

Intelligence Clinical Decision Support System for Diabetes Management

TASNEEM ABU-KABEER,
Department of Computer Science,
King Abdullah II for Information Technology,
The University of Jordan, Amman 11942, JORDAN,

MOHAMMAD ALSHRAIDEH,
Department of Computer Science,
King Abdullah II for Information Technology,
The University of Jordan, Amman 11942, JORDAN,

FERIAL HAYAJNEH
Faculty of Nursing,
Department of clinical nursing.
The University of Jordan, Amman 11942, JORDAN,

Abstract:- Diabetes is the most common endocrine disease in all populations and all age groups. The diabetes patient should use correct therapy to live with this disease; there are several of important things to record about the patient and disease that help the doctors to make an optimal decision about the patient treatment. To improve the ability of the physicians, several tools have been proposed by the researchers for developing effective Clinical Decision Support System (CDSS), one of these tools is Artificial Neural Networks(ANN) that are computer paradigms that belong to the computational intelligence family. In this paper, a multilayer perceptron (MLP) feed-forward neural is used to develop a CDSS to determine the regimen type of diabetes management. The input layer of the system includes 25 input variables; the output layer contains one neuron that will produce a number that represents the treatment regimen. A Resilient backpropagation (Rprop) algorithm is used to train the system. In particular, a 10-fold cross-validation scheme was used, an 88.5% classification accuracy from the experiments made on data taken from 228 patient medical records suffering from diabetes (type II).

Key-Words: - Clinical Decision Support System, Artificial Neural Network, Diabetes Management.

Received: December 18, 2019. Revised: April 29, 2020. Accepted: May 9, 2020.

Published: May 18, 2020.

1. Introduction

A disease treatment, based upon various symptoms, tests and previous state performed upon the patient. When several symptoms and tests are involved, the ultimate treatment decision may be difficult to make, even for a medical expert. Consequently, over the past few decades, this has given rise to computerized medical applications. These computerized tools are called clinical decision support systems (CDSSs) which are types of computer programs that can be developed through several kinds of machine learning (ML) techniques and used to assist clinicians at the point of care [24]. Machine Learning (ML) studies computer algorithms for learning to complete a task, make accurate predictions and behave intelligently. This learning is done based on observations or data and direct experience or instructions. In general, ML is about learning to do better in the future based on what was experienced in the past. In some tasks, we could specify input/output pairs but not a concise relationship so we need machines to approximate the relationship in these I/O pairs. The amount of knowledge available about certain tasks might be too large for explicit encoding by humans.

In medical diagnosis, some of the area for ambiguity in the inputs is the history (patient's description of the disease condition), physical examinations (especially in cases of uncooperative patients) and laboratory tests (faulty methods or equipment). Moreover, in the treatment, there are chances of drug reactions and specific allergies [2, 20, 21, 22]. All these factors may not be taken into account by busy clinics in which hundreds of patients attending it daily. Therefore, computers can help the clinician to reach an accurate diagnosis faster because they make fewer errors than humans in making analyses of complex data [19]. In the literature many researchers have given their definitions of Clinical Decision Support System(CDSS); in [4] defined a CDSS be software that designed to be direct aid to clinical decision-making, in which the characteristics of an individual patient are matched to a computerized clinical knowledge base and patient-specific assessments or recommendations are then presented to the clinician or the patient for a decision. Miller and Geissbuhler [13] defined a CDSS providing diagnostic decision support as a "computer-based algorithm that assists a clinician with one or more component steps of the diagnostic process".

The remainder of this paper is organized as following: section 2 will describe Artificial Neural Networks ANN from its basic model through its architecture, algorithms and design to end with using ANN in medical problems; section 3 illustrates Intelligence Clinical Decision Support System for diabetes management; section 4 the experiments evaluation, section 5 System Results; and section 6 conclusion and future work.

2. Artificial Neural Networks Overview

ANN is one of the techniques that belong to the CI family (usually called Neural Network (NN) that was introduced in the 1940s. and further developed in the 1980s, as an attempt to emulate the features of biological neural networks to address a range of difficult information processing, analysis and modelling problems[5]. It is a powerful data modelling tool that can capture and represent complex input/output relationships.

2.1 The Basic Model

The basic model of ANNs is an input layer consisting of nodes that simply accept the input values. The outputs of neurons in a layer are inputs to neurons in the next layer. The last layer is called the output layer. Layers between the input and output layers are known as hidden layers. The output of a particular artificial neuron is determined by the type of activation function, known sometimes transfer function. The activation function is used to limit the amplitude of the output of a neuron and is commonly known as a squashing function[6]. The most common type of activation functions is: Linear activation function (LF): The outputs of a linear activation function, also known as a linear combiner, are determined as follows: $a = n$ where n is the sum of the weighted inputs. Hyperbolic tangent sigmoid activation function (tanh): The hyperbolic tangent sigmoid activation function (tanh) has the following formula:

$$a = \frac{2}{1 + e^{-2n}} - 1$$

2.2 ANNs Architectures

ANN architecture can be divided into two main categories based on their processing structure (known as topology): feedback and feed-forward neural networks [12]. In feedback neural networks signals can travel in both directions, the network has loops in its topology. In Feed-forward neural networks signals are allowed to travel one way only; from input to output with no loops. These architectures can be further subdivided into two main classes based on the number of layers of processing nodes used in the model; single-layer perceptron and multi-layer perceptron [6]. In Single-Layer Perceptrons or Perceptron, there is a layer of input nodes and a layer of output nodes, which are the only processing nodes in the model [8, 9]. The perceptron rule has two main stages; Feed-forward stage Perceptron generates initially randomly weights, then the weights (w_j) and input values x_j are summed to compose activation function (v). That produces output. Error backward stage; In this stage, the output error is calculated as the following: $\varepsilon_i = t_i - y_i$, where t_i is the target output and y_i is the output, if the error is equal to zero then no weight update is performed, else the weight must be updated. In Multi-Layer Perceptron (MLP), or multilayer feed-forward networks, there are one or more hidden layers between the input and the output nodes [10].

2.3 Linearly and Non-Linearly Separable Problems

A problem is said to be linearly separable if a straight line can be drawn to separate the input vectors into two categories. If more than one line is needed to separate the input vectors, then we say that this problem is non-linearly separable. The single-layer perceptron is powerful to be used with linearly separable problems, while MLP is used with non-linearly separable problems.

2.4 The Back Propagation and Resilient Back Propagation Algorithm

The backpropagation (BP) neural network algorithm is a multi-layer feed-forward network trained according to the error backpropagation algorithm and is one of

the most widely applied neural network models. BP networks can be used to learn and store a lot of mapping relations of input-output models, with no need to disclose in advance the mathematical equation that describes these mapping relations. Its learning rule is to adopt the steepest descent method in which the backpropagation is used to regulate the weight value and threshold value of the network to achieve the minimum error sum of the square.

Back Propagation algorithm will experience a local minimum that occurs because the algorithm always changes the weights based on the value of the partial derivative. So to solve this problem; the Resilient Back Propagation (Rprop) was devised [17], where it eliminates the harmful influence of the magnitude of the partial derivative. As a consequence, only the sign of the derivative is considered to indicate the amount of the weight update. To achieve this,

an update-value Δ_{ij} is introduced for each weight w_{ij} , which solely determines the size of the weight-update [23]. This adaptive update-value evolves during the learning process, according to the following learning-rule:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ \times \Delta_{ij}^{(t-1)} & , \text{ if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \times \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \eta^- \times \Delta_{ij}^{(t-1)} & , \text{ if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \times \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)} & , \text{ else} \end{cases}$$

where $0 < \eta^- < 1 < \eta^+$

The adaptation-rule works as follows: Every time the partial derivative of the corresponding weight w_{ij} changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-value Δ_{ij} is decreased by the factor η^- (which was found by experiments that the best value for it is equal to 0.5). If the derivative retains its sign, the update-value is slightly increased by the factor

η^+ (which was found by experiments that the best value for it is equal to 1.2) to accelerate convergence in shallow regions.

2.5 Designing issues of Artificial Neural Network

Selection of an ANN Topology

The decision as to what ANN topology is to be used and which training algorithm to apply to this structure is not easy. Selection of a suitable combination of these two design components is often done experimentally and is problem-dependent [16]. Other issues closely related to the network topology choice include the number of hidden layers and hidden units to be used.

Pre-Processing and Post-Processing

The input data should be in such a form that a neural network can process it, which means numeric values. The choice of the pre-processing method will be one of the most significant factors in determining the performance of the ANN [14]. One commonly used pre-processing method is data standardization; standardizing a vector most often means subtracting a measure of location and dividing by a measure of scale, as follows:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^N a_i$$

while the standard deviation shows how much variation or "dispersion" there is from the "average" (mean). A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data are spread out over a large range of values [7]. The standard deviation S_N is defined as follows:

$$S_N = \sqrt{\frac{1}{n} \sum_{i=1}^N (x_i - \bar{x})^2}$$

To obtain a dataset with zero mean and unit standard deviation, we use the following:

X_i = Value of the raw input variable X for i th training case.

S_i = Standardized value corresponding to X_i .

N = number of training cases.

Standardize X to mean 0 and standard deviation 1 by:

$$S_i = (X_i - \text{mean}(X)) / \text{Standard Deviation}(X).$$

Generalization Performance of an ANN

One of the major advantages of neural networks is their ability to generalize; this means that a trained network could classify data that it has never seen before from the same class as the learning data. To reach the best generalization, the dataset should be split into three parts:

- *The training set* is used to train a neural network. The error of this dataset is minimized during training.
- *The validation set* is used to determine the performance of a neural network on patterns that are not trained during learning.
- *A test set* for finally checking the overall performance of a neural network.

The learning should be stopped at the minimum of the validation set error. At this point, the network generalizes best. When learning is not stopped, overtraining occurs and the performance of the network on the whole data decreases.

Early Stopping

This technique is used for improving the generalization of a neural network during training and preventing it from overfitting (Haykin 2001). An overfitted network is illustrated in Fig. 1. It achieves that by splitting the available data into three subsets which are used separately for training, validation and testing. These sets are used in the following manner: at each iteration through the training set, the ANN runs a test over the validation set; this is done iteratively while the performance of the network continues to improve. When the performance does not show further improvement with more iteration, the training is completed and the ANN runs a test over the test set. The result gives the final performance of the classifier [18].

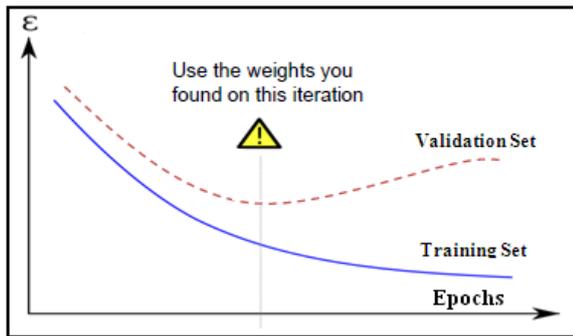


Fig. 1: Overfitting in Neural Networks, Training error is a showdown and validation error is shown up[18].

2.6 Using Neural Networks in Medical Problems

Previous studies sought ways of capitalizing the use of neural networks in several diseases. It proved that Information Technology is helpful in breast cancer where it can improve the results of breast cancer screenings by evaluating the performance ANN to predict an outcome (cancer/not cancer). Paulin and Santhakumaran [15] presented a study on the classification of breast cancer using feed-forward ANNs; the performance of the network is evaluated using Wisconsin Breast Cancer Data set (WBCD) where the accuracy of 99.28% using the Levenberg-Marquardt (LM) training algorithm was obtained.

In hepatitis, several studies proved that neural networks are useful for the differentiation of two forms of chronic active hepatitis. Bascil and Temurtas [3] presented a study for hepatitis diagnosis using a multilayer neural network as a machine learning technique and LM algorithm as a training algorithm. The architecture was on the data set that was taken from the University of California Irvine (UCI) machine learning database. The study obtained a classification accuracy of 91.87% via ten-fold cross-validation.

In heart disease, several diseases share the same symptoms which increase the ability of the wrong diagnosis so that ML can help in making correct diagnoses. Yan et al. (2006) presented a multilayer perceptron based decision support system to support the diagnosis of heart diseases; the system was applied to five different heart diseases. The results showed that the proposed system could achieve very high diagnosis accuracy (90%),

proving its usefulness in support of clinical diagnosis decision of heart diseases.

AlTimemy and Al Naima [1] in their study presented the prediction of kidney dysfunction using different neural network approaches. PNN, SOM and MLP neural network well trained with backpropagation algorithm; these networks are used as a classifier to predict whether the kidney is normal or it will have dysfunction applied on a data set collected from one of the private clinical laboratories in Baghdad. In this study, the PNN gave a more accurate prediction of kidney dysfunction with accuracy 99% while the accuracy of SOM was 98% and the accuracy for MLP was 95%.

Applications of neural networks in the diabetes field

Many researchers study diabetes since it is a chronic disease, for that it includes multiple fields that possible to be computerized. Polat and Gunes,(2007) have reported 89.47% classification accuracy using 10 fold cross-validation on Pima Indian diabetes disease diagnosis using principal component analysis (PCA) and adaptive neuro-fuzzy inference system (ANFIS).

Temurtas et al, (2009), realized a comparative Pima diabetes disease diagnosis For this purpose, a multilayer neural network structure which was trained by Levenberg–Marquardt (LM) algorithm and a probabilistic neural network structure was used. The study achieved 79.62 % accuracy using 10 fold cross-validation approach and it achieved 82, 37% accuracy using the conventional validation approach (one training and one test).

Other researchers give attention to the psychological state of diabetic patients Manda et al, [11] purpose feed-forward backpropagated neural network with graphical user interface based icons. They collected data from 241 diabetic patients with 33 variables and found that four variables are very important for predicting the well being (age, gender, weight, fasting plasma glucose). Predefined values were considered as a set of inputs and found the output for measuring the wellbeing are energy, positive wellbeing, depression and anxiety.

Since there are multiple types of insulin treatment as shows below:

1. Short-acting insulin mixed with intermediate-acting insulin, given twice daily before meals
2. Short-acting insulin, given three times daily AND intermediate-acting insulin, given at bedtime.
3. Intermediate-acting insulin, given once daily.

Gogou et al (2001) proposed a system which will further facilitate decision making for diabetic patient management by insulin administration based on a neural network (NN) approach. The factors participating in the decision making were included in diabetes type, patient age, current treatment, glucose profile, physical activity, food intake, and desirable blood glucose control. The resulting system was trained with 100 cases and tested on 100 patient cases. The system proved to apply to this particular problem, classifying correctly 92% of the testing case.

3. Intelligence Clinical Decision Support System for Diabetes Management

ANN performs quite well with complex tasks such as classification problems using abstract data. ANN has been applied to many areas of the medical field for analysis of various diseases and conditions. Therapeutic Decision Support System for Type 2 Diabetes using ANN is a fairly new area, and not much analysis has been performed with ANN. Section 3.1 will discuss how data was collected and prepared to build diabetes management CDSS. Section 3.2 will discuss the system architecture for the diabetes management CDSS

3.1 Data Preparation

When a patient report to a physician, a large number of possibly relevant inputs must be considered during the session. A physician reaches an accurate diagnosis or treatment decision based upon observations, the patient's answers to questions, and physical examinations and lab results.

In this paper, the diabetes datasets used for testing and training the system consists of a total of 228 cases, in which each medication type has a total of 76 cases, gathered from Jordan University hospital, located in Amman, Jordan. We have trained and tested the system using these medical records. Moreover, 25 variables are essential to give stable treatment. These variables can be divided into four categories:

Basic information of a patient (6 factors in total).

The patient's previous status and test (4 factors in total).

Diabetes complication (7 factors in total)

Lab results and physical examination (8 factors in total).

In Fig.2 in the Appendix I, the datasheet that was used in gathering the patient's variables is illustrated; as can be seen from this Figure that there are some attributes are assigned a yes or no value to indicate the presence or absence of an attribute, while other attributes are assigned one value such as gender and previous treatment. The value of the gender has to be either female or male and the value of the previous treatment has to be single medication, dual medication or insulin therapy. The remaining attributes of age and all basic information, diastolic blood pressure, systolic blood pressure, and all blood lab results were assigned the actual values the patient has for these attributes.

1.2 System Architecture

The following subsections explain how the input variables that were used in the training and testing process were encoded for building the diabetes treatment CDSS, and how the number of hidden layers and hidden neurons in each layer was chosen.

1.2.1 Input Variables Classes Encoding Scheme

All neural networks take numeric inputs, therefore the 25 variables were encoded into numeric values, as well as the predicted treatment type using the following encoding schemes:

- i. Numerical variables such as age, diastolic blood pressure, systolic blood pressure, and all blood lab results were reserved the same with no changes.
- ii. Variables with two independent attributes, such as the gender, For instance, 1 represents the female and 0 male. Some attributes from the diabetes complication are encoded with binary values (0, 1). 1 is adopted for the mentioned categories for the presence of the attribute in that category, and 0 for the absence of it.
- iii. Previous treatment that represented by three independent attributes is encoded using the ternary values (1,2,3); with 1 representing that the patient uses single medication, 2 representing that the patient using the dual medication, and 3 for the patient treated by insulin.

After encoding all variables, the training dataset was standardized to have a zero mean and a unit standard deviation as explained in subsection 2.5, and based on the information that was collected from the training dataset during the standardization; the validation and test datasets were standardized also to have a zero mean and a unit standard deviation. As we mentioned before, the standardization method must be applied for both input and output variables.

1.2.2 Number of Hidden Layers and Hidden neuron

Optimizing the number of hidden layer and neurons in each hidden layer for a feed-forward neural network remains one of the unsolved tasks in this research area. Thus several researchers have proposed some general rules for determining an optimal number of hidden neurons for any application. Kavzoglu as cited in Yan et al., (2006) has shown that in most situations, there is no way to determine the best number of hidden layers without training several networks and estimating the generalization error of each, because the best number of hidden layers and units depends in a complex way on the number of input and output units, the number of training cases and the complexity of the classification to be learned [19]. Based on this, we used an iterative process to determine the best number of hidden layers and neurons in each hidden layer. In the iterative process, a ten-fold cross-validation technique (more details will be provided in Section 4) is used to choose the best neural model that will provide the highest classification accuracy. The whole process works as follows:

Step1. Starting from one hidden layer; the network architecture has been tested with one hidden neuron up to 25 neurons.

Step2. Add another layer; the number of neurons in this layer will be half the number of neurons in the previous layer.

Step3. Repeat step two until the number of the neurons in the layer is one.

4. Experiments evaluation

In this Section, Subsection 4.1 introduces the evaluation method that was used to assess the generalization of the ANN models; 2 presents multiple ANN architectures that were built to choose the best ANN model for building the CDSS for diabetes treatment. Finally, Subsection 4.3 introduces the detailed results for the best ANN model we obtained in Subsection 4.2.

4.1 Cross-Validation and Classification

Accuracy

Cross-validation is a technique used for estimating the performance of a predictive model. Also, it can be referred to as an estimate of accuracy, sometimes referred to as rotation estimation. It is determined by the overall number of correct classifications divided by the total number of instances in the dataset (Yan et al., 2006). One round of cross-validation involves partitioning a sample of data into subsets, performing the analysis on one subset (called the training set), and testing the analysis on the other subset (called the testing set). A third subset called the validation set is used to avoid overfitting. Moreover, to reduce variability and to avoid bad splits that may produce overfitting, multiple rounds of cross-validation are performed using different partitions, and the tested results are averaged over the rounds.

In this case study, ten-fold cross-validation was used to assess the generalization of the network models that were built. A percentage of 80%, 10% and 10% data split of the complete data samples were applied to represent the training, validation and test subsets respectively. This provides for each fold a total of 186, 21 and 21 sample cases for the training validation and test subsets respectively. Each medication type in the training set will be represented by 62 samples of the total samples of the training set. Also, each medication type in the validation set will be represented with 7 samples and 7 samples for the test set.

The classification accuracy is the percentage of the cases that were correctly classified. ANN classification accuracy is the most important issue since this performance metric gave some indication of the ANN performance. Accuracy (A) of an individual testing dataset depends on the number of samples correctly classified and is evaluated by the formula:

$$A = (T/N) * 100\%$$

Where T is the number of cases correctly classified and N is the total number of cases.

4.2 Models' Building

To investigate the best system architecture for building the CDSS for diabetes treatment; multiple system architectures were built using the iterative process that was explained in subsection 3.2.2. The different architectures were compared according to the classification accuracy which each of them has obtained. The model with the highest classification accuracy (less generalization error) was chosen as the final model for building the CDSS.

The different ANN architectures have been built and tested using MATLAB version 2010a, which was run under windows Microsoft Windows XP Professional version 2002 OS with a Genuine intel® CPU, and 2.49 GB of RAM.

For all models the following parameters were used:

- A Feedforward Backpropagation neural network was used for building all models.
- The number of neurons in the input layer is 25 (representing the number of features for each sample).
- The number of neurons in the output layer is 1 (representing one class of the treatment the neural will generate).
- The training algorithm that was used for training the models is Resilient Backpropagation.
- The activation function that was used for all the hidden layers is HTSF and the activation function that was used for the output layer is LF.
- The following values were used 0.08, 0.0000001, and 6 for the delta initial, the performance goal error and the number of validation checks to avoid the overfitting of the network respectively.

4.2.1 Models' Building using One Hidden Layer

Since the input layer contains 25 neurons, we built network architecture with one hidden layer and with one up to 25 neurons.

Table 1 illustrates the results of 25 different architectures. The best architecture that produced the highest classification accuracy is (25-11-1); this architecture represents 25 input variables, one hidden layer of 11

neurons and one output layer, the classification accuracy for this architecture is 88.5714%. All the remaining architectures produced low classification accuracy varying in an interval [73.33-87.14].

Table 1: Classification Accuracy with one hidden layer

Network Architecture	Classification Accuracy
25-1-1	80.4762
25-2-1	82.8571
25-3-1	73.3333
25-4-1	86.6667
25-5-1	86.6667
25-6-1	84.2857
25-7-1	81.9048
25-8-1	81.4286
25-9-1	86.1905
25-10-1	84.7619
25-11-1	88.5714
25-12-1	85.7143
25-13-1	85.2381
25-14-1	85.2381
25-15-1	86.6667
25-16-1	79.5238
25-17-1	84.2857
25-18-1	78.5714
25-19-1	83.8095

25-20-1	86.6667
25-21-1	87.1429
25-22-1	76.6667
25-23-1	77.1429
25-24-1	85.7143
25-25-1	80.4762

4.2.2 Models' Building using Two Hidden Layers

For the first hidden layer, we obtained 25 architectures, so by applying step 2 from the iterative process, the number of neurons in the second hidden layer will be as follows:

- $N/2$ if N is even.
- $\lceil N/2 \rceil$ and $\lfloor N/2 \rfloor$ if N is odd.

(Assuming that the number of neurons in the first hidden layer is N)

Consequently, 37 different architectures will be tested when the number of hidden layers is two. As noticed from Table 2 in the Appendix I, the best classification accuracy is 86.6667% is obtained when the network architecture is 25-9-5-1 and when the network architecture is 25-13-7-1. The results for the other architectures vary in an interval [69.04-85.2381].

4.2.3 Models' Building using three Hidden Layers

We obtained 37 different architectures when the number of hidden layers was two so that when a third hidden layer is added, the number of possible architectures will be increased and it will be hard to manage. To decrease the number of possible architectures we need to modify the step 1 in the iterative process (mentioned before) to be as follow:

Step1. We start testing the network architecture with only one hidden layer; the following equation will be applied to find the number of neurons in the first hidden layer:

$$nf = (ni) / 2$$

Where n_f is the number of neurons in the first hidden layer, n_i is the number of neurons in the input layer.

Since the input layer contains 25 neurons, the number of neurons in the first hidden layer will be 12.5 after applying the equation in step 1 from the new iterative process. Because the result is not a fixed number, applying the ceil operation will produce 13, applying the floor operation will produce 12. Therefore the network can be trained with two different numbers of neurons (in the first hidden layer), and to make the testing more precise another number is used, which is 11. Thus, we are concerned with five different architectures when the number of hidden layers was two, 25-11-5-1, 25-11-6-1, 25-12-6-1, 25-13-6-1, 25-13-7-1

After applying step 2 and W.R.T. even and the odd number of neurons in the second hidden layer, 7 different architectures were tested. Table 3 illustrates the classification accuracy when using three hidden layers; the highest accuracy was achieved with the network architecture of the form 25-13-6-3-1 is 83.3333% classification accuracy. The other architectures achieved results varying in an interval [76.6667-82.8571].

Table 2: Classification Accuracy with three hidden layers

Network Architecture	Classification Accuracy
25-11-5-2-1	80.4762
25-11-5-3-1	81.4286
25-11-6-3-1	80
25-12-6-3-1	82.8571
25-13-6-3-1	83.3333
25-13-7-3-1	76.6667
25-13-7-4-1	82.8571

4.2.4 Models' Building using Four Hidden Layers

Building the network with four hidden layers we need to use the architectures that were obtained using three hidden layers. Table 4 illustrates the results obtained using four hidden layers. The highest accuracy was achieved with the network architecture of the form 25-11-5-3-2-1 is 85.2381% classification accuracy. The other architectures achieved results varying in an interval [67.619-84.7619].

Table 4: Classification Accuracy with four hidden layers

Network Architecture	Classification Accuracy
25-11-5-2-1-1	78.0952
25-11-5-3-1-1	67.619
25-11-5-3-2-1	85.2381
25-11-6-3-1-1	70
25-11-6-3-2-1	84.7619
25-12-6-3-1-1	82.8571
25-12-6-3-2-1	70.9524
25-13-6-3-1-1	76.6667
25-13-6-3-2-1	82.381
25-13-7-3-1-1	82.381
25-13-7-3-2-1	82.381
25-13-7-4-2-1	80.4762

4.2.5 Summary of the results

To summarize, we noticed from the results of all models that:

- The best classification accuracy was 88.57% using one hidden layer and 11 hidden neurons.
- We conclude that experiments are the only way to determine the best network

architecture that can be used to solve a specific problem.

5. System Results

The effectiveness of the chosen neural network architecture was evaluated on the diabetes test set for each fold independently.

After that, the average accuracy was computed; the ANN model with the highest classification accuracy is (25-11-1). A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such a system is commonly evaluated using the data in the matrix. In a confusion matrix, columns represent the predicted data, while rows represent the actual data.

Table 5 shows the result of the first run; Table 6 in the Appendix I, shows the experimental results of the 10 test sets

Table 5: Confusion Matrix of the First Run

Regimen Type	One oral medication	Dual oral medication	Insulin medication	Classification Accuracy
One oral medication	7	0 0		100.00
Dual oral medication	0	6 1		85.71
Insulin medication	0	0 7		100.00
Total Accuracy			95.24	

and comparing the results obtained with the one obtained using the ANN may be useful.

The accuracy for each class can be calculated as follow in Table 7.

Table 7: Proportion of correctly classified Treatment

Treatment type	Accuracy %
One oral medication	66/70 = 94.29%
Dual oral medication	55/70 = 78.57%
Insulin medication	65/70 = 96.86%
Total Accuracy	186/210 = 88.57%

6. Conclusion and Future Work

In this paper, we have developed a clinical decision support system for therapy regimen selection of therapy for diabetes type two patients. Three treatment regimens have been considered (single oral medication “metformin”, dual oral medication “Metformin Plus Sulfonylurea”, insulin medication) An accuracy of 94.29%, 78.57% and 96.86% were obtained for the one oral medication, dual oral medication and insulin medication respectively

In the future, this work can be extended by using other types of ANNs such as GRNN, PNN and RBF to compare the results obtained from them with the one obtained using the MLP. Also, we can apply one of the feature selection algorithms on the diabetes disease dataset to choose the medical variables that play a major role in the treatment process which may give more accurate results. Moreover, applying other ML methods such as Artificial Immune Systems, Neuro-Fuzzy and Decision Trees

References

- [1] Timmy.A. and Al Naima.F.Comparison of Different Neural Network Approaches for the Prediction of Kidney Dysfunction, International Journal of Biological and Life Sciences, 6 (2), 2010, pp.84-90.
- [2] Qatawneh, Z., M. Alshraideh, N. Almasri, L. Tahat and A. Awidi, Clinical decision support system for venous thromboembolism risk classification. Applied Comput. Information, 2017, DOI: 10.1016/j.aci.2017.09.003
- [3] Bascil.S., and Temurtas.F. A Study on Hepatitis Disease Diagnosis Using Multilayer Neural Network with Levenberg Marquardt Training Algorithm, Journal of Medical Systems, 33 (1), 2009, pp. 1-4.
- [4] Farhan, S., Alshraideh, M., Mahafza, T. A Medical Decision Support System for ENT Disease Diagnosis using Artificial Neural Networks, 4(2), 2015,pp.45-54.
- [5] Fausett, L Laurene, Fundamentals of Neural Networks, Architectures, Algorithms and applications, (1st ed), New Jersey: Pearson Prentice Hall Education, Inc. 2001.
- [6] Hawamdeh, Ziad M.; Alshraideh, Mohammad A.; Al-Ajlouni, Jihad M.; Salah, Imad K.; Holm, Margo B., Otom, Ali H. Development of a decision support system to predict physicians’ rehabilitation protocols for patients with knee osteoarthritis, International Journal of Rehabilitation Research. 35(3), 2012, pp.214-219.
- [7] Ghahramani, Saeed, Fundamentals of Probability, (2nd ed). New Jersey: Prentice-Hall, 2000.
- [8] Kong, G., Xu, D. and Yang, J. Clinical Decision Support Systems: A Review on Knowledge Representation and Inference Under Uncertainties, International Journal of Computational Intelligence Systems, 1(2), 2008, pp. 159-167.

- [9] Krose, B and Smagt, P. An Introduction To Neural Networks, (8th ed), Amsterdam: The University of Amsterdam, 19960
- [10] Machado, Lucila Ohno. Medical Application of Artificial Neural Networks: Connectionist Models of Survival, Master's thesis, Department of Computer Science, University of Stanford, USA, 1996.
- [11] Manda R., Rao.N, Sridhar.GR., Madhu.K., Rao.A., A clinical decision support system using multi-layer perceptron neural network to predict the quality of life in diabetes, Diabetes and Metabolic Syndrome: Clinical Research and Reviews;2010, pp. 57–59
- [12] Mehrotra, K, Chilukuri, K, Mohan and Ranka, S. Elements of Artificial Neural Networks,(1st ed), New York: Addison-Wesley Publishing Company, Inc, 1996.
- [13] Miller R. and Geissbuhler A. Diagnostic Decision Support Systems In Berner E. (Ed), Clinical Decision Support Systems, (2nd ed), New- York: Springer, 2007, pp.99-125.
- [14] Musen Mark A. Patient Record. In: J.van Bommel and Mark A. Musen (Ed), Handbook of medical informatics, Houten: Springer, 1997, pp. 99-115.
- [15] Paulin, F. and Santhakumaran, A. Classification of Breast Cancer by Comparing Backpropagation Training Algorithms, International Journal on Computer Science and Engineering, 3(1), 2011, pp.327-332.
- [16] Rasheed, Zeesham, Adaptive Fuzzy Logic Based Framework for Handling Imprecision and Uncertainty in Pattern Classification of Bioinformatics Datasets, Master's thesis, Department of Computer Science, King Fahd University of Petroleum and Minerals, Saudi Arabia, 2009.
- [17] Riedmiller, M. and Braun, H A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, Neural Networks, IEEE International Conference, San Francisco, CA, USA, 1993,586-591.
- [18] Veropoulos, Konstantinos, Machine Learning Approaches to Medical decision Making, Doctoral thesis, Department of Engineering Mathematics, University of Bristol, United Kingdom, 2001.
- [19] Xu, S. and Chen, L. A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN's and Its Application in Data Mining, In the 5th International Conference on Information Technology and Applications, Cairns, Australia, 2008,pp.683-686
- [20] Zaitoun, I., Almasri, N., Taha, L., Alshraideh, M. Satellite Image Cloud Classification Using Fuzzy Logic. International Journal of Imaging & Robotics,19(1), 2019.
- [21] AL-Dreabi E., Otoom M., Salah B., Hawamdeh Z., Alshraideh M., Automated Detection of Breast Cancer Using Artificial Neural Networks and Fuzzy Logic, International Journal of Sciences: Basic and Applied Research (IJSBAR) 35 (3), 2017.
- [22] Abu-Arida A., Salah I., Alshraideh M., Hayajneh F., Multi-Standard Accreditation as Expert System Tool in Jordan University Hospitals, International Journal Of Computers And Communications, 6(4), 2012, pp. 202- 2017.
- [23] Al Switi M., Alshraideh B., Alshraideh A., Massad A., Alshraideh M., Treatment of Diabetes Type II Using Genetic Algorithm, International Journal of Online and Biomedical Engineering (iJOE), 15(11), 2019, pp. 53-68.
- [24] Salah B., Alshraideh M., Beidas R., Hayajneh F., Skin cancer recognition by using a neuro-fuzzy system. Cancer Inform, 2011, pp.1–11.

Appendix I :

Patients Data Set

System Output: Treatment Type:

- II. Single medication
- III. Dual medication
- IV. Insulin /insulin + Oral medication

System Input:

i. Patient's Basic Information:

- Age: _____
 - BMI: _____
 - Height: _____
-
-

- *Gender: Male / Female*
 - *Duration:_____*
 - *Weight:_____*
-
-

ii. Diabetes Complication

- *Hypertension: Yes/No*
 - *Dyslipidemia: Yes/No*
 - *Diabetic Neuropathy: Yes/No*
 - *Hypoglycemic Attack: Yes/No*
 - *Diabetic Nephropathy: Yes/No*
 - *Diabetic Retinopathy: Yes/No*
 - *Cardiac Heart Disease: Yes/No*
-
-

iii. The patient's previous status and test :

- *Previous HBA1C:_____*
 - *Previous FBS:_____*
 - *Previous weight:_____*
 - *Previous Treatment Type:_____*
-
-

iv. Lab results and physical examination:

- *Current HBA1C:_____*
 - *Current FBS:_____*
 - *Blood Pressure:_____ /_____*
 - *Cholesterol: _____*
 - *Triglyceride:_____*
 - *Low-Density lipoprotein:_____*
 - *High-Density lipoprotein:_____*
-
-

Figure 2: Patient's Datasheet.

Table 3: Classification Accuracy with two hidden layer

Network Architecture	Classification Accuracy
25-1-1-1	77.619
25-2-1-1	79.5238
25-3-1-1	82.8571
25-3-2-1	73.8095
25-4-2-1	76.6667
25-5-2-1	83.8095
25-5-3-1	79.5238
25-6-3-1	83.3333
25-7-3-1	78.5714
25-7-4-1	82.8571
25-8-4-1	80.9524
25-9-4-1	81.4286
25-9-5-1	86.6667
25-10-5-1	80.9524
25-11-5-1	85.2381
25-11-6-1	74.2857
25-12-6-1	79.5238
25-13-6-1	78.0952
25-13-7-1	86.6667
25-14-7-1	74.7619

Table 6: The confusion matrix
 Diabetes: 10- fold cross-

25-15-7-1	80.4762
25-15-8-1	78.0952
25-16-8-1	82.381
25-17-8-1	81.9048
25-17-9-1	78.0952
25-18-9-1	77.1429
25-19-9-1	70.9524
25-19-10-1	79.5238
25-20-10-1	82.381
25-21-10-1	81.9048
25-21-11-1	78.0952
25-22-11-1	84.2857
25-23-11-1	82.381
25-23-12-1	83.8095
25-24-12-1	83.3333
25-25-12-1	80
25-25-13-1	69.0476

of the therapeutic result for
 validation

Regimen Type	One oral medication	Dual oral medication	Insulin medication	Row Sum
One oral medication	66	4	0	70
Dual oral medication	9	55	6	70
Insulin medication	0	5	65	70
Column Sum	75	64	71	210