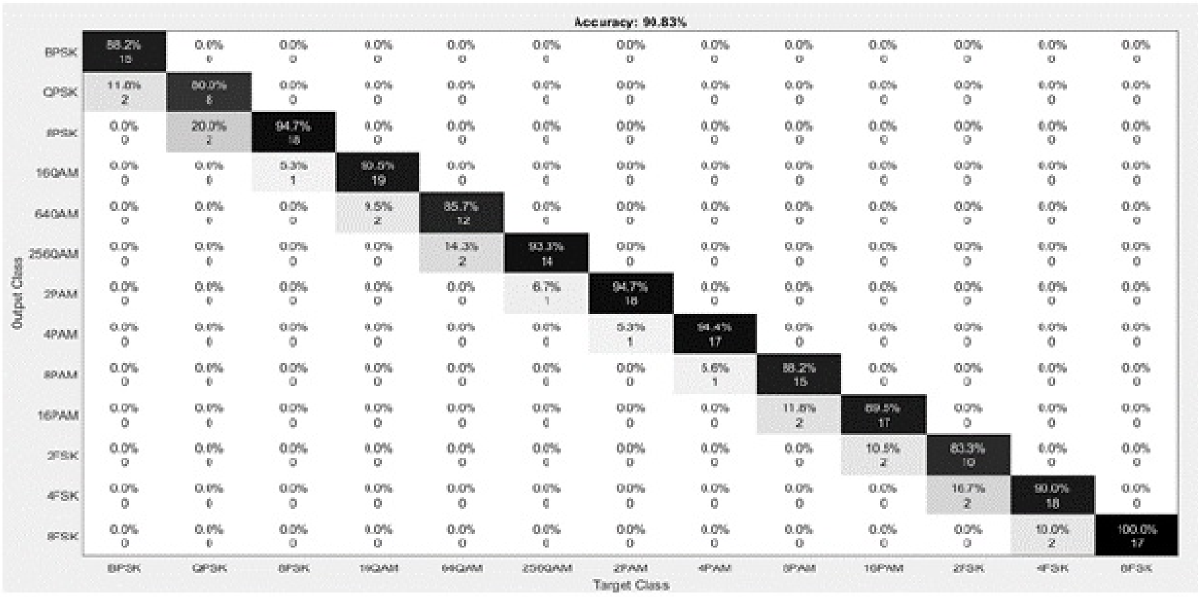

Figure 4: Experimental (target) versus classified modulation using the QIGA over

Table 3: Comparative result - 20% Testing and 80% Training

Method	*Accuracy*	TP	TN	FP	*FN*	*F- Score*
QIGA	**96.97**	**211**	**200.69**	**7**	**7**	**0.9697**
CGA	90.86	198	199.69	20	20	0.9083

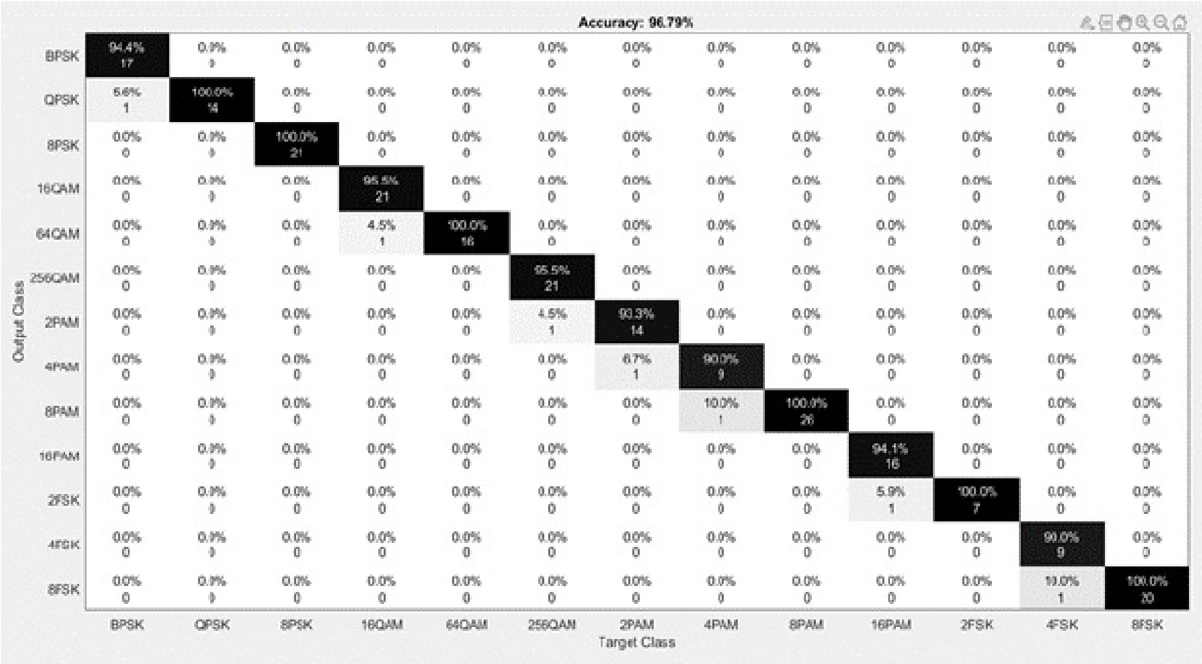

Figure 5: Experimental (target) versus classified modulation using the GA

Table 4: Comparative results (CGA Vs QIGP) -20% Testing and 80% Training

	CGA	QIGP
Generations	20	5
Populations	20	5
Accuracy	90.86	**96.97**
Elapsed time in seconds	5.817	**0.51**

Table 5: SNR Values by increasing Sensors

Sensors / Signal	SNR(dB)									
	1	2	3	4	5	6	7	8	9	10
BPSK	7.4465	7.4581	7.45814	7.4581	7.4581	7.4759	7.4759	7.4759	7.4759	7.4759
QPSK	3.2082	3.3712	3.37124	3.3712	3.3712	3.3712	3.3712	3.3712	3.3712	3.3712
8PSK	5.8725	5.8725	5.87257	6.0166	6.0337	6.0337	6.0337	6.0337	6.0337	6.0337
16QAM	5.2687	5.2782	5.27828	5.2782	5.2782	5.2861	5.2861	5.2861	5.2861	5.2861
64QAM	2.9132	2.9132	2.91325	2.9209	2.9209	2.9209	2.9209	2.9242	2.9242	2.9266
256QAM	5.3413	5.3413	5.34136	5.3413	5.3413	5.3413	5.3413	5.3413	5.3413	5.3413
2PAM	6.3293	6.3473	6.34738	6.3473	6.3473	6.3608	6.3608	6.3608	6.3608	6.3608
4PAM	9.0197	9.0197	9.01977	9.0544	9.0590	9.0590	9.0590	9.0590	9.0590	9.0590
8PAM	3.6250	3.6665	3.66657	3.6665	3.6665	3.7093	3.7093	3.7093	3.7093	3.7093
16PAM	4.4606	4.5320	4.53201	4.5320	4.5320	4.5878	4.5878	4.5878	4.5878	4.5878
2FSK	5.8923	5.8923	5.89230	6.0144	6.0350	6.0350	6.0350	6.0350	6.0350	6.0350
4FSK	5.5270	5.5907	5.59074	5.5907	5.5907	5.6383	5.6383	5.6383	5.6383	5.6383
8FSK	5.4328	5.4859	5.48592	5.4859	5.4859	5.5225	5.5225	5.5225	5.5225	5.5225

Table 6: SNR, Time and Distance in case of 1 Sensor.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.446574	1	0.00000001700	5.1	47	0.515056
QPSK	3.208236	1	0.00000001700	5.1	47	0.47305
8PSK	5.87258	1	0.00000001700	5.1	47	0.507613
16QAM	5.268744	1	0.00000001700	5.1	47	0.47236
64QAM	2.913256	1	0.00000001700	5.1	47	0.514605
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.329324	1	0.00000001700	5.1	47	0.540983
4PAM	9.019776	1	0.00000001700	5.1	47	0.495276
8PAM	3.625063	1	0.00000001700	5.1	47	0.543898
16PAM	4.460692	1	0.00000001700	5.1	47	0.519479
2FSK	5.892306	1	0.00000001700	5.1	47	0.45631
4FSK	5.527001	1	0.00000001700	5.1	47	0.486476
8FSK	5.43285	1	0.00000001700	5.1	47	0.483184

Table 7: SNR, Node Number Time and Distance in case of 2 Sensors.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.458143	2	0.00000002567	7.7	41	0.485527
QPSK	3.371242	2	0.00000002567	7.7	41	0.488052
8PSK	5.87258	1	0.00000001700	5.1	47	0.507613
16QAM	5.278288	2	0.00000002567	7.7	41	0.50025
64QAM	2.913256	1	0.00000001700	5.1	47	0.514605
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.347384	2	0.00000002567	7.7	41	0.508336
4PAM	9.019776	1	0.00000001700	5.1	47	0.495276
8PAM	3.666571	2	0.00000002567	7.7	41	0.525369
16PAM	4.532015	2	0.00000002567	7.7	41	0.471417
2FSK	5.892306	1	0.00000001700	5.1	47	0.45631
4FSK	5.590746	2	0.00000002567	7.7	41	0.544094
8FSK	5.485925	2	0.00000002567	7.7	41	0.485646

Table 8: SNR, Node Number Time and Distance in case of 3 Sensors.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.458143	2	0.00000002567	7.7	41	0.485527
QPSK	3.371242	2	0.00000002567	7.7	41	0.488052
8PSK	5.87258	1	0.00000001700	5.1	47	0.507613
16QAM	5.278288	2	0.00000002567	7.7	41	0.50025
64QAM	2.913256	1	0.00000001700	5.1	47	0.514605
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.347384	2	0.00000002567	7.7	41	0.508336
4PAM	9.019776	1	0.00000001700	5.1	47	0.495276
8PAM	3.666571	2	0.00000002567	7.7	41	0.525369
16PAM	4.532015	2	0.00000002567	7.7	41	0.471417
2FSK	5.892306	1	0.00000001700	5.1	47	0.45631
4FSK	5.590746	2	0.00000002567	7.7	41	0.544094
8FSK	5.485925	2	0.00000002567	7.7	41	0.485646

Table 9: SNR, Node Number Time and Distance in case of 4 Sensors.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.458143	2	0.00000002567	7.7	41	0.485527
QPSK	3.371242	2	0.00000002567	7.7	41	0.488052
8PSK	6.016658	4	0.00000000667	2	45	0.44591
16QAM	5.278288	2	0.00000002567	7.7	41	0.50025
64QAM	2.920909	4	0.00000000667	2	45	0.462641
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.347384	2	0.00000002567	7.7	41	0.508336
4PAM	9.054424	4	0.00000000667	2	45	0.49813
8PAM	3.666571	2	0.00000002567	7.7	41	0.525369
16PAM	4.532015	2	0.00000002567	7.7	41	0.471417
2FSK	6.014456	4	0.00000000667	2	45	0.462192
4FSK	5.590746	2	0.00000002567	7.7	41	0.544094
8FSK	5.485925	2	0.00000002567	7.7	41	0.485646

Table 10: SNR, Node Number Time and Distance in case of 5 Sensors.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.458143	2	0.00000002567	7.7	41	0.485527
QPSK	3.371242	2	0.00000002567	7.7	41	0.488052
8PSK	6.033716	5	0.00000000567	1.7	57	0.51972
16QAM	5.278288	2	0.00000002567	7.7	41	0.50025
64QAM	2.920909	4	0.00000000667	2	46	0.462641
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.347384	2	0.00000002567	7.7	41	0.508336
4PAM	9.05909	5	0.00000000567	1.7	57	0.459881
8PAM	3.666571	2	0.00000002567	7.7	41	0.525369
16PAM	4.532015	2	0.00000002567	7.7	41	0.471417
2FSK	6.035066	5	0.00000000567	1.7	57	0.508111
4FSK	5.590746	2	0.00000002567	7.7	41	0.544094
8FSK	5.485925	2	0.00000002567	7.7	41	0.485646

Table 11: SNR, Node Number Time and Distance in case of 6 Sensors.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.475967	6	0.00000003300	9.9	31	0.507666
QPSK	3.371242	2	0.00000002567	7.7	41	0.488052
8PSK	6.033716	5	0.00000000567	1.7	57	0.51972
16QAM	5.286169	6	0.00000003300	9.9	31	0.485394
64QAM	2.920909	4	0.00000000667	2	45	0.462641
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.360841	6	0.00000003300	9.9	36	0.486841
4PAM	9.05909	5	0.00000000567	1.7	57	0.459881
8PAM	3.709381	6	0.00000003300	9.9	31	0.4622
16PAM	4.587853	6	0.00000003300	9.9	32	0.54191
2FSK	6.035066	5	0.00000000567	1.7	57	0.508111
4FSK	5.63833	6	0.00000003300	9.9	31	0.501048
8FSK	5.522541	6	0.00000003300	9.9	31	0.521431

Table 12: SNR, Node Number Time and Distance in case of 7 Sensors.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.475967	6	0.00000003300	9.9	30	0.507666
QPSK	3.371242	2	0.00000002567	7.7	41	0.488052
8PSK	6.033716	5	0.00000000567	1.7	57	0.51972
16QAM	5.286169	6	0.00000003300	9.9	31	0.485394
64QAM	2.920909	4	0.00000000667	2	45	0.462641
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.360841	6	0.00000003300	9.9	30	0.486841
4PAM	9.05909	5	0.00000000567	1.7	57	0.459881
8PAM	3.709381	6	0.00000003300	9.9	32	0.4622
16PAM	4.587853	6	0.00000003300	9.9	27	0.54191
2FSK	6.035066	5	0.00000000567	1.7	57	0.508111
4FSK	5.63833	6	0.00000003300	9.9	32	0.501048
8FSK	5.522541	6	0.00000003300	9.9	32	0.521431

Table 13: SNR, Node Number Time and Distance in case of 8 Sensors.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.475967	6	0.00000003300	9.9	31	0.507666
QPSK	3.371242	2	0.00000002567	7.7	41	0.488052
8PSK	6.033716	5	0.00000000567	1.7	57	0.51972
16QAM	5.286169	6	0.00000003300	9.9	32	0.485394
64QAM	2.924215	8	0.00000000800	2.4	39	0.532019
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.360841	6	0.00000003300	9.9	30	0.486841
4PAM	9.05909	5	0.00000000567	1.7	57	0.459881
8PAM	3.709381	6	0.00000003300	9.9	30	0.4622
16PAM	4.587853	6	0.00000003300	9.9	31	0.54191
2FSK	6.035066	5	0.00000000567	1.7	57	0.508111
4FSK	5.63833	6	0.00000003300	9.9	30	0.501048
8FSK	5.522541	6	0.00000003300	9.9	31	0.521431

Table 14: SNR, Node Number Time and Distance in case of 9 Sensors.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.475967	6	0.00000003300	9.9	29	0.507666
QPSK	3.371242	2	0.00000002567	7.7	41	0.488052
8PSK	6.033716	5	0.00000000567	1.7	57	0.51972
16QAM	5.286169	6	0.00000003300	9.9	32	0.485394
64QAM	2.924215	8	0.00000000800	2.4	39	0.532019
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.360841	6	0.00000003300	9.9	31	0.486841
4PAM	9.05909	5	0.00000000567	1.7	57	0.459881
8PAM	3.709381	6	0.00000003300	9.9	31	0.4622
16PAM	4.587853	6	0.00000003300	9.9	32	0.54191
2FSK	6.035066	5	0.00000000567	1.7	57	0.508111
4FSK	5.63833	6	0.00000003300	9.9	30	0.501048
8FSK	5.522541	6	0.00000003300	9.9	30	0.521431

Table 15: SNR, Node Number, Time and Distance in case of 10 Sensors.

Signal type	SNR(dB)	Node NO.	Time Second	Distance(KM)	AoA°	RSSI(dB)
BPSK	7.475967	6	0.00000003300	9.9	31	0.507666
QPSK	3.371242	2	0.00000002567	7.7	41	0.488052
8PSK	6.033716	5	0.00000000567	1.7	57	0.51972
16QAM	5.286169	6	0.00000003300	9.9	31	0.485394
64QAM	2.926688	10	0.00000000900	2.7	60	0.513533
256QAM	5.34136	1	0.00000001700	5.1	47	0.500605
2PAM	6.360841	6	0.00000003300	9.9	31	0.486841
4PAM	9.05909	5	0.00000000567	1.7	57	0.459881
8PAM	3.709381	6	0.00000003300	9.9	31	0.4622
16PAM	4.587853	6	0.00000003300	9.9	30	0.54191
2FSK	6.035066	5	0.00000000567	1.7	57	0.508111
4FSK	5.63833	6	0.00000003300	9.9	32	0.501048
8FSK	5.522541	6	0.00000003300	9.9	33	0.521431

Table 5-15 demonstrates SNR, Time and Distance in case 1,2,3,4,5,6,7,8,9 and 10 sensors 13 type of signal modulation. Table 4 demonstrates for both the CGA and QIGA algorithms for runtime and fitness evaluations. The generation number is 20, while the size of the generation number in QIGA is 5. The findings indicate that QIGAs with 5 individuals will have greater utility with respect to the best fitness as well as a mean fitness of CGA with 20 individuals with only 1/4 of CGA's elapsed time with simulation setup configuration as number of generations=20, crossover likelihood (0.7) and mutation probabilities (0.3).

8. Conclusion

In this paper, we provide a method for classifying the QIGA model using a feature selection strategy, which we call the feature selection strategy. By constructing a model based on the quantum rotational gate, researchers were able to take use of the unpredictability of quantum chromosomes represented by qubits, which was very simple in contrast to most traditional training methods. The model that was developed for a data set with thirteen modulations was shown below. We find that the best model generated by QIGA provides a more accurate categorization than the pre-specified model optimized by GA in our instance. This is due to the larger solution space which can be investigated in QIGA but is left to the evolutionary mechanism with varying probabilities resulting from the qubit overlay by means of the quantum rotation gate, rather than being pre-determined. We were able to significantly increase the accuracy of the AMC by combining the GA and the superposition principle and the results indicate that the measurement efficiency of the QIGA is much better than that of the CGA.

References

[1] M. Farooq-i-Azam and M.N. Ayyaz, *Location and position estimation in wireless sensor networks*, Curr. Status Future Trends, CRC Press, 2016.

[2] S. Kharbech, I. Dayoub, M. Zwingelstein-Colin and E.P. Simon, *On classifiers for blind feature-based automatic modulation classification over multiple-input–multiple-output channels*, IET Commun. 10(7) (2016) 790–795.

[3] O.A. Dobre, A. Abdi, Y. Bar-Ness and W. Su, *Survey of automatic modulation classification techniques: Classical approaches and new trends*, IET Commun. 1(2)(2007) 137–156.

[4] C. Mair, G. Kadoda, M. Lefley, K. Phalp, C. Schofield, M. Shepperd and S. Webster, *An investigation of machine learning based prediction systems*, J. Syst. Software 53(1) (2000) 23–29.

[5] M. O'Neill, L. Vanneschi, S. Gustafson and W. Banzhaf, *Open issues in genetic programming*, Genet. Program. Evolvable Mach. 11(3-4)(2010) 339–363.

[6] J. Kobashigawa, H.S. Youn, M. Iskander and Z. Yun, *Comparative study of genetic programming vs. neural networks for the classification of buried objects*, IEEE Antennas Propag. Soci. Int. Symp. 2009, pp. 1–4.

[7] M. Brezocnik, M. Kovacic and L. Gusel, *Comparison between genetic algorithm and genetic programming approach for modeling the stress distribution*, Materials Manuf. Processes, 20(3) (2005) 497–508.

[8] H. Guo, L.B. Jack and A.K. Nandi, *Feature generation using genetic programming with application to fault classification*, IEEE Trans. Syst. Man. Cybern. Part B (Cybern.) 35(1) (2005) 89–99.

[9] N.M. Razali and J. Geraghty, *Genetic algorithm performance with different selection strategies in solving TSP*, Proc. World Congress Eng. Hong Kong, Int. Assoc. Eng. 2(1) (2011) 1–6.

[10] D. Ristè, M.P. Da Silva, C.A. Ryan, A.W. Cross, A.D. Córcoles, J.A. Smolin, J.M. Gambetta, J.M. Chow and B.R. Johnson, *Demonstration of quantum advantage in machine learning*, npj Quantum Inf. 3(1) (2017) 1–5.

[11] Z. Laboudi and S. Chikhi, *Comparison of genetic algorithm and quantum genetic algorithm*, Int. Arab J. Inf. Technol. 9(3) (2012) 243–249.

[12] L. Wang, F. Tang and H. Wu, *Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation*, Appl. Math. Comput. 171(2) (2005) 1141–1156.

[13] S.Y. Kuo, Y.H. Chou and C.Y. Chen, *Quantum-inspired algorithm for cyber-physical visual surveillance deployment systems*, Comput. Networks 117 (2017) 5–18.

[14] A.O. Pittenger, *An Introduction to Quantum Computing Algorithms*, Springer Sci. Bus. Media, 2012.

[15] J. L. Xu, W. Su and M. Zhou, *Likelihood-ratio approaches to automatic modulation classification*, IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) 41(4)-(2010) 455–469.

[16] W. Su, *Feature space analysis of modulation classification using very high-order statistics*, IEEE Commun. Lett. 17(9) (2013) 1688–1691.

[17] F. Hameed, O.A. Dobre and D.C. Popescu, *On the likelihood-based approach to modulation classification*, IEEE Trans. Wirel. Commun. 8(12) (2009) 5884–5892.

[18] A.K. Nandi and E.E. Azzouz, *Algorithms for automatic modulation recognition of communication signals*, IEEE Trans. Commun. 46(4) (1998), 431–436.

[19] F. Wang and X. Wang, *Fast and robust modulation classification via Kolmogorov-Smirnov test*, IEEE Trans. Commun. 58(8) (2010) 2324–2332.

[20] K.C. Ho, W. Prokopiw and Y. Chan, *Modulation identification of digital signals by the wavelet transform*, IEE Proc.-Radar Sonar Navig. 147(4(2000) 169–176.

[21] L. Hong and K.C. Ho, *Identification of digital modulation types using the wavelet transform* , IEEE Mil. Commun. Conf. Proc. 1999, pp. 427-431.

[22] Z. Fucai, H. Yihua and H. Shiqi, *Classification using wavelet packet decomposition and support vector machine for digital modulations*, J. Syst. Eng. Electron. 19(5) (2008) 914–918.

[23] P. Li, F. Wang and Z. Wang, *Algorithm for modulation recognition based on high-order cumulants and subspace decomposition*, Int. Conf. Signal Process. 2006.

[24] M.R. Mirarab and M.A.Sobhani, *Robust modulation classification for PSK/QAM/ASK using higher-order cumulants*, Int. Conf. Inf. Commun. Signal Process. 2007, pp. 1–4.

[25] L. Shen, S. Li, C. Song and F. Chen, *Automatic modulation classification of MPSK signals using high order cumulants*, Int. Conf. Signal Process. 1 (2006).

[26] N. An, B. Li and M. Huang, *Modulation classification of higher order MQAM signals using mixed-order moments and Fisher criterion*, Int. Conf. Comput. Autom. Eng.(ICCAE), 3 (2010), pp. 150-153.

[27] M.W. Aslam, Z. Zhu and A. K. Nandi, *Automatic digital modulation classification using genetic programming with K-nearest neighbor*, Mil. Commun. Conf., 2010, pp. 1731–1736.

[28] Z. Shan, Z. Xin and W. Ying, *Improved modulation classification of MPSK signals based on high order cumulants*, Int. Conf. Future Compu. Commun. 2 V2-444.

[29] M. Zhang, V.B. Ciesielski and P.Andreae, *A domain-independent window approach to multiclass object detection using genetic programming*, EURASIP J. Adv. Signal Process. 8 (2003) 1–19.

[30] L. Zhang, L. B. Jack and A. K. Nandi, *Extending genetic programming for multi-class classification by combining k-nearest neighbor*, Proc. IEEE Int. Conf. Acoust. Speech Signal Process. 5(2005) v-349.