

Research Article

An Improved Artificial Neural Network Model for Effective Diabetes Prediction

Muhammad Mazhar Bukhari,¹ Bader Fahad Alkhamees ,² Saddam Hussain ,³
Abdu Gumaei ,⁴ Adel Assiri ,⁵ and Syed Sajid Ullah ³

¹Department of Computer Science, National College of Business Administration & Economics, Lahore, Pakistan

²Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

³IT Department, Hazara University, Mansehra 21120, KP, Pakistan

⁴Research Chair of Pervasive and Mobile Computing, Department of Information Systems,

College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

⁵Management Information Systems Department, College of Business, King Khalid University, Abha 61421, Saudi Arabia

Correspondence should be addressed to Saddam Hussain; saddamicup1993@gmail.com and Abdu Gumaei; abdugumaei@gmail.com

Received 5 February 2021; Revised 19 March 2021; Accepted 1 April 2021; Published 22 April 2021

Academic Editor: Michela Gelfusa

Copyright © 2021 Muhammad Mazhar Bukhari et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data analytics, machine intelligence, and other cognitive algorithms have been employed in predicting various types of diseases in health care. The revolution of artificial neural networks (ANNs) in the medical discipline emerged for data-driven applications, particularly in the healthcare domain. It ranges from diagnosis of various diseases, medical image processing, decision support system (DSS), and disease prediction. The intention of conducting the research is to ascertain the impact of parameters on diabetes data to predict whether a particular patient has a disease or not. This paper develops an improved ANN model trained using an artificial backpropagation scaled conjugate gradient neural network (ABP-SCGNN) algorithm to predict diabetes effectively. For validating the performance of the proposed model, we conduct a large set of experiments on a Pima Indian Diabetes (PID) dataset using accuracy and mean squared error (MSE) as evaluation metrics. We use different number of neurons in the hidden layer, ranging from 5 to 50, to train the ANN models. The experimental results show that the ABP-SCGNN model, containing 20 neurons, attains 93% accuracy on the validation set, which is higher than using the other ANNs models. This result confirms the model's effectiveness and efficiency in predicting diabetes disease from the required data attributes.

1. Introduction

The revolution in artificial neural networks (ANNs) within the medical discipline research domain emerged in data-driven applications, particularly in the healthcare sector. It ranges from diagnosis of various kinds of diseases, image-processing in the medical field, decision support system (DSS), and disease prediction. Many contemporary ANN models such as deep learning models, recurrent neural networks, and genetic algorithms are significant in artificial intelligence, robotics, image processing, and several other

cutting-edge technologies, especially in the health sector. ANN is an ideal tool that is used in the identification, analysis, and prediction of general and health sectors involving in Thrombo-embolic stroke disease, bone densitometry, hepatitis B, and breast cancer [1]. This paper presents the predictive framework for a successful diagnosis of diabetes using a dataset comprising female patients with corresponding attributes. It is quite pertinent to mention that typical regression models have been employed to solve the problem. Usually, these models' basis includes inference of statistical independence and interdependencies of their

input and output variables and uniformity of continuity and presence of external variables.

On the other hand, in most applications, these inferences are often offended or disregarded due to complicated physiological characteristics. To enhance patients' diabetes self-management, research in designing various models is in progress [2]. However, several models and programs have been developed to provide medical research benefits; rigorous testing of these models is partial and inadequate, resulting in very hard to perform and difficult to manage. Here, in this research, a well-organized and refined regression model, i.e., artificial backpropagation scaled conjugate gradient neural network (ABP-SCGNN), is proposed that predicts the underlying disease since ANN has been employed in medical research and analytical studies [3]. Backpropagation is supervised learning of ANN using gradient descent. The algorithm computes the gradient of the error function in connection with weights.

Diabetes is one of the diseases that affect problems associated with the hormone insulin. It may appear when the human body acts in response, particularly in a negative manner. Unfortunately, it has obtained the cure for nothing up till now, but people with diabetes can adopt certain precautions to manage their disease and enjoy a healthy life. This paper focuses on the successful prediction using ANN in the diagnostic approach by choosing some attributes carefully. Upon the positive predicted result, people may consult their doctors for prior precautions treatment. Typically, suffering from diabetes, patients become the victims of other serious diseases ranging from heart attack to kidney failure if not appropriately controlled [4]. In many countries, a significant cause of cardiovascular disease, blindness, kidney failure, and lower limb amputation is diabetes. The number of patients with diabetes has been increasing worldwide, nearly 200 million people are targeted, and more than half of the population was female. Females can quickly become victims of this disease during the age range from 25 to 44 and at high risk accordingly. Several research studies are in progress to predict diabetes for early diagnoses and cure hence [4, 5].

Such models can be designed and implemented using ANN that proved to be more helpful, efficient, and effective in several fields of medicine like analysis, diagnosis, and prediction and assist not only professionals but also ordinary people [5]. ANN is a representation of the human neural system mathematically, demonstrating the strength of training and generalization. Most of the ANN techniques are based on nonlinear functions in which the link or association of input features is either a bit complex or unknown. A series of nodes also called neurons that form an ANN is organized in different layers. In a typical statistical model of an ANN, each neuron is directly connected to the neurons of the other layers employing some weighted values that illustrate the strength or power of the connection between them [6, 7]. Each neuron input is affected with the weighted permutation of several input signals that may contain distinct computations and finally on the resultant output. These neurons apply transfer function to the weighted inputs to evaluate the threshold value. Using the activation function, the message

is sent to the next concerned neuron if the threshold value exceeds.

Based on the functions of ANNs, it is significant to understand when it presents prediction, perception classification, and pattern recognition along with training accordingly [7]. Despite this, on the other hand, significant work is carried out towards ANN development for the applications in medical fields, for example, classifications, clustering, data optimization, and prediction based on a given set of inputs. Any ANN model consists of several components comprising layers, specifically an input layer, one or more hidden layers, an output layer, several neurons, and their corresponding interaction. Choosing these features is sensitive as few features result in slow training, and on the contrary, many features decrease the overall network processing power. There are different ways to determine these components, typically the pruning method and growing method. One can go with many features in the pruning method and reduce the network size with those not important components [8]. The growing process that was used in the proposed work adds desired components gradually. Several algorithms have been applied to perform various operations on the dataset used in this paper to select the attributes, process the data, and predict diabetes accordingly. Diagnosis in the medical sector using the same diabetes dataset utilizing general regression neural networks (GRNNs), multilayer perceptron (MLP) neural networks, radial basis function (RBF), and feed-forward neural networks for comparing the performance of the MLP was tested for different types of backpropagation training algorithms [9]. The adaptive learning routine has been employed in [10].

The rest of the paper is organized as follows. Section 2 presents the previous related work of diabetes prediction. Section 3 explains the proposed methodology in which the ABP-SCGNN-based diabetes prediction framework is given in detail. Experiments and results are introduced in Section 4, and then a conclusion of the study with future work is shown in Section 5.

2. Literature Review

In 2011, Sapon et al. [1] took 250 diabetes patients who were both male and female ranging 25–78 years with 27 input variables to training the network to identify the disease pattern. Among the three algorithms, the Bayesian regulation algorithm presented the most excellent result in the prediction of diabetes to Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm and Quasi-Newton and Levenberg–Marquardt algorithms. BFGS Quasi-Newton possesses 0.86714 correlation coefficients with 578 epochs while Bayesian regulation acquires 0.99579 for 37 epochs only and Levenberg–Marquardt holds 0.6051 for only five epochs. In [1], the Bayesian regulation algorithm presented a good correlation between estimated targets and actual outputs (i.e., 0.99579) with 88.8% prediction accuracy that affirms the validation that exhibits the appropriateness of this algorithm to carry out the successful diabetes prediction.

For the sake of dataset classification, in 2012, Choubey et al. [8] applied a naïve Bayes (NBs) classifier, as well as a genetic algorithm (GA) with NBs methods to predict diabetes in females aged from 21–78 years. The total number of instances was 768. First, using naïve Bayes, the process of classification has been performed on PIDD, and a genetic algorithm was used to add and remove attributes from the dataset. It comparatively decreased the computational cost as well as time and increased ROC and classification accuracy. The result comparison on PIDD regarding accuracy with ROC, GA, and NB highlights the most accurate result and better ROC compared with other methods.

In 2003, Kayaer and Yildirim [9] applied MLP neural networks, RBF, and GRNNs on the Pima Indians Diabetes data. The Levenberg–Marquardt training algorithm has demonstrated the best result against the training data. The accuracy of RBF was not better than that of MLP, even utilizing the all-underlying values. The GRNNs attained the best result using test data (i.e., 80.21%). This algorithm proves to be an excellent and pragmatic selection for the successful classification of diabetes data.

In 2016, Florez et al. [11] used 700 instances as training data that were selected randomly using software *R* to predict the intensity of diabetes. MSE was 2.952. Next, the higher risk subgroups of characteristics that lead to diabetes were obtained. The unmodified model is the number of times pregnant (PRG), plasma concentration in saliva (PLASMA), BP, body mass index (BODY), and diabetes pedigree function (PEDIGREE). PRG, PLASMA, BODY, and PEDIGREE have certain effects on predicting diabetes according to their coefficient. A large number of variables with a higher probability of containing diabetes possess the MSE to be 3.21068. Therefore, the concluding variables include PRG, PLASMA, BODY, and PEDIGREE.

Iqbal et al. [12] proposed an intelligent system for smart-cities using clusters and fuzzy inference systems to predict traffic congestion. The solution for traffic congestion is presented using smart mobile technologies. It motivates the idea for the prediction of diabetes using computational intelligence technologies.

Since the proposed framework is based on the ABP-SCGNN algorithm, the experiment might proceed with a small number of components, and a decision could be made on the performance of the learning curve. The features of the network can be concluded upon generating a small error rate when training complete.

3. Proposed Methodology

People of any age group may become the victim of diabetes. The reasons may vary among different age groups, gender, living style, glucose and insulin level, BP, and so on. At present, numerous algorithms, such as ANNs, SVM, and naïve Bayes (NBs) with fuzzy logic (FL), are incorporated for predicting the diagnosis of diabetes. These algorithms have their trade-offs regarding their processing time and accuracy rate, obtaining hidden information from the given data. Since in complex problem domains, finding an optimum solution may not require a trivial approach [13–16] and the

proposed framework drawn in the following figure demonstrates the working process of ANN algorithms for predicting, validating, and systematically testing the network for the intended purpose to improve the self-reliance and significant certainty. Operations and parameters are yielded during ANN training, and a careful comparison is made on predicted and obtained values used for ANN optimization. The whole course of action is explained in Figure 1 schematically.

According to the framework, the network obtains dataset that consists of female diabetes patients' detail for predicting the disease diagnosis. Since the dataset contains different attributes/features (glucose, insulin, and BP) relevant to the patients' detail, a careful selection of these features must be performed as some features may contribute to misleading the result because of noise or null data. Sometimes, the value range among attributes is high. It is recommended to apply a formalized process for the sake of minimizing the erroneous result while finding a relatively improved result [9]. Data that need to train must be preprocessed before the evaluation process. Network architecture varies from the classifier to classifier, exhibiting the underlying algorithm parameters that are dependent on the classifier that is supposed to train the network. Other diabetes patients can use this network for predicting the diagnoses of their disease. To end with it, a general physician of diabetes or doctor of medicine performs an evaluation and appraisal process on this expected diagnosis. A detailed explanation of the proposed framework is provided as follows.

3.1. Data Collection and Representation. The dataset used in the study is obtained from the National Institute of Diabetes and Digestive and Kidney Diseases [17]. The purpose behind this is to predict the disease considering some selected diagnostic key attributes included in the dataset whether a person is a diabetes patient or not. The dataset contains the data for female patients only with minimum age 21 years for the resident of Arizona USA. The responder includes a binary value of either 0 (negative test representing diabetes) or 1 (positive test representing diabetes). Approximately 35% (268) patients are included in a class containing value 1. On the other hand, 65% (500) patients are included in another class containing a value of zero [9]. The researchers have mentioned many vital factors that may vary from patient to patient for diabetes currently or after a predefined period [18].

3.2. Feature Selection/Reduction. To diagnose diabetes correctly, key attributes are normally independent of each other. Medical experts examine the attributes carefully and decide if each is required in identifying the viable diagnosis. These attributes, called features in ANN, may be identified as diabetes symptoms and another relevant piece of information that assists in predicting diabetes. Choubey et al. [8] used GA for selecting attributes (features) and NBs for classification on PID dataset.

Since feature selection is a process of identifying the most relevant features extracted from the complete set, it

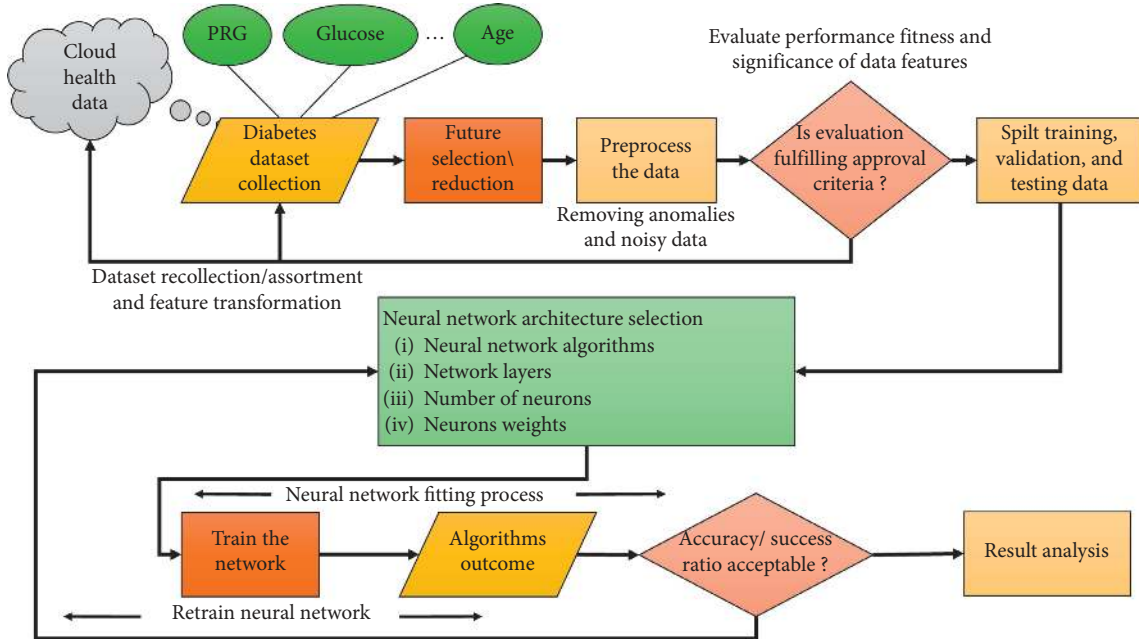


FIGURE 1: The proposed framework of diabetes prediction using neural network architecture selection.

increases the success ratio of predicting the diagnosis of diabetes patients [8]. Given the fact [18], these features must be robust and noise-free; hence, a careful selection is made in choosing these features with the entire dataset. Women may have diabetes from the sixth month of their pregnancy, and in addition to it, a certain level of glucose and insulin play a vital role in the diagnosis of diabetes. Eventually, there are eight explanatory input variables with one output responder variable that are included in the selected dataset. Each variable plays a vital role in diagnosing diabetes individually. Accumulating their values enables the network to be trained effectively and perform the diabetes prediction in return. The detail of both types of variables is explained in Table 1.

3.3. Preprocessing. Many methods are available to preprocess the data in the dataset before model evaluation [19, 20]. The preprocessed result must demonstrate the anticipated output. Learning rate, momentum, and time were taken; all are obtained in the response to the preprocessing process. Data must be transformed in the form of fulfilling the acceptance criterion. Accumulating the provided values in various features like blood pressure, skin thickness, insulin, and BMI must map with the outcome value. The diabetes dataset contains 768 instances with 8 input variables that are enough to predict the diabetes diagnosis using the simulation model. Noisy data have already been eliminated before receiving the dataset from the concerned authority [19].

3.4. Neural Network Fitting Process. The objective to accumulate the abovementioned information is to design such a neural network model that can predict precisely whether a particular patient possesses diabetes or not. Since ANN is the combination of many algorithms, some

algorithms are used in the paper to train the network, obtain the prediction ratio, and given that compare the mutual results. After choosing the patients' dataset, the next step is to segregate the data according to the requirements. In our case, data are needed to be divided into three main segments as training, validation, and test data. Training data are presented to the network when the training session needs to commence while adjusting the network with an error reference. The diabetes dataset consists of 768 patients, and 70% (approximately 537) instances are used for training purposes. Multiple pieces of training may generate different results each time regarding various conditions as well as data. On the other hand, validation data need to measure the network generalization. When it reaches its optimum level, the validation process stops. Approximately 115 instances are selected to perform the validation process to evaluate the training impact pragmatically. In so far as test data are concerned, there is no effect or change on training data due to testing data and after training process. It enhances the independent measurement of the desired performance of the underlying network. A set of 115 instances are chosen for testing purposes, and each algorithm's overall performance is observed and recorded accordingly.

3.5. Proposed System Model. The ABP-SCGNN method architecture is segregated into four distinct parts, as depicted in Figure 2. These include initialization of weight, forward and backward propagation of error, updating of weight, and bias. The hidden layer consists of a number of neurons, and every neuron has an activation function as $f(x) = \text{Sigmoid}(x)$. Activation function gets the sum of weighted input $(w_{11} * x_1 + w_{21} * x_2 + w_{31} * x_3 + \dots + w_{81} * x_8 + 1 * b)$ argument as follows:

TABLE 1: The input and output variables for the proposed system framework.

S/N	Input attributes	Description	Range values
1	Pregnancies	Number of times pregnant	0–17
2	Glucose	Plasma glucose concentration 2 hours in an oral glucose tolerance test	0–199
3	Blood pressure	Diastolic blood pressure (mm Hg)	0–122
4	Skin thickness	Triceps skinfold thickness (ram)	0–99
5	Insulin	2-hour serum insulin (mu U/ml)	0–846
6	BMI	Body mass index (weight in kg/(height in m) ²)	0–67.1
7	Diabetes pedigree function	Diabetes pedigree function	0.078–2.42
8	Age	Age (years)	21–81

Sr.	Input attributes	Output/responder variable	Range values
1	Outcome	Diabetes, yes or no	01

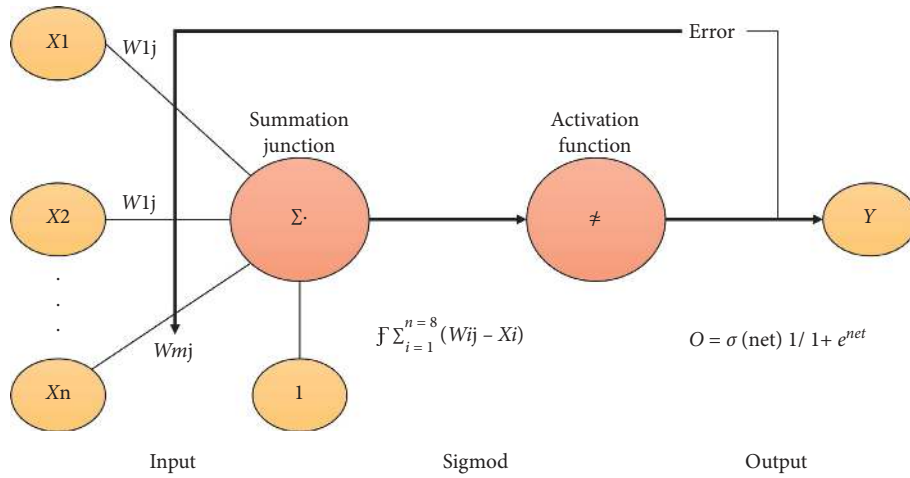


FIGURE 2: The architecture of the ABP-SCGNN model.

$$f \sum_{i=1}^{n=8} W_{ij} X_i + b. \quad (1)$$

Activation function was used to build a nonlinear transformation that is allocated to correspond nonlinear proposition or for assessing the complex functions. The inputs of the activation function were calculated by multiplying the weight by input and then added bias value. In the following section, the proposed neural network is explained how it performed its intended operations. To begin with, forward propagation is performed. First, backward propagation is made after comparing predicted Y output with actual output \hat{Y} during the calculation of the gradient error at the output layer.

A neural network is trained with diabetes datasets by using the following algorithms:

- (1) Multilayer perceptron (MLP)
- (2) Bayesian regularized
- (3) Scaled conjugate gradient

The proposed framework contains one input and one output layer with a single hidden layer containing eight input features (neurons) that produce one of the two outputs mentioned in the proposed research methodology. The

dataset comprises input and output layers (X_i, Y_i), where X_i is the input layer and Y_i is the predicted output. If n is the total size of the entire set, then

$$X_n = \{(x_i, y_i), \dots, (x_n, y_n)\}. \quad (2)$$

The parameters of feed-forward are denoted collectively as θ . Since training a neural network involves the gradient of the error function $E = (X, \theta)$ relevant to the weight w_{ij}^k (weight at node j in layer lk for node i), biases are b_i^k (bias at node i in layer lk), so as per learning rate, gradient descent updates the weights at each iteration as follows:

$$\theta_t = \theta_t - \alpha \frac{\partial E(X, \theta)}{\partial \theta_t}, \quad (3)$$

where θ_t represents ANN parameters at iteration t . The MSE in backpropagation is

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^N (\hat{y}_i - y_i), \quad (4)$$

where \hat{y}_i is the predicted output and y_i is the actual output of the input x_i . The derivative of $f(x)$ is represented as $f'(x)$, and the derivative of the sigmoid function is $\sigma'(x)$. $b(k/i)$ (bias in layer k th at node i th) is included into the weight as w_{0i}^k with an output $\sigma_0^{k-1} = 1$ at node 0 in layer $k-1$; hence,

$$w_{0i}^k = b_i^k, \quad (5)$$

$$\vec{W} = \begin{pmatrix} w_{00} & \cdots \\ w_{0m} & \cdots \\ w_{10} & \cdots \\ w_{1m} & \cdots \\ \vdots & \vdots \\ w_{n0} & \cdots \\ w_{nm} & \cdots \\ w_{n+1,0} & \cdots \\ \vdots & \vdots \end{pmatrix}. \quad (6)$$

Weight and bias values are initialized with random values as one-time initiation; an updated weight value is used in subsequent iterations. Equation (6) can be written as follows:

$$\vec{W} = \begin{pmatrix} \vec{W}_{IH} \\ \vdots \\ \vec{W}_{HO} \end{pmatrix}. \quad (7)$$

Equation (7) demonstrates that \vec{W}_{IH} contains the weight from the input layer to hidden layer where as \vec{W}_{HO} is the weight representing from hidden layer to output layer.

$$\vec{W}_{IH} = \begin{pmatrix} w_{00} & w_{01} & \cdots & w_{0m} \\ w_{10} & w_{11} & \cdots & w_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,0} & w_{n,1} & \cdots & w_{n,m} \\ w_{00} & w_{11} & \cdots & \cdot \end{pmatrix}. \quad (8)$$

Equation (8) contains the weights between the input and hidden layer. It contains $n \times m$ elements. $W_{a,b}$ is the weight between element.

$$\vec{W}_{HO} = \begin{pmatrix} w_{n+1,0} & w_{n+1,1} & \cdots & w_{n+1,m} \\ w_{n+2,0} & w_{n+2,1} & \cdots & w_{n+2,m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n+k,0} & w_{n+k,1} & \cdots & w_{n+k,m} \\ w_{00} & w_{11} & \cdots & \cdot \end{pmatrix}. \quad (9)$$

In our case, only one output neuron is used. Equation (9) can also be written as

$$\vec{W}_{HO} = (w_{n+1,0} \ w_{n+1,1} \ \cdots \ w_{n+1,m}). \quad (10)$$

Corresponding to the original formulation \mathcal{F}_{k-1} ,

$$a_i^k = b_i^k + \sum_{j=1}^{\mathcal{F}_{k-1}} w_{ji}^k o_j^{k-1} = \sum_{j=0}^{\mathcal{F}_{k-1}} w_{ji}^k o_j^{k-1}. \quad (11)$$

Then, the error can be calculated as

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^N (\hat{y}_i - y_i)^2. \quad (12)$$

After taking the derivative of Equation (12),

$$\frac{\partial E(X, \theta)}{\partial w_{ji}^k} = \frac{1}{N} \sum_{d=1}^N \frac{\partial}{\partial w_{ij}^k} \left(\frac{1}{2} (\hat{y}_i - y_i)^2 \right) = \frac{1}{N} \sum_{d=1}^N \frac{\partial E_d}{\partial w_{ij}^k}. \quad (13)$$

The backpropagation algorithm is concerned with one I/O pair, and all I/O pairs in X may be produced in merging each gradient. For derivation, the error function is

$$E = \frac{1}{2} (\hat{y} - y)^2. \quad (14)$$

3.5.1. Error Function Derivatives. After applying the chain rule,

$$\frac{\partial E}{\partial w_{ji}^k} = \frac{\partial E}{\partial a_j^k} \frac{\partial w_{ji}^k}{\partial w_{ij}^k}, \quad (15)$$

where a_j^k treats as activation of node j in layer k .

Here,

$$\delta_j^k \equiv \frac{\partial E}{\partial a_j^k}. \quad (16)$$

The second term from the equation for a_j^k is as follows:

$$\frac{\partial a_j^k}{\partial w_{ji}^k} = \frac{\partial}{\partial w_{ij}^k} \left(\sum_{l=0}^{\mathcal{F}_{k-1}} w_{lj}^k o_l^{k-1} \right) = o_i^{k-1}. \quad (17)$$

Error function E partial derivative with w_{ij}^k (weight) is as follows:

$$\frac{\partial E}{\partial a_j^k} = \delta_j^k o_i^{k-1}. \quad (18)$$

So, the partial derivative of weight is a product of the error term δ_j^k at node j in layer k and the output o_i^{k-1} of node i in layer $k-1$.

3.5.2. The Output Layer. Backpropagation characterizes the value ϕ_1^m , where m is the final layer. Four-layer neural network possesses $m = 3$ for the ending layer and $m = 2$ for the second to the last layer. Expressing E (error function) in terms of the value a_1^m (since ϕ_1^m is a partial derivative for ϕ_1^m) gives

$$E = \frac{1}{2} (\hat{y} - y)^2 = E \frac{1}{2} (g_o(a_1^m) - y)^2. \quad (19)$$

Then, applying the partial derivative,

$$\delta_i^m (g_o(a_1^m) - y) g_o'(a_1^m) = (\hat{y} - ty) g_o'(a_1^m). \quad (20)$$

The partial derivative of E and the error function w_{i1}^m is

$$\frac{\partial E}{\partial w_{i1}^m} = \delta_1^m o_i^{m-1} = (\hat{y} - y) g_o'(a_1^m) o_i^{m-1}. \quad (21)$$

3.5.3. The Hidden Layer. The hidden layers' error can be calculated as follows:

$$\delta_j^k = \frac{\partial E}{\partial a_j^k} = \sum_{l=1}^{N^{\mathcal{F}^{k+1}}} \frac{\partial E}{\partial a_j^{k+1}} \frac{\partial a_j^{k+1}}{\partial a_j^k}, \quad (22)$$

where l ranges from $rk+1$ to the number of nodes.

Error term φ_l^{k+1} gives the following equation:

$$\begin{aligned} \delta_j^k &= \sum_{l=1}^{\mathcal{F}^{k+1}} \delta_l^{k+1} \frac{\partial a_l^{k+1}}{\partial a_j^k}, \\ a_l^{k+1} &= \sum_{j=1}^{\mathcal{F}^k} w_{jl}^{k+1} g(a_j^k), \end{aligned} \quad (23)$$

remembering the definition of a_i^{k+1} where the $g(x)$ is the activation function.

$$\frac{\partial a_l^{k+1}}{\partial a_j^k} = w_{jl}^{k+1} g'(a_j^k). \quad (24)$$

As we know that

$$\delta_j^k = \sum_{l=1}^{\mathcal{F}^{k+1}} \delta_l^{k+1} w_{jl}^{k+1} g'(a_j^k) = g'(a_j^k) \sum_{l=1}^{\mathcal{F}^{k+1}} w_{jl}^{k+1} \delta_l^{k+1}, \quad (25)$$

the partial derivative of the Error function w_{ij}^k for $1 \leq k < m$ is

$$\frac{\partial E}{\partial w_{ij}^k} = \delta_i^k o_i^{k-1} = g'(a_j^k) o_i^{k-1} \sum_{l=1}^{\mathcal{F}^{k+1}} w_{jl}^{k+1} \delta_l^{k+1}, \quad (26)$$

$$\frac{\partial E_d}{\partial w_{ij}^k} = \delta_i^k o_i^{k-1}. \quad (27)$$

For the partial derivatives and for the final layer's error term,

$$\begin{aligned} \delta_1^m &= g_o'(a_1^m) (\hat{y}_d - y_d), \\ \delta_1^m &= g'(a_j^k) \sum_{l=1}^{\mathcal{F}^{k+1}} w_{jl}^{k+1} \delta_l^{k+1}. \end{aligned} \quad (28)$$

For the hidden layers' error term,

$$\frac{\partial E(X, \theta)}{\partial w_{ij}^k} = \frac{1}{N} \sum_{d=1}^N \frac{\partial}{\partial w_{ij}^k} \left(\frac{1}{N} (\hat{y}_d - y_d)^2 \right) = \frac{1}{N} \sum_{d=1}^N \frac{\partial E_d}{\partial w_{ij}^k}. \quad (29)$$

For combining the partial derivatives,

$$\Delta w_{ij}^k = a \frac{\partial E(X, \theta)}{\partial w_{ij}^k}. \quad (30)$$

For updating the weights,

$$w_{ij}^{k+1} = w_{ij}^k + \lambda \Delta w_{ij}^k. \quad (31)$$

3.6. Algorithm of ABP-SGCNN Model. Assume a as the learning rate and parameter initialization as w_{ij}^k , following steps are used to keep progress in the proposed algorithm.

3.6.1. Calculate the Forward Phase. For each pair of input and output (\vec{x}_d, y_d) , store the results (\hat{y}_d, a_j^k) and (o_j^k) for each node (j) in layer (k) by proceeding from layer zero, input layer, to layer (m), the output layer.

3.6.2. Calculate the Backward Phase. For each pair of input and output (\vec{x}_d, y_d) , store the results $(\partial E_d / \partial w_{ij}^k)$ for each weight (w_{ij}^k) connecting node (i) in layer ($k-1$) to node (j) in layer (k) by proceeding from layer (m), the output layer, to layer (1), the input layer.

Evaluate the error term for the final layer using equation (2).

Backpropagate the error terms for the hidden layers δ_l^k , working backward from the $k = m-1$, by repeatedly using equation (3).

Evaluate the partial derivatives of the individual error E_d regarding w_{ij}^k by using equation (1).

3.6.3. Combine the Individual Gradients. Combine the individual gradients for each input-output pair $(\partial E_d / \partial w_{ij}^k)$ to get the total gradient $(\partial E(X, \theta) / \partial w_{ij}^k)$, for the entire set of input-output pairs $X = \{(\vec{x}_1, y_1), \dots, (\vec{x}_N, y_N)\}$ by using equation (4) (a simple average of the individual gradients).

3.6.4. Update the Weights. According to the learning rate a and total gradient $(\partial E_d / \partial w_{ij}^k)$ and by using equation (5) (moving in the direction of the negative gradient), the weights are updated using equation (27).

Using equation (27), weights are updated accordingly.

4. Experimental and Analytical Verification

Probably, it is not likely to achieve the desired result during a single training iteration. However, sometimes, the model needs to train several times till it approaches near to the predicted outcome. After completing thousands of iterations, the measurement and evaluation results of predicting the diagnosis of diabetes are described in Table 2 for analysing the actual performance. Taking into account, optimizing the framework along with training the network for 5000 epochs and using 768 instances, the experiment demonstrates the potential benefit and worth of neural network algorithms. For all algorithms used in training, the learning rate was 0.25, and the momentum coefficient was 0.5. During the simulation, a number of neurons that were

TABLE 2: The performance analysis of the ABP-SCGNN model compared with standard ANNs models using an accuracy measure.

Algorithms	Training accuracy (%)			Validation accuracy (%)		
	5 neurons	20 neurons	40 neurons	5 neurons	20 neurons	40 neurons
Multilayer perceptron	91.6	93.6	94.9	90.8	91.9	93.2
Bayesian regularized	77.5	79.6	81	71	76.5	78.7
Proposed ABP-SCGNN	92.94	94.44	95.71	92.073	92.708	93.34

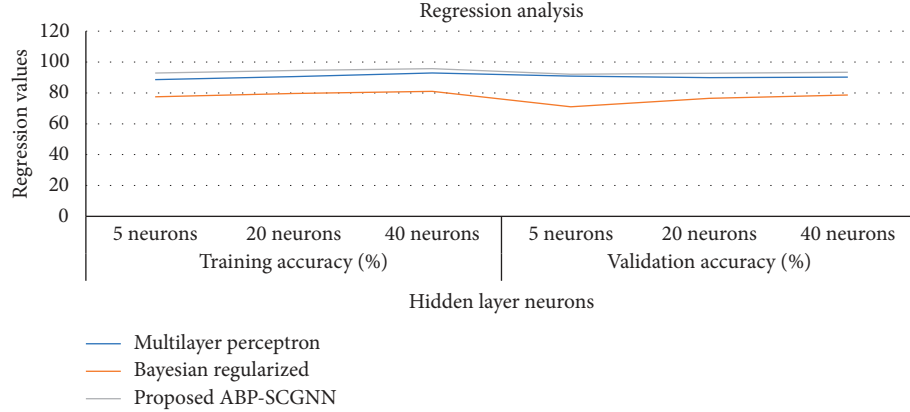


FIGURE 3: The regression accuracy results of the ABP-SCGNN model at different neurons of hidden layer.

taken in the hidden layer fall in the range 5, 10, 20, 30, 40, and 50 neurons. Results prove that the value of correlation coefficient particularly for proposed artificial back propagation scaled conjugate gradient neural network (ABP-SCGNN) is near to 1 that indicates pretty accuracy based on the regression accuracy plot among all other algorithms. Mean squared error (MSE) is calculated on all algorithms. The prediction accuracy is calculated with the ratio of the total number of predictions that are correct to the complete predictions. The proposed algorithm (ABP-SCGNN) produces more correct predictions with 93% prediction accuracy.

Figure 3 presents the MSE during the learning process for the hidden layer using different number of neurons in the range 5 to 50 which is computed against all algorithms. Table 3 reveals that the MLP algorithm contains 0.0026 using five neurons at the hidden layer; no doubt by increasing the number of neurons, i.e., 20 and 40, MSE was approaching low, but network complexity increased along with decreasing MSE rate. The Bayesian regularization algorithm has approximately high MSE, which is the same for a different number of neurons of the hidden layer. On the contrary, the proposed algorithm, ABP-SCGNN, attains relatively a lower MSE result when five neurons are used at the hidden layer, and this result is reduced more upon increasing the number of neurons, as shown in Figure 4. It confirms that using 20 neurons gets a lowest MSE and a high accuracy results. Moreover, the model achieves a regression accuracy value of 93%, as presented in Figure 3. Consequently, a trade-off between network complexity and performance of the model is approaching for 20 hidden neurons in both MSE and accuracy measures.

The regression accuracy plot in Figure 3 depicts a significant correlation of the proposed algorithm between

targets and predicted values that are identified by the dashed line that exhibits the highest accuracy. Three different numbers of neurons were used in the hidden layer, as mentioned in Table 2. Prediction accuracy of MLP remains the same for all neurons used. In the same case with the Bayesian regularization algorithm, its regression accuracy result ranges from 70 to 78 using the same number of hidden neurons taken for the other two algorithms.

Since forward propagation concerns the inference phase of a feed-forward neural network and the learning phase deals with the backpropagation neural network, the learning phase is relatively slower than the inference phase because gradient descent needs to repeat several times during the whole process. Using parallel running, the work performance of underlying algorithms can be efficient. The computational complexity of the feed-forward neural network is calculated by splitting the computation in the training and inference phase. As shown in Figure 5, the overall time complexity will increase with the number of epochs. However, the efficiency of algorithms remains stable.

The following equation calculates the factor of complexity (Fc) according to Figure 5.

$$\frac{2 * it}{2 * (j * i + j * it)}, \quad (32)$$

where i is the input layer neurons, j is hidden layer neurons, and it is the number of iterations.

Likewise, The MSE time complexity can be calculated for the same algorithms as follows:

$$2 * ((j * i + j * k) * it), \quad (33)$$

where Tc is time complexity, i is the input layer neurons, j is hidden layer neurons, and k is the number of iterations. Figure 5 demonstrates the time complexity based on the

TABLE 3: The performance analysis of the ABP-SCGNN model compared with standard ANNs models using an MSE measure.

Algorithms	Training MSE rate			Validation MSE Rate		
	5 neurons	20 neurons	40 neurons	5 neurons	20 neurons	40 neurons
Multilayer perceptron	$2.59E-03$	$1.99E-03$	$1.55E-03$	$2.61E-03$	$2.29E-03$	$2.13E-03$
Bayesian regularized	$4.17E-1$	$3.8E-1$	$3.54E-01$	$4.03E-01$	$3.96E-01$	$3.63E-01$
Proposed ABP-SCGNN	$2.6286-e3$	$2.3071E-3$	$1.6265E-3$	$2.381E-03$	$2.190E-03$	$2.054E-03$

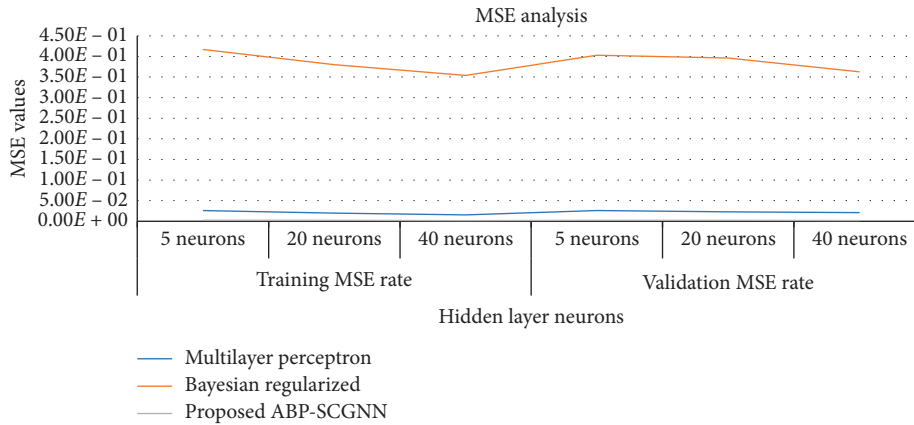


FIGURE 4: The MSE results of the ABP-SCGNN model at different neurons of hidden layer.

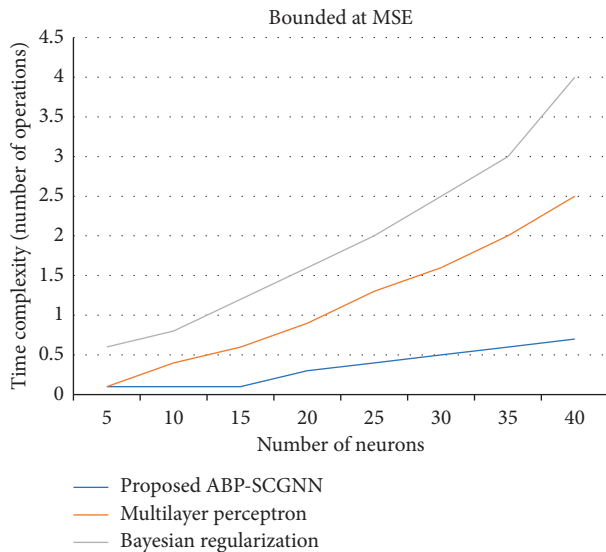


FIGURE 5: The time complexity of the ABP-SCGNN model at different neurons of hidden layer.

number of epochs by choosing three configurations of different neurons, such as 5, 20, and 40, as follows.

Indeed, with the increase in neurons, the success ratio increases to some extent. However, after examining 20 neurons, the success ratio remains the same in conjunction with increase in time complexity. Table 4 shows the comparison among various neural net algorithms with the proposed (ABP-SCGNN) algorithm that demonstrates its highest success ratio, such as 93%. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm contains an

TABLE 4: The accuracy results of the ABP-SCGNN model compared with some other models.

Algorithm	Accuracy (%)
BFGS [1]	88.8
Genetic algorithm [8]	87
GRNN [9]	80.21
ABP-SCGNN (proposed)	93

88.8% success ratio higher than ever used earlier in the underlying problem. On the other hand, the proposed algorithm shows the best result regarding success ratio and regression accuracy.

5. Conclusion

Using artificial backpropagation neural network (ABPNN), a pragmatic framework has been proposed to predict the diagnosis of diabetes. The presented results have been demonstrated in the paper that it can be viable to model an adaptive framework using APBNN. Since the applications of ANN have been organized differently from the classical approach, the features are provided as input variables rather than to provide them into the ANN.

The proposed ABP-SCGNN framework is effective and efficient, with a 93% success ratio when simulated with a test PIDD dataset [17]. To check the effective performance of ANNs and computational analysis, many performance indicators were calculated. Among these indicators, mean squared error (MSE) and regression accuracy analysis were examined anxiously. All neural networks were trained using 5 to 50 hidden layer neurons with the same

dataset. The best result was demonstrated at a hidden layer containing 20 neurons, specifically at the proposed ABP-SCGNN. The rest of the two algorithms remain with high MSE at the same 20 hidden layer neurons, whereas the presented ABP-GCGNN gives attractive results for the same dataset.

The experimental results demonstrated that ABP-SCGNN was the best algorithm that exhibited the highest prediction result among all algorithms. Since ANN possibly presents new approaches, strategies, and methodologies to remove uncertainties and reduce the potential instability of correlation, the intensity of network performance achieved using the proposed framework verified that ANN is functionally beneficial for successfully predicting the diagnose of diabetes with suitable disease input features and a corresponding set of instances.

Data Availability

The dataset used in this study is obtained from the National Institute of Diabetes and Digestive and Kidney Diseases, as cited in [17] (<http://archive.ics.uci.edu/ml>).

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

The authors are grateful to the Deanship of Scientific Research, King Saud University, for funding through Vice Deanship of Scientific Research Chairs.

References

- [1] M. A. Sapon, K. Ismail, and S. Zainudin, "Prediction of diabetes by using artificial neural network," in *Proceedings of the 2011 International Conference on Circuits, System and Simulation*, vol. 2829, Singapore, Singapore, May 2011.
- [2] A. Chavey, M.-D. Ah Kioon, D. Bailbé, J. Movassat, and B. Portha, "Maternal diabetes, programming of beta-cell disorders and intergenerational risk of type 2 diabetes," *Diabetes & Metabolism*, vol. 40, no. 5, pp. 323–330, 2014.
- [3] D. Manzella, R. Grella, A. M. Abbatecola, and G. Paolisso, "Repaglinide administration improves brachial reactivity in type 2 diabetic patients," *Diabetes Care*, vol. 28, no. 2, pp. 366–371, 2005.
- [4] E. O. Olaniyi and K. Adnan, "Onset diabetes diagnosis using artificial neural network," *International Journal of Scientific and Engineering*, vol. 5, no. 10, pp. 754–759, 2014.
- [5] P. Rahimloo and A. Jafarian, "Prediction of diabetes by using artificial neural network, logistic regression statistical model and combination of them," *Bulletin de la Société Royale des Sciences de Liège*, vol. 85, pp. 1148–1164, 2016.
- [6] S. S. Abbasi, B. Dashtbozorg, B. M. T. H. Romeny, and F. Fleuret, "Exploratory study on direct prediction of diabetes using deep residual networks," *European Congress on Computational Methods in Applied Sciences and Engineering*, vol. 1, pp. 797–802, 2017.
- [7] O. Karan, C. Bayraktar, H. Gümüşkaya, and B. Karlık, "Diagnosing diabetes using neural networks on small mobile devices," *Expert Systems with Applications*, vol. 39, no. 1, pp. 54–60, 2012.
- [8] D. K. Choubey, S. Paul, S. Kumar, and S. Kumar, "Classification of Pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection," in *Proceedings of the Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing System (ICCCS 2016)*, pp. 451–455, Gurgaon, India, September 2016.
- [9] K. Kayaer and T. Yıldırım, "Medical diagnosis on Pima Indian diabetes using general regression neural networks," in *Proceedings of the International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP)*, pp. 181–184, Istanbul, Turkey, June 2003.
- [10] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, p. 261, Rockville Pike, MD, USA, April 1988.
- [11] J. C. Florez, K. A. Jablonski, N. Bayley et al., "TCF7L2 Polymorphisms and progression to diabetes in the diabetes prevention program," *New England Journal of Medicine*, vol. 355, no. 3, pp. 241–250, 2006.
- [12] K. Iqbal, M. A. Khan, S. Abbas, Z. Hasan, and A. Fatima, "Intelligent Transportation System (ITS) for smart-cities using mamdani fuzzy inference system," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 2, pp. 94–105, 2018.
- [13] M. A. Khan, M. Umair, and M. A. S. Choudhry, "GA based adaptive receiver for MC-CDMA system," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, pp. 2267–2277, 2015.
- [14] M. Umair, M. A. Khan, and M. A. Choudry, "GA backing to STBC based MC-CDMA systems," in *Proceedings of the Intelligent Systems Modelling & Simulation (ISMS), 2013 4th International Conference*, pp. 503–506, Bangkok, Thailand, January 2013.
- [15] A. Atta, S. Abbas, M. A. Khan, G. Ahmed, and U. Farooq, "An adaptive approach: smart traffic congestion control system," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, pp. 1012–1019, 2018.
- [16] M. A. Ullah, M. A. Khan, S. Abbas, A. Athar, S. S. Raza, and G. Ahmad, "Blind channel and data estimation using fuzzy logic-empowered opposite learning-based mutant particle swarm optimization," *Computational Intelligence and Neuroscience*, vol. 13, pp. 1–12, 2018.
- [17] D. Dua and C. Graff, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, CA, USA, 2019.
- [18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [19] N. M. Nawwi, W. H. Atomi, and M. Z. Rehman, "The effect of data pre-processing on optimized training of artificial neural networks," *Procedia Technology*, vol. 11, pp. 32–39, 2013.
- [20] C. Boos, C. Scolaro, M. Pereira, and F. Azevedo, "Analysis of pre-processing methods for artificial neural network pattern recognition of EEG signals," in *Proceedings of the World Congress on Medical Physics and Biomedical Engineering*, pp. 558–561, Beijing, China, May 2012.