

Received May 20, 2019, accepted June 10, 2019, date of publication July 5, 2019, date of current version August 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926989

# Advanced Quantum Based Neural Network Classifier and Its Application for Objectionable Web Content Filtering

OM PRAKASH PATEL<sup>1</sup>, NEHA BHARILL<sup>1</sup>, ARUNA TIWARI<sup>2</sup>, VIKRAM PATEL<sup>3</sup>,  
OJAS GUPTA<sup>4</sup>, JIAN CAO<sup>5</sup>, JUN LI<sup>6</sup>, AND MUKESH PRASAD<sup>6</sup>

<sup>1</sup>Department of Computer Science and Engineering, Mahindra Ecole Centrale College of Engineering, Hyderabad 500043, India

<sup>2</sup>Department of Computer Science and Engineering, Indian Institute of Technology Indore, Indore 453552, India

<sup>3</sup>Chota Hospital Private Limited, Indore 452011, India

<sup>4</sup>VMWare, Palo Alto, CA 94304, USA

<sup>5</sup>Department of Computer Science and Engineering, Shanghai Jiatong University, Shanghai 200240, China

<sup>6</sup>Centre for Artificial Intelligence, School of Computer Science, FEIT, University of Technology Sydney, Ultimo, NSW 2007, Australia

Corresponding author: Mukesh Prasad (mukesh.prasad@uts.edu.au)

This work was supported in part by the China National Key Research and Development Plan under Grant 2018YFB1003800, and in part by the Science and Technology Program of Guangzhou under Grant 201704030133, A Knowledge-Connection and Cognitive-Style-based Mining System for Massive Open Online Courses and Its Application (PRO16-1300).

**ABSTRACT** In this paper, an Advanced Quantum-based Neural Network Classifier (AQNN) is proposed. The proposed AQNN is used to form an objectionable Web content filtering system (OWF). The aim is to design a neural network with a few numbers of hidden layer neurons with the optimal connection weights and the threshold of neurons. The proposed algorithm uses the concept of quantum computing and genetic concept to evolve connection weights and the threshold of neurons. Quantum computing uses qubit as a probabilistic representation which is the smallest unit of information in the quantum computing concept. In this algorithm, a threshold boundary parameter is also introduced to find the optimal value of the threshold of neurons. The proposed algorithm forms neural network architecture which is used to form an objectionable Web content filtering system which detects objectionable Web request by the user. To judge the performance of the proposed AQNN, a total of 2000 (1000 objectionable + 1000 non-objectionable) Website's contents have been used. The results of AQNN are also compared with QNN-F and well-known classifiers as backpropagation, support vector machine (SVM), multilayer perceptron, decision tree algorithm, and artificial neural network. The results show that the AQNN as classifier performs better than existing classifiers. The performance of the proposed objectionable Web content filtering system (OWF) is also compared with well-known objectionable Web filtering software and existing models. It is found that the proposed OWF performs better than existing solutions in terms of filtering objectionable content.

**INDEX TERMS** Quantum computing, Web crawler, neural network classifier, objectionable Web content.

## I. INTRODUCTION

Inspired by the human brain to classify complex patterns of the real world, an artificial neural network has been introduced. In recent decades, many neural architectures have been proposed like Perceptron, Feedforward Neural Network, and Backpropagation, etc. These models have been used to solve problems of many areas like mathematics, bio-science, economics, and prediction system [1]–[5]. The neural network is a mathematical model which performs parallel

information processing by adjusting the connections amongst the number of nodes. However, the performance of the neural network depends on many factors such as network architecture, weights, the number of hidden layer neurons, and learning strategy, etc. [6]. Forming a neural network architecture with optimization of all parameters is a difficult task. In the neural network learning algorithm, finding optimal connection weights is an important factor for its performance. For proper selection of these connection weights, several evolutionary algorithms have been proposed [7]–[12]. To solve the problem of evolving weights of neural network Chen et al. [11], proposed a classification technique of neural

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

network using an improved genetic algorithm. In this method, the authors use shift reverse logic crossover operation and improved genetic algorithm for optimizing Backpropagation algorithm. One more similar concept is introduced by Lu et. al. [12], in which the quantum computing concept is used to optimize connection weights and architecture of the neural network. In this method, the connection weights, and the number of connections have been decided using the quantum computing concept. However, in this approach, the number of neurons in the hidden layer is selected randomly, which is not the correct way to decide the number of neurons. Determining the number of neurons in an above-mentioned way increases the training time and also it may not guarantee to achieve the global optima. The problem of determining the appropriate number of hidden layer neurons can be solved by forming neural network architecture constructively by adding neurons one by one as per requirement. Kim and Park [13], proposed a neural network algorithm which forms neural network architecture constructively. In constructive neural network algorithm, the neurons are being added to the network during the learning process as per requirements. To determine the optimal connection weights and the number of hidden layer neurons, a quantum based binary neural network learning algorithm has been presented by Patel and Tiwari [14]. In this proposed algorithm, the connection weights have been decided using quantum computing concept, and neural network architecture is formed constructively.

As discussed above, the neural network has been widely used for pattern recognition or data classification [15], [16]. In pattern recognition, the neural network is used in the area of computer networks especially for providing network security in Web. Today, the information age is characterized by the continuous massive production and dissemination of information. Nowadays, the Internet is being used as one of the necessities of our mundane life, and it is more than just emails and web browsing. The number of Internet users is increasing day by day, for example from social networking to an e-commerce platform, a knowledge library to an e-market, Internet has captured everyone in its charisma. The boon of the worldwide Web is the openness and the accessibility that is provided to its users to access any information. However, the availability of all the information to all the users is inappropriate. There are lots of malicious content present on the Internet which may affect the users, especially children. A 2000 US congressional study shows that one in five of the 24 million children has been solicited online for porn when surfing on the Internet [17]. However, protection from objectionable Web and free speech on the Web are a contradictory issue and there must be a tradeoff between them. The following example can be understood with the help of a scenario. There are some websites which provide awareness about sex education, which is not objectionable. On the contrary, there are other websites which have objectionable pornographic content. Both kinds of websites may have objectionable words. When the analysis is done on the basis of keywords, it blocks both the types of websites which is a

drawback in this case. Including these problems as mentioned above, there are many other issues like, websites which do not contain objectionable content as a whole but containing some hyperlinks which might contain objectionable content. In such cases, blocking of parent website is not an intelligent solution, the particular hyperlink must be blocked. All these issues belong to the features selected from website. Along with this, the dynamic nature of the Web is also an important issue. As discussed above, several new websites are launched every day. Analyzing all websites against objectionable and non-objectionable content at run time is not an intelligent solution. There should be a system which is able to identify new objectionable websites before any user access it. This system must regularly check the newly coming objectionable websites and make a database of those websites which helps to speed up the process of blocking objectionable websites. Apart from this, the classification of these features is also an important task. To classify accurately between objectionable contents and non-objectionable contents, there must be a proper selection mechanism for selecting the appropriate classifiers.

To overcome the issues and drawbacks discussed here, Patel et. al. [18] proposed a method in which quantum based neural network has been used to classify objectionable Web request. In this algorithm, the connection weights and threshold is evolved using quantum computing concept. Furthermore, the algorithm is tested on objectionable Web content. Although the performance of QNN-F is good still there are possible improvements that can be done in the proposed work [18]. Firstly, the threshold of neurons has been selected randomly in this method which may lead the algorithm to stuck into local minima or maxima. Secondly, the connection weights and threshold is evolved in one population using several generations in it. Finding neural network parameters in this way may lead to evolved locally optimal parameters. Therefore, to handle these issues of an improved algorithm, an Advanced Quantum based Neural Network Learning Algorithm (AQNN) is proposed here. The AQNN is used to form an Objectionable Web Content Filtering system (OWF). The proposed algorithm forms a neural network classifier constructively by adding the neuron at the hidden layer, whereas, the connection weights and threshold of a neuron is decided using the quantum computing concept. In this algorithm, the threshold boundary parameter is introduced which helps to find optimal value of threshold. The value of threshold is find out from a particular range rather than randomly. In this a minimum and maximum range of threshold are found which provide an optimal range of exploration. Thus, it reduces the number of iterations to find an optimal value of the threshold. Along with the threshold boundary parameter here, we find connection weights and threshold of a neuron using quantum and genetic concept in more than one population rather than a single population. Finding parameters in this way helps to find optimal parameters. The proposed algorithm is trained and tested on Web dataset to form objectionable Web content filtering system. In the

proposed OWF, for accurate classification of objectionable Web content, some unique features have been added. These features have been extracted from the content of websites. This OWF mainly consists of five modules: Web crawler, requested Web page analyzer, hyperlink analyzer, the feature extractor, and the quantum neural network. The module Web crawler requested Web page analyzer, and hyperlink analyzer help to get websites for Web content analysis. The feature extractor module helps to extract features from websites contents, collected from the different modules. These features are given as input into the proposed AQNN to decide whether websites are objectionable or not.

This paper is organized as follows. The existing work related to blocking of objectionable Web content filtering system is discussed in Section II. The required preliminaries about the neural network and quantum computing concept are described briefly in Section III. In Section IV, the proposed algorithm is discussed in detail. In Section V, the proposed work AQNN and OWF is illustrated with the help of an example. The experimental work and results are discussed in detail in Section VI. Section VII, is presented with the concluding remarks.

## II. EXISTING WORK

In 1995, to resolve the problem of blocking objectionable content, a set up named as Netscape Communications and Progressive Networks was established by the coalition of Microsoft Corporation [19]. In addition to this, several international organizations and countries have taken similar action to protect the public and children from accessing objectionable websites [20]. The Internet content filtering software (ICFS) became very popular and widely adopted by the school, parents and some governments to protect children from the harmful effect of objectionable content [21], [22]. These Softwares prevent objectionable Web content by monitoring data flow between end-user and Internet connection. The existing work in this area is done by many researchers and can be categorized mainly into two categories that are text-based filtering, and other is text and image-based filtering.

The first category is text-based search in which searching of objectionable content is done with the help of objectionable keywords [23]. In this approach, some objectionable keywords are defined by the administrator which is to be searched in Web content or request website. If it is found then, website is blocked otherwise it is not blocked. However, the drawback of this method is that it blocks the whole Website rather than a particular page or hyperlink in the website which is objectionable. As there are several Web contents available which account as objectionable like sex education but it is useful for students and society. This approach also blocks such type of contents.

The other filtering objectionable content methods are based on image and text feature. Several researchers have proposed various methods based on text and image [24]–[28]. Filtering based on text is done on the basis

of objectionable keywords. In addition to this, the filtering of objectionable images is done using skin features. Here, we discuss related work in this field.

An approach is proposed based on text and visual features for filtering objectionable contents by Hammami et. al. [24]. In this approach, text features along with visual feature based on skin color detection are taken. However, objectionable image detection on the basis of skin based features is not sufficient to judge nudity or objectionability in the image. Thus, the classification accuracy achieved for visual objectionable content is not promising. Also, the way text and visual features are combined is not clear. Chen et. al. [27], also proposed a method of filtering objectionable content on the basis of textual and visual contents. This approach is categorized by three filtering techniques that are based on keywords, based on sentences, and based on images. In these methods, the ROI method is used in place of skin segmentation for objectionable image classification. Classification of objectionable Web content on the basis of image classification, the filtering rate is also not very high. For most of the approaches, skin detection is an initial step for finding nudity in the image. Jones et. al. [31], proposed an approach in which emphasis on the correlation between the percentage of skin and the possibility of nudity in the image. For skin detection in the image, two methods are common. One of them is pixel-based and another is region based. In pixel based, pixels are considered as features and in the region based, the classification is done on the basis of spatial information of pixels. However, finding an objectionable image on the basis of these features is not an optimal method. The features based on skin vary due to illumination, the background of a person, age, angle, etc. To solve this issue, several methods have been proposed. A statistical skin detection method has been proposed by Zeng et. al. [32]. A region based approach is proposed by Xu et. al. [33], in which texture and image features are taken from arbitrary-shaped segmented. To judge nudity in an image, the skin detection is done on the basis of YCbCr color space by Shih et. al. [34]. To detect objectionable image online, a BHO (browser helper object) is proposed by Girgis et. al. [26]. The BHO works with Internet Explorer (IE5) in the background which extracts all the images and links of the image and checks against nudity. To judge nudity in this method, the skin color is checked using YUV technique.

Along with textual and visual based filtering, some more techniques like semantic analysis of Web content were also proposed. Several approaches have been proposed using machine learning algorithm which checks website content semantically [35]–[37]. The classification model used for filtering the objectionable Web content are SVM, Logistic Regression, and K-nearest neighbor. A lot of research has been done, and many methods have been proposed by researchers to block objectionable images and video, but they have failed to block entire objectionable content because of 85% of the information, present are in the form of text [38]–[41].

### III. PRELIMINARIES

In this section, preliminaries related to the neural network, and quantum computing concept is discussed in brief, which helps to illustrate the proposed algorithm.

#### A. NEURAL NETWORK

The proposed algorithm forms a three-layer neural network architecture consists of an input layer, a hidden layer, and output layer. Let  $X = \{X_1, X_2, X_3, \dots, X_u\}$ , where  $u$  is the number of input samples and  $X_l = (x_1^l, x_2^l, x_3^l, \dots, x_n^l)$  where  $n$  is the number of attributes in one instance of the input sample. Thus, in the proposed neural network architecture the input nodes equal to  $n$ . For  $j^{th}$  hidden layer neuron, connection weights are denoted as follows:

$$W_j^{real} = (w_{j1}, w_{j2}, w_{j3}, \dots, w_{jn}) \quad (1)$$

The number of hidden layer neurons is decided constructively. The number of neurons in the output layer is one, which helps to identify whether the input is objectionable or not. For forming a neuron, we make use of the step function which is given as follows:

$$net_j = \sum_{i=1}^n w_{ji} \times x_i \quad (2)$$

$$f(net_j) = \begin{cases} 1 & \text{if } net_j \leq \text{Threshold} \\ 0 & \text{if } net_j > \text{Threshold} \end{cases} \quad (3)$$

The value of  $net_j$  is compared with the threshold to find out whether the input is objectionable or not. In the proposed algorithm the weights ( $W_j^{real}$ ) and threshold ( $\lambda_j^{real}$ ) are decided using the quantum computing concept. The required preliminary for the quantum computing concept is presented subsequently.

#### B. QUANTUM COMPUTING

The quantum computing concept uses quantum bits  $Q$  in place of classical bits. The weight matrix and the threshold of  $j^{th}$  hidden layer neuron corresponding to Eqs (1), and (3) in the form of quantum bit  $Q$  can be represented as:

$$W_j' = (Q_{j1}, Q_{j2}, Q_{j3}, \dots, Q_{jn}); \quad (4)$$

$$\lambda_j' = (Q_j^{Th}); \quad (5)$$

The quantum bit ( $Q_j$ ) of weight matrix and threshold  $Q_j^{Th}$  is being decided by the quantum concept, where any quantum bit ( $Q_j$ ) and  $Q_j^{Th}$  are represented by several qubits ( $q$ ).

$$Q_j = (q_{j1}|q_{j2}|\dots|q_{jk}) \quad (6)$$

Here, the  $k$  number of qubits represents a quantum bit ( $Q$ ). The quantum bits uses probabilistic features for its representation [42].

$$q_{ji} = \alpha_{ji} |0\rangle + \beta_{ji} |1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (7)$$

where,  $\alpha$  and  $\beta$  is a complex number representing the probability of qubit in “0” state and in “1” state.

However, the proposed algorithm works on a classical computer, therefore, these qubits are converted into real coded value using a conversion process which is discussed next.

#### 1) CONVERSION FROM QUANTUM BITS TO REAL VALUE

This process starts by taking random number matrix  $R_j$ , where  $R_j = [r_{j1}r_{j2}\dots r_{jk}]$ , corresponding to  $Q_j = (\alpha_{j1} | \alpha_{j2} | \dots | \alpha_{jk})$ . The value of  $r_{ji}$  is selected with the help of random function which generates a uniform number between 0 to 1. Then, further mapping is done by using binary matrix  $S_j$  where  $S_j = [s_{j1}s_{j2}\dots s_{jk}]$  [42]. The value of the matrix  $S_j$  is generated as follows:

$$\text{if } (r_j \leq (\alpha_{ji})^2) \text{ then } s_{ji} = 1 \text{ else } s_{ji} = 0. \quad (8)$$

where  $j = 1, 2, \dots, k$  and  $k$  denote the number of qubits. To get the real coded value of the required parameter from binary matrix  $S_j$ , we have used a uniform random number generator (*urg*). The individual quantum bit with  $k$  qubits can represent the linear superposition of  $2^k$  states. Thus, to get an optimal value of the real coded value of the qubit  $q_{ji}$  from  $2^k$  subspaces, we use a formula  $\text{bin2dec}(S_j) + 1$ . This formula helps to get a real-coded value from a particular subspace from the available  $2^k$  subspaces. The whole conversion process has been described in terms of pseudo-code as follows:

---

#### Algorithm 1 Conversion Process

---

**begin**

Step-1:  $q_{ji}$ , link=0 and random number matrix  $r_{ji}$

**for i = 1:k**

$q_{ji} = \alpha_{ji}; 0 \leq \alpha_{ji} \leq 1$

$r_{ji} = \text{rand}();$

This rand function generate uniform value between 0 and 1.

**endfor**

Step-2: **for i = 1:k**

**if** ( $r_{ji} < (\alpha_{ji}^2)$ )

$s_{ji} = 1;$

**else**

$s_{ji} = 0;$

**endif**

**endfor**

Step-3:  $\text{link} = \text{bin2dec}(S_j) + 1$

$q_{ji}^{real} = \text{urg}();$

**end**

---

The conversion process is explained with the help of an example. Let us consider a quantum bit having the length of two qubit  $Q = \langle 0.807 | .45 \rangle$ . Now, take any random number matrix having two random numbers  $R = [0.330, 18]$ . Using Eq. (8) the binary matrix is achieved as  $S = [11]$ . After obtaining the binary matrix, a formula ( $\text{bin2dec}(S)+1$ ) is used to obtain decimal value from a binary value which is achieved as 4. A quantum bit having two qubits will provide 4 search subspaces. Let for example, four search subspaces



are (0, 0.25), (0.25, 0.5), (0.5, 0.75), and (0.75, 1). The binary to the decimal value achieved as 4 it means the qubit will explore fourth subspaces, i.e., (0.75, 1). Thus, quantum bits of connection weights  $W'_j$  and quantum threshold  $\lambda'_j$  are converted into real coded values of connection weight  $W_j^{real}$  and threshold  $\lambda_j^{real}$ . Once, the real coded value of connection weights and threshold are achieved, these are applied to the learning process. However, the value achieved in the first generation may or may not produce optimal results. Therefore, there is the requirement of updating the qubits.

## 2) QUBIT UPDATE PROCESS

To generate a new value of qubit for quantum weights  $W'_j$  and quantum threshold  $\lambda'_j$ , qubit update process is used which utilizes the fitness value (F) and a binary bit (S) corresponding to each qubit. By using fitness value and binary bits, the rotation angle  $\Delta\theta$  is decided. Now using Eq (9), Eq (10), and Table 1, the qubits are updated [12].

$$U(\Delta\theta) = \begin{bmatrix} \cos \Delta\theta & -\sin \Delta\theta \\ \sin \Delta\theta & \cos \Delta\theta \end{bmatrix} \quad (9)$$

$$q_{ji} = \begin{bmatrix} \alpha_{ji}^{g+1} \\ \beta_{ji}^{g+1} \end{bmatrix} = \begin{bmatrix} \cos \Delta\theta & -\sin \Delta\theta \\ \sin \Delta\theta & \cos \Delta\theta \end{bmatrix} * \begin{bmatrix} \alpha_{ji}^g \\ \beta_{ji}^g \end{bmatrix} \quad (10)$$

TABLE 1. Qubit updation.

$s_{ji}^g$	$s_{ji}^{*g}$	$F_{current\ generation} < F_{best\ in\ all\ previous\ generations}$	$\Delta\theta$
0	0	false	0
0	0	true	0
0	1	false	$-0.03 * \Pi$
0	1	true	0
1	0	false	$0.03 * \Pi$
1	0	true	0
1	1	false	0
1	1	true	0

The learning process of AQNN with pseudo code and QWF architecture is discussed in detail in the next section.

## IV. PROPOSED APPROACH

In this section, an advanced quantum based neural network classifier (AQNN) is proposed and it is used to form objectionable Web filtering system (OWF) is presented. First, the proposed algorithm AQNN is discussed in detail, then each module of OWF is discussed.

### A. INPUT DATASET

Firstly, the database has been created for training and testing of proposed AQNN. In database preparation, total 2000 websites contents have been taken, out of which 1000 are objectionable websites and the remaining 1000 are non-objectionable websites. To uniquely identify objectionable and non-objectionable content, in all total 8 features have been selected as mentioned in Table 2. The dataset is prepared on the basis of the above-mentioned features. The dataset extracted from objectionable websites is called as class “A” dataset, and dataset belongs to non-objectionable websites called as class “B” dataset.  $X(i)$  where  $i = 1, 2, 3, \dots, c_1$ ;

TABLE 2. Details of features.

Features	Description
nw	Number of words in Web page
nw_o	Number of Objectionable words in Web page
r_nw	Ratio of total number of words to the objectionable word
sen	Number of sentences in the web page
sen_o	Number of sentences having objectionable words
img	Number of images in the Web page
img_o	Number of images having objectionable words as title
url_o	number of objectionable word in Websites

number of sample of class “A” ( which gives output “0”).  $Y(j)$  where  $j = 1, 2, 3, \dots, c_2$ ; number of sample of class “B” ( which gives output “1”).

### B. LEARNING OF AQNN

As discussed above the input dataset is  $X(i)$  belongs to class “A” and  $Y(j)$  belongs to class “B”. Once processing of input dataset is done, then initialization of neural network parameters like connection weights, threshold, and objective functions are done. As discussed in section I, to evolve neural network parameters optimally, we use quantum computing concept along with the genetic concept. In the neural network, hidden layer neurons are added incrementally, therefore first, a hidden layer neuron is taken in the hidden layer. The connection weights  $(W'_g)^h$  of a hidden layer neuron in terms of quantum bits are represented in three different populations ( $h = 1, 2, 3$ ). Here,  $g$  represents the number of generations needed to evolve connection weights in each population. Finding neural network parameters from three different populations genetically helps to achieve the optimum values. Once, the initialization of quantum weights are done for this neuron, some more parameters are also initialized. Here,  $F_{global}$  is the objective function value corresponding to the whole population ( $h = 1, 2, 3$ ),  $(F_g^*)^h$  is the objective function value corresponding to each population ( $h$ ), and  $(F_g)^h$  is objective function corresponding to each generation ( $g$ ). Similarly, initialize the binary bit matrix  $S_{global}^*$ ,  $(S_g^*)^h$ , and  $(S_g)^h$  of qubits corresponding to the objective function values  $F_{global}$ ,  $(F_g^*)^h$ , and  $(F_g)^h$ , respectively. The binary bit matrix  $S_j$  corresponding to quantum bit  $Q_j$  is discussed in subsection III-B.1. To evolve the connection weights, these parameters are required to update qubit using qubit update process which is discussed in subsection III-B.2. Once all the parameters are defined, the generation  $g$  is initialized as 1 for all three populations ( $h = 1, 2, 3$ ) and then the conversion process is called to convert quantum weights into the real coded value of connection weights. Thus, in this generation,  $(W'_1)^h$  is converted into the real value weight matrix  $(W_1^{real})^h$ . Here, after converting weights into real coded value, the quantum threshold function is called.

Once, the initialization and conversion of connection weights are done, the initialization of quantum threshold  $(\lambda'_t)^h$  in terms of qubits is done for all three populations ( $h = 1, 2, 3$ ). Along with the initialization of quantum threshold

some more parameters are also initialized. Here,  $(F_{\lambda}^{global})$  is objective function value for threshold corresponding to the whole population ( $h = 1, 2, 3$ ),  $(F_{\lambda}^*)^h$  is the objective function value corresponding to each population ( $h$ ) and  $(F_{\lambda}^t)^h$  is the objective function for each generation ( $t$ ). Similarly, initialize the binary bit matrix  $S_{\lambda}^{global}$ ,  $(S_{\lambda}^*)^h$ , and  $(S_{\lambda}^t)^h$  corresponding to objective function values  $(F_{\lambda}^{global})$ ,  $(F_{\lambda}^*)^h$ , and  $(F_{\lambda}^t)^h$ , respectively. The binary bits matrix  $S_j$  corresponding to quantum bit  $Q_j$  is discussed in subsection III-B.1. The selection of the objective function for connection weights and threshold is further discussed in this section. Here, to check the number of the learned sample corresponding to each class, two parameters i.e.  $count1 = 0$ , and  $count2 = 0$  are taken which show the number of learned samples of class “A” dataset and class “B” dataset, respectively.

Here, using the conversion process the real coded value of the threshold  $(\lambda_t^{real})_{con}^h$  is generated. Here, the parameter  $con$  is added in the subscript to represent the real coded value of the threshold which is generated using the conversion process. However, the real coded value of the threshold  $(\lambda_t^{real})_{con}^h$  generated using the conversion process is not a final value of the threshold. To get an optimal value of the threshold, a threshold boundary parameter is proposed here. The threshold boundary parameter helps to find the threshold value of a neuron in a particular range. Thus, the chance of getting an optimal value of the threshold is more.

### 1) THRESHOLD BOUNDARY PARAMETER

The selection of the boundary parameter is done with the help of connection weight and input dataset of each class. Here, parameters  $(net_A(i))_h^{max}$ , and  $(net_B(j))_h^{max}$  are initialized as 0 and parameters  $(net_A(i))_h^{min}$ , and  $(net_B(j))_h^{min}$  are initialized as  $\infty$ . In this process, multiplication of the real coded value of connection weights and input dataset as shown in Eq (2), are done for both the classes, then the minimum and maximum values are found from that. Now, the threshold value of the neuron is decided within the range of this minimum and maximum value. The selection of the boundary parameter for the threshold of each class is done using the following formulation:

### Boundary Parameter Selection of Each Class (CLASS A)

for  $h=1$  to 3

for  $i=1$  to  $c_1$

$$\begin{aligned} (net_A(i))_h &= \sum (W_g^{real})^h \times X_i \\ (net_A(i))_h^{max} &= \max((net_A(i))_h^{max}, (net_A(i))_h) \\ (net_A(i))_h^{min} &= \min((net_A(i))_h^{min}, (net_A(i))_h) \end{aligned}$$

endfor

endfor

$$\begin{aligned} &\max((\lambda_t^{real})^h) \\ &= \max((net_A(i))_h^{max}, (net_B(j))_h^{max}); \end{aligned} \quad (11)$$

$$\begin{aligned} &\min((\lambda_t^{real})^h) \\ &= \min((net_A(i))_h^{min}, (net_B(j))_h^{min}); \end{aligned} \quad (12)$$

### Boundary Parameter Selection of Each Class (CLASS B)

for  $h=1$  to 3

for  $j=1$  to  $c_2$

$$\begin{aligned} (net_B(j))_h &= \sum (W_g^{real})^h \times Y_j \\ (net_B(j))_h^{max} &= \max((net_B(j))_h^{max}, (net_B(j))_h) \\ (net_B(j))_h^{min} &= \min((net_B(j))_h^{min}, (net_B(j))_h) \end{aligned}$$

endfor

endfor

$$\begin{aligned} &(\lambda_t^{real})^h \\ &= \begin{cases} \min((\lambda_t^{real})^h) & \text{if } (\lambda_t^{real})_{con}^h < \min((\lambda_t^{real})^h); \\ (\lambda_t^{real})_{con}^h & \text{if } \min((\lambda_t^{real})^h) \leq (\lambda_t^{real})_{con}^h \leq \max((\lambda_t^{real})^h); \\ \max((\lambda_t^{real})^h) & \text{if } (\lambda_t^{real})_{con}^h > \max((\lambda_t^{real})^h); \end{cases} \end{aligned} \quad (13)$$

Here,  $(\lambda_t^{real})_{con}^h$  is a value of threshold achieved after the conversion process. As discussed above the selection of the objective function value is discussed next.

### 2) SELECTION OF OBJECTIVE FUNCTION

As discussed above the objective function for weights and threshold in each generation are  $(F_g)^h$  and  $(F_{\lambda}^t)^h$  respectively. The objective function for weights and threshold in each population are  $(F_g^*)^h$  and  $(F_{\lambda}^*)^h$ , respectively. Whereas, the global objective function for weight and threshold are  $F_{global}$  and  $(F_{\lambda}^{global})$ . The  $(F_g^*)^h$  and  $(F_{\lambda}^*)^h$  having the best value of the objective function in a particular population. The calculation of these values is done as follows:

$$(F_g^*)^h = \max((F_g)^1, (F_g)^2, \dots, (F_g)^h) \quad (14)$$

$$(F_{\lambda}^*)^h = \max((F_{\lambda}^t)^1, (F_{\lambda}^t)^2, \dots, (F_{\lambda}^t)^h) \quad (15)$$

The calculation of the global parameter, which shows the value of the best objective function in all whole populations and generation are calculated as follows:

$$F_{global} = \max(F_{global}, (F_g^*)^h) \quad (16)$$

$$F_{\lambda}^{global} = \max(F_{\lambda}^{global}, (F_{\lambda}^*)^h) \quad (17)$$

### 3) LEARNING PROCESS

The threshold value of neuron  $(\lambda_t^{real})^h$  evolved through Eqs (11), (12), and Eqs (13) and connection weights are used to learn the neuron for input samples. Here, a multiplication value of an input dataset of both classes and connection weights are compared with threshold  $(\lambda_t^{real})^h$ . Using these calculations,  $count1$ , and  $count2$  values are achieved which shows the number of samples learned of class “A” and class “B” respectively by objective function  $(F_{\lambda}^t)^h$ . Now, if the number of samples learned using connection weights and threshold value are not equal to the total number of samples of class “A” and class “B”, then the qubits of quantum threshold  $(\lambda_t^t)^h$  are updated using quantum update process, which is discussed in subsection III-B.2. For updation of qubits of threshold of the objective functions  $(F_{\lambda}^{global})$ ,  $(F_{\lambda}^*)^h$ ,

and  $(F'_\lambda)^h$  and their corresponding binary bits are required. Now, for  $(W_1^{real})^h$ , ( $g = 1$ ) the threshold is updated from  $(\lambda_1^{real})^h$ , ( $t = 1$ ) to  $(\lambda_z^{real})^h$  ( $t = z$ ) or till the whole sample learned of both the classes. After completion of this process, the best objective functions and corresponding binary bit matrix of quantum threshold  $(\lambda_t^{real})^h$  are stored. Here, to get an optimal value of the threshold and connection weights, the concept of the genetic algorithm is used. After some predefined interval, the qubits of quantum threshold and quantum weights are exchanged with its adjacent population and mutated. Finding the solution in this way gives large subspace to get optimal value of connection weights and threshold. If there are unlearned sample remain after completion of the whole generation of the quantum threshold, then quantum weights  $(W'_1)^h$  are evolved in the next generation to get  $(W'_2)^h$  using the quantum update function discussed in subsection III-B.2. To update quantum bits of connection weights, the objective function value  $(F_g)^h$ ,  $(F_g^*)^h$ ,  $F_{global}$  and corresponding binary bits matrix are used.

If there are still samples remains, which not learned after updating weights  $(W'_g)^h$  ( $g = 1$  to  $m$ ) and the threshold of the neuron ( $t = 1$  to  $z$ ) for all three populations, then one more neuron is added in the hidden layer. The learning of this hidden layer neuron for the number of unlearned samples of both classes is done in the same way as done for the first hidden neuron at hidden layer by evolving connection weights and threshold using quantum computing concept. The AQNN in the form of pseudo code is presented next:

#### 4) PSEUDO CODE OF AQNN

In this section, the pseudo code of the learning algorithm is discussed. The algorithm mainly divided into two parts, the first part shows AQNN algorithm and the second part shows the quantum threshold function.

#### C. TESTING OF MALICIOUS WEB REQUEST

Once, the hidden layer formation is done, the formation of the output layer starts. The connection weights between the hidden layer and output layer are initialized as unity. The threshold of the output layer neuron is kept in such a way that, if the input to the output neuron is positive then the dataset belongs to an objectionable class. If the input to the output neuron is negative, then the dataset belongs to non-objectionable class. Here, hidden layer neuron is trained in such a way to fire output accordingly.

The proposed AQNN is used to develop an objectionable Web filtering system (OWF). The AQNN works as a classifier for the objectionable and non-objectionable Web content collected from different modules of OWF. The basic architecture along with the working of each module of OWF is discussed next.

#### D. OWF ARCHITECTURE

OWF has five basic module. These modules are the Web crawler, requested Web page analyzer, hyperlink analyzer,

---

#### Algorithm AQNN Algorithm

---

Step-1: Take input dataset as X(i) and Y(j)

Take first neuron at hidden layer and initialize the population by specifying quantum weights  $(W'_g)^h$  in terms of qubits.

$$(W'_g)^h = (Q_{w1}^h, Q_{w2}^h, Q_{w3}^h, \dots, Q_{wn}^h)$$

where  $g = 1, \dots, m$ ;  $m$  is the number of generations to update weights in each population.

$i = 1, \dots, n$ ;  $n$  is the number of features of sample.

Initialization of more parameters.

$$(F_0)^h = 0$$

$$(S_0)^h = 0;$$

$$F_{global} = 0$$

$$S_{global} = 0;$$

$$(F_g^*)^h = 0$$

$$(S_g^*)^h = 0;$$

For each population

Step-2: for  $h = 1$  to 3

for  $g = 1$  to  $m$

Call conversion process  $((W'_g)^h)$ .

Call Quantum threshold function  $((W_g^{real})^h)$ .

if  $((F_g)^h \geq (c_1 + c_2))$

Stop learning.

else

$$(F_g^*)^h = \max((F_g)^1, (F_g)^2, \dots, (F_g)^h)$$

update quantum bits by using quantum

update  $(F_{global}, (F_g^*)^h)$ , Eqs (9), (10), and Table 1.

endif

if  $((g == m) \wedge ((F_g^*)^h \leq (c_1 + c_2)))$

Add new neuron for unlearned sample  $(c_2) - (F_g^*)^h$  and finalize its weight by

using

Step-1 and Step-2.

$$F_{global} = \max(F_{global}, (F_g^*)^h)$$

endif

endfor

endfor

---

a feature extractor, and AQNN as shown in Fig. 1. The module Web crawler, requested Web page analyzer, and hyperlink analyzer fetches the content from the requested website on the basis of objectionable keywords. It does not fetch each requested website data, but only those website content which having objectionable keywords. Having objectionable keywords in the websites content does not certify that they are objectionable as discussed in section I. Now, the feature extraction module extract all the features as described in Table 2 from Web content. Once, all the features are extracted, these are provided as the input to the AQNN. The AQNN classifies it into objectionable or non-objectionable website. If the website is found objectionable, then the objectionable website/URL's database is updated by adding the

## Quantum Threshold Function()

Step-1: Initialization of different parameters

```

for  $t = 1$  to  $z$ 
     $z$  is user defined variable to update  $((\lambda')_t)^h$ .
     $(\lambda'_t)^h = ((\alpha_i^{Th})^h | (\alpha_{i+1}^{Th})^h)$ ;
     $(count1)^h = 0$ ;
     $(count2)^h = 0$ ;
     $(F_\lambda^0)^h = 0$ ;
     $(S_\lambda^0)^h = 0$ ;
     $(F_\lambda^*)^h = 0$ ;
     $(S_\lambda^*)^h = 0$ ;
     $F_\lambda^{global} = 0$ 
     $(S_\lambda^{global})^h = 0$ 
    Call conversion process $((\lambda'_t)^h)$  to generate real
    value  $(\lambda_t)^h$  using Eqs (11), (12) and Eqs (13)
    for  $i = 1$  to  $c_1$ 
         $net_A(i) = \sum (W_g^{real})^h \times X_i$ 
        if $(net_A(i) > (\lambda_t)^h)$ 
            increase  $(count1)^h$  by 1;
        endif
    endfor
    for  $j = 1$  to  $c_2$ 
         $net_B(j) = \sum (W_g^{real})^h \times Y_j$ 
        if $(net_B(j) \leq (\lambda_t)^h)$ 
            increase  $(count2)^h$  by 1;
        endif
    endfor
     $((F_\lambda^t)^h = (count1)^h + (count2)^h)$ ;
     $(F_\lambda^*)^h = \max((F_\lambda^t)^1, (F_\lambda^t)^2, \dots, (F_\lambda^t)^h)$ 
    update quantum bits for  $(\lambda'_{t+1})^h$  by using Eqs (9),
    (10) and Table 1.
    generate updated real coded value of  $(\lambda_{t+1})^h$  using
    conversion process.
     $(F_g)^h = (F_\lambda^*)^h$ 
endfor
return  $(F_g)^h$ ;
  
```

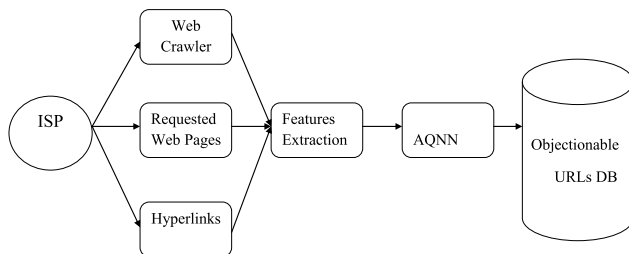


FIGURE 1. Outline of OWF.

particular website or its objectionable hyperlink websites. Each module of OWF is explained in detail next.

### 1) WEB CRAWLER

The efficiency of our OWF can further be enhanced by speeding up the process of finding objectionable websites

and making our objectionable URLs database rich. For this task, a web crawler is added to our proposed OWF. The primary objective of this Web crawler is to generate a list of suspicious websites with the help of objectionable keywords. Crawler works by searching on known search engines like Google, Yahoo, and Ask on the basis of objectionable keywords, thus, augmenting the list of websites. Consequently, the contents of these websites are fetched for feature selection and analysis. After extracting all the features of these selected websites, it is given to AQNN for classification of these websites as objectionable or non-objectionable. If a website is found objectionable, then it is added to the objectionable website/URL's database. Thus, if a user tries to access any objectionable website, then there is a higher chance of it to be present in our objectionable URLs database.

### 2) REQUESTED WEB PAGE ANALYSIS MODULE

This module works by analyzing the source code of the web pages requested by the user. The website requested by the user is first searched in the objectionable database. If the website is found in the database, then the message is sent back to the user specifying that this website is blocked and cannot be accessed. Otherwise, its content is extracted for features analysis. These features are provided as input to AQNN which classifies it into objectionable or non-objectionable website depending upon the results. If the website is found objectionable, then the objectionable website/URL's database is updated with this website.

### 3) HYPERLINK WEB PAGE ANALYSIS MODULE

Hyperlinks often referred to as just "links", are common in Web pages. These links act the same way as they do on the Web, allowing the user to jump from page to page. There are many websites which contain hyperlinks redirected to other websites. Therefore, to make our system more efficient, analyzing these hyperlinks for objectionable content has become an essential task. Therefore, each hyperlinked are extracted from Web content. After extracting these hyperlinks, this module fetches the content one by one. Once, the features are extracted from these hyperlinks, it is given to AQNN. If the hyperlink is found objectionable, then it is added to the objectionable URLs database. Many of the times it might be the case that a website may contain an objectionable hyperlink. Therefore, it will not be justified to block parent websites. During such situations, this module comes in handy as it only blocks the hyperlinks irrespective of their parent websites.

### 4) DETECTION OF MALICIOUS WEB REQUEST USING AQNN

This module with the help of AQNN, classify websites in the category of objectionable and non-objectionable. The classification into an objectionable and non-objectionable website is done on the basis of features extracted from the modules discussed above. The proposed AQNN is already discussed in detail in this section.



## V. ILLUSTRATION OF AQNN AND OWF

### A. WORKING OF AQNN

Working of AQNN can be understood by taking a simple example. Let us consider a database of three objectionable and three non-objectionable websites. The number of features of each website is two. The dataset belongs to objectionable websites is class *A*, which generate output 1 and dataset belongs to non-objectionable websites is class *B* which generate output 0. Thus class *A* dataset is ((0.23, 0.38), (0.19, 0.45), (0.11, 0.09)) and class *B* dataset is ((0.55, 0.85), (0.66, 0.72), (0.75, 0.88)). Now, initialization of quantum weights are done in each population. Let consider three numbers of population and their quantum weights are  $(W_1')^1 = (0.707|0.707, 0.707|0.707)$ ,  $(W_1')^2 = (0.5|0.5, 0.6|0.6)$ , and  $(W_1')^3 = (0.8|0.3, 0.1|0.4)$  for first neuron at hidden layer. Now, for all three populations, the instance conversion process is called to get the real coded value of weights. Let the value achieved after conversion process are  $(W_1^{real})^1 = (0.14, 0.25)$ ,  $(W_1^{real})^2 = (0.35, 0.44)$ , and  $(W_1^{real})^3 = (0.48, 0.55)$ . Now, these weight values are used to find out the boundary of the threshold. The boundary values achieved as (0.0379, 0.325), (0.0781, 0.6497), and (0.01023, 0.844) using multiplication of weight and input dataset, for three populations respectively. Now, after deciding the boundary parameter for threshold, the quantum threshold is initialized in terms of qubits for all three populations. Let quantum bits for threshold are  $(\lambda_1')^1 = (0.707|0.707)$ ,  $(\lambda_1')^2 = (0.5|0.8)$ , and  $(\lambda_1')^3 = (0.3|0.6)$ . After that conversion process is called and use boundary parameter to get the appropriate value of the threshold. The value achieved as real coded value of the threshold are  $(\lambda_1^{real})^1 = (0.256)$ ,  $(\lambda_1^{real})^2 = (0.277)$ , and  $(\lambda_1^{real})^3 = (0.504)$ . Now by using the threshold boundary parameter, the final value of threshold is achieved. Once, the threshold value is achieved, the weights are applied to input dataset, and it is compared with a threshold. The objective functions values achieved after comparison as  $(F_\lambda^1)^1 = 4$ ,  $(F_\lambda^1)^2 = 4$ , and  $(F_\lambda^1)^3 = 6$ . It can be observed from the results that, the whole sample is learned for the third population. Thus the connection weights and threshold of the third population is considered as final parameters for the neural network. The above example is a minor depiction of the bigger problem; therefore a proper solution is achieved in only one generation. If the sample is not learned in the first generation of the threshold, then it is updated until all samples are learned, or counter reaches to the user-defined number. The exchange of qubit is also done in this duration after some user-defined interval, which helps to get optimal value for connection weights and threshold.

### B. WORKING OF OWF

The working of OWF starts from the Web crawler. The Web crawler fetches all the objectionable websites on the basis of objectionable keywords mentioned in the list. There may be a possibility that all the websites fetched by Web crawler

may not be objectionable, therefore, to properly investigate websites, their contents are extracted into a separate file. Now, all the hyperlinks are separate out from this content. The feature extraction module extract features from these contents. To check whether the website is objectionable or not, the value of the features are given as input to AQNN. If the website is found objectionable then, the database of objectionable Website/URL is updated by adding this particular website. Thus, on the basis of objectionable keywords, and semantic analysis by AQNN, the database will have the number of objectionable links. Now, if any user request for any website first it is checked in the database of objectionable website. If the requested website is found in the database, then an error message is delivered else it is forwarded to Internet Service Provider (ISP). The response from ISP is forwarded to the user and requested Web page analyzer also keeps its content. Now, the requested Web page analyzer separates all hyperlink from content and forwarded to hyperlink module. The feature extraction module finds features from content extracted by requested Web page analyzer and gives input to AQNN. If it is found objectionable, then the database of objectionable websites is updated by the requested website. The hyperlink extractor module also accesses the content of each website/hyperlink one by one and its feature are extracted from the feature extraction module. These all features from hyperlink are given to AQNN to check whether hyperlink content is objectionable or not. If it is found objectionable, then the database of objectionable websites is updated by these hyperlinks. In this way, each module of OWF help to filter objectionable Web content.

## VI. EXPERIMENT AND RESULTS

In this section, the experimental setup and results are discussed in detail. Firstly, the experimental setup, the description of the system, and the experimental dataset are discussed. In the next subsection, after the experimental setup, the results of experiments on several evaluation parameters are discussed. Firstly, the evaluation of AQNN as the classifier is done, and then the performance of OWF is checked with existing software and objectionable content filtering models.

### A. EXPERIMENT SETUP

The proposed AQNN and OWF are implemented in Java (Version7) and on a computer with an i7 processor, 2.5GHz, and 8GB RAM. For the testing and training purpose, total 1000 objectionable and 1000 non-objectionable websites have been randomly selected. The dataset has been prepared in two classes, one is objectionable, and another is non-objectionable, both are having eight numbers of features as described in Table 2. In the proposed algorithm, objectionable content has been considered as pornographic. In the machine learning field, it is common to partition the dataset into two separate sets: a training set and a testing set. To evaluate the efficiency of AQNN, the dataset has been divided into the training and testing data in three different partitions.

Here, the first partition is done by 60-40, where 60 % of the samples are used for training of the AQNN and the rest of 40% for testing. The second partition is 70-30, 70 % of the samples are used for training and the remaining 30% used for testing. In addition to this, a 10-fold cross validation scheme is also used. In this method, the entire dataset is divided into ten blocks of approximately equal size. During the implementation of our algorithm, 90% of the data is used to train AQNN while the rest 10% data is used for testing. The details of the training and testing partitions are given in Table 3.

**TABLE 3. Data partition table.**

Data Partition			
Data Partition	Total	Training Set	Testing Set
60-40	2000	1200	800
70-30	2000	1400	600
10-fold cross validation	2000	1800	200

## B. RESULTS AND DISCUSSION

To evaluate the performance of proposed algorithm AQNN, the measures as mean classification accuracy, sensitivity, specificity, confusion matrix are calculated using the parameters such as true positive, true negative, false positive, and false negative [43] as shown in Table 4. Along with these evaluation parameters some more statistical evaluation parameters like precision, recall, and F1 score is calculated.

**TABLE 4. Sample confusion matrix.**

Predicted Class			
Actual Class	Class	A	B
	A	True Positive(TP)	False Negative(FN)
	B	False Positive(FP)	True Negative(TN)

As shown in Table 5, for 60-40 training-testing partition, the mean classification accuracy achieved by the proposed (AQNN) algorithm is 96.875% which is very remarkable, it shows that even for smaller training samples, AQNN has good adaptation and generation capability and can provide better solutions. The mean classification accuracy of the proposed models for 70-30 training-testing partition is 98%, and for 10-fold cross-validation, the classification accuracy achieved is 99% which confirms the importance of proper training data. AQNN achieves maximum accuracy of 97.9%, 98.6%, and 100% for 60-40, 70-30 training-testing partition and 10-fold cross validation, respectively. The minimum, maximum, and mean values of sensitivity and specificity of all training and testing partitions for AQNN are presented in Table 6. It shows a comparison between sensitivity and specificity. For the 60-40 training-testing partition, the best value achieved by the neural network for sensitivity and specificity is reported in terms of mean and standard deviation (std. dev.) is 99.75%, 0.16, and 94%, 0.44, respectively.

**TABLE 5. Comparison of classification accuracy.**

Classification Accuracy				
Classifiers	Training-Testing Partition (%)	Mean	Std. Dev.	Max
AQNN	60-40	96.875	0.55	97.9
	70-30	98	0.23	98.6
	10-fold cross validation	99	0.76	100
QNN-F	60-40	96.375	0.86	97.53
	70-30	97.83	0.36	99.2
	10-fold cross validation	98	0.57	98.5
Back Propagation	60-40	91.5	0.46	94.12
	70-30	91.34	0.83	94.8
	10-fold cross validation	92	0.31	97.2
SVM	60-40	94.125	0.4	95.63
	70-30	95	0.58	97.84
	10-fold cross validation	94	0.23	98.3
QNN	60-40	95.125	0.32	96.4
	70-30	96	0.51	97.1
	10-fold cross validation	97	0.25	99
ANN	60-40	94.125	0.33	95.4
	70-30	95.34	0.8	97
	10-fold cross validation	96	0.27	98.3
MLP	60-40	93.375	0.13	95.2
	70-30	93.34	0.56	95.8
	10-fold cross validation	92.5	0.49	97.1
Decision Tree	60-40	92.875	0.21	94
	70-30	93.167	0.41	94.8
	10-fold cross validation	94	0.32	97.4

**TABLE 6. Sensitivity and specificity.**

Sensitivity and Specificity					
Classifiers	Training-Testing Partition (%)	Sensitivity		Specificity	
		Mean	Std. Dev.	Mean	Std. Dev.
AQNN	60-40	99.75	0.16	94	0.44
	70-30	99.34	0.29	96.67	0.15
	10-fold cross validation	99	0.13	99	0.84
QNN-F	60-40	98	0.53	92.75	0.26
	70-30	98.75	0.62	95.67	0.74
	10-fold cross validation	97.05	0.22	98.9	0.53
Back Propagation	60-40	97.75	0.36	85.25	0.5
	70-30	96.67	0.2	86	0.23
	10-fold cross validation	91	0.41	93	0.82
SVM	60-40	98.25	0.57	90	0.43
	70-30	97.67	0.42	92.33	0.77
	10-fold cross validation	93	0.12	95	0.32
QNN	60-40	99	0.74	91.25	0.61
	70-30	98.67	0.29	93.34	0.41
	10-fold cross validation	97	0.3	97	0.37
ANN	60-40	98.75	0.51	89.5	0.55
	70-30	98.34	0.48	92.34	0.09
	10-fold cross validation	96	0.21	96	0.4
MLP	60-40	98	0.56	88.75	0.31
	70-30	97	0.88	89.67	0.12
	10-fold cross validation	92	0.11	93	0.48
Decision Tree	60-40	98.25	0.36	87.5	0.35
	70-30	97.67	0.6	88.67	0.19
	10-fold cross validation	94	0.74	94	0.91

For the 70-30 training-testing partition, the best value achieved by the neural network for sensitivity and specificity is reported in terms of mean and std. dev. is 99.34%, 0.29 and 96.67%, 0.15, respectively. For the 10-fold cross validation, the best value achieved by the neural network for sensitivity and specificity is reported in terms of mean and std. dev. are 99%, 0.13 and 99%, 0.84, respectively.

Table 7, represents the classification accuracies of AQNN in the form of the confusion matrix. It is clear from Table 7, the sum of true positive and true negative increases with the increase in training data. Especially, for 70-30 training-testing partition and 10-fold cross validation, the true positive rate and the true negative rate is around 100.

**TABLE 7. Confusion matrix.**

Confusion Matrix							
Classifiers		60-40		70-30		10-fold Cross Validation	
		Objectionable	Non-objectionable	Objectionable	Non-objectionable	Objectionable	Non-objectionable
AQNN	Objectionable	399	1	298	2	99	1
	Non-objectionable	24	376	10	290	1	99
QNN-F	Objectionable	400	0	300	1	99	1
	Non-objectionable	29	371	13	287	3	97
Back Propagation	Objectionable	391	9	290	10	91	9
	Non-objectionable	59	341	42	258	7	93
SVM	Objectionable	393	7	293	7	93	7
	Non-objectionable	40	360	23	277	5	95
QNN	Objectionable	396	4	296	4	97	3
	Non-objectionable	35	365	20	280	3	97
ANN	Objectionable	395	5	295	5	96	4
	Non-objectionable	42	358	23	277	4	96
MLP	Objectionable	392	8	291	9	92	8
	Non-objectionable	45	355	31	269	7	93
Decision Tree	Objectionable	393	7	293	7	94	6
	Non-objectionable	50	350	34	266	6	94

**TABLE 8. Comparison of Precision, Recall, and F1 Score.**

Precision, Recall, and F1 Score									
Data Partition	Parameter/Approache	AQNN	QNN-F	Backpropagation	SVM	QNN	ANN	MLP	Decision Tree
60-40	Precision	0.988	0.985	0.9775	0.9825	0.9875	0.9875	0.978	0.9825
	Recall	0.9975	0.98	0.9775	0.9825	0.99	0.9875	0.98	0.9825
	F1 Score	0.9927	0.9824	0.9775	0.9825	0.9887	0.9875	0.979	0.9825
70-30	Precision	0.989	0.988	0.967	0.9766	0.9866	0.9833	0.968	0.9766
	Recall	0.9934	0.9875	0.9667	0.9767	0.9867	0.9834	0.97	0.9767
	F1 Score	0.9912	0.9877	0.9666	0.9766	0.9866	0.9833	0.969	0.9766
10-fold cross validation	Precision	0.99	0.99	0.91	0.929	0.97	0.96	0.92	0.94
	Recall	0.99	0.9705	0.91	0.93	0.97	0.96	0.92	0.94
	F1 Score	0.99	0.9801	0.91	0.9295	0.97	0.96	0.92	0.94

In Table 8, The performance of the proposed algorithm is compared with other classification algorithms in terms of parameters like precision, recall, and F1 score. Here, it can be observed that the value of precision and recall and score for the data partition 60-40 and 70-30 are 0.988, 0.9975, 0.9927 and 0.989, 0.9934, 0.9912. The value of precision, recall, and F1 score for 10-fold cross validation scheme are 0.99, 0.99, and 0.99.

### C. COMPARISON AND DISCUSSION

To judge the performance of proposed AQNN, it is compared with QNN-F algorithm [18], machine learning algorithms like Backpropagation, Support Vector Machine (SVM), multilevel perceptron, decision tree algorithm, artificial neural network, and quantum neural network algorithm. Since, AQNN is an advanced version of QNN-F [18], it is observed that AQNN is improving results significantly which is discussed subsequently. Table 5, shows the classification accuracy of AQNN in terms of parameters like mean, standard deviation (std. dev.), and maximum (max) value. The mean, std. dev., and max values for the dataset ratio of 60-40, and 70-30 are 96.88%, 0.55, 97.9%, and 98%, 0.23, 98.6% respectively. Whereas, with dataset ratio of 60-40 and 70-30 the value of the mean, std. dev., and max for QNN-F

are 96.375%, 0.86, 97.53% and 97.83%, 0.36, 99.2% respectively. For 10-fold cross validation the values of mean, std. dev., and max are 99%, 0.76 and 100% respectively. The of mean, std. dev., and max for QNN-F are 98%, 0.57, and 98.5% respectively. This clearly shows that AQNN performs better than QNN-F, and other existing classifiers. Similarly, the comparison is made in terms of sensitivity and specificity as discussed in Table 6. It shows that, for the dataset ratio of 60-40, 70-30, and 10-fold cross validation are 99.75%, 94%, 99.34%, 96.67%, and 99%, 99% respectively. The mean value of sensitivity and specificity of QNN-F for dataset ratio of 60-40, 70-30, and 10-fold validation are 98%, 92.75%, 98.75%, 95.67%, and 97.05%, 98.9% respectively. Table 7 shows the confusion matrix corresponding to all methods. The results show that with respect to the objectionable dataset, each algorithm performs well, but corresponding to the non-objectionable dataset, AQNN performs better than other classification algorithms in both the data partitions.

The performance of the proposed AQNN is better than QNN-F and existing classifiers, this is mainly due to two reasons. Firstly, introducing here threshold boundary parameter, which helps to find an optimal value of threshold within a particular range. As finding an optimal value of the threshold

**TABLE 9. Features comparison of OWF with other method and renowned softwares.**

Method	Location (user/admin)	URL Block	Keyword based	Content analysis	Filtering domain
URLfilterDB	Server	Yes	Yes	Yes	General
DansGuardian	Server	Yes	Yes	Yes	General
Net Nanny	Client	Yes	Yes	No	General
Web Shield	Server	Yes	Yes	Yes	General
QNN-F	Server	Yes	Yes	Yes	General
OWF	Server	Yes	Yes	Yes	General

is important for any neural network architecture. Due to improper selection of this parameter, the algorithm may get trapped into local minima or maxima. Thus, with an optimal value of the threshold, a neural network is formed which gives high classification accuracy. Secondly, in the AQNN finding the connection weights and threshold using genetic concept with three different populations. Thus, it gives the better value of connection weights and threshold from one population without genetic concept. As in QNN-F the connection weights and the threshold of the neurons are searched in one population using several generations. In AQNN the parameters of the neural network are found in more than one population in several generations in each population. The genetic concept is used here to exchange and muted qubits of connection weights and threshold thus, it helps to find the optimal value of connection weights and threshold globally.

However, the proposed AQNN is compared with QNN-F, Backpropagation, Support Vector Machine (SVM), multi-level perceptron, decision tree algorithm, artificial neural network, and quantum neural network algorithm but it is hard to make a practical comparison between the existing work of filtering objectionable Web content with the proposed OWF. There are some reasons for this. Firstly, the unavailability of the standard dataset on which the other algorithms are tested. These datasets are prepared in different-different ways from the Web. Secondly, the number of features of extracted Web content is not uniform. Many researchers have opted their own methodology to extract features from the content. Thirdly, many researchers used standard or library classifier like Backpropagation, Support Vector Machine (SVM), multi-level perceptron, decision tree algorithm for classification their Web contents [45]–[47]. Therefore, to judge the performance of the OWF, the comparison is done with existing well known objectionable Web blocking Softwares as URLfilterDB, DansGuardian, and Net Nanny [48]–[50] and existing some objectionable Web blocking model Webshield and QNN-F [18], [23]. First, the features of the Softwares and existing model is compared with our proposed OWF in Table 9. The comparison is made on the basis of features like Location (user/admin), URL Block, Keyword based, Content analysis, and Filtering domain. To evaluate the performance of the proposed OWF with existing well-known Softwares and existing model, the testing is done by taking 800 (400 objectionable + 400 non-objectional) websites. Table 10, shows the testing results in terms of confusion matrix and accuracy. It can be easily observed from Table 10, that the proposed OWF achieves the highest accuracy that is 96.87% in comparison to existing Softwares and existing method.

**TABLE 10. Comparison of OWF with other method and renowned softwares.**

	Websites				Accuracy (%)
	Objectionable		Non-Objectionable		
	Correctly classified	Incorrectly classified	Correctly classified	Incorrectly classified	
URLfilterDB	384	16	369	31	94.125
DansGuardian	367	33	375	25	92.75
Net Nanny	359	41	372	28	91.38
Web Shield	360	40	352	48	89
QNN-F	400	0	371	29	96.375
OWF	399	1	376	24	96.87

## VII. CONCLUSION

In this paper, an Advanced Quantum based Neural Network Classifier (AQNN) is presented, and it is used to form an Objectionable Web filtering system (OWF). The proposed algorithm uses quantum computing and genetic concept to decide connection weights of the neural network and threshold of neurons. The quantum concept is characterized by the representation of the individual and the population dynamics. It gives better population diversity to get the optimal value of required parameters (connections weights and threshold). The network architecture is formed constructively by adding neurons one by one at the hidden layer. The proposed AQNN architecture is used to form OWF. The OWF analyzes the content of websites collected from the different module like Web crawler, requested Web page analyzer, and hyperlink analyzer. The requested Web page analyzer gives the content of websites which are requested by users, and the hyperlink analyzer helps to find out the content from hyperlink available in websites content. To avoid repeat analysis of each requested websites, the collection of all objectionable Websites declared by AQNN is done, and a database is created as objectionable Website/URL database. The Web crawler module helps to speed up the process of blocking objectionable Web request by updating the database of objectionable Website/URLs.

To judge the performance of the proposed AQNN algorithm and OWF, these are tested on objectionable and non-objectionable Web dataset. The performance of the AQNN is also checked against well-known classifiers with the parameters like accuracy, specificity, sensitivity, precision, recall, and F1 score. Here, the performance of proposed OWF is also compared with objectionable Web filtering Softwares as URLfilterDB, DansGuardian, and Net Nanny and existing methods. It is found from the result that both AQNN and OWF performs better than existing solutions.

## REFERENCES

- [1] J.-H. Xia, Rusli, and A. Kumta, "Modeling of silicon carbide ECR etching by feed-forward neural network and its physical interpretations," *IEEE Trans. Plasma Sci.*, vol. 38, no. 5, pp. 1091–1096, May 2010. [Online]. Available: [http://research.ntu.edu.sg/expertise/academicprofile/pages/StaffProfile.aspx?ST\\_EMAILID=ERUSLI](http://research.ntu.edu.sg/expertise/academicprofile/pages/StaffProfile.aspx?ST_EMAILID=ERUSLI)
- [2] R. Batruni, "A multilayer neural network with piecewise-linear structure and back-propagation learning," *IEEE Trans. Neural Netw.*, vol. 2, no. 3, pp. 395–403, May 1991.



- [3] E.-S. M. El-Alfy, N. T. Ratrou, and U. Gazder, "Impact of stock market indices and other regional exogenous factors on predictive modeling of border traffic with neural network models," *Arabian J. Sci. Eng.*, vol. 40, no. 2, pp. 303–312, 2015.
- [4] M. Alshayeb, Y. Eisa, and M. A. Ahmed, "Object-oriented class stability prediction: A comparison between artificial neural network and support vector machine," *Arabian J. for Sci. Eng.*, vol. 39, no. 11, pp. 7865–7876, 2014.
- [5] M. Deriche, "An image quality index based on mutual information and neural networks," *Arabian J. Sci. Eng.*, vol. 39, no. 3, pp. 1983–1993, 2014.
- [6] S. Kumar, *Neural Networks: A Classroom Approach*. New York, NY, USA: McGraw-Hill, 2004.
- [7] W. A. Farag, V. H. Quintana, and G. Lambert-Torres, "A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 756–767, Sep. 1998.
- [8] M. B. Takahashi, J. C. Rocha, and E. G. F. Núñez, "Optimization of artificial neural network by genetic algorithm for describing viral production from uniform design data," *Biochemistry*, vol. 51, no. 3, pp. 422–430, 2016.
- [9] D. Rivero, J. Dorado, J. Rabuñal, and A. Pazos, "Generation and simplification of artificial neural networks by means of genetic programming," *Neurocomputing*, vol. 73, nos. 16–18, pp. 3200–3223, 2010.
- [10] H.-C. Tsai and Y.-H. Lin, "Modular neural network programming with genetic optimization," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11032–11039, 2011.
- [11] M. Chen and Z. Yao, "Classification techniques of neural networks using improved genetic algorithms," in *Proc. IEEE Int. Conf. Genetic Evol. Comput. (WPEC)*, Hubei, China, Sep. 2008, pp. 115–119.
- [12] T.-C. Lu, G.-R. Yu, and J.-C. Juang, "Quantum-based algorithm for optimizing artificial neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1266–1278, Aug. 2013.
- [13] J. H. Kim and S.-K. Park, "The geometrical learning of binary neural networks," *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 237–247, Jan. 1995.
- [14] O. P. Patel and A. Tiwari, "Quantum inspired binary neural network algorithm," in *Proc. IEEE Int. Conf. Inf. Technol. (ICIT)*, Bhubaneswar, India, Dec. 2014, pp. 270–274.
- [15] J.-J. Wong and S.-Y. Cho, "Recognizing human emotion from partial facial features," in *Proc. IEEE Int. Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2006, pp. 166–173.
- [16] E.-S. M. El-Alfy, "A hierarchical GMDH-based polynomial neural network for handwritten numeral recognition using topological features," in *Proc. IEEE Int. Conf. Neural Netw. (IJCNN)*, Barcelona, Spain, Jul. 2010, pp. 1–7.
- [17] T. D. Do, K. Chang, and S. C. Hui, "Web mining for cyber monitoring and filtering," in *Proc. IEEE Int. Conf. Cybern. Intell. Syst.*, Singapore, Dec. 2004, pp. 399–404.
- [18] O. Patel, A. Tiwari, V. Patel, and O. Gupta, "Quantum based neural network classifier and its application for firewall to detect malicious Web request," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Cape Town, South Africa, Dec. 2015, pp. 67–74.
- [19] N. Agarwal, H. Liu, and J. Zhang, "Blocking objectionable Web content by leveraging multiple information sources," *ACM SIGKDD Explor. Newsl.*, vol. 8, pp. 17–26, 2006.
- [20] W. P. Stol, H. K. W. Kaspersen, J. Kerstens, E. R. Leukfeldt, and A. R. Lodder, "Governmental filtering of websites: The Dutch case," *Comput. Law Secur. Rev.*, vol. 25, no. 3, pp. 251–262, 2009.
- [21] Y.-S. Kim, "A study of PICS/RDF-based Internet content rating system: Issues related to freedom of expression," *J. Korean Soc. Inf. Manage.*, vol. 24, no. 3, pp. 271–297, 2007.
- [22] C.-H. Chou, A. P. Sinha, and H. Zhao, "Commercial Internet filters: Perils and opportunities," *Decis. Support Syst.*, vol. 48, no. 4, pp. 521–530, 2010.
- [23] U. Thakar, O. Patel, and L. Purohit, "Web shield: A modified firewall to detect malicious request," in *Proc. Int. Conf. Inf. Syst. Design Intell. Appl. (INDIA)*, New Delhi, India, Berlin, Germany: Springer, 2012, pp. 497–506.
- [24] M. Hammami, Y. Chahir, and L. Chen, "WebGuard: A Web filtering engine combining textual, structural, and visual content-based analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 2, pp. 272–284, Feb. 2006.
- [25] M. Hammami, D. Tsishkou, and L. Chen, "Adult content Web filtering and face detection using data-mining based kin-color model," in *Proc. IEEE Int. Conf. Multimedia Expo*, Taipei, Taiwan, vol. 1, Jun. 2004, pp. 403–406.
- [26] M. R. Girgis, T. M. Mahmoud, and T. Abd-El-Hafeez, "An approach to image extraction and accurate skin detection from Web pages," in *Proc. World Acad. Sci., Eng. Technol.*, 2007, pp. 367–375.
- [27] Z. Chen, O. Wu, M. Zhu, and W. Hu, "A novel Web page filtering system by combining texts and images," in *Proc. IEEE Web Intell.*, Dec. 2006, pp. 732–735.
- [28] X. Shen, W. Wei, and Q. Qian, "The filtering of Internet images based on detecting erotogenic-part," in *Proc. IEEE Natural Comput.*, vol. 2, Aug. 2007, pp. 732–736.
- [29] M. B. Garcia, T. F. Revano, B. G. M. Habal, J. O. Contreras, and J. B. R. Enriquez, "A pornographic image and video filtering application using optimized nudity recognition and detection algorithm," in *Proc. 10th IEEE Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Nov./Dec. 2018, pp. 1–5.
- [30] W. F. M. Yaqub, M. Mohanty, and N. D. Memon, "Encrypted domain skin tone detection for pornographic image filtering," in *Proc. 15th IEEE Int. Conf. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–5.
- [31] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *Int. J. Comput. Vis.*, vol. 46, no. 1, pp. 81–96, Jan. 2002.
- [32] H. Zheng, M. Daoudi, and B. Jedynak, "Blocking adult images based on statistical skin detection," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 4, no. 2, pp. 1–14, 2004.
- [33] Y. Xu, B. Li, X. Xue, and H. Lu, "Region-based pornographic image detection," in *Proc. IEEE Multimedia Signal Process.*, Oct./Nov. 2005, pp. 1–4.
- [34] J.-L. Shih, C.-H. Lee, and C.-S. Yang, "An adult image identification system employing image retrieval technique," *Pattern Recognit. Lett.*, vol. 28, no. 16, pp. 2367–2374, 2007.
- [35] V. S. Jacob, R. Krishnan, and Y. U. Ryu, "Internet content filtering using isotonic separation on content category ratings," *ACM Trans. Internet Technol.*, vol. 7, no. 1, p. 1, 2007.
- [36] J. P. Caulkins, W. Ding, G. Duncan, R. Krishnan, and E. Nyberg, "A method for managing access to Web pages: Filtering by statistical classification (FSC) applied to text," *Decis. Support Syst.*, vol. 42, no. 1, pp. 144–161, 2006.
- [37] J. Zeng, J. Duan, and C. Wu, "Adaptive topic modeling for detection objectionable text," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. (WI) Intell. Agent Technol. (IAT)*, Atlanta, GA, USA, Nov. 2013, pp. 381–388.
- [38] H. Zheng, H. Liu, and M. Daoudi, "Blocking objectionable images: Adult images and harmful symbols," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Taipei, Taiwan, vol. 2, Jun. 2004, pp. 1223–1226.
- [39] W. Zeng, W. Gao, T. Zhang, and Y. Liu, "Image guarder: An intelligent detector for adult images," in *Proc. Asian Conf. Comput. Vis.*, Seoul, South Korea, 2004, pp. 1080–1084.
- [40] D. Li, N. Li, J. Wang, and T. Zhu, "Pornographic images recognition based on spatial pyramid partition and multi-instance ensemble learning," *Knowl.-Based Syst.*, vol. 84, pp. 214–223, 2015.
- [41] C.-Y. Kim, O.-J. Kwon, W.-G. Kim, and S. Choi, "Automatic system for filtering obscene video," in *Proc. IEEE Int. Conf. Adv. Commun. Technol.*, Gangwon-Do, South Korea, Feb. 2008, pp. 1435–1438.
- [42] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.
- [43] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.
- [44] G. W. Corder and D. I. Foreman, *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*. Hoboken, NJ, USA: Wiley, 2009.
- [45] J. Sharma and V. K. Pathak, "Automatic pornographic detection in Web pages based on images and text data using support vector machine," in *Proc. Int. Conf. Soft Comput. Problem Solving (SocProS)*. Roorkee, India: Springer, 2011, pp. 473–483.
- [46] R. Suvorov, I. Sochenkov, and I. Tikhomirov, "Method for pornography filtering in the Web based on automatic classification and natural language processing," in *Proc. Int. Conf. Speech Comput.*, Pilsen, Czech Republic. Cham, Switzerland: Springer, 2013, pp. 233–240.
- [47] F. V. Rigo, M. N. Maraes, and E. T. Matsubara, "Smart saint: An active semi-supervised learning Internet filter," in *Proc. IEEE Int. Conf. Intell. Syst. (BRACIS)*, Fortaleza, Brazil, Oct. 2013, pp. 136–142.
- [48] M. Kool. *URLfilterDB*. Accessed: Dec. 15, 2018. [Online]. Available: <https://www.urlfilterdb.com/>
- [49] SmoothWall. *DansGuardian*. Accessed: Dec. 16, 2018. [Online]. Available: <http://dansguardian.org/>
- [50] G. Ross. *NetNanny*. Accessed: Dec. 18, 2018. [Online]. Available: <https://www.netnanny.com/products/netnanny/>

**OM PRAKASH PATEL** received the Ph.D. degree in computer science and engineering from IIT Indore, India, in 2018, and the M.E. degree from the Department of Computer Science and Engineering, Shri Govindaram Sakseria Institute of Technology and Science, Indore, India, in 2011. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Mahindra Ecole Centrale College of Engineering, Hyderabad, India. His current research interests include quantum-based neural network learning algorithm, pattern recognition, data mining, and fuzzy-based soft computing algorithm. He is a Reviewer of the *Neural Processing Letter* (Springer).

**NEHA BHARILL** received the B.E. degree in information technology from the Bansal Institute of Science and Technology, Bhopal, India, in 2008, the M.E. degree in computer engineering from the Shri Govindaram Sakseria Institute of Technology and Science, Indore, India, in 2011, and the Ph.D. degree in computer science and engineering from IIT Indore, India, in 2018. She is currently an Assistant Professor with the Department of Computer Science and Engineering, Mahindra Ecole Centrale College of Engineering, Hyderabad, India. Her current research interests include fuzzy sets and systems, big data, pattern recognition, data mining, machine learning, and scalable machine learning approaches for genome identification. She has published various refereed journals and conferences of International repute. She is the Reviewer of the IEEE TRANSACTIONS ON CYBERNETICS and the IEEE TRANSACTIONS ON FUZZY SYSTEMS.

**ARUNA TIWARI** received the B.E. and M.E. degrees in computer engineering and the Ph.D. degree in computer science and engineering. She has been an Associate Professor of computer science and engineering with IIT Indore, since 2012. Her research interests include soft computing, neural networks, fuzzy clustering, evolutionary computation, genome analysis, and health-care applications. She has many publications in peer-reviewed journals (of the IEEE Transactions, Elsevier, and Springer), international conferences, and book chapters. She has been a Reviewer for many reputed journals and conferences. She has research collaboration with CSIR CEERI Pilani and the Indian Institute of Soybean Research (Indian Council of Agriculture and Research).

**VIKRAM PATEL** received the B.Tech. degree in computer science and engineering from IIT Indore, India, in 2016. He is currently the Co-Founder and the Chief Technology Officer of Chota Hospital Private Limited. Prior to pursuing his startup full-time, he has published research on quantum-based neural network classifier and had interned at UpsQool, and Refresh.io (acquired by LinkedIn, in 2015). His current research interests include quantum-based neural network learning algorithm, pattern recognition, natural language processing, and automatic speech recognition (ASR).

**OJAS GUPTA** received the B.Tech. degree in computer science and engineering from IIT Indore, India, in 2016, and the M.Sc. degree in computer science from the University of California San Diego, in 2018. He is currently a member of the technical staff with VMware, Palo Alto, CA, USA. He is working on orchestration and networking with VMware's next-generation network slicing solution in 5G. His current research interests include software-defined networking, orchestration, telco networks, quantum-based neural networking, and convolution neural networks.

**JIAN CAO** received the Ph.D. degree from the Nanjing University of Science and Technology, in 2000. He was a Postdoctoral Research Fellow of Shanghai Jiaotong University (SJTU) and also a Visiting Scholar of Stanford University. He is currently a tenured Professor with the Department of Computer Science and Engineering, Shanghai Jiaotong University (SJTU), China. He is the Director of the SJTU&Morgan Stanley Joint Research Center on Financial Service Innovation. He is also the Leader of the Lab for Cloud and Service Computing. His research interests include intelligent data analytics, service computing, and network computing. He has published over 200 papers in refereed conference and journals, such as SIGKDD, IJCAI, AAAI, VLDB, WWW, INFOCOM, TMC, TOIS, TPDS, and TKDD. He has undertaken more than ten projects funded by the National High Technology Research and Development and the National Natural Science Foundation of China. His research projects are also widely founded by many industry partners, including Ctrip, Morgan Stanley, Docomo, Samsung, and Shanghai International Port Group, to name but a few. His research has won eight provisional Science and Technology Progress Awards.

**JUN LI** received the B.S. degree in computer science and technology from Shandong University, Jinan, China, in 2003, the M.Sc. degree in information and signal processing from Peking University, Beijing, China, in 2006, and the Ph.D. degree in computer science from the Queen Mary University of London, London, U.K., in 2009. He is currently a Lecturer with the School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia.

**MUKESH PRASAD** received the Ph.D. degree in computer science from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2015, and the master's degree in computer application from Jawaharlal Nehru University, New Delhi, India, in 2009. He was a Principle Engineer (Research and Development) with Taiwan Semiconductor Manufacturing Company, Hsinchu. He is currently a Lecturer with the School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. He has published 25 peer-reviewed journals and 30 conference articles. His current research interests include machine learning, data analytics, pattern recognition, fuzzy systems, neural networks, artificial intelligence, and brain-computer interface.

...