Jewel Journal of Scientific Research (JJSR) 6(1&2): 111–120, 2021
©Federal University of Kashere-Nigeria (Print ISSN: 2384 – 6267, Online ISSN 2756-651X)
jjsr.fukashere.edu.ng

Multi-Layer Perceptron for Classification of Diabetic Patients: A Case of Federal Medical Centre, Yola

***[1]Ahmed, H., [2]Muhammad, M. B. and [1]Yakubu, N.**

[1]Department of Statistics and Operations Research, Modibbo Adama University, Yola, Adamawa State.

[2] Department of Mathematics and Computer Science, Federal University of Kashere, Gombe, Gombe State, Nigeria.

***Corresponding Author**: hassanahmed.official@gmail.com; +2348032417537

Abstract

Multi-Layer Perceptron (MLP) are used to handle regression analysis when the dependent response variable is categorical. Therefore, this study assesses the performance of MLP in terms of classification of object/observations into identified component/groups. A data set consists of 553 cases of diabetes mellitus were collected at Federal Medical Center, Yola. The variables measured: Age(years), Mass of a patient (kg/meters), glucose level (plasma glucose concentration, a 2-hour in an oral glucose tolerance test), pressure (Diastolic blood pressure mmHg), insulin (2-hour serum insulin mu U/ml) and class variable (0 or 1) treating 0 as false or negative and 1 treated as true or positive test for diabetes. The method used in the study is the Multi-Layer perceptron, a type of Artificial Neural Network, confusion matrix, classification, network algorithm and SPSS version 21 for Windows 10.1. The result of the study showed that MLP classifies diabetic patients with 88.6% accuracy while it classifies Non-diabetic patients with 93.2% classification accuracy. Overall, MLP classifies better with 91% accuracy. This study complements other literatures where MLP, a type Artificial neural network classifies and predicts better than other Non-neural network classifiers. Remarkably, the results indicate the effectiveness of applying MLP as significant method to predict and classify diabetic and non-diabetic patients' to help in understanding variables that may influence diabetic treatment.

Keyword: Multi-layer perceptron, Neural networks, classification, diabetes mellitus, medical data.

Received: 16th Aug, 2021 *Accepted:* 11th Oct., 2021 *Published Online:* 25th Dec., 2021

Introduction

Many statistical techniques are available for handling various problems. Some of these techniques come as models such as linear, exponential and quadratic models. These models have become integral components concerned with describing the relationship between a response variable and one or more explanatory variables (Hosmer and Leweshow, 2000). If there is a reason to believe that a linear relationship exists between a variable of interest (response variable) and other variables (predicator variables) in a study, the ordinary linear model is one technique that is often used for predicting outcomes (Agresti, 2002). This technique is mostly adopted due to its flexibility for analyzing the relationship between multiple independent variables and a single dependent variable. Much of its flexible is due to the way in which all sorts of independent variables can be accommodated (Joaquim and Marques, 2007).

It is meaningful to address how the analyst can deal with data representing multiple independent variables and a categorical dependent variable, how independent variables can be used to contribute to the discovery of differences in the categories. The assignment of observations or objects into predefined homogenous groups is a problem of major practical and research interest. For example, we may use quantitative information in predicting who will or will not graduate from a college. This would be an example of simple binary classification problems, where the categorical dependent variable can only assume two distinct values. In other cases, there are multiple categories or classes for the categorical dependent variable. For example, when we are ill, we want a doctor to diagnose our disease from the symptoms of the illness, the outcome maybe more than two.

All the above are classification problems where we attempt to predict values of a categorical dependent variable from one or more continuous and/or categorical predictor variables. In statistics, it is the process of allocating an observation p in one of several predefined groups or categories and an ideal classification method which distinguishes different classes from each other. The basic objective is to build a discriminant function that takes the information to summarize the p variables on an indicator that yields the optimal discrimination between the classes − the goal of classification in this case − also known as supervised pattern recognition (Wehrens,2010). In order to derive the decision rule that yields the optimal discrimination between the classes, one assumes that a training set of pre-classified cases −the data sample− is available, and can be used to determine the model applicable to new cases. The decision rule can be derived in a model-based approach, whenever a joint distribution of the random variables can be assumed, or in a model-free approach (Joaquim and Marques, 2007).

There are numerous algorithms for predicting continuous or categorical variables from a set of continuous predictors and/or categorical factor effects (Lewicki and Hill, 2006). For example, in GLM (General Linear Models) and GRM (General Regression Models), we can specify a linear combination design of continuous predictors and categorical factor effects to predict a continuous dependent variable. In GDA (General Discriminant Function Analysis), we can specify such designs for predicting categorical variables to solve classification problems.

A neural-network is a classification algorithm in the field of artificial intelligence. It is a very powerful tool with the capability of pattern recognition. Artificial Neural Networks (ANNs) were designed to model the functioning of human brain. Linear classifiers separate objects by the value of a linear combination of their features. The feature of an object is represented by a vector. There is another vector to be trained with known observations. This is called weight vector. There are several algorithms in this category such as Support Vector Machines (SVM, Multi-layer Perceptron (MLP) and the radial Basis Function (RBF).

The healthcare industry generates and stores a massive amount of data that can be used to forecast and analyse the overall healthcare ratio. It is possible to extract hidden and usable information from datasets known as Knowledge Discovery in Database (KDD) and Computer-based information system (CBIS) utilizing data mining (DM) approaches (Sohail, 2017). The healthcare field is notorious for its ontological challenges and a mix of medical data standards and varied data excellence (Witten, 2011; Bhattacharyya, 2006; Thomas, 2005; Karegar, 2008). The modern clinical practice is undergoing a significance shift not just in terms of diagnosis and treatment, but also in terms of understanding health and illness concepts, moving away from disease-oriented problem resolution, toward a patient-centered approach, with computer-aided knowledge finding techniques playing a significant role (Chandamona, 2016).

Hypertension, diabetes, and coronary artery disease are currently the most common chronic health disorders in Sub-Saharan Africa (Haibinf, 2004). Infectious diseases such as HIV, tuberculosis (TB), and malaria are the leading causes of death in Sub-Saharan Africa (SSA); however, with increased international attention to these issues, treatment options are increasing and mortality rates are down (Haibinf, 2004); Joint United Nations program on HIV/AIDS and WHO. Although, curing infectious diseases has resulted in greater life expectancy and an increase in the prevalence of noncommunicable diseases (Goodwin, 1997). Diabetes is a chronic illness that has become a worldwide epidemic. Traditional tribal groups' in developing countries are embracing a modern lifestyle while suffering chronic health problems associated with developed countries (Illayaraja, 2013). In SSA, the direct and indirect disease load surpasses the healthcare system's financial and human resources (Prather, 1997).

The objective of this work is to evaluate the implementation and performance of classification techniques, a multi-layer perceptron in order to predict the presence of diabetes in a collected data from Federal Medical Center, Yola, Adamawa State, Nigeria. This paper describes how these techniques have been applied to the data and presents a comparison analysis. The study used confusion matrix, classification, SPSS 21 for windows 10.1 The results are reported and discussed according to this technique and future work.

Materials and Methods

Between 1st August, 2016 to 31st October, 2017, a total of five hundred and fifty-three (553) women were tested for diabetes at FMC, Yola. Three hundred and six were diabetic while two hundred and forty-seven were non-diabetic. The data collected was from records of patients at FMC, Yola.

Observation with missing data were dropped from the analysis. The final dataset consists of 553 subjects, described by several clinical characteristics.

The classification task consists of predicting whether a patient would test positive for diabetes. The class labels of the data are 1 for diabetes and 0 otherwise. There are 8 predictor variables for 553 patients.

The data set have the following numeric attributes and they are:

1. “glucose”: Plasma glucose concentration 2 hours in an oral glucose tolerance test.
2. “pressure”: Diastolic blood pressure (mm Hg).
3. “insulin”: 2-Hour serum insulin (mu U/ml).
4. “mass”: Body mass index (weight in kg/(height in meters).
5. “age”: Age in years.
6. Class variable (0 or 1). The Class variable (6) is treated as 0 (false), 1 (true – tested positive for diabetes).

The Multi-layer perceptron

We used a common feedforward backpropagation multilayer perceptron (MLP) simulator developed in SPSS software package. The prediction method is based on the nonlinear weighted combination of input units (i.e. predictive variables) to predict one or more output units (i.e. outcome variable). The learning process is iterative and essentially consists in adjusting the weights to decrease the output error. The network was specified with one input layer (representing the five predictive variables), one hidden layer (including five hidden units) and one output layer (with one output unit representing a binary diabetic event). Several sensitivity analyses were performed to test how the prediction results could be influenced by the variations of learning parameters and to elicit the most optimized network. These parameters refer to the architecture of the network (number of hidden units), the method of internal validation (number of iterations and data-splitting processes), the options of data pre-treatment (i.e. normalization of inputs), the activation function for hidden units, and the "Score Threshold" used by the system to classify a case from its predicted probability. The multilayer perceptron is the most known and most frequently used type of neural network. On most occasions, the signals are transmitted within the network in one direction: from input to output. There is no loop, the output of each neuron does not affect the neuron itself. This architecture is called feedforward Layers which are not directly connected to the environment are called hidden. In the reference material, there is a controversy regarding the first layer (the input layer) being considered as a standalone (itself a) layer in the network, since its only function is to transmit the input signals to the upper strata, without any processing on the inputs. In what follows, we will count only the layers consisting of stand-alone neurons, but we will mention that the inputs are grouped in the input layer. There are also feed-back networks, which can transmit impulses in both directions, due to reaction connections in the network. These types of networks are very powerful and can be extremely complicated. They are dynamic, changing their condition all the time, until the network reaches an equilibrium state, and the search for a new balance occurs with each input change. Introduction of several layers was determined by the need to increase the complexity of decision regions. As shown in the previous paragraph, a perceptron with a single layer and one input generates decision regions under the form of semi planes. By adding another layer, each neuron acts as a standard perceptron for the outputs of the neurons in the anterior layer, thus the output of the network can estimate convex decision regions, resulting from the intersection of the semi planes generated by the neurons. In turn, a three-layer perceptron can generate arbitrary decision. Regarding the activation function of neurons, it was found that multilayer networks do not provide an increase in computing power compared to networks with a single layer, if the activation functions are linear, because a linear function of linear functions is also a linear function. The power of the multilayer perceptron comes precisely from non-linear activation functions. Almost any non-linear function can be used for this purpose, except for polynomial functions. Currently, the functions most commonly used today are the single-pole (or logistic) sigmoid, shown in Figure 1:

sed today are the single-pole (or logistic) sigmoid, shown in Figure 1:

$$f(s) = \frac{1}{1+e^{-s}} . \qquad (1)$$

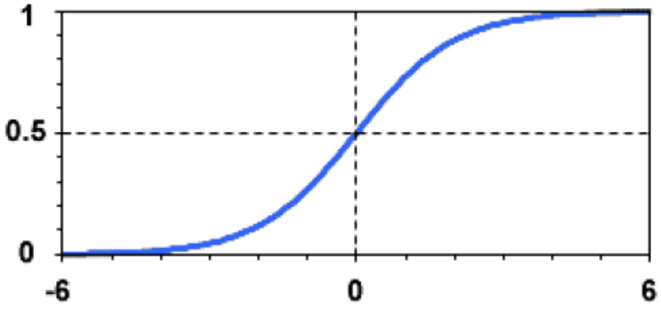

Figure 1: Sigmoid function

And the bipolar sigmoid (the hyperbolic tangent) function, shown in Figure 1 for a=2:

$$f(s) = \frac{1-e^{-a\cdot s}}{1+e^{-a\cdot s}} . \qquad (2)$$

It may be noted that the sigmoid functions act approximately linear for small absolute values of the argument and are saturated, somewhat taking over the role of threshold for high absolute values of the argument. It has been shown (Cybenko, 1999) that a network (possibly infinite) with one hidden layer is able to approximate any continuous function. This justifies the property of the multilayer perceptron to act as a universal approximator. Also, by applying the Stone-Weierstrass theorem in the neural network, it was demonstrated that they can calculate certain polynomial expressions: if there are two networks that calculate exactly two functions f1, namely f2, then there is a larger network that calculates exactly a polynomial expression of f1 and f2. Multi Perceptron is the best known and most used type of neural networks are trained units of the type. Each of these units forms a weighted sum of its inputs to which are added a constant. This amount is then passed through a nonlinear function which is often called the activation function. Most units are interconnected in a manner "feed forward" ie interconnections which form a loop. Learning networks is typically achieved through a supervised manner. It can be assumed to be available a learning environment that contains both the learning models and models of desired output corresponding to input (this is known as "target models"). As we will see, learning is typically based on the minimization of measurement errors between network outputs and desired outputs. This implies a back propagation through a network similar to that which is learned. For this reason, algorithm learning is called back-propagation. The method was first proposed (Bryson, 1969), but at that time it was virtually ignored, because it supposed volume calculations too large for that time. It was then rediscovered by Silver, 1990, but only in the mid-'80s was launched by Williams (Principe, et. al, 2000) as a generally accepted tool for training of the multilayer perceptron. The idea is to find the minimum error function e(w) in relation to the connections weights. The algorithm for a multilayer perceptron with a hidden layer is the following (Negnevitsky, 2002):

The algorithm

Step 1: Initializing. All network weights and thresholds are initialized with random values, distributed evenly in a small range. If these values are 0, the gradients which will be calculated during the trial will be also 0 (if there is no direct link between input and output) and the network will not learn. More training attempts are indicated, with different initial weights, to find the best value for the cost function (minimum error). Conversely, if initial values are large, they tend to saturate these units. In this case, derived sigmoid function is very small. It acts as a multiplier factor during the learning process and thus the saturated units will be nearly blocked, which makes learning very slow.

Step 2: A new era of training. An era means presenting all the examples in the training set. In most cases, training the network involves more training epochs. To maintain mathematical rigor, the weights will be adjusted only after all the test vectors will be applied to the network. Therefore, the gradients of the weights must be memorized and adjusted after each model in the training set, and the end of an epoch of training, the weights will be changed only one time (there is an „on-line" variant, more simply, in which the weights are updated directly, in this case, the order in which the vectors of the network are presented might matter. All the gradients of the weights and the current error are initialized with 0 (Δwij = 0 and E = 0).

Step 3: The forward propagation of the signal

An example from the training set is applied to the to the inputs.

The outputs of the neurons from the hidden layer are calculated:

$$y_j(p) = f\left(\sum_{i=1}^{n} x_i(p) \cdot w_{ij} - \theta_j\right), \qquad (3)$$

where n is the number of inputs for the neuron j from the hidden layer, and f is the sigmoid activation function.

The real outputs of the network are calculated:

$$y_k(p) = f\left(\sum_{i=1}^{m} x_{jk}(p) \cdot w_{jk}(p) - \theta_k\right), \qquad (4)$$

where m is the number of inputs for the neuron k from the output layer.

The error per epoch is updated:

$$E = E + \frac{(e_k(p))^2}{2}. \qquad (5)$$

Step 4: The backward propagation of the errors and the adjustments of the weights.

The gradients of the errors for the neurons in the output layer are calculated:

$$\delta_k(p) = f' \cdot e_k(p), \qquad (6)$$

where f' is the derived function for the activation, and the error. If we use the single-pole sigmoid (equation 1, its derived is:

$$f'(x) = \frac{e^{-x}}{\left(1 + e^{-x}\right)^2} = f(x) \cdot (1 - f(x)). \qquad (7)$$

If we use the bipolar sigmoid (equation 2, its derived is:

$$f'(x) = \frac{2a \cdot e^{-a \cdot x}}{\left(1 + e^{-a \cdot x}\right)^2} = \frac{a}{2} \cdot (1 - f(x)) \cdot (1 + f(x)) . \qquad (8)$$

Further, let’s suppose that the function utilized is the single-pole sigmoid. Then the equation (6) becomes:

$$\delta_k(p) = y_k(p) \cdot (1 - y_k(p)) \cdot e_k(p) . \qquad (9)$$

The gradients for the weights between the hidden layer and the output layer are updated:

$$\Delta w_{jk}(p) = \Delta w_{jk}(p) + y_j(p) \cdot \delta_k(p) . \qquad (10)$$

The gradients of the errors for the neurons in the hidden layer are calculated:

$$\delta_j(p) = y_j(p) \cdot \left(1 - y_j(p)\right) \cdot \sum_{k=1}^{l} \delta_k(p) \cdot w_{jk}(p) , \qquad (11)$$

where l is the number of outputs for the network. 4.4 The gradients of the weights between the input
layer and the hidden layer are updated:

$$\Delta w_{ij}(p) = \Delta w_{ij}(p) + x_i(p) \cdot \delta_j(p) . \qquad (12)$$

Step 5: A new iteration.
If there are still test vectors in the current training epoch, pass to step 3. If not, the weights all the connections will be updated based on the gradients of the weights:

$$w_{ij} = w_{ij} + \eta \cdot \Delta w_{ij} , \qquad (13)$$

where η is the learning rate. If an epoch is completed, we test if it fulfils the criterion

The algorithm involved in MLP are as follows

1. Start with an initial network of k hidden units. The default is k = min (g (R, P), 20, h (R, P)), where,

g (R, P) =

$$\begin{cases} \frac{4.5}{P+R} & R < 5, P \geq 8 \\ 0.5 + 0.5(P + R) & \text{otherwise} \end{cases}$$

and h(R,P)=[M−R/P+R+1]. If k < k_{min}, set K = k_{min}. Else if K > k_{max}, Set k = k_{max}.

2. If K > k_{min}, Set DOWN = TRUE. Else if training error ratio > 0.01, DOWN = FALSE. Else stop and report the initial network.
3. If DOWN=TRUE, remove the weakest hidden unit (see below); k=k−1. Else add a hidden unit; k=k+1.

for termination (E<Emax or a maximum number of training epochs has been reached). If not, we pass to step 2. If yes, the algorithm ends.

So the SPSS algorithm is:

Input Layer: $J_0 =$ P units or variable; $a_{0:1}, \ldots, a_{0:5}$;
with $a_{0:j} = x_j$,
and $J_i =$ Number of units in layer i, $a_{i:j}$ unit i of layer j
i^{th}hidden layer: J_i units, $i = 1 \ldots 5$, $a_{i:1}, \ldots, a_{i:j}$ with $a_{i:k} = \gamma_i(C_{i:k})$
and $C_{i:k} = \sum_{j=0}^{J_{i-1}} w_{i:j,k} a_{i-1:j}$ where $a_{i-1:0} =$
1. $\gamma_i(C) =$ activation function for layer i
Output layer: $J_I =$ R units, $a_{i:1}, \ldots, a_{I:j}$ with $a_{i:k} = \gamma_i(C_{i:k})$
and $C_{i:k} = \sum_{j=0}^{J_{i-1}} w_{I:j,k} a_{I-1:j}$ where $a_{i-1:0} = 1$.

The activation function of the hidden layer is the hyperbolic tangent given as

$$\gamma(C) = \tanh(c) = \frac{e^c - e^{-c}}{e^c + e^{-c}}$$

The activation function of the output layer is the sigmoid function given as

$$\gamma(C) = \frac{\exp(C_k)}{\sum \exp(C_j)}$$

4. Using the previously fit weights as initial weights for the old weights and random weights for the new weights, train the old and new weights for the network once through the alternated simulated annealing and training procedure (steps 3 to 5) until the stopping conditions are met.
5. If the error on test data has dropped:

If DOWN=FALSE, if k< k_{max} and the training error has dropped but the error ratio is still above 0.01, return to step 3. Else if k> k_{min}, return to step 3. Else, stop and report the network with the minimum test error.

Else if DOWN=TRUE, if |k−k0|>1, stop and report the network with the minimum test error. Else if training error ratio for k=k0 is bigger than 0.01, set DOWN=FALSE, k=k0 return to step 3. Else stop and report the initial network.

Else stop and report the network with the minimum test error.

If more than one network attains the minimum test error, choose the one with fewest hidden units.

If the resulting network from this procedure has training error ratio (training error divided by error from the model using average of an output variable to predict that variable) bigger than 0.1, repeat the architecture selection with different initial weights until either the error ratio is <=0.1 or the procedure is repeated 5 times, then pick the one with smallest test error.

Using this network with its weights as initial values, retrain the network on the entire training set.

Results

Multi-layer Perceptron (MLP) was applied. In MLP, the Input layer has 5 factors with 315 units excluding the bias unit as seen from Table 1. The hidden layer has 2 units excluding the bias unit. The Hyperbolic tangent was the activation function in the hidden layer. The output layer has 1 dependent variable which is 'class', with 2 units. Sofmax was the activation function. Cross-entropy was the error function used.

368 cases were used in the training sample. the network weights that corresponded to the lowest mean squared error on the validation set were used for evaluation on the test data. The test data has 115 cases and 61 hold-out cases. 9 cases had factors that do not occur in the training sample, as a result they were excluded from the analysis. For class 0 in the training sample, the network has 97.7 % correct classification and 99.5% correct classification for class 1. The testing sample has 84.1% correct classification and 84.5 % correct classification for classes 0 and 1 respectively.

Table 1: Multi-Layer Perceptron Network Information

Network Information				
Input Layer	Factors	1	Glucose	
		2	Pressure	
		3	Insulin	
		4	Age	
		5	Weight	
	Number of Units[a]			315
Hidden Layer(s)	Number of Hidden Layers			4
	Number of Units in Hidden Layer 1[a]			1
	Activation Function		Hyperbolic tangent	
Output Layer	Dependent Variables	1	Class	
	Number of Units			2
	Activation Function		Softmax	
	Error Function		Cross-entropy	

Table 2: MLP CLASSIFICATION

			Predicted Group Membership		
	Actual Group	No. of cases	Diabetic(1)	Non-diabetic(0)	Percent Correct
Training	Diabetic(1)	192	191	1	88.60%
	Non-diabetic(0)	176	4	172	93.20%
	Overall				91.0%
Testing	Diabetic(1)	71	60	11	93.0%
	Non-diabetic(0)	42	7	37	93.0%
	Overall				93.0%

Table 3: Comparing proposed model with other models

Authors	Best Performing Classifier	Database	Records	Performance
Zou *et al.* (2018)	RF	Hospital, Luzhou, China	220,680	80.84%
Maniruzzaman *et al.* (2017)	Gaussian RBF	PIDD		82%
Maniruzzaman *et al.* (2017)	RF	PIDD		92.26%
Ahuja *et al.* (2019)	MLP	PIDD	768	78.7%
Sisodia and Sisodia (2018)	NB			76.30%
Yu *et al.* (2010)	SVM	1999–2004 US NHANES	6214	83.50%
Semerdjian and Frank (2017)	GB GB	1999–2004 US NHANES	5515	
Mohapatra *et al.* (2019)	MLP			77.50%
Pei *et al.* (2019)	DT			94.20%
Maniruzzaman *et al.* (2020)	RF	2009-2012 US NHANES	6561	94.25%
Proposed Study	RBF	Hospital, Yola, Nigeria	553	94.30%

Discussion

The study was carried out to see the classification power of the MLP model. 553 records of data were collected on diabetic patients who were tested. The studied variables comprise of glucose level of each patient, diastolic pressure, insulin level, weight of each patient and their ages. The aim was to examine which of the two techniques classifies better. First, at the implementation stage, we chose to evaluate the method at its best performance, i.e. after optimization of the modeling specifications. This required to understand the meaning of each learning parameter and to test its influence on final results. In the analysis, (70%) of the data was used to train the network and 30% were used for testing the trained network. Remarkably, MLP predicts Diabetic patients with 93% in this study.

The study proposed a machine learning model based on MLP model which classified patients into two groups: diabetes and non-diabetic. Numerous articles in the literature are devoted to identifying high-risk factors for diabetes as well as sophisticated classification of the condition. Zou *et al.* (2018) used dataset derived from physical examination at a hospital in Luzhou, China. The dataset consisted of 220, 680 patients and had 14 attributes. 69% of the patients were diabetic, whereas 31% were not. The study used three classifiers to classify diabetic patients, reporting the RF-based classifier with the highest classification accuracy of 80.84%. In Maniruzzaman *et al.* (2017) study, classified diabetic patients using difference classification method, Gaussian process classification (GPC) was one of them. The study demonstrated that using a GP-based classifier with a radial basis kernel (RBF) resulted in the highest classification accuracy of roughly 82%. Additional, Maniruzzaman *et al.* (2018) used eleven classifiers to classify diabetic patients. The result revealed that the RF-based feature selection technique with the RF-based classifier resulted in the best classification accuracy of 92.26%.

Ahuja *et al.* (2019) used PIDD dataser with 768 observations and ten attributes. The result demonstrated that the MLP had the maximum classification accuracy of 78.70%. Moreover, Sisodia, D., & Sisodia, D. S. (2018) applied SVM, NB, and DT classifiers and found that the NB classifier had the best accuracy of 76.30%. Yu et al. (2010) developed an SVM model to categorise diabetes patients using data from the 1999–2004 US NHANES. The results revealed that the SVMs with RBF kernels performed the best, with an accuracy of 83.50%. Similarly, Semerdjian and Frank (2017) analysed 5515 total samples from the 1999–2004 NHANES dataset. They determined the most significant risk factors using RF with the Gradient boosting (GB) classifier performing the best. Furthermore, Mohapatra et al. (2019) employed MLP and discovered that it provided 77.50% classification accuracy and Pei *et al.* (2019) applied DT and reported a classification accuracy of 94.20 percent. Recently, Maniruzzaman *et al.* (2020) applied RF-based classifiers and achieves the highest of 94.25% (see Table 1). However, in this work, we used the RBF classifier and achieved the highest classification accuracy of 94.3%. Interestingly, this study improved on and supported the findings by Mohapatra *et al.* (2019) who demonstrated that using MLP yield 77.50%.

Conclusion

Diabetes is one of the most common problems affecting people recently. It is a set of metabolic disorders characterised by elevated blood sugar levels. This study used feed forward MLP to classify. Our results indicated that our model achieved 93% classification accuracy. Additionally, a comparison analysis was performed with existing studies and our model slightly outperformed other classifiers. It would be intriguing to see other types of medical data classified similarly in the future, thereby establishing a cost-effective and time-saving solution for both diabetic patients and physicians.

References

Ahuja, R., Vivek, V., Chandna, M., Virmani, S. and Banga, A. (2019). Comparative study of various machine learning algorithms for prediction of insomnia. In Advanced classification techniques for healthcare analysis (pp. 234-257). IGI Global.

Agresti Alan. (2002). Categorical Data Analysis. 2nd Edition. University of Florida. New York: Wiley

Bryson, A.E. and Ho, Y.C. (1969). Applied Optimal Control, Blaisdell, New York.

Chao, S. and Wong, F. (2009). An incremental decision tree learning methodology regarding attributes in medical data mining. Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding.

Cybenko, G. (1969). Approximation by superpositions of a sigmoidal function, Math. Control, Signal Syst. 2, pp.303-314.

Hosmer, D. W., Lemeshow, S., and Klar, J. (1988). A goodness-of- fit test for the multiple regression model. *Communications in Statistics*, A10, 1043-1069.

Hosmer, D. W. and Lemeshow, S. (2000). Applied Logistic Regression, 2nd ed, 1-2. Wiley and sons inc.

Joaquim, P., Marques de Sá. (2007). Applied Statistics Using SPSS, STATISTICA, MATLAB and R. 2nd Edition. Springer-Verlag Berlin.

Karegowda, A.G. and Jayaram, M.A. (2009). Cascading GA & CFS for feature subset selection in medical data mining. *IEEE*: 1-4.

Lewicki, P, Hill. T. (2006). Statistics: Methods and Applications: Comprehensive Reference for Science, Industry, and Data Mining. StatSoft, Inc.

Maniruzzaman, M., Rahman, M. J., Ahammed, B., and Abedin, M. M. (2020). Classification and prediction of diabetes disease using machine learning paradigm. *Health Information Science and Systems*, 8(1): 1-14.

Maniruzzaman, M., Rahman, M. J., Al-MehediHasan, M., Suri, H. S., Abedin, M. M., El-Baz, A., and Suri, J. S. (2018). Accurate diabetes risk stratification using machine learning: role of missing value and outliers. *Journal of Medical Systems*, 42(5): 1-17.

Maniruzzaman, M., Kumar, N., Abedin, M. M., Islam, M. S., Suri, H. S., El-Baz, A. S., and Suri, J. S. (2017). Comparative approaches for classification of diabetes mellitus data: Machine learning paradigm. *Computer Methods and Programs in Biomedicine*, 152: 23-34.

Mohapatra, S. K., Swain, J. K. and Mohanty, M. N. (2019). Detection of diabetes using multilayer perceptron. In International Conference on Intelligent Computing and Applications (pp. 109-116). Springer, Singapore.

Muhammad, M. U., Jiadong, R., Sohail, M. N., Irshad, M., Bilal M. and Osi, A. A. (2018). A logistic regression modeling on the Prevalence of Diabetes mellitus in the North Western Part of Nigeria, *Benin Journal of Statistics*, 1: 1- 10.

Negnevitsky, M. (2002). Artificial Intelligence: A Guide to Intelligent Systems, Addison Wesley, England.

Pei, D., Zhang, C., Quan, Y. and Guo, Q. (2019). Identification of potential type II diabetes in a Chinese population with a sensitive decision tree approach. *Journal of Diabetes Research*. 2019:12-21

Podgorelec, V. and Maribor, H.M. (2005). Improving mining of medical data by outliers' pre-dictions. *IEEE*: 1-6.

Principe, J.C., Euliano, N.R. and Lefebvre, W.C. (2000). Neural and Adaptive Systems. Fundamentals Through Simulations, John Wiley & Sons, Inc.

Semerdjian, J. and Frank, S. (2017). An ensemble classifier for predicting the onset of type II diabetes. arXiv preprint arXiv:1708.07480.

Sisodia, D. and Sisodia, D. S. (2018). Prediction of diabetes using classification algorithms. *Procedia Computer Science*, 132: 1578-1585.

Silva, F. M. and Almeida, L. B. (1990). Acceleration techniques for the backpropagation algorithm in L.B. Almeida, C.J. Wellekens (eds.), Neural Networks, Springer, Berlin, pp.110–119

Stephen. V. (1997) Selecting and Interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 77-89.

Tang, P.H. and Tseng, M.H. (2009). Medical data mining using BGA & RGA for weighting of features in Fuzzy KNN classification. *IEEE*, 5: 1-6.

Yu, W., Liu, T., Valdez, R., Gwinn, M., and Khoury, M. (2010). Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes. *BMC Medical Informatics and Decision Making*. 10:16-23

Wang, S. and Zhou, G.G. (2005). Application of fuzzy clusters analysis for medical image data mining. *IEEE*, 2: 1-6.

Wehrens, R. (2010). Chemometrics with R Multivariate Data Analysis in the Natural Sciences and Life Sciences. (pp. 70-106) Springer, England.

Xue, W. and Yanan S. Y. (2006). Research and application of data mining in traditional Chinese medical clinic diagnosis. *IEEE*, 4: 1-4.

Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y. and Tang, H. (2018). Predicting diabetes mellitus with machine learning techniques. *Front Genet*, 9: 515- 525