International Journal Of Artificial Intelegence Research
Vol 3, No 1, Juny 2019, pp. 33-46 ISSN: 2579-7298

Quantum Inspired Genetic Programming Model to Predict Toxicity Degree for Chemical Compounds

Saad Mohamed Darwish

Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Egypt
saad.darwish@gmail.com

ARTICLE INFO

Article history:
Received: 11may 2018
Revised: 06 June 2018
Accepted: 03 August 2018

Keywords:
Cheminformatics;
Quantum Computing;
Prediction;
Genetic Programming

ABSTRACT

Cheminformatics plays a vital role to maintain a large amount of chemical data. A reliable prediction of toxic effects of chemicals in living systems is highly desirable in domains such as cosmetics, drug design, food safety, and manufacturing chemical compounds. Toxicity prediction topic requires several new approaches for knowledge discovery from data to paradigm composite associations between the modules of the chemical compound; such techniques need more computational cost as the number of chemical compounds increases. State-of-the-art prediction methods such as neural network and multi-layer regression that requires either tuning parameters or complex transformations of predictor or outcome variables are not achieving high accuracy results. This paper proposes a Quantum Inspired Genetic Programming "QIGP" model to improve the prediction accuracy. Genetic Programming is utilized to give a linear equation for calculating toxicity degree more accurately. Quantum computing is employed to improve the selection of the best-of-run individuals and handles parsimony pressure to reduce the complexity of the solutions. The results of the internal validation analysis indicated that the QIGP model has the better goodness of fit statistics and significantly outperforms the Neural Network model.

I. Introduction

Cheminformatics is a part of computer science that plays an important role in collecting, storing and analyzing the chemical data. Cheminformatics mixes biology, chemistry, biochemistry, physics, statistics, mathematics, and informatics [1]. Toxicology deals with the quantitative calculation of toxicant effects to organisms in relation to the extent, duration, and frequency of exposure [2]. Toxicity prediction is considered one of the major disciplines of cheminformatics [1]. As the experimental determination of properties could be a pricey and time-consuming process, it is essential to develop mathematical predictive relationships to measure the toxicity scale [2]. Assessment of biological stimulates with a fast, unsophisticated, susceptible and cost- applicable technique can specify explicit information on toxicity [3].

Within the toxicity prediction, accuracy, explanatory value and configurability are used in assessing the utility and quality of prediction techniques [4]. Noticeably, a prediction method that not succeed to sustain a certain level of accuracy will not be adequate. However, the researchers believe that accuracy, by itself, is not a sufficient condition for acceptance. Furthermore, if a particular prediction is in some sense, surprising to the end user, it is harder to establish any rationale for the value generated (has no explanatory value). Regarding configurability, how much effort is required to build the prediction system in order to generate useful results. Regression examination is a well-recognized procedure including well-intentioned tool assistance. Nevertheless, it requires significant endeavor to form the neural net and it calls for a reasonable level of knowledge. Even though several groups of heuristics have been issued on this subject, these procedures' process largely to be one of trial and error. Consequently, it is complex to understand in what way ANN methods could be straightforwardly employed inside the estimation task setting by end-users [4].

In the literature, several statistical prediction approaches have been utilized for Quantitative Structure-Toxicity Relationship (QSTR), including discriminant analysis, principal component analysis, multiple linear regression, factor analysis, multivariate analysis, partial least squares, cluster analysis, and adaptive least squares [5]. These techniques are easy to implement and do not need a large computational cost, yet they have a less accurate prediction. Furthermore, Radial basis function (RBF) can be exploited to estimate QSTRs that verified to have a substantial predictive ability, as it is rapidly and cyclical, contrary to the major of current traditional training procedures [2]. RBFs have better generalization capabilities compared to linear regression models at the expense of the increased complexity of the model compared to a simple structure of a linear model. RBF is expensive and needs more time-consuming tests for resolving toxicity. The NN systems are usually employed once the connections between items cannot be inferred exactly by linear operates [2].

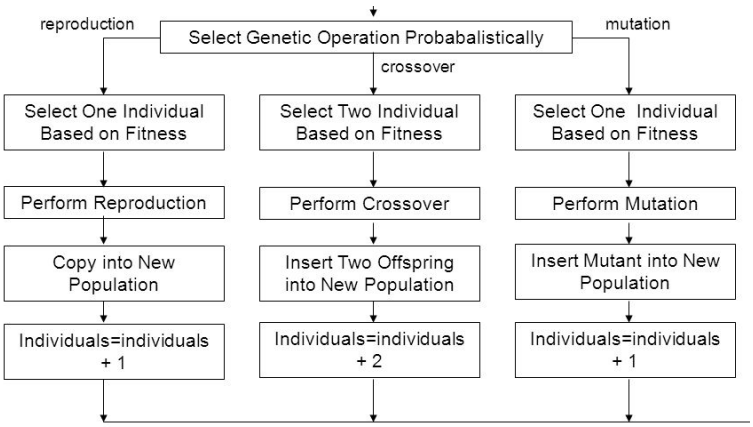

Fig. 1: Genetic Programming Flowchart

Genetic Programming (GP) is a form of biologically inspired automatic program induction where evolutionary algorithms are used to build computer programs (e.g. Prediction) and complex data structures [6]. GP showed better performance than NN in the various levels of problem difficulty. GP also revealed robustness to untrained data, which initiated problems for the NNs. The optimization of the NN's structure was observed to be integral in obtaining both convergence and acceptable performance. A well-defined style for construction optimization is not manifest in the case of NNs, and an overall ideal solution may not be applied. On the other hand, because of the global searching nature of GP, these problems with NN could be solved by using GP [7]. Genetic algorithm (GA) is a general approach for solving problems or "teaching" the machine to react to specific things in a specific way; whereas GP is a specific niche in GA, which lets the computer, write code by itself. Fig.1 illustrates the GP flowchart; see [8] for more details regarding the complete steps of GP.

One of the variables that actually affect the efficiency of GP is the selection. The selection operator is precisely prepared to confirm that appropriate participants of the population (with superior fitness) have a better chance of being nominated for reproducing or modify. Nevertheless, inferior participants of the population yet have a minor chance of being elected, and this is essential to guarantee that the exploration procedure is global and does not easily converge to the closest local optimum [9]. There are three main categories of selection; roulette wheel, rank-based roulette wheel, and tournament selection. See [10] for more details regarding these types. As stated in [10] the GA based tournament selection is more efficient in obtaining a minimum total distance with less number of generations and fastest iteration time matched to the other two policies [10]. Still, this is only valid to trivial problem size. As the size of problem growth, tournament selection, in addition to proportional roulette wheel turns out to be vulnerable to early convergence [10].

The key ability of quantum computing is to powerfully resolve specific problems that are computationally cost for a classical computing [11]. The power of the genetic-inspired quantum computing is in that the integration of micro-space and macro-space based search along with the synthesis of multiple various genetic operators; i.e. it explores large search spaces while preserving the relationship between efficiency and performance [12][13]. Conventional quantum-inspired exploration procedures employ the idea of superposition state to deal with combinatorial difficulties

that adjust each variable individually [14]. Superposition is the aptitude of a quantum system to be in numerous positions (states) at the same time while waiting for measuring. The actual strength in arrears of quantum computing is precisely the superposition of states. Classical computers are in one state at each instant. Quantum computers can be retained in a superposition of states. This is the final in parallel processing [15].

The main purpose of this paper is to investigate the accuracy of an adapted Quantum-Inspired Genetic programming (QIGP) for estimation of toxicity degree of chemical compounds. The suggested scheme relies on the concept of superposition fitness selection to enhance the traditional GP selection strategy, reduce the computational cost, and evade early convergence. Furthermore, it adopts a superposition in both of mutation (divergence) and crossover (convergence) operations to increase diversity and handle populations' selection at one time [15]. In general, the contributions in this paper are presented as follows: (1) this study attempts to expedite the traditional chemical compounds' toxicity prediction techniques by replacing the statistical algorithms with QIGP with the aim of producing an accurate mathematical linear prediction equation, and (2) addressing the issues associated with the computational cost of the traditional optimization algorithms, and an effort has been hired to build a new bio-inspired quantum computation model and enhancing its productivity as well. A chain of experimentations proofs that the suggested QIGP procedure is meaningfully accurate and faster than other widespread prototypes.

The remaining of this paper is structured as follows. Section 2 displays the current related work. Section 3 offers the in-depth process of the suggested QIGP algorithm. The experiment results and their discussion are given in Section 4. To close, in Section 5, we conclude this paper.

II. Research Method

In the literature, different methodologies are introduced for predicting the toxicity degree of chemical compounds. The most common methods are based on statistical analysis to discover the major associations among variables, i.e. latent variables to forming the covariance layouts in these spaces [16][17]. Despite the simplicity of these methods, there are certain restrictions and assumptions like the independence of the variables, and inherent normal distributions of the variables. For instance, Relevance Vector Machine (RVM) technique was employed to build the regression models for the prediction of oral acute toxicity rate [21]. However, the disadvantage of RVM includes non-parametric, in other words, the classifier is deduced directly from the data without assumptions about a probabilistic distribution.

Numerous statistical-based prediction methods have been utilized inside in recent years, among them discriminant analysis, principal component analysis (PCA), factor and cluster analysis [18]. These techniques focus on finding orthogonal projections of the dataset that contains the highest variance possible in order to 'find hidden linear correlations' between variables of the dataset. Therefore, if you have several parameters in the dataset that are linearly correlated, you can realize guidelines that characterize your data, but if the data is not linearly correlated, these approaches are not sufficient.

With the same objective, the neural network has also been used successfully in QSAR. The NN systems are normally exploited when the relationships cannot be inferred precisely by linear equations [19]. NNs regularly reveal configurations analogous to those obtained by persons. Nevertheless, shortcomings include its "black box" class, more computational load, and the inclination towards overfitting [22]. One more research including NN based on the radial basis function manner is presented with the intention of creating QSTR models for the prediction of toxicity. However, the prediction accuracy was not optimal; this is due to difficulties in the loading of training samples and learning process.

The existing neuro-prediction methods fail to handle the problem of minimizing the differences between the data values and their corresponding modeled values since they have a major limitation in selecting optimal factors (chemical compounds descriptors). Recently a lot of research interest is being shown in optimization techniques that can obtain a linear formulation for prediction schemes based on cross- correlation maximization which can alleviate the problem of local minima and at the same time reduce computational complexity [19] [23] [24].

An insight into the potential benefits of using optimization-based prediction models for toxicity real valued-data is provided in [25]. In this case, genetic programming is used in the induction of decision trees for application to two Eco- toxicity datasets of organic compounds, both with a large number of inputs and four classes obtained from equal frequency splitting of the endpoint. GP can handle a vast number of correlated descriptors; so that no form of feature extraction is required prior to forming the decision trees. However, the efficiency of this system depends mainly on the configuration parameters of genetic programming, which may often require considerable time to achieve its purpose.

Predicting the toxicity of chemical compounds is an important process in various fields such as drug manufacturing and chemicals industry. Yet, traditional toxicity prediction methods are not accurate enough. Moreover, current GP-based prediction approaches suffer from large computational cost, non-convergence to a global optimum and premature convergence. To avoid the potential errors in GP-based chemicals toxicity prediction, the formal QIGP paradigm is fit for precisely describing the toxicity degree. This paradigm able to address the multimodal functions, without that the population diversity tends to gradually disappear and may make the algorithm stagnate in local optima.

This section describes in details the proposed quantum inspired prediction system that aims to form an accurate linear equation to estimate phenols toxicity degree. The system's inputs are the five phenols descriptors in addition to the value of the toxicity degree obtained from laboratory experiments called Ciliate Tetrahymena Pyriformis. The system adapts GP to obtain the optimal tree representation for toxicity linear equation; this tree is formed based on normalized Euclidean objective function. The quantum computing is utilized inside the suggested system to exploit randomness offered by the probabilistic models of quantum chromosomes described by qubits to realize discrepancy in the population's assembly. This great variation in each generation leads to reduce the required number of GP generations to reach the optimal solution. Fig. 2 shows the main components of the suggested prediction system and how these components are linked together and the following subsections discuss its steps in details.

Step1: Building Database for Chemical Descriptors

Given the chemical data set that consists of four chemical compounds; the phenol descriptors are calculated; see [27] for a supplementary complete information about these descriptors. These descriptors are stored in a central database beside the corresponding toxicity value of each phenol. So there exist 221 phenols "records"; each record has five attributes (descriptors) and the last field contains the toxicity degree. These descriptors are used later to build GP tree structure.

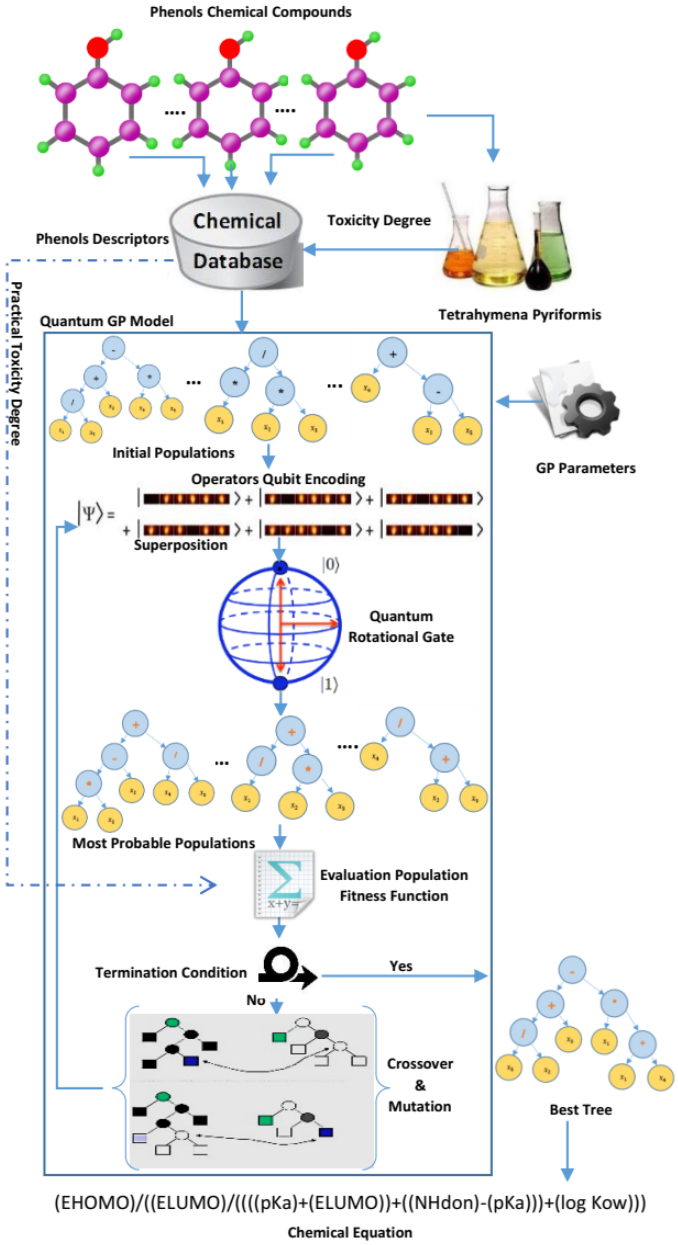

Fig. 2.Quantum inspired genetic programming Predication model

Step2: Quantum Genetic Programming Model

Genetic Programming is a symbolic optimization technique based on so-called "tree representation". This representation is extremely flexible, as trees can symbolize computer programs, mathematical equations or comprehensive prototypes of a process. A population in GP is a hierarchically organized tree involving functions and terminals. The functions and terminals are carefully chosen from a set of functions (operators) and a set of terminals. In our case, the set of operators F contains the basic arithmetic operations: $F=\{+,-,/,*\}$ for simple implementation (low computational cost). The set of terminals T contains the arguments for the functions. Herein, $T=\{x_i, i=1,2,\dots,5\}$ and x represent the phenol descriptor.

Given these initial populations, the next step is to reform the population set according to qubit representation. The QGP is based on the representation of the quantum state vector. It applies the probabilistic amplitude representation of qubit to the coding of the tree, which makes one tree represent the superposition of many states, and uses quantum rotation gates to fulfill the update operation, to overcome the premature convergence by employing quantum crossover and finally accomplishes the optimal determination of the goal [28]. QGP is qubit based encoding for the GP

population tree, such that each internal node can be found in the superposition of *n* states at quantum population and it represents only one state in classic population by applying measurement. According to the information of the optimal individual in each operator "internal node", quantum gates can lead populations to update themselves.

Qubit is the minimum information component in quantum computing. In our case, it represents a four-state quantum system described as:

$$|\psi_s\rangle = \alpha_1|00\rangle + \alpha_2|01\rangle + \alpha_3|10\rangle + \alpha_4|11\rangle \quad (1)$$

$$|\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 + |\alpha_4|^2 = 1 \quad (2)$$

in which 00, 01, 10, and 11 are employed to encode +, -, / and * operators respectively. In QGP, multi-qubit is used to store and represent one gene. Each qubit may be in the '1' state, the '0' state, or any superposition of the two. To be exact, the information embodied by this gene is not steady, nevertheless possible; as a result, when an operation is passed on this gene, it may be terminated to all probable information concurrently. Herein, each gene has two-qubits. The multi-dimensional unitary transform is very difficult to design. The simpler solution is to adopt the binary coding technique in GP to encode these qubits of multi-states; i.e. using two qubits to represent multi-states. This method has better adaptability and is easier to understand. Herein, the suggested system uses tensor product $\otimes$, a way of putting vector spaces together to form larger vector spaces, to handle the difficulty of representing multi-state, so that:

$$|VW\rangle = |V\rangle \otimes |W\rangle = |V\rangle|W\rangle,$$

$$|01\rangle = |0\rangle \otimes |1\rangle = |0\rangle|1\rangle \quad (3)$$

so, two qubits are used to represent one gene; and each qubit can stay in the superposition of the two quantum states simultaneously, e.g.

$$|\psi_Q\rangle = \gamma|0\rangle + \beta|1\rangle \quad (4)$$

$|0\rangle$ represents the state of spin up, while $|1\rangle$ represents the state of spin down. For general case, the multi-qubits are applied to represent the multi-state operator node, as follows:

$$q_j^t = \begin{bmatrix} \gamma_{11}^t & \gamma_{12}^t & \cdots & \gamma_{1k}^t & \gamma_{21}^t & \gamma_{22}^t & \cdots & \gamma_{2k}^t & \gamma_{m1}^t & \gamma_{m2}^t & \gamma_{mk}^t \\ \beta_{11}^t & \beta_{12}^t & \cdots & \beta_{1k}^t & \beta_{21}^t & \beta_{22}^t & \cdots & \beta_{2k}^t & \beta_{m1}^t & \beta_{m2}^t & \beta_{mk}^t \end{bmatrix} \quad (5)$$

q_j^t represents the chromosome of the *t*-th generation and the *j*-th tree, *k* is the qubit number of every coding state, and *m* is the operator node number in each tree. The adoption of qubit coding enables one tree to represent the superposition of multi-states simultaneously, making the QGP better in diversity to the classic GP algorithm. As stated in [28], convergence can be also obtained with the qubit representation. As $|\gamma|^2$ or $|\beta|^2$ approaches to 0 or l, the qubit chromosome (tree data structure) converges to one single state.

Each qubit is initialized to $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$. This clarifies that one qubit gene may represent the superposition of all possible states with the same probability. For the updating execution mechanism, Quantum rotation gate can be used [29]

$$U(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \quad (6)$$

where U is an arbitrary single qubit unitary operation, and θ_i is the rotation angle for each qubit, defined as:

$$\theta_i = S(\gamma_m, \beta_m) * \Delta\theta_i \quad (7)$$

$S(\gamma_m, \beta_m)$ is the sign of θ_i that determines the direction, and $\Delta\theta_i$ is the magnitude of rotation gate illustrated in Fig. 3, and So, γ_m^* and β_m^* are calculated as [9].

$$\begin{bmatrix} \gamma_m^* \\ \beta_m^* \end{bmatrix} = U(\theta) \begin{bmatrix} \gamma_m \\ \beta_m \end{bmatrix} \quad (8)$$

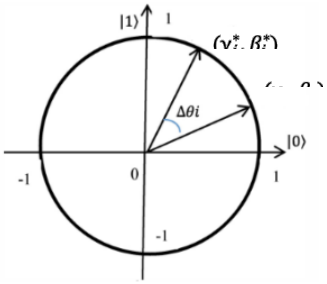

Fig. 3: Rotation Gate for Qubit

Table 1: Rotation Angle Selection Strategy

x_i	b_i	$f(x_i) > f(b_i)$	$\Delta\theta_i$	$S(\gamma_m, \beta_m) * \Delta\theta_i$			
			δ	$\gamma_m * \beta_m > 0$	$\gamma_m * \beta_m < 0$	$\gamma_m = 0$	$\beta_m = 0$
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

The next step is to measure all the updated populations (each qubit is updated using rotation gate) and obtain a group of definite solution. The measuring execution is as follows: generate a random number τ between 0 and 1. If $\tau > |\gamma_{ij}^t|^2$ the measuring result is 1; otherwise 0. Then evaluate the group of the solution with its fitness, the best tree and its fitness among the binary solution $P(t)$ is then selected and stored for next generations. In Table 1, the updating policy is to match the fitness $f(x_i)$ of the recent quantified value of the item q_j^t with the current evolutionary aim's fitness $f(b_i)$. If $f(x_i) > f(b_i)$, then fine-tune the qubit of the related bit $f(x_i) \neq f(b_i)$ to force the likelihood value progress near the track of promoting the appearance of x_i. In contrast, if $f(x_i) < f(b_i)$, then regulate the qubit of the equivalent bit to attain the probability amplitude go forward in the direction of aiding the presence of b_i. As well, δ is the angle step of every updating. The value of δ has an influence on the convergence speed; if the value is too large, the resolution may move away or have an early convergence to a local optimum. In this, the dynamic tune of δ is approved, so that, it receipts a value flanked by 0.2π and 0.8π by dynamic tuning as stated by the variance of the genetic generations.

The most difficult section in GP is determining an objective function; different environment may have different fitness function, and so in this research, the system uses the normalized Euclidean distance metric as the most effective values that affect phenol toxicity prediction defined as function:

$$f_i = \sqrt[2]{\sum_{i=1}^{n}(|x_i - y_i|)^2} \quad (9)$$

where f_i is the calculated fitness value, x_i , y_i are the corresponding result and target and n is the phenols number. During generations, the solution of the generation is converged little by little to the optimum solution. In each generation, get a group of solution $P(t)$ through measuring $Q(t)$, calculate the fitness of every solution, carry out the crossover and mutation on the individuals of the generation, revise them by employing the quantum gates to obtain $Q(t+1)$, warehouse the updated ideal solution and equate it with the current individual. If the optimum solution is bigger than the present individual, the present individual is replaced by the optimum solution; otherwise, the present individual remains unchanged. Termination condition is responsible in designating the individual program that is identified with the best fitness. This outcome may be a solution to the problem.

In formal, to localize an optimal point of an objective function $: p \to W$, an GP uses a population of elements of p, which is modified by genetic operators, like mutation, crossover, and selection. Assume that μ signifies the number of parents and λ the number of descendants in one generation. The implementation of an GP requires that the abstract elements of p be represented by a data structure, i. e. elements of a space $\mathcal{G}$. The set p is called the phenotype space and $\mathcal{G}$ the genotype space. For instance, when *applying* *S* terminologies in a GP-system, $\mathcal{G}$ contains all *S* terminologies, whereas the phenotype space p involves all functions $f : A \to B$ (where *A* and *B* are defined by the problem to be resolved)[30]. Concerning GP, it is adequate to warehouse an individual in genotype mode, as the genotype-phenotype mapping $h : \mathcal{G}^{\lambda} \to p^{\lambda}$ is deterministic and environmental impacts are not considered. Hence, we can represent a population at generation *t* by $Pop(t) \in \mathcal{G}^{\lambda}$.

The mapping $h : \mathcal{G}^{\lambda} \to p^{\lambda}$ can be composed simply from λ reduced mappings $h' : \mathcal{G} \to p$, which represent the geno-phenotype mapping for single individuals: $\mathrm{h(g)} = \left(h'(\mathrm{g}_1), \ldots, h'(\mathrm{g}_{\lambda})\right),\ \mathrm{g} = (\mathrm{g}_1, \ldots, \mathrm{g}_{\lambda}) \in \mathcal{G}^{\lambda}$.The mapping $h' : \mathcal{G} \to p$ determines the abstract element $h'(x)$ of the search space being signified by $x \in \mathcal{G}$. Therefore, the mapping h' defines the relationship between genotype and phenotype space. The crossover operator $r : \mathcal{G}^{\mu} \times \Omega_m \to \mathcal{G}^{\lambda}$ produces λ descendants from the parent population by merging the parental genetic information. The probabilistic effect for the period of crossover is defined by the probability space (Ω_r, P_r), i.e. the result of the crossover be subject to the random choice of $\omega_r \in \Omega_r$ in keeping with P_r.

The mutation $m : \mathcal{G}^{\lambda} \times \Omega_m \to \mathcal{G}^{\lambda}$ is functioning on the genotype space $\mathcal{G}$ only. Here (Ω_m, P_m) is the primary probability space. The new population $Pop(t+1) \in \mathcal{G}^{\mu}$ is elected from the set of offspring of $Pop(t)$, where the election of an individual is established explicitly or implicitly based on the objective function $f : p \to W$. The objective function evaluates only the phenotype. This is formalized by the selection operator $s : \mathcal{G}^{\lambda} \times p^{\lambda} \times \Omega_s \to \mathcal{G}^{\mu}$ (with probability space (Ω_s, P_s)). With the support function $h^* : \mathcal{G}^{\lambda} \to \mathcal{G}^{\mu} \times p^{\lambda},\ h^*(g) := (g, h(g))$ for $g \in \mathcal{G}^{\lambda}$ the equation:

$$Pop(t+1) = s(h * \left(m(r(Pop(t), \omega_r), \omega_m)\right), \omega_s) \qquad (10)$$

holds, where $\omega_r \in \Omega_r,\ \omega_m \in \Omega_m$ and $\omega_s \in \Omega_s$ are chosen randomly according to $P_r, , P_m$ and P_s. In general, the phenols are structurally heterogeneous and represent a variety of mechanisms of toxic action. The pseudo code of the suggested system is as follows. Furthermore, Table 2 and 3 list GP parameters, terminal and functions respectively.

Algorithm 1: QGP

*Input: Dataset T; No. of Generation t=0; set of arithmetic operator F= {+, -, /, *}; Initial Populations Pops.*

```
1-  while t < MAX_GENS do
2-  t ← t+1
3-  Pops ← Qubit_Encoding(Pops)
4-  Fitness_Vals=Fitness_Evaluation(Pops)
5-  Pop ← Selection_Best (Pops, Fitness_Vals)
6-  If Termination_Condition is False, then
7-  for i ← 0 to (POP_SIZE − 1) do
8-  New_Pops (i) ←Crossover(Pop)
```

9- New_Pops (i) ← Mutation(New_Pops)
10- end
11- Pops= New_Pops
12- end
13- If Termination_Condition is True, then
14- Return Pop
15- end
16- end
17- Best Tree= Pop

Table 2. Genetic Programming Parameters

Parameter	Value
Population size	50
Maximum number of evaluated individuals	2500
Selection	Rotational Gate
Mutation	one point
Crossover	two point
Generation gap	0.6
Likelihood of Crossover	0.7
Likelihood of Mutation	0.3
Termination Condition	200 Generation

Table 3. Genetic Programming terminals and functions

Objective	Determining the least in Euclidean Distance
Terminal set	$log\ K_{ow}$, $log\ D_{owu}$, pK_a, E_{LUMO}, E_{HOMO}
Function Set	$\{+, -, /, *\}$
Fitness function	$f_i = \sqrt[2]{\sum_{i=1}^{n}(\lvert x_i - y_i \rvert)^2}$

III. Analysis and Result

This section validates the efficiency of the suggested system by performing many experiments on a benchmarked realistic dataset [27]. Furthermore, the performance is compared with traditional statistical prediction approach in order to evaluate the predicted accuracy of proposed approach. The system is implemented in a form of MATLAB library, which was designed to be easy to use in custom applications. The tests are conducted on a machine with Intel(R) Zeon(R) CPU E5430@ 2.66GHz (2 Processor), 16 GB RAM PC running Microsoft windows 8.1 Enterprise 64 bit. The simulation outcomes approve the capability of the suggested technique to achieve precise prediction of toxicity degree.

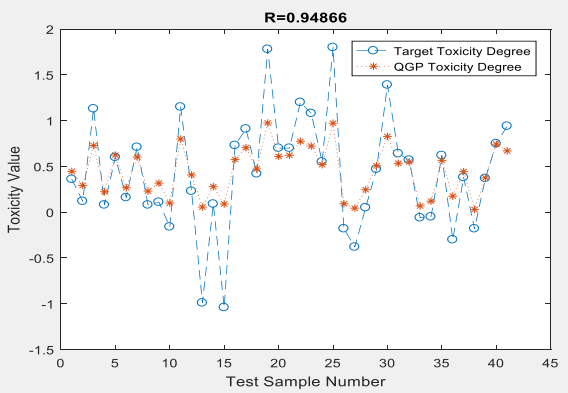

Fig. 4. Experimental (target) versus predicted toxicity using the QGP

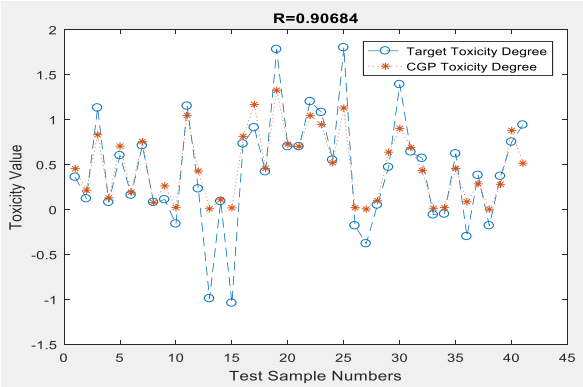

Fig. 5. Experimental versus predicted toxicity using the CGP

In our evaluation using standard benchmarks, our best prediction achieved a correlation coefficient (*R*-value) over 0.94. It is confirmed that the proposed QGP algorithm proved superior toxicity prediction performance when compared with GP, RBF NN algorithms that archive 0.91, 0.79 correspondingly in *R* metric as illustrated in Fig. 4,5, and 6 sequentially. One possible explanation for that results is that utilizing GP for prediction creates a diversity of solutions with unlimited search ability according to different GP parameters such as selection, crossover, mutation and tree depth. This diversity is usually associated with an objective function that can produce the optimal tree structure (linear equation). Unlike the neural network-based prediction that mainly depends on the weighting matrix to build a hidden nonlinear relationship between input samples and output (toxicity prediction degree). However, the quality of NN prediction often stacks with architecture complexity, generalization ability, noise-tolerant ability, and limited search-ability.

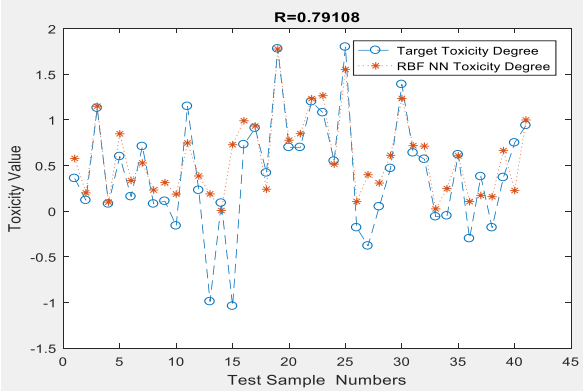

Fig. 6: Experimental versus predicted toxicity using the RBF NN.

Overall, GP had the aptitude of successfully modelling composite real-world relations in comparison to the traditional regression approaches. Despite the enhancements in the prediction accuracy achieved by employing GP, the result reveals that GP can rebuild the transparent functional relationship as a linear equation which is convenient to use later (e.g., unlike RBF NN). However, we noticed that one possible disadvantage of utilizing GP for prediction is to produce extremely linear complex functions, which may not be useful for knowledge induction (e.g., like black-box modeling techniques). The results demonstrate that QGP-based predication can offer further improvement in terms of *R* with low complexity of the generated regression equation as illustrated in the two generated regression function form both CGP and QCP respectively.

CGP generated equation

$$\log(1/IGC_{50}) = ((E_{LUMO})/(E_{HUMO})) * (\left(\frac{N_{Hdon}}{N_{Hdon}}\right) + ((\left(\left(\log K_{ow}\right) + (\log K_{ow})\right) - (E_{LUMO})) + ((pK_a) - (E_{HUMO}))))$$

QGP generated equation

$$\log(1/IGC_{50}) = ((((\log K_{ow}) * (E_{LUMO})) - (\log K_{ow})) - (\log K_{ow}))/(E_{HUMO})$$

In general, inspiring quantum computing concept inside GP leads to the diversity of the populations than classical GP. These diversity helps to obtain optimal solutions with best fitness functions that can be used later to select the optimal linear equation for toxicity prediction. Furthermore, in quantum chromosomes, the linear superposition of all possible binary states provides great variety over classical representation. To converge the chromosome individuals toward optimal solutions, quantum rotation gate is incorporated. Table 4 summarizes the results of all prediction methods.

Table 4. Comparative result using testing set = 41 phenols

Method	R
QGP	0.9486
GP	0.9068
RBF NN	0.7910

Table 5 displays the run time and fitness evaluation of the two algorithms CGP and QGP. The population size of CGA is 50, while the QGA's population size is chosen to be 10. The table offers the mean value of the best fitness, the average fitness, the worst fitness and the elapsed time per generation. Over 20 runs with the simulation setup configuration as number of generations = 200, tree depth =12, generation gap =0.6, crossover probability = 0.7 and mutation probability = 0.3, the results reveal that QGP with 10 individuals can reach a better effect regarding both of best fitness and mean fitness of CGP with 50 individuals, but QGP's elapsed time is only 1/3 of that of CGP.

Table 5. Comparative results between CGP & QGP

	CGP	QGP
Population size	50	10
Best fitness	1.57	0.8976
Average fitness	179.53	147.41
Worst fitness	1753.2	2681.6
Elapsed time	289.01	88.89

Fig.7 shows the progress of the average fitness of QGP with 10 individuals using the fixed rotation angle and dynamic adjusting rotation angle. From the figure, we can see the superiority of the dynamic adjusting rotation angle. The experimental results demonstrate that the convergence speed of the dynamic tuning of δ is higher than that of the fixed rotation angle. Dynamic quantum rotation gate enhances the prediction accuracy by increasing number of populations through increasing qubits possible probability (i.e. diversity).

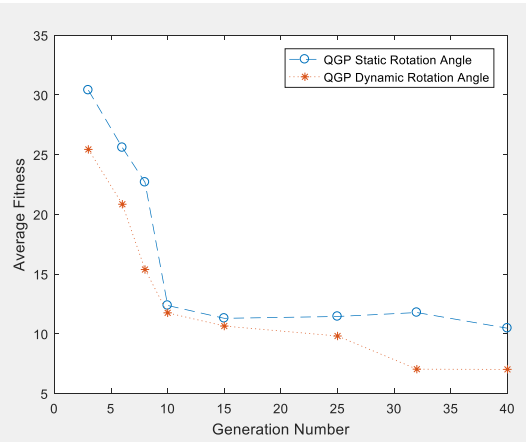

Fig.7: *The progress of the average fitness of QGP with the fixed and dynamic rotation angle*

IV. Conclusion

In this work, we presented a novel QSTR methodology based on the QGP model for toxicity prediction by extracting linear equations to calculate phenol toxicity. The QGP model were produced based on the quantum rotational gate in order to exploit randomness offered by quantum chromosomes described by qubits, which is fast and gives a linear equation, in contrast to most traditional training techniques. The model generated for the data set required five descriptors. In our case, the best model developed by QGP gives a more accurate prediction than the pre-specified model optimized by both GP and RBF NN. This is due to the fact that in QGP a much wider solution space can be analyzed because the structure of the models is not prescribed in advance but is left to the evolutionary procedure with different likelihoods arising from qubit superposition by means of quantum rotation gate. By combining the GP and superposition concept, we successfully enhanced the toxicity prediction accuracy, and the result shows that the calculation efficiency of QGP is obviously better than that of CGP and RBF NN.

In terms of the R, the QGP models proved to have a significant predictive potential. The results obtained illustrate that the QGP architecture can be used to derive QSTRs, which are more accurate and have better generalization capabilities compared to other models. QGP could be a substitute for costly and time-consuming experiments for toxicity determination. The results imply that the GP approach could be successfully used if there are relatively simple relations between input and output variables. If the relations would be more complex (e.g., much more complicated geometry), the QGP would be a much more suitable approach. In the future work, the plan is to enhance the toxicity prediction accuracy by applying the quantum rotation gate on the terminals.

References

[1] [1] Begam, B., Kumar, J.: 'A study on cheminformatics and its applications on modern drug discovery', Procedia engineering, 2012, 38, pp.1264-1275.
[2] [2] Melagraki, G., Afantitis, A., Makridima, K., Sarimveis, H., et al.: 'Prediction of toxicity using a novel RBF neural network training methodology', Journal of molecular modeling, 2006, 12, (3), pp 297-305.
[3] [3] O'Neill, M., Vanneschi, L., Gustafson, S., Banzhaf, W.: 'Open issues in genetic programming', Genetic Programming and Evolvable Machines, 2010, 11, (3-4), pp 339-363.
[4] [4] Mair, C., Kadoda, G., Lefley, M., Phalp, K., et al.: 'An investigation of machine learning based prediction systems', Journal of Systems and Software, 2000, 53, (1), pp 23-29.
[5] [5] Debnath A.: 'Quantitative structure-activity relationship (QSAR): A Versatile Tool in Drug Design, Combinatorial Library Design and Evaluation: Principles, Software Tools, and Applications in Drug Discovery', (Marcel Dekker, New York, Chapter3, 2001), pp. 73-129.
[6] [6] Khan, M., Ahmad, A., Khan, G., Miller, J.: 'Fast learning neural networks using cartesian genetic programming', Neurocomputing, 2013, 121, pp 274-289.
[7] [7] Kobashigawa, J., Youn, H., Iskander, M., Yun, Z.: 'Comparative study of genetic programming vs. neural networks for the classification of buried objects', Antennas and Propagation Society International Symposium, APSURSI'9, IEEE, 2009, pp 1-4.
[8] [8] Brezocnik, M., Kovacic, M., Gusel, L.: 'Comparison between genetic algorithm and genetic programming approach for modeling the stress distribution', Materials and Manufacturing Processes, 2005, 20, (3), pp 497-508.
[9] [9] Guo, H., Jack, L., Nandi, A.: 'Feature generation using genetic programming with application to fault classification', IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2005, 35, (1), pp 89-99.
[10] [10] Razali, N., Geraghty, J.: 'Genetic algorithm performance with different selection strategies in solving TSP', Proceedings of the world congress on engineering, Hong Kong: International Association of Engineers, 2011, 2, pp. 1134-1139.
[11] [11] Ristè, D., Da Silva, M., Ryan, C., Cross, A., et al.: 'Demonstration of quantum advantage in machine learning', npj Quantum Information, 2017, 3, (1), pp 16.
[12] [12] Laboudi, Z., Chikhi, S.: 'Comparison of genetic algorithm and quantum genetic algorithm', Int. Arab J. Inf. Technol, 2012, 9, (3), pp 243-249.
[13] [13] Wang, L., Tang, F., Wu, H.: 'Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation', Applied Mathematics and Computation, 2005, 171, (2), pp 1141-1156.
[14] [14] Kuo, S., Chou, Y., Chen, C.: 'Quantum-inspired algorithm for cyber-physical visual surveillance deployment systems', Computer Networks, 2017, 117, pp 5-18.
[15] [15] Yanofsky N. S.: 'An introduction to quantum computing', ArXiv Preprint ArXiv, 2007, 0708, (0261).
[16] [16] Darnag R., Minaoui, B., Fakir, M.: 'QSAR models for prediction study of HIV protease inhibitors using support vector machines, neural networks and multiple linear regression', Arabian Journal of Chemistry, 2017, 10, pp S600-S608.
[17] [17] Askari, H., Ghaedi, M., Dashtian, K., Azghandi, M.: 'Rapid and high-capacity ultrasonic assisted adsorption of ternary toxic anionic dyes onto MOF-5-activated carbon: artificial neural networks, partial least squares, desirability function and isotherm and kinetic study', Ultrasonic Sonochemistry, 2017, 37, pp71-82
[18] [18] Cronin, M., Aptula, A., Duffy, J., Netzeva, T., et al.: 'Comparative assessment of methods to develop QSARs for the prediction of the toxicity of phenols to Tetrahymena pyriformis', Chemosphere, 2002, 49, (10), pp 1201-1221.
[19] [19] Zupan, J., Gasteiger J.: 'Neural networks in chemistry and drug design', John Wiley & Sons, Inc.., 1999.
[20] [20] Koç, D., Koç, M.: 'A genetic programming-based QSPR model for predicting solubility parameters of polymers', Chemometrics and Intelligent Laboratory Systems, 2015, 144. pp 122-127.
[21] [21] Lei, T., Li, Y., Song, Y., Li, D., et al.: 'ADMET evaluation in drug discovery: 15. Accurate prediction of rat oral acute toxicity using relevance vector machine and consensus modeling', Journal of cheminformatics, 2016, 8, (1), pp 6.
[22] [22] Chokshi, P., Dashwood, R., Hughes, D.: 'Artificial neural network (ANN) based microstructural prediction model for 22MnB5 boron steel during tailored hot stamping', Computers & Structures, 2017, 190, pp 162-172.

[23] [23] Spedicato, E., Xia, Z., Zhang, L.: 'ABS algorithms for linear equations and optimization', Journal of computational and applied mathematics, 2000, 124, (1-2), pp 155-170.
[24] [24] Aalizadeh, R., Peter, C., Thomaidis, N.: 'Prediction of acute toxicity of emerging contaminants on the water flea Daphnia magna by Ant Colony Optimization–Support Vector Machine QSTR models', Environmental Science: Processes & Impacts, 2017, 19, (3), pp 438-448.
[25] [25] Buontempo, F., Wang, X., Mwense, M., Horan, N., et al.: 'Genetic programming for the induction of decision trees to model ecotoxicity data', Journal of chemical information and modeling, 2005, 45, (4), pp 904-912.
[26] [26] McKay, B., Willis, M., Barton, G.: 'Steady-state modelling of chemical process systems using genetic programming', Computers & Chemical Engineering, 1997, 21, (9), pp 981-996.
[27] [27] Aptula, A., Netzeva, T., Valkova, I., Cronin, M. T., et al.: 'Multivariate discrimination between modes of toxic action of phenols', Quantitative Structure-Activity Relationships, 2002, 21, (1), pp12-22.
[28] [28] Yang, J., Li, B., Zhuang, Z.: 'Research of quantum genetic algorithm and its application in blind source separation', Journal of Electronics (China), 2003, 20, (1), pp 62-68.
[29] [29] Mohammed, A., Elhefnawy, N., El-Sherbiny, M., Hadhoud, M., et al.: 'Quantum crossover based quantum genetic algorithm for solving non-linear programming', Informatics and Systems (INFOS), International Conference on. IEEE, 2012.
[30] [30] Droste S, Wiesmann D.: 'Metric based evolutionary algorithms', InEuropean Conference on Genetic Programming, Springer, Berlin, Heidelberg, 2000, 15, pp 29-43.