Arab American University

Faculty of Graduate Studies

Using Enhanced Artificial Neural Networks Model for Classification and Prediction of Diabetes

By

Haneen Omar Ismail Qteat

Supervisor

Prof. Dr. Mohammed Awad

This thesis was submitted in partial fulfillment of the requirements for the Master`s degree in

Computer Science

December/2019

Using Enhanced Artificial Neural Networks Model for Classification and Prediction of Diabetes

By

Haneen Omar Ismail Qteat

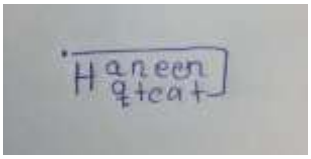
This thesis was defended successfully on **14/12/2019** and approved by:

Committee Members	Signature
1. Supervisor: **Prof. Dr. Mohammed Awad**	
2. Internal Examiner: **Dr. Osama Salameh**	
3. External Examiner: **Dr. Mohammad Al Dasht**	

Declaration

I declare that the work in this thesis entitled " Using Enhanced Artificial Neural Networks Model for Classification and Prediction of Diabetes" has been implemented by me in the department of Computer Science, it is supervised by Dr.Mohammad Awad. All information and results mentioned in this thesis are the result of my own work except for the works associated with reference. This thesis has not been submitted to any other degree.

Signature: Haneen omar ismail qteat



Dedication

I am grateful to my kind family, my mother, my father, my brothers and sisters, my friends and all my loved ones for supporting me during this work.

Acknowledgments

All praise and gratitude to Allah Almighty the creator, the creator of the intelligence. Many thanks and gratitude to Prof. Dr. Mohammed Awad, who guided me with his wise advice to complete this work to the fullest.

I would like to thank the Palestinian Diabetes Institute for facilitating the process of acquiring the dataset for diabetic patients in Palestine.

Abstract

Diabetes mellitus is one of the deadliest and chronic diseases that affect the persons who have an increase in their blood glucose levels. Type 1 Diabetes Mellitus “T1DM” is considered as one of the most dangerous types of diabetes as it is the reason why diabetes is called the silent killer. Due to the common symptoms of type 1 and 2 diabetes, diabetes specialists face doubts about their diagnosis of the type of diabetes in the patient where the uncertainty about the diagnosis of the disease may lead to delays in controlling the potential complications, especially if they have T1DM.

In this thesis, we have collected a local Palestinian dataset "DataPal" With the assistance of the Palestinian Diabetes Institute. A local Palestinian dataset was applied using machine learning algorithms to predict diabetes. The DataPal consists of 9 predictors used to diagnose diabetes types. The dataset consists of 314 instances of diabetic females. Where the women are more likely to die due to diabetes, so we gave them priority in this thesis. Thus, our samples were for females with both types 1 and 2 diabetes, aged between 5 and 89 years. The local dataset "DataPal" and the global dataset "PIDD" were preprocessed using the K-nn algorithm to fill their missing values because in medical diagnosis there is no room for error. The SVM algorithm also applied to the datasets to select the most optimal features to predict diabetes. Both the two-fold and four-fold cross-validation methods were applied to the datasets to evaluate the applied models fairly. A hybrid model "PSO-FFNN" uses the PSO evolutionary algorithm to train an FFNN and find the optimal weight values of the trained network. The PSO-FFNN model was applied to the

preprocessed datasets. Then the performance of the model was evaluated using different metrics such as the overall accuracy, recall, specificity, G-mean, and AUC and others. The obtained results show that the proposed PSO-FFNN model outperformed all models applied in this thesis in the classification of diabetes and its types.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

DM	Diabetes Mellitus
ML	Machine Learning
T1DM	Type one Diabetes Mellitus
T2DM	Type two Diabetes Mellitus
SVM	Support Vector Machine
K-NN	K Nearest Neighbor
FFNN	Feedforward Neural Networks
MLP-BPNN	Multiple Layer Perceptron Backpropagation Neural Networks
FP	False Positive
FPR	False Positive Rate
G-mean	Geometric mean
TP	True Positive
TPR	True Positive Rate
AUC	Area Under Curve
ROC	Receiver operating Characteristic
NB	Naive Bayes
RF	Random Forest
PSO	Particle Swarm Optimization
GAs	Genetic Algorithms
ACO	Ant Colony Optimization
DT	Decision Tree
DA	Discriminant Analysis
PIDD	Pima Indian Diabetes Dataset
FN	False Negative
TN	True Negative
RBF	Radial Base Function
UCI	University of California, Irvine

ADA	American Diabetes Association
SCGD	Scaled Conjugate Gradient Descent

Chapter 1

Introduction

1 Introduction

Diabetes mellitus is a chronic disease and one of the 10 deadliest diseases in countries with a strong economy [1]. Diabetes causes many complications that greatly affect human life [2]. Diabetes is one of the main causes of myocardial infarction, diabetic retinopathy, diabetic bone necrosis, kidney failure, amputation, diabetic neuropathy, diabetic coma and other complications that have not been discovered yet. The number of diabetic people with both types of diabetes has been increasing over the last forty years. According to the World Health Organization [1], diabetes is an epidemic disease that poses a threat to human life. 80% of diabetes deaths are in low-income countries that occur in low and middle-income countries [1]. Diabetes caused 4.2 million deaths [70]. Diabetes can be controlled whenever it is detected in the early stages. In addition, the correct diagnosis of the type of disease plays a major role in reducing the symptoms of the disease. T1DM is the most dangerous level of diabetes as it is the reason why diabetes is called the silent killer. Diabetes affects people with high blood sugar in the body. There are two different types of diabetes: T1DM and T2DM, which affect people who are unable to produce insulin or are unable to use it to convert the glucose in the blood into energy [2].

T1DM usually affects young people, mostly under the age of 30. The most common symptom of T1DM is high blood sugar and constant thirst [3]. This type of diabetes cannot be prevented and can only be treated with insulin injections. T2DM is the most common diabetes type and often affects adults. Usually associated with high blood pressure, obesity, atherosclerosis, and other diseases [6].

Usually, doctors and diabetes specialists use some of the tests and factors that help them in the diagnosis of diabetes and its type. Diabetes is usually diagnosed by a person's weight, blood pressure, fasting blood glucose FBS, random blood glucose RBS, the Hemoglobin A1C, and many other tests [3][4][5]. To help detect diabetes early to be able to control it, machine learning mechanisms have proven their ability to help doctors make an initial diagnosis and It can also be a source to confirm their diagnoses [7][8].

To predict diabetes accurately, all attributes that will be used to predict the disease have to contain the correct values without any missing values. The features that are best suited to predict diabetes must be selected. A set of preprocessing steps has to be applied on datasets to fill in the missing values and select the optimal features. To select the optimal features, different feature selection algorithms can be applied such as the genetic algorithm [15], Linear discriminant [53] and some of the evolutionary algorithms [37][38].

Several Previous works have applied different machine learning methods to predict diabetes. A large number of common learning algorithms have been applied in diabetes prediction [9] including Support Vector Machine "SVM", Decision Tree "DT" and others. Olaniyi and Adnan [4] have applied Back Propagation Neural Network "BPNN" to distinguish between diabetic and non-diabetic people. N. Mohana Sundaram [9] proposed an approach of applying the Elman neural network with MLP neural network to predict diabetes. The Scaled conjugate gradient back-propagation algorithm used to train the network.

Support Vector Machine "SVM" with different kernel functions (i.e.Linear, Radial Base [54], Polynomial, and Gaussian) functions, K-Nearest Neighbor "KNN", Discriminant Analysis Classifier "DA", Naive Bayes "NB", Decision Tree "DT" and Random Forest "RF" algorithms are some of the most common machine learning techniques used to predict diabetes.Zou Q et.al [18] collected a data set from Luzhou, China, consisting of 12 attributes that have been used in diabetes predictions. They have noticed that fasting glucose is the most important predictor but it does not eliminate the role of the rest of the attributes in diabetes prediction.

They have applied the Random Forest "RF", MLP-BPNN and J48 decision tree to the PIDD dataset [24] and a local dataset that they have collected to predict and diagnose diabetes.

K-NN, SVM, Naïve Bayes, and Decision trees automatic classification algorithms were applied by Thirumal P. C. and Nagarajan N. [52] to predict diabetes. The algorithms have been applied and tested to the PIDD dataset. The K-NN is a lazy algorithm given that it doesn't get rid of the previous classified instances until the coming of the other. So, the K-NN algorithm got one of the worst overall accuracy values. The distinction between the presence of diabetes and non-diabetes is very important, but misdiagnosis of its type poses a greater risk. Several previous studies have used machine learning algorithms to predict type 2 diabetes. XL Xiong et.al. [10] have applied AdaBoost "AD", MLP, Random Forest "RF", and Gradient Tree Boosting "GTB" algorithms in addition to their proposed algorithm to their own local dataset of the Chinese adult population consisting of 11845 cases to classify non-diabetic and people with T2DM, where they have achieved accuracy up to 87%.

Razavian et.al. [11] have proposed a machine learning methodology to classify high dimensional datasets of multiple features to predict T2DM, using the logistic regression mechanism. In contrast, previous studies have put the type of diabetes as future work [18].

It has proven that Machine learning techniques can make a medical diagnosis, especially diabetes prediction. The support vector machine "SVM" is one of the most powerful supervised machine learning methods. SVM has been applied in various fields including the medical diagnosis. Multi-Layer Perceptron Backpropagation Neural Network MLP-BPNN has been widely applied to diagnosis diabetes, with different training algorithms that are used to adjust the network weight values. The evolutionary algorithms [36] such as Particle Swarm Optimization "PSO", Genetic Algorithm "GA", and Ant Colony Optimization "ACO" algorithms have been applied to solve machine learning problems. The evolutionary algorithms used in training the ANNs to search the optimal weight values in the problem search space. Some of the above-mentioned models and others have been applied in this thesis to demonstrate the ability of the proposed model to predict diabetes.

In this thesis, we have collected a local Palestinian dataset "DataPal" With the assistance of the Palestinian Diabetes Institute. The DataPal consists of 314 cases of diabetic people and 9 predictors are presented in section 2.1.1. Diabetes in the dataset is classified into two types T1DM and T2DM. Another dataset called Pima Indian Diabetes Dataset "PIDD" the most applied dataset in diabetes prediction also has applied in this thesis to validate the performance of the proposed model. Both datasets were applied to a set of preprocessing steps to prepare them for prediction. The K-nearest neighbor algorithm was applied to the

datasets to clean their missing values. The missing values were estimated using the K nearest instances. Then the SVM classifier was applied as a feature selection method, to select the most optimal attributes.

The preprocessed datasets were applied to the K-fold cross-validation method to split the datasets into two subsets for training and testing as presented in section 3.2.3. In this thesis, we have applied both Two-fold and four-fold cross-validation methods to validate the performance of the applied models. Each of support vector machine SVM, K-nearest neighbor KNN, Decision Tree DT, MLP-BPMN, and other machine learning algorithms were applied in this thesis. In addition, the PSO-FFNN proposed a hybrid model that uses one of the most powerful evolutionary algorithms used with the high-dimensional search spaces. In this thesis, we propose to use a hybrid approach of the Feed-forward neural network FFNN model with the PSO optimization algorithm to improve the diagnosis of Diabetes Mellitus and its types. The PSO algorithm is supposed to be the training method of the FFNN model to adjust the weights of the network. To prove the ability of the applied models including the PSO-FFNN, a set of performance metrics was used [45][46]. Confusion matrix, classification accuracy, classification error rate, precision, recall, specificity, g-mean, and area under curve ”AUC” are some of the main measurements used to evaluate the performance of the applied models in this work. Where all results show that the PSO-FFNN model outperforms all models in predicting diabetes and types with accuracy with a total accuracy by up to 89.09% and 99% respectively.

1.1 Objectives

Through this thesis, we aim to improve the accuracy of diagnosing diabetes compared to previous studies. In medical diagnoses, there is no room for error because this will affect the human life. Misdiagnosis of diabetes may lead to death because diabetic people will not be able to receive proper preventive treatment. In addition, the Misdiagnosis of diabetes type is very dangerous as well. Where people with diabetic type I may develop serious complications and other diseases caused by diabetes. In case they are not provided with insulin to prevent these complications. Thus, for example, if people with type 1 diabetes are misdiagnosed as having type 2 diabetes, they are prevented from receiving the necessary insulin doses, which is a big risk to their lives.

This thesis presents the hybrid PSO-FFNN prediction model to predict diabetes types. The PSO-FFNN, K-NN, SVM, and other machine learning models were trained using local (i.e. the DataPal) and global (i.e. the PIDD) datasets in this thesis. The trained models are then used to obtain an accurate diagnosis of the types of diabetes.

During the work in this thesis, we have faced a set of limitations such as our inability to find any medical data related to following up patients 'conditions for long periods of time, which made it almost impossible to predict the state of diabetes for a particular patient.

1.2 Contribution

In this thesis, several machine learning techniques have been applied to improve the diagnosis of the type of diabetes. The models were applied to two datasets, local datasets

collected from the Palestinian Diabetes Institute and global dataset collected from the UCI Machine Learning Repository. The local "DataPal" dataset is the first Palestinian diabetes dataset that has been applied to predict diabetes.

The performance of the applied models was evaluated by a set of measures where the proposed PSO-FFNN model had the best results. The model was the best among the applied models in terms of total accuracy and in terms of recall that proves the ability of the model in diagnosing T2DM and the specificity that proves the ability of the model in diagnosing T1DM. It was the best in terms of the G-mean values and the value of the prediction tradeoff between T1DM and T2DM. In addition to the area under the ROC curve that illustrates the ability of the models to predict T2DM correctly.

1.3 Overview

The content of the thesis is arranged as follows. Chapter 2 will introduce a backgIteration about the DataPal and PIDD datasets that will be used in this thesis, and then a set of previous works within the same research field will be discussed in the related work section. Chapter 3 will illustrate the proposed approach, where the dataset selection considerations will be discussed in section 3.1, the dataset collection and preprocessing steps will be presented in section 3.2 then the rest of the chapter will discuss the main three applied models, the model procedures and the performance metrics used to evaluate the machine learning techniques. The preprocessing phase, the implementation platform, and the system parameters are presented in Chapter 4. The results of applying the PSO-FFNN and the other models that have been applied in some of the previous works to the DataPal and the PIDD

datasets will be illustrated in Chapter 4 as well. Where the conclusions and some future works will be discussed in Chapter 5. At the end of this thesis, the results of the experiments of selecting the PSO-FFNN optimal parameters will be presented in Appendix A.

1.4 Summary

In this chapter, we have introduced the main purpose of this work. The motivation and the contribution of this thesis in solving the problem statement were discussed as well. Also, the main objectives of this work were discussed. The next chapter will present backgIteration about the datasets applied in this thesis and various previous works in predicting diabetes.

Chapter 2

Background

2 Background

Given that diabetes is a widespread chronic disease that affects adults and children, it has to be predicted and diagnosed accurately. All diseases have to be predicted sensitively and accurately because they may affect human lives. Diabetes Mellitus "DM" has many types such as Gestational diabetes GDM, T2DM and T1DM. DM types differ in terms of complications and side effects. T1DM is the most dangerous level of diabetes that may cause death [64]. So, the early diagnosis of the disease can slow down or alleviate the side effects of the disease and can also prevent them from developing a lot of side effects. A lot of methodologies were applied to support the diabetes diagnosis. Machine Learning techniques have been applied by several researchers in diabetes prediction as presented in section 2.2.

Diabetes occurs when the body does not produce insulin or it cannot convert insulin into glucose, the main source of energy for the human body [64].

Diabetes has many types, where two of the most common types are T1DM and T2DM.

1-Type I diabetes T1DM: Type I Diabetes Mellitus is the most dangerous level of diabetes mellitus that may cause death. T1DM infects the persons who do not produce insulin at all, the hormone necessary for the absorption and utilization of glucose. Where this type of diabetes constitutes less than 20% of diabetic people[1]. T1DM is mostly common among young people.

2-Type II diabetes T2DM: Type II Diabetes Mellitus is less dangerous compared to T1DM. This type affects individuals directly and it is common for older people. It occurs when the

patients are unable to metabolize glucose [12]. Where the majority of diabetic people are carriers of the disease type II by up to 80%.

3- Gestational Diabetes Mellitus GDM: Gestational Diabetes is caused by a glucose tolerance disorder during pregnancy or it may be the beginning of symptoms of permanent diabetes. Where 7% of pregnant women develop gestational diabetes, equivalent to 200,000 cases annually, most of them return to the way they were before pregnancy [75].

2.1 Datasets Description

There is a set of diabetes datasets that have been used in diagnosing diabetes in different domains whether it is in the field of diagnosing the diabetes disease or in diagnosing the type of diabetes. On the one hand, a global dataset the Pima Indians Diabetes mellitus PIDD dataset collected from the UCI Machine Learning Repository is the most common applied dataset in diabetes prediction [21] [6] [7] [10][14]. On the other hand, there are some locally collected datasets such as the dataset collected from several clinics in Kerala [56] and a dataset collected in [62].

In this thesis, we have collected and prepared a new dataset DataPal to be applied to different models. Besides the Pima Indians Diabetes Mellitus PIDD that will be used to prove the ability of the proposed model to predict diabetes. In the following subsections, we describe the two datasets applied in this thesis.

2.1.1 DataPal Dataset

DataPal dataset is a Palestinian dataset locally collected from the Palestinian Institute of Diabetes within two weeks. It consists of 9 predictors used to diagnose diabetes types.The dataset consists of 314 diabetic females . Where the women are more likely to die due to diabetes, so we gave them priority in this thesis. Thus, our samples were for females with both type 1 and type 2 diabetes, aged between 5 and 89 years. Figure 2.1 illustrates the distribution of diabetes types in the DataPal dataset. The following are the attributes used to diagnosis diabetes types:

1- Age: Age of patient is one of the most powerful predictors in predicting diabetes types. Where the T1DM is mostly common among young people while T2DM is common for older people

2- Diabetes Mellitus DM-family history: Having first and second-degree relatives with diabetes have the effect of transmitting the disease from generation to generation.

3- Pregnancies: Number of pregnancies. The number of pregnancies affects the determination of the type of sugar because pregnant women are more likely than others to develop diabetes.

4- Body mass index "BMI": The body fat index according to the height and the weight of patients. Obesity is one of the leading causes of diabetes because of high blood sugar levels. Thus, BMI has a role in the classification of the type of diabetes.The following equation is used to calculate the BMI value [71]:

$$BMI = weight\ in\ kilograms/(height\ in\ meters\ X\ height\ in\ meters) \quad 2.1$$

5- Hypertension "HTN": Hypertension in family history. Blood pressure has a direct relationship with diabetes, where blood pressure is one of the most important factors affecting diabetes.Thus, the presence of high levels of blood pressure in the first and second degree relatives makes people vulnerable to the same disease, which is one of the diseases most linked to diabetes.

6- Ischemic Heart Disease "IHD": People with both diabetes and cardiovascular disease are more likely to die [6] [66] where both diseases are closely linked to each other. Thus, the genetic factor has a role in the transmission of the disease between generations.

7- Fasting Blood Sugar "FBS": Blood tests are done after fasting a full night to find out the blood sugar level [67].

8- Human Glycated Hemoglobin A1c "HbA1c": Test of the average blood sugar associated with hemoglobin during the last two months or three months without the need to fast before the test [13][67].

9- Blood Pressure "BP": The patient diastolic blood pressure value. According to ADA, at high blood pressure, the effort on the heart increases, so the incidence of heart disease and diabetes increases [68]. Hypertension is closely related to diabetes, as the combination of both type 2 diabetes and hypertension is one of the most fatal cases [68]. In addition to the high incidence of other side effects such as blindness, renal virtue or retinopathy. The max and min values for the DataPal dataset attributes are shown in table 2.1.

Table 2.1: The DataPal dataset features List.

Attribute	Minimum Val	Maximum Val
Age (years)	5	89
DM (The number of diabetic people in the family **[0:2]**)	0	2
Pregnancies (Number of times pregnant)	0	17
BMI (weight in kg/(height in m)^2)	19	40
HTN (The number of people with high blood pressure in the family **[0:2]**)	0	2
IHD (Number of people with cardiovascular disease in the family **[0:2]**)	0	2
FBS (mg/dL)	58	612
BP (mm Hg)	45	110
HbA1c (mmol/L)	5	15

To predict diabetes, the Fasting Blood Sugar FBS test can be used to distinguish the T1DM and T2DM [69]. It can be distinguished between T1DM and T2DM according to the value of the FBS test, so the patient has T1DM or T2DM if FBS value < 130 mg/dL or > 130 mg/dL for both types, respectively.

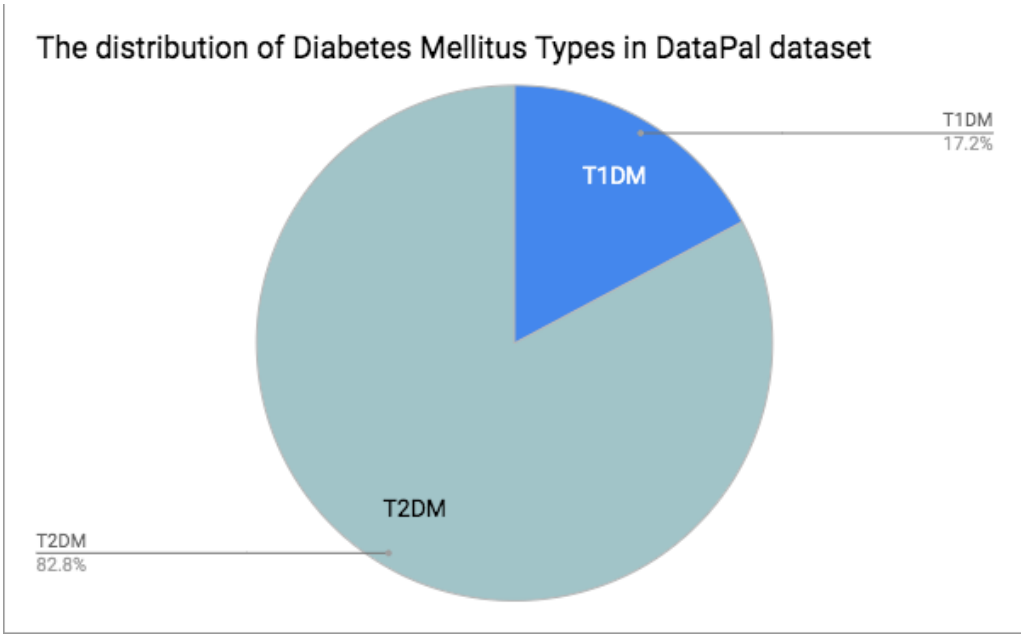


Figure 2.1: A chart illustrates the distribution of diabetes types in the DataPal dataset.

2.1.2 Pima Indian Diabetes Mellitus Dataset "PIDD"

The most applied dataset for diabetes mellitus prediction research is the PIDD dataset as we will present in section 2.2. Diabetes is prevalent among the Pima Indians, where the only type of diabetes that affects them is T2DM. The PIDD dataset collected from the UCI Machine Learning Repository [24].The dataset contains 8 attributes that help diagnose diabetes consist of 768 diabetic females over 21 years old. Where 500 of them are non-diabetic and 268 are diabetic as shown in figure 2.2

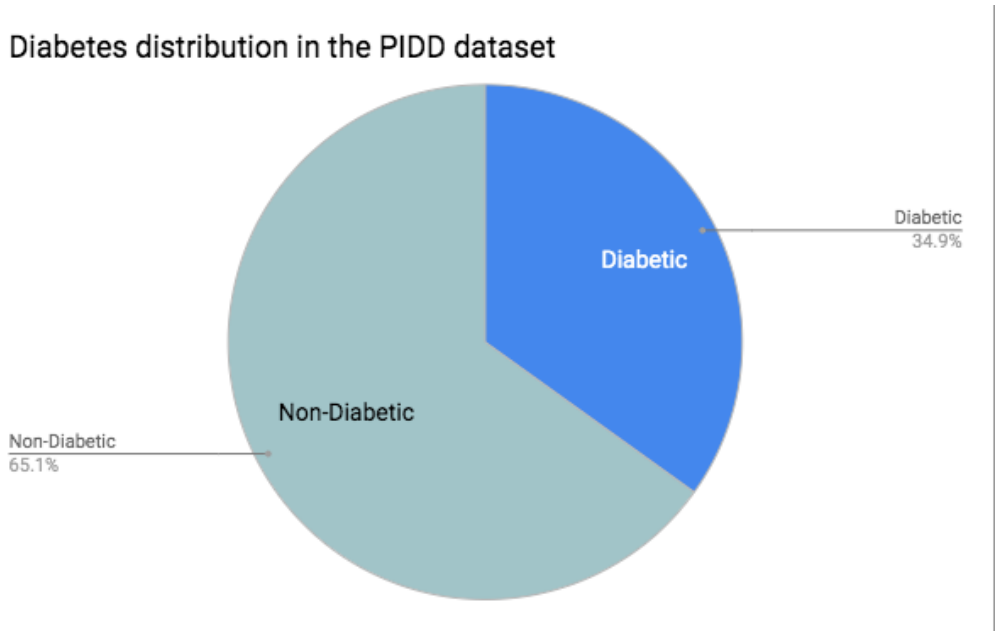


Figure 2.2: A chart illustrates diabetes distribution in the PIDD dataset.

The PIDD will be applied to the proposed model and various machine learning models to validate the proposed model's ability to predict diabetes. Where the following attributes are the predictors that will be used to diagnose diabetes in the PIDD [24]:

1- Pregnancies: Number of pregnancies as we have mentioned in sec 2.1.1 .

2 - Glucose: Plasma glucose concentration 2 hours in an oral glucose tolerance test.

3 - Blood Pressure "BP" - (mmHg): The patient diastolic blood pressure value as we have mentioned in sec 2.1.1.

4 - Skin Thickness - (mm): triceps skinfold thickness.

5 - Insulin 2-Hour serum insulin (mu U/ml).

6 - Body mass index "BMI": BMI is calculated according to [71] as we have mentioned in sec 2.1.1.

7 - Diabetes Pedigree Function.

8 - Age.

The PIDD dataset contains many missing fields, so we will present in section 3.2.2 the preprocessing steps that we follow to correct the dataset and estimate the missing values and how to select the optimal attributes. The max and min values for the PIDD dataset attributes are shown in table 2.1.

Table 2.2: The PIDD dataset features List.

Attribute	Minimum Val	Maximum Val
Pregnancies (Number of times pregnant)	0	17
Glucose (mmol/L)	44	199
Blood Pressure (mm Hg)	19.6	122
Skin Thickness (Triceps skin fold thickness (mm))	7	99
Insulin (2-Hour serum insulin (mu U/ml))	14	846
BMI (weight in kg/(height in m)^2)	18.2	69
Diabetes Pedigree Function (Diabetes pedigree function)	0،078	2.42
Age (years)	21	81

2.2 Related Works

The problem that will be raised in this thesis is the model used to classify and predict diabetes mellitus types.The main objective of applying machine learning methods to classify medical datasets is to help the medical specialist and doctors to process the non-linear data automatically and find the correct diagnosis. The medical diagnoses are very sensitive, especially with chronic diseases such as Diabetes Mellitus.

manipulate

Artificial Intelligence methods have proved to be effective in analyzing non-linear data more than statistical methods [72][73] . In previous years, many researchers have used different machine learning techniques to predict and diagnose diabetes. Olaniyi and Adnan [4] have applied Back Propagation Neural Network (BPNN) on Pima Indians Diabetes Data Set after normalizing the sample with values between 0 and 1. They have got an overall accuracy of up to 82 %. Pradhan and Sahu [5] have also applied their proposed methodology to the Pima Indians Diabetes Data Set. They have got an overall accuracy of up to 73.5% using a Genetic algorithm GAs and the backpropagation neural network approach.

Zou Q et.al. [18] Have collected a set of instances from a hospital in Luzhou, China consists of 12 attributes used in diabetes predictions. They have noticed that fasting glucose is the most important predictor but it does not eliminate the role of the rest of the attributes in diabetes prediction. Random Forest "RF" the method of some decision trees where the average of all trees output is the final output of the RF model. The MLP-BPNN neural network that learns in a backpropagation manner as we will present in chapter 3. The J48 decision tree or it called C4.8 [20][19] was one of all attributes selected to be the root node, a branch will be generated for each attribute, the samples separated into different subsets, where the subsets will be distributed over branches of the root node. The algorithm stops when all samples got the same classification [21]. The three above algorithms were applied to both the collected Luzhou dataset and the PIDD dataset. The best results obtained by

applying Luzhou and the PIDD datasets were 80.84% and 77.2%, respectively. They also expressed their inability to predict the type of diabetes and made it their future work, the problem that we will fix in our thesis.

SomnathRakshit, et .al [6] have achieved an 83.3% overall accuracy. They have applied their model on women over 20 to predict T2DM. Where a two-class neural network was applied to the global PIDD dataset. While Akm Ashiquzzaman, et al [7] have used a new approach, where they have trained the PIDD using a dropout of neurons approach with a neural network by extracting some neurons from some layers while the network weights are updating to avoid overfitting. They have noticed that the overall accuracy achieved from applying the dropout methodology is better than the standard regression networks, by up to 88.41% and 76% respectively.

The neural network methodologies proved to be one of the best machine learning techniques in diabetes predictions. M. Durairaj and G. Kalaiselvi [8] have evaluated various machine learning techniques to classify diabetes, thus the ANN model was the winner with overall accuracy up to 89%. N.MohanaSundaram [9] proposed an approach of applying the Elman neural network with MLP neural network. The Scaled conjugate gradient backpropagation algorithm used by Olaniyi and Adnan [4] has applied Back Propagation Neural Network "BPNN" to distinguish between diabetic and non-diabetic people. N.MohanaSundaram [9] proposed an approach of applying the Elman neural network with the MLP neural network to predict diabetes. The Scaled conjugate gradient backpropagation algorithm used to train the network.

Train the network. While the softmax activation function used to calculate the final output of the layers instead of using the sigmoid activation function. A 95.7% accuracy achieved by training an ELMAN Softmax model on the PIDD to predict diabetes mellitus. While the Neural networks have proven their predictability in diabetes by applying different approaches to train the neural networks and tune the connection weights. The evolutionary algorithms (i.e.GA, PSO and ACO) proved their ability to support the neural networks in diabetes predictions as well. As Karamath Ateeq and Dr. Gopinath Ganapathy [10] have proposed a modified PSO "MPSO" model to predict diabetes. Both a Multi-Layer Perceptron Network "MLPN" and a Radial Basis Function Network "RBFN" models were applied to validate their proposed model. The three mentioned models were applied to both the PIDD and US datasets. Where the proposed hybrid MPSO-NN model was the winner achieved an 84.2% and 81.8% overall accuracy by applying the US and the PIDD datasets, respectively. A lot of evolutionary algorithms used in medical diagnoses in general and in diabetes pattern recognition specifically.

Vaishali et . al. [14] has proposed a hybrid approach consisting of a Fuzzy classifier with Multi-Objective Evolutionary to predict diabetes. The genetic algorithm used as a feature selection classifier to select the most relevant attributes from the PIDD dataset, where a lot of features have been eliminated. The model is based on rules which are suitable to apply with the critical decision support systems. Using the proposed hybrid model they got an overall accuracy of up to 83%. The genetic algorithm proposed by Choubey, D.K. et.al. [15] as a feature selection method as well. The Naive Bayes classification algorithm has been

applied to The PIDD dataset to diagnose diabetes. The results show that the Naive Bayes algorithm got 78% overall accuracy. It can be noticed that medical datasets became larger and larger, making the search space more complex to predict. Thus, Asha Gowda Karegowda et al. [17] have applied the Genetic algorithm to train an MLP-BPNN to find the optimal weights. Where the GA is considered as one of the best search methods through large spaces. The PIDD was applied to a Decision Tree and GA_CFS feature selection methods to select the most optimum features in predicting diabetes Mellitus.

The above-mentioned works that have applied various evolutionary algorithms to improve the predictions were a bit weak. The GA is time-consuming in an iterative manner algorithm and needs a lot of parameters to tune. The mentioned modified PSO algorithm has additional adjustments that may affect the results of the prediction negatively. The thing that we try to avoid through our applied PSO-FFNN model.

It is undeniable that the preprocessing phases are an integral part of the success of any pattern recognition methodologies. Collecting the datasets, cleaning the datasets and selecting the most optimal features are three of the main phases of any pattern recognition approaches. The datasets can contain missing values as with the PIDD dataset. Several previous studies [16] have proposed to clean these missing values with the mean of all column values. Use the K-nearest Neighbor “K-NN” algorithm of K neighbor columns to estimate the missing values [16]. Or remove the samples with missing values. In this thesis, we will prove that the optimal way of cleaning the PIDD dataset is by removing the samples with missing values. The medical diagnosis is very sensitive, where any error in the

diagnosis may affect human life so there is no room for error. T. Jayalakshmi et al. [16] have preprocessed the PIDD dataset using the K-nearest Neighbor algorithm “K-NN” by replacing the missing values from nearby columns. The K-NN algorithm applied as a classifier to predict diabetes mellitus. Krati Saxena et.al. [50] have applied the K-NN algorithm to a dataset [51] consist of 200 instances with 11 predictors. K-NN algorithm applied to train and test the instances for a different number of "K" nearest neighbors to predict diabetes. The overall accuracy and error rate metrics have been calculated to evaluate the results of the model. Where both K=5 and K=3 have been used to test the model.

K-NN, SVM, Naïve Bayes, and Decision trees automatic classification algorithms were applied by Thirumal P. C. and Nagarajan N. [52] to predict diabetes. The algorithms have been applied and tested to the PIDD dataset. The K-NN is a lazy algorithm given that it is don't get rid of the previous classified instances until the coming of the other. So, the K-NN algorithm got one of the worst overall accuracy values. The overall accuracy values obtained by applying the K-NN, SVM, Naïve Bayes, and Decision trees models are 77.73%, 77.4%, 77.86%, and 78.25%, respectively. In this thesis, we have applied the models presented at [52] to validate the PSO-FFNN.

Parashar A. et al. [53] have applied the Support Vector Machine to classify diabetes mellitus and the linear discriminant feature selection method to select the optimal predictors. They have applied a Feed-forward Neural Network "FFNN" to verify their applied model. Their combination model of Support vector Machine with Linear discriminant feature selection

method outperformed the FFNN in predicting the PIDD dataset, by up to 77.6%. Also, Kumari V. Anuja and Chitra R. [54] have applied the Support Vector Machine algorithm using a Radial base function kernel “RBF-SVM" to predict diabetes. The RBF-SVM model applied to the PIDD dataset to classify diabetic and non-diabetic instances. thus the model has achieved classification accuracy up to 78%.

Sarwar, M. A et.al. [55] have applied six machine learning algorithms to the PIDD dataset to classify diabetes. Support Vector Machine "SVM", K Nearest Neighbor "KNN", Decision Tree "DT", Logistic Regression "LR", Naive Bayes "NB" and Random Forest "RF" were the algorithms used to classify the PIDD dataset. The results obtained by training all the 8 features of the PIDD dataset where they have achieved classification accuracy up to 77% by applying both KNN and SVM models. Veena and Anjali [56] have proposed a set of machine learning classifiers to diagnose diabetes. A global dataset collected from the UCI repository of machine learning and a local dataset collected from several clinics in Kerala used to evaluate the models. Decision Tree "DT", Support Vector Machine "SVM", Naive Bayes "NB", and Decision Strumb "DS" were applied to both datasets. The DS got the highest accuracy of up to 80.72%. Minyechil Alehegn et.al. [57] have applied Naive Net "NN", Decision Stump "DS", and Support Vector Machine "SVM" algorithms to validate their proposed model. They have applied the mentioned models to the PIDD dataset. A set of pre-processing techniques have been applied to the dataset such as filling the missing data, removing duplication and data conversion. Their proposed model achieved an overall accuracy of up to 90%.

Evolutionary algorithms have the main role in medical dataset diagnosis. Several previous datasets have applied various evolutionary algorithms in predicting diabetes. The Genetic Algorithm "GA" and a Back Propagation Neural Network have been applied as a hybrid prediction model by A.G. Karegowda et.al. [58] to predict the PIDD dataset. The GA has been used to optimize the MLP-BPNN neural Network. The hybrid model of the GA and an MLP-BPNN has achieved an accuracy of up to 84%. The PSO evolutionary algorithm has been applied to predict medical diagnosis in general and diabetes in particular. Yuan et al [59] have applied each of the GA and the PSO evolutionary algorithms to optimize the parameters of the SVM classifier. G.Kranthi Kumar and K.Swathi [60] also proposed the PSO algorithm to adjust the SVM classifier parameters. Then the optimized SVM has been used to classify the PIDD dataset. Alaa Badr Eysa et al [61] have applied two models to predict diabetes. The MLP-BPNN and PSO-NN hybrid approach have been applied to the PIDD dataset to classify diabetes. Where the implementation of both models achieved an accuracy of 77.8% and 88.2% respectively. Many studies have applied different machine learning techniques to predict diabetes and the status of the disease (i.e. Diabetic, Non-diabetic or Pre-diabetic). Sejdinović, D et.al.[62] have proposed an approach to classify the T2DM and Prediabetic instances according to two predictors. A dataset consisting of 190 samples was applied by an ANN to classify T2DM and Pre-diabetic using both the FPG and the HbA1c predictors (i.e.Fasting blood sugar and cumulative diabetes during the last three months). They rated 94.1% and 93.3% of cases of Pre-diabetic and T2DM correctly.

In this thesis, we will conduct a set of data mining methods using machine learning techniques to predict Diabetes Mellitus. The proposed PSO-FFNN model and a set of machine learning techniques will be used to classify the Diabetes mellitus and its types (i.e.T1DM and T2DM). A Palestinian dataset DataPal has been collected to be used in this thesis to evaluate applied models in classifying the Diabetes types (i.e.T1DM and T2DM). In addition to a global dataset, the PIDD dataset to be applied using applied models in this thesis to classify the diabetic and non-diabetic cases.

PSO, GA, and ACO are some of the applied evolutionary algorithms on the optimization problems of machine learning to optimize the ANNs and classify variant datasets. Also, evolutionary algorithms can be used as a feature selection method to extract valuable data from all the huge amounts of data in databases to improve the resolution of prediction [37][38]. Both the GA and the PSO algorithms are very good optimization algorithms with the large search spaces for large datasets. But, if we compare the implementation of PSO algorithm based on particle's velocity with another evolutionary algorithm such as GA, it is easier to implement PSO than GA, since the GA has a lot of factors to take into consideration when implementing an optimization problem such as mutation and crossover which is time-consuming [41]. Thus in this thesis, we propose to use a hybrid approach of the Feedforward neural network FFNN with the PSO optimization algorithm to improve the diagnosis of Diabetes Mellitus and its types. The PSO algorithm is supposed to be the training method of the FFNN model to adjust the weights of the network. Chapter 4 will

present the experiment results of applying various models that will be used to demonstrate the ability of the PSO-FFNN model in diabetes predictions.

Chapter 3

The Proposed Method

3 The Proposed Method

In this chapter, we present the whole scheme of dataset considerations and models used and built to validate our results in this thesis. The methods used to prepare valuable datasets and the models that used these datasets to improve the pattern recognition accuracy in diabetes mellitus are described in this chapter. In section 3.1, the dataset selection considerations are described. The data collection method used to collect the Palestinian data set “DataPal" that mentioned in chapter 2 section 2.2.1, the preprocessing phase and the datasets partitioning method were described in section 3.2. Then the proposed hybrid model procedure and the other models (i.e. SVM and MLP-BPNN) will be described in section 3.3. Finally, the analysis techniques and measurements used to evaluate the performance of the models in this thesis are presented in section 3.4 of this chapter.

3.1 Dataset Selection Considerations

The development of biotechnology has resulted in a huge amount of medical data [22][23], resulting from clinical and laboratory tests of patients. Different datasets have been applied to diagnosis diabetes, global data sets such as the PIDD dataset which have been applied by most researchers [24] and a local diabetes dataset such as “LDD” that have been customized by Choubey and others for their study [25]. Both local and global data contain a set of data for a group of patients that is converted from data to knowledge using data mining techniques such as machine learning algorithms. Data mining techniques (e.g. SVM, K-NN .etc) have been applied to the databases of a group of patients to distinguish between

diabetic and non-diabetic people. The data sets used to detect diabetes are divided into two groups,the first group to classify the people to diabetic and non-diabetic and the second group to determine the type of diabetes in diabetic people. Most of the databases available are from the first group, but in contrast, the databases available from the second group are very few. When taking the risk of death due to diabetes into account, the females are more vulnerable to the risk of death compared to males [26][27]. Females are more likely to develop serious diabetes complications than males, such as heart disease, which can cause death [28].Thus, in this study, we prioritize diabetic females so we have collected a diabetes dataset of diabetic females that will be applied in this thesis. With the help of some Palestinian doctors and specialists in the field of diabetes, we have collected samples for diabetic females of the two types (i.e.T1DM and T2DM) to be used to validate the model presented in this thesis. The data set consists of some features that help specialists distinguish between the two types of diabetes to determine the level of risk of the disease to the patient. The selection of the data set has an impact on the efficiency of the evaluation. Therefore, we have collected a medium-sized data set with no shortage of content. The selection of the data set has an impact on the efficiency of the evaluation. Therefore, we have collected a medium-sized data set with no shortage of content. Our dataset is divided into two groups, with T1DM diabetes being the largest 82% compared to 16% for T2DM. With such a large proportion of T1DM, we have a great chance to limit the development of T2DM to T1DM if possible by accurately detecting the type of diabetes and informing patients of the seriousness of their disease.

3.2 Dataset Collection

Before starting our process, we have collected a new diabetes dataset from the Palestinian Diabetes Institute, which specializes in the classification of types of diabetes in diabetic people. Randomly, we have collected 314 data samples of diabetic females. In addition to the lack of time that we faced because all government and private institutions refused to give us any data related to their patients except the Palestinian Diabetes Institute where they were our only option. Thus, our samples were for females with both types 1 and 2 diabetes, aged between 5 and 89 years. Data collection using questionnaires was not possible because diabetes tests are very accurate. No ordinary person can give accurate information about his or her health, so we have to take them from their original and exact source.

3.3 Data Preprocessing

After collecting the Palestinian dataset DataPal and downloading the most popular used global dataset PIDD, many data pre-processing steps [29] have to be applied to extract valuable data values. Preprocessing involves some steps such as fill the missing values (data cleaning), feature selection and data partitioning , most of which are used to prepare the used dataset in different machine learning methods. The following are the main steps for data preprocessing:

3.3.1 Data Cleaning

Preprocessing involves some steps to fill the missing values, where the PIMA Indian diabetes global data set present in table 2.2 contains a large number of missing values that must be considered before use to verify the efficiency of the proposed model in this thesis. Where the data set that we customized for this thesis consists of 10 attributes presented in table 2.1, as it contains a few missing values compared to global data sets. The data set consists of 2 classes (T1DM and T2DM) that distinguish it from other datasets. The local dataset "DataPal" is applied in this thesis to conduct the first study on a Palestinian dataset, in particular, to distinguish the two types of diabetes using the model proposed in this thesis. Both datasets applied in this thesis contain only numeric data. As mentioned earlier, the data sets, especially the general PIMA, contain missing values.

Most of the previous studies did not mention that PIDD has a lot of missing values, where approximately 57% of the PIDD cases have at least one missing value. Where they have verified their models using all dataset values including missing values, with a total of 768 rows. So, the KNN algorithm [16] has been applied to fill the missing values in both datasets applied in this thesis. The PIDD dataset contains at least 400 instances that have a missing value in one or more attributes. So in this thesis, we have applied the KNN algorithm to estimate the missing values from the K nearest instances [16].

3.3.2 Feature Selection

Some features in the applied datasets are irrelevant based on specialists in the field of diabetes. So, optimal features must be selected. In this thesis, we have applied the SVM [74] method to select the relevant features. By applying feature selection to both global and local data sets, some features that contain duplicate or non-useful records for the classification of diabetes and its types were excluded.

3.3.3 Dataset Partitioning (K-Fold Cross-Validation)

After performing the KNN algorithm to fill the missing values and selecting the most optimum features of datasets, the data should be divided into groups to evaluate the classification model performance. The K-Fold cross-validation methodology will be used to evaluate the applied models in this thesis. Where the datasets samples are separated into K-Folds randomly to train and test the groups through K number of Iterations/Iteration s. At each Iteration , one of the K folds will represent the testing dataset and the rest will be used to train the model. This method is optimal in the case of limited data [42][43][44]. For example, if we apply this method to the DataPal of 314 samples and we apply the two-fold cross-validation method, each record of the 314 records will have a random label of 1 or 2 because we have two-folds to separate in this case, the training and testing groups will be 50% of the dataset. The generated folds are evaluated alternately where in the first Iteration the contents of the first fold for testing the data set and the rest for training and vice versa in the second Iteration. In this thesis, we evaluated the performance of the models using both

two-fold and four-fold cross-validation because the distribution of our limited data would be illogical if we applied ten-fold cross-validation with 10% of samples for testing.

3.4 Building Models Phase

In this section, we present the models that we have applied to the diagnosis of diabetes existence and their types. The SVM and MLP-BPNN are two of the strongest and most applied models in pattern recognition fields, so we applied them to prove the merit of our hybrid model PSO-FFNN by applying them to both data mentioned in chapter 2. The PSO is a fast convergence algorithm where it has a few parameters to adjust, short computational duration and a high probability to get the problem optimized solution in a non-iterative way.

3.4.1 Support Vector Machine

It is one of the most powerful supervised machine learning methods. SVM has been applied in various fields including the medical diagnosis [9][54]. Find the maximum geometrical margins between hyperplanes are the robustness of the SVM method, to determine the optimal hyperplane that separates the dataset classes in the middle of the maximum margin. Both binary and multi-class classification can be applied using the SVM. A Kernel trick is what allows the SVM to perform well in non-linear classification by mapping the inputs of the dataset on high-dimensional feature spaces [30].

To build a classification model of a high-dimensional dataset consisting of n features, it needs an n-dimensional feature space. The s samples vectors are separated by the optimal

hyperplane into classes. A maximum margin between the support vectors which are the closest points of both positive and negative classes to the optimal hyperplane has to be taken into consideration. The maximum margins mean the minimum risk of misclassifying of new data. The training data points presented in the form of:

$$d = [(x1, y1), (x2, y2), \ldots\ldots (xs, ys)] \quad 3.1$$

Where d is the dataset of x_s the data point that belongs to class y_s = -1/1. Where

s = number of data samples/support vectors.

Each X is represented by an n-dimensional vector mapped into a high-dimensional feature space. The two data points that have no other data points between them considered as the initial hyperplanes with maximum margins to be separated with an optimal hyperplane w. Where the way to maximize the margins m=2/||W|| is to minimize the norm of w vector by finding the value of (||W||^2)/2.

$$w.\ xs + b = 0 \quad 3.2$$

Where b is a scalar proved to help finding the maximum margins. While the threshold values of the two classes ys=-1/1 are:

$$w.\ xs + b = 1 \quad 3.3$$

for the positive class hyperplane and

$$w.\ xs + b = -1 \quad 3.4$$

for the negative class hyperplane as shown in figure 3.1 the boundaries between the two classes. The final decision is made based on the results of the optimization problem for both minimize the error of the classification using the regularization parameter and

maximizing the margin of hyperplanes according to the formula 3.5 after reformulating it using Lagrange Method [32]:

$$sgn(w.\ (x) + b) = sgn\left(\sum_{i=1}^{s} y_i \alpha_i K\left(x_i, x\right) + b\right) \qquad \textbf{3.5}$$

Each xi data point joins the class yi=1/-1 using the step function sgn. Where $K\left(x_i, x\right)$is the kernel function such as sigmoidal and Gaussian kernels andα_i is a Lagrange constant SVM that has a whole library called SVMLIB [31], that supports some SVM algorithms with different purposes including binary and multiclass classification. Each of Matlab, C#, C++, Python, and other platforms and programming environments support the SVMLIB. Where in this work we use the built-in tool of SVM in the Matlab environment to compare its result with our proposed model.

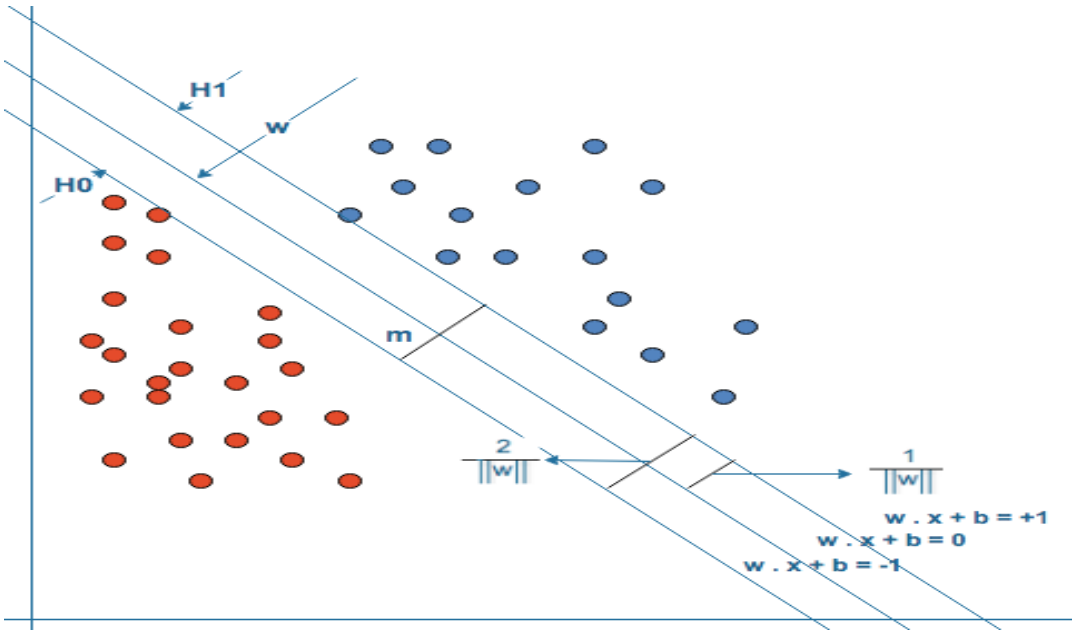


Figure 3.1: SVM data points separated by an optimal hyperplane.

3.4.2 Artificial Neural Networks

Artificial neural networks are one of the most widely used classification methods. Despite some disadvantages, the most important of which is its ability to deal with data sets in terms of increasing the number of properties and how to find the correct transfer function to classify datasets in the right way, which is not so easy. After collecting the data sets to be classified, they are divided into training and testing datasets so that each of them is applied individually.The optimal weight vectors are obtained by applying the largest set of the dataset which is usually 2/3 of the whole dataset, where the optimal weights will be used to test the trained network with the remaining dataset

After completing the network training, it is necessary to calculate the classification error to adjust the network parameters to obtain the lowest error and the highest accuracy. The network parameters are usually adjusted using various methods so that each of them has the advantage of distinguishing it from the others, the most important of which are applied in this thesis:

3.4.2.1 Scaled Conjugate Gradient Descent Algorithm “SCGD”:

The basic of the back-propagation learning algorithm is to train a multilayer feed-forward neural network iteratively using the gradient descent algorithm. The values of initial weights and biases are given to allow the designed neural network to start the training process. The network weights W's are tuned and the inputs X's are updated at the steepest descent direction of the function (i.e. when the performance of the function is reduced) [33]. But if we compare the convergence of the gradient descent algorithm with the

Conjugate Gradient Descent algorithm which is a fast convergence algorithm, it will be much less convergent [34]. The following formula shows how it is tuned the weights and biases using the basic gradient descent:

$$x_{n+1} = x_n - \alpha_n \Delta g_n \qquad 3.6$$

where x_{n+1} and $n = 1, 2,..$is the vector of weights and biases after updating them from the current vector of weights and biases $x_n$to the new values in back-propagation way as shown in figure 3.2, α_n is the learning rate and Δg_n is the current gradient of the function.

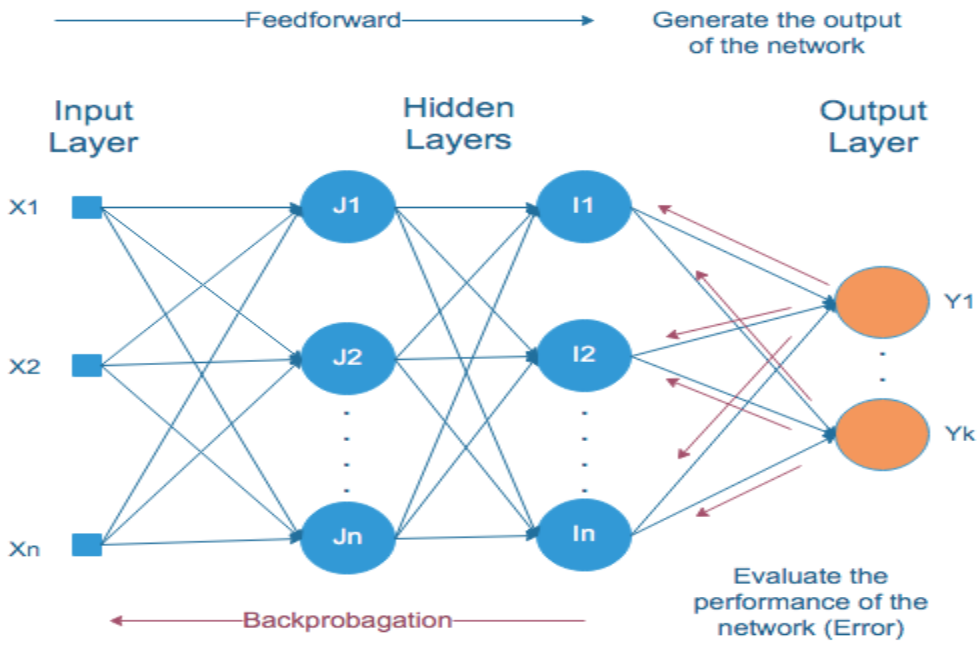


Figure 3.2: The basic structure of the Artificial Neural Network-MLP.

The Conjugate Gradient Descent algorithms characterized by the way in finding the minimum performance of the function. Where the algorithm based on the search by the

conjugate gradient direction to find out the step size to get the minimum function performance. Also, there is an algorithm called the Scaled Conjugate Gradient Descent algorithm which outperforms the Conjugate Gradient Descent algorithm by being inexpensive in time searching for the steepest descent to adjust the weights [34]. A lot of approaches can be used to estimate the step size including the Levenberg-Marquardt method. The algorithm as we mentioned earlier in this section begins from the direction of the steepest descent of the function (i.e. toward the negative gradient) as the following formula shows:

$$f(x) = min\, f(x) \rightarrow d_i = -g_k,\ k = 0 \qquad 3.6$$

Where g_k is the initially negative gradient vector and i=1,2,3,...The new values of weights and biases calculated according to the formula:

$$x_{i+1} = x_i + \alpha_i d_i \qquad 3.7$$

Where $\alpha_i$is the step size to the next steepest descent point, which is usually determined by the Levenberg-Marquardt algorithm [35]. x_{i+1} is the next point. The search vector d_i is calculated as follow if the function is non-linear:

$$f(x_i + \alpha_i d_i) = min\, f(x_i + \alpha_i d_i), \alpha_i \geq 0 \qquad 3.8$$

As shown in formula 3.9 the new search direction d_i is associated with the optimal point (i.e. the new steepest descent point) and the previous search direction d_{i-1} in addition to β_i

, the coefficient of the conjugate descent algorithm. The following formula is to calculate d_i when ≥ 0 :

$$d_i = - g_i + \beta_i d_{i-1} \qquad 3.9$$

Many ways have been proposed to calculate the coefficient of the conjugate descent algorithm β_i, [34] gives the value of the coefficient of the conjugate descent as:

$$\beta_i = \frac{|g_{i+1}|^2 - g_{i+1}\, g_i}{\mu_i} \qquad 3.10$$

, $$\mu_i = (d_i)^T g_i \qquad 3.11$$

The SCGD algorithm is the training algorithm that the Matlab tools used in pattern recognition to classify a lot of the high-dimensional features datasets in record time compared with the other training algorithms. So in this thesis, we have applied our datasets on these tools to validate the proposed hybrid model، by comparing the results of both models.

3.4.2.2 The Evolutionary Algorithms - Particle Swarm Optimization PSO:

Many evolutionary algorithms have been applied to optimize the ANNs to find the optimal weights of the network [36]. PSO, GA, and ACO are some of the applied evolutionary algorithms on the optimization problems of machine learning to train the ANNs and classify variant datasets. Also, evolutionary algorithms can be used as a feature selection method to extract valuable data from all the huge amounts of data in databases to improve

the resolution of prediction [37][38].In this thesis, we propose to use a hybrid approach of the Feed-forward neural network FFNN model with the PSO optimization algorithm to improve the diagnosis of Diabetes Mellitus and its types. The PSO algorithm is supposed to be the training method of the FFNN model to adjust the weights of the network. The inspiration for evolutionary algorithms, in general, is the process of evolution of nature, the process of seeking solutions evolutionarily.

The PSO algorithm Inspired by the social behavior of the movement of birds flocks and how each particle plays a role in finding a possible solution to the optimization problem [39][40]. The PSO algorithm particles fly over a search space with its memory to share the rest of the particles with potential solutions, where each particle has a velocity to fly over the search space [41]. If we compare the implementation of PSO algorithm based on particle's velocity with another evolutionary algorithm such as GA, it is easier to implement PSO than GA, since GA has a lot of factors to take into consideration when implementing an optimization problem such as mutation and crossover which is time-consuming.

The PSO algorithm consists of $n, n = [1, 2,\dots\ n]$ particles within an S- dimensional search space, $S = [1, 2,\dots\ S]$. Each particle x_i, $i \in n$ has an initial position x_{il} $l \in S$, an initial velocity v_i and a personal best position $P_{best,i}$ chosen based on the best value of the fitness function within a search space. The global best position $G_{best,i}$ is the position of the particle with the best fitness value among all particles.

Solving an optimization problem means that a set of particles fly over a search space to

evaluate the personal and global possible solutions. Where the velocity and the position of each particle updated according to the best global and personal best fitness values. According to the model used in our thesis, the personal best position $P_{best,i}$ is calculated according to the following formula:

$$P_{best,i} = \left\{ P_{best,i} \, , \; if \quad f(x_i) > P_{best,i} \right\} else \left\{ x_{il} \, , if \quad f(x_i) \leq P_{best,i} \right\} \qquad 3.12$$

Then to calculate the global best position $G_{best,i}$ the following formula used:

$$G_{best,i} = \left\{ min(P_{best,i}) \, , \; where \; i \in n = [1, 2, \ldots \; n] \right\} \qquad 3.13$$

and the following one used to calculate the velocity of the particles:

$$v_{i,new} = w \, v_i + c_1 r_1 (P_{best,i} - x_i) + c_2 r_2 (G_{best,i} - x_i) \qquad 3.14$$

Where w is the inertia weight and $(w \, v_i)$ is the inertia component that helps the particles to move in the correct direction. The higher the value of the inertia component the higher the chance to explore the whole search space. Therefore, the inertia component value gives the poise between the effect of the individual component in exploiting the search space and the effect of the social component in exploring the whole search space. The value of the inertia component lies between $[0.8 \, , \; 1.2]$ [39], in our model we have supposed that it can be calculated as $w = rand * 0.35$. r_1 and r_2 are random values between $[0, 1]$, $c1$and $c2$ are coefficients of the individual learning component (i.e. $c_1 r_1 (P_{best,i} - x_i)$) and social learning component (i.e. $c_2 r_2 (G_{best,i} - x_i)$) where c_1 and c_2 usually lies between 0 and

2 [39].

After calculating the velocity of particles the new position of them will be determined by the equation:

$$x_{i,new} = x_i + v_{i,new} \quad 3.15$$

As for how this evolutionary algorithm-PSO trains neural networks, each particle represents a set of weights (i.e. the neural feed-forward network weights) values which are the dimension of the particle. The n particles fly over the weights search space to minimize the classification error (i.e.maximum fitness) to update the local and global best positions.

The training process of the PSO-FFNN is shown in figure 3.3. A set of particles with random positions (i.e. initial random weights) is initialized at the first step. An FFNN is trained in the second step using the initialized positions of the PSO algorithm. Then, the learning error is calculated to evaluate the performance of the training process in the third step. Where in the fourth step, the velocity and positions of the particles updated according to the new best local and global positions. Hence the new positions of the particles are the new weights that will be used to train the PSO-FFNN again, and so on until the minimum classification error is reached.

The new position (i.e. new weights) of each particle is updated by adding the new velocity to the old position value. Where the best global position with minimum classification error is the solution to our optimization problem.

The output of each node of the trained FFNN is calculated according to the following formula:

$$y = \sum_{i=0}^{j} (w_i x_i + b_i),\ i \in [0, 1, \ldots j]\ and\ j\ is\ \#\ of\ node\ inputs \quad 3.16$$

The ANN input layer has no function to apply to the network inputs, where the inputs forward directly to the hidden layer. But the hidden and the output layers have to apply one or more functions to pass their neurons outputs (i.e. the output of equation 3.16). A function called activation function used to calculate the output of the hidden and output layers. A lot of activation functions can be used to train the ANN models [63]. The activation functions are chosen according to both the problem that needs to be solved and the model training algorithm. In this thesis, we are solving a pattern recognition problem to classify medical datasets. The Sigmoidal and Binary Threshold or step activation functions are two of the most commonly applied activation functions with pattern recognition problems [63].

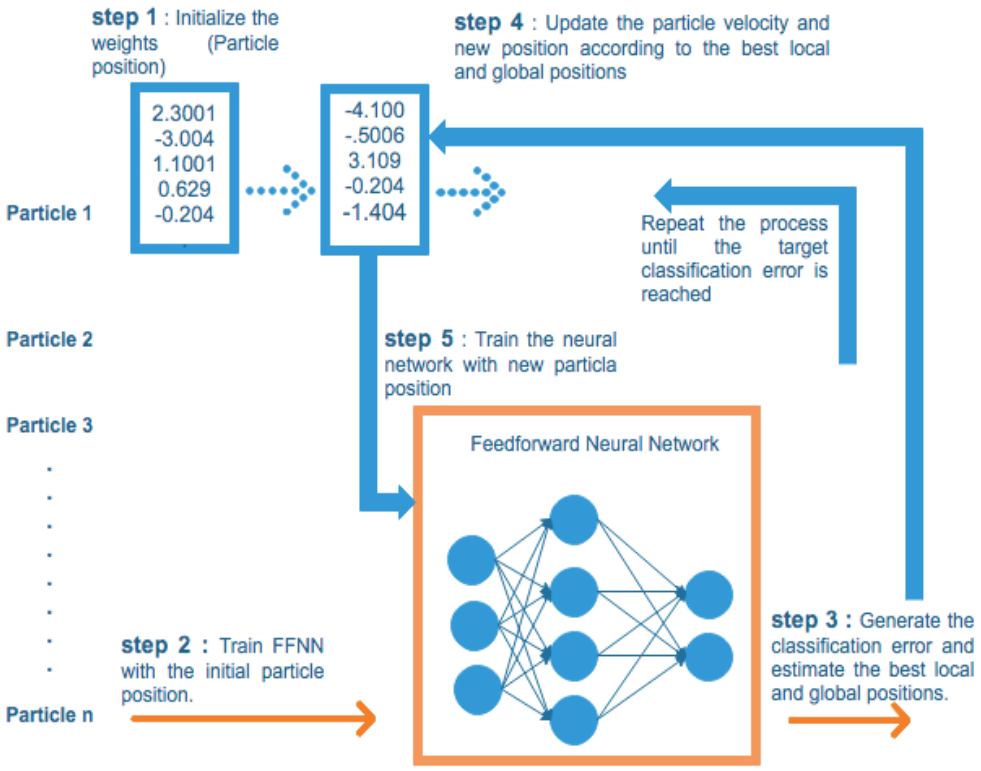


Figure 3.3: The process of training PSO-FFNN.

In this thesis, the Sigmoid function has been used to calculate the output of the FFNN hidden layer to estimate the probability of disease occurs as shown in equation 3.18. While the Binary Threshold or step function has been used to predict the final output of the FFNN as shown in equation 3.17.

- Binary Threshold / Step Activation Function:

$$f(y) = \begin{cases} 1 & \text{if, } y \geqslant 0 \\ 0 & \text{if, } y < 0 \end{cases} \qquad 3.17$$

- Sigmoidal Activation Function:

$$Y = 1/(1 + exp^{-x}) \qquad 3.18$$

3.4.3 The General Method Procedures

The whole procedures that used to predict and diagnose both DataPal and PIMA datasets are shown in **Figure 3.4** and **Algorithm 1**.

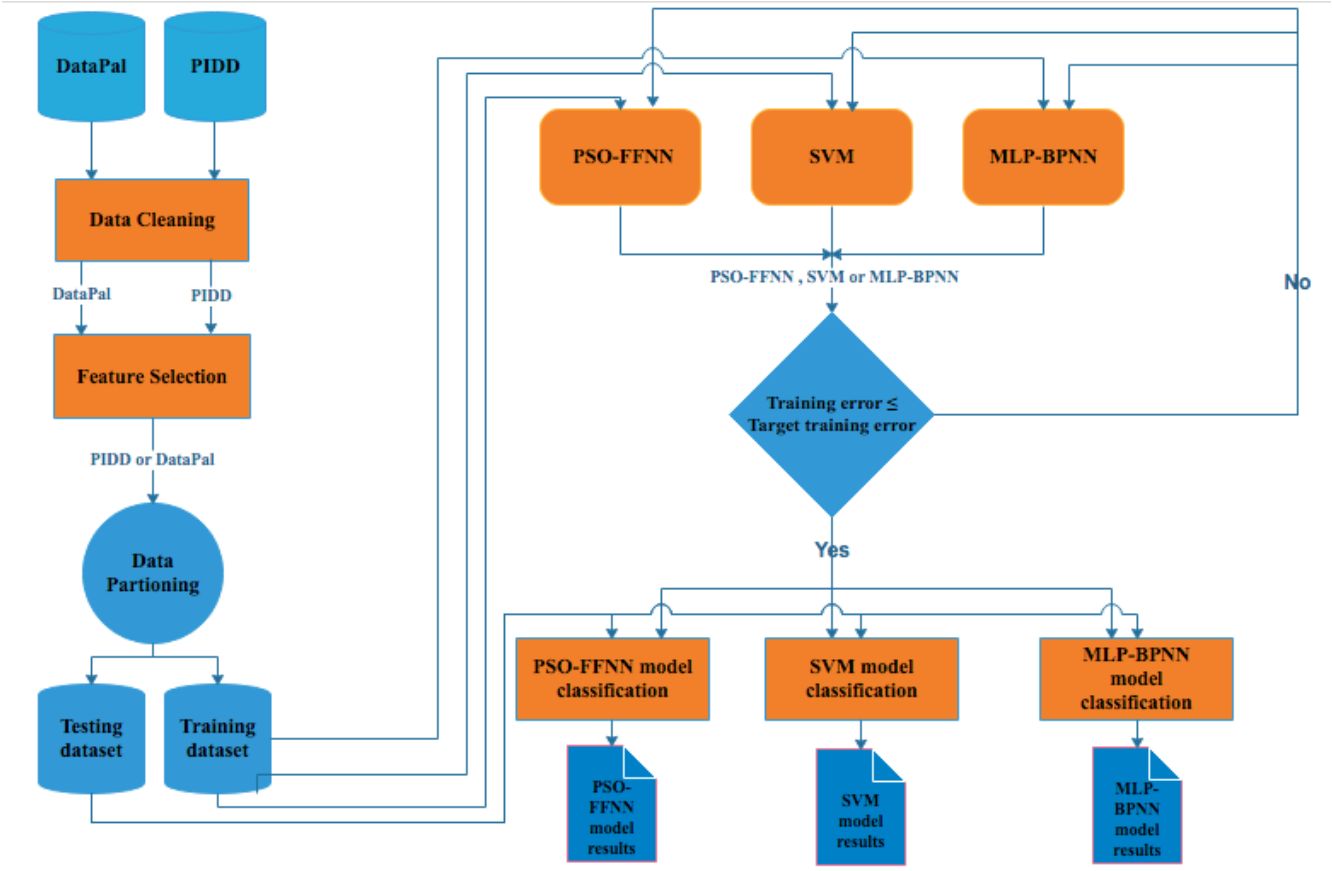


Figure 3.4: The general procedures of the PSO-FFNN, SVM and MLP-BPNN models.

Algorithm 1: The general procedure of the models used in this thesis

Input: Dataset-input, Dataset-target, the number of Folds F, number of hidden layers neurons n (for MLP-BPNN and PSO-FFNN models), particle position x (for PSO-FFNN model);
Output: Training and Testing Results;

Data Preprocessing:
*// **Applying the Data Cleaning/Filling method on the dataset fields***
*for sample **in the** dataset, **do***
*for variable = **1 to size**(sample), **do***
if sample [variable]=0
Apply KNN

```
            model_KNN ← Build KNNModel(sample)
            Estimate sample [variable]
            Fill sample [variable]
    // Feature selection
    f = # features
    train = train sub-dataset
    test = test sub-dataset
    array a[5],sorted[f]
        for i = 1 to f , do
            model_svm ← Build SVMModel(train)
            model_svm Results= Test SVMModel(model_svm, test)
             sorted[i] =sortpriority (model_svm Results)
               for j = 1 to 5 , do
                a[j]= select (sorted)
// Cross-Validation, K-Fold Cross-Validation
        K=# of Folds
        array targLabel[size(Dataset-target)]
        for i= 1 to size(Dataset-target) , do
          if Dataset − target [i]=0
           targLabel[i]="T1"
          else
         targLabel[i]="T2"
        endif
      array indices[size(Dataset-target)]
     size=size(Dataset-target)/F

      for i= 1 to size(Dataset-target) , do
         indices[i] ← crossvalind("K-fold",DataPalSpecies,K)
      for x= 1 to size(Dataset-target) , do
          for j= 1 to F , do
            array Fold j [size]
            if indices[x]=j
                for n= 1 to F , do
                  Fold j[size] ← indices[x]
            endif
    return Fold , errorRate
// Distribute the samples into training, testing sub-datasets and determine some general parameters
```

// R= # of Iterations
R=F
array tr [size]
n=# hidden neurons
x= particle position/ neural network weights
net-f = feedforward neural network
for** i= **1 to R , do
TrainD $\leftarrow dataset - \{ Fold\ j[size] \}$
TestD $\leftarrow dataset[Fold\ j[size]]$
// PSO-FFNN model
$model_{PSOFFNN} \leftarrow BuildPSOFFNNModel(TrainD,\ n, x,\ net - f)$
$testing\ results = TestPSOFFNN(model_{PSOFFNN}, TestD)$
// Linear SVM model
$model_{SVM} \leftarrow TrainSVMModel(TrainD)$
$testing\ results = TestSVM(model_{SVM}, TestD)$

// MLP-BPNN model
$model_{MLPBPNN} \leftarrow TrainMLPBPNNModel(TrainD, n)$
$testing\ results = TestMLPBPNN(model_{MLP-BPNN}, TestD)$

***return** training results, testingResults.*

3.5 Performance Metrics

The performance of Pattern recognition models is evaluated using different measurements [45][46]. Confusion matrix, Classification Accuracy, Classification error or Misclassification Rate, Precision, Recall, Specificity, G-mean, and Area Under Curve ”AUC” are some of the main measurements used to evaluate the performance of different pattern recognition models. Some important measures have to be clarified before start evaluating the diabetes types classification models such as:

True Positive TP: Number of records that are in the positive class and it is classified as a positive class.

False Positive FP: Number of records that are not in the positive class and it is classified as a positive class.

True Negative TN: Number of records that are in the negative class and it is classified as a negative class.

False Negative FN: Number of records that are not in the negative class and it is classified as a negative class.

Confusion Matrix: As shown in Table 3.1, the confusion matrix is a two-dimensional matrix, where the rows represent the actual/target classes value and the columns represent the predicted class values. Table 3.2 and Table 3.3 represent the confusion matrices of DataPal and PIDD datasets respectively.

Table 3.1: General Confusion matrix.

	Classified Positive	**Classified Negative**
Positive	TP	FP
Negative	TN	FN

Table 3.2: DataPal Confusion matrix.

	Classified T1DM	**Classified T2DM**
T1DM	TP	FP
T2DM	TN	FN

Table 3.3: PIDD Confusion matrix.

	Classified Diabetic	**Classified Non-Diabetic**
Diabetic	TP	FP
Non-Diabetic	TN	FN

The **Accuracy** of pattern recognition models is calculated according to the following formula:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} * 100\ \% \qquad 3.4.1$$

where the **Misclassification Rate** or **Classification error** (1-Accuracy) is calculated as the following formula:

$$Classification\ error = \frac{FP+FN}{TP+FP+TN+FN} * 100\ \% \qquad 3.4.2$$

Precision: Is the percentage of samples have been predicted as positive class correctly to all samples that have been predicted as a positive class as shown in the following formula:

$$Precision = \frac{TP}{TP+FP} * 100\ \% \qquad 3.4.3$$

Sensitivity or Recall: Is the percentage of samples has been predicted as positive class correctly to all samples that have been predicted as positive class correctly and incorrectly as a negative class. Is the measurement used to evaluate the accuracy of positive cases, as the following equation shows:

$$Sensitivity = \frac{TP}{TP+FN} * 100\ \% \qquad 3.4.4$$

Specificity: Is the percentage of samples predicted as negative class correctly to all samples that have been predicted as negative class correctly and incorrectly as a positive class. Is the

measurement used to evaluate the accuracy of negative cases, as the following equation shows:

$$Specificity = \frac{TN}{TN+FP} * 100\ \% \qquad 3.4.5$$

Geometric Mean G-mean: It is a measurement used to evaluate the classification performance in imbalanced datasets. The smallest value of G-mean means there are deficiencies in the performance of the classification of the class that has the lowest number of samples even if the other class is classified correctly.

$$G - mean = (sensitivity * specificity)^{\frac{1}{2}} \qquad 3.4.6$$

F-measure or F-score: It is the harmonic mean of precision and sensitivity [47]. It is making a balance between the sensitivity and the precision to measure the testing accuracy. When F-measure value =0 that means that both precision and sensitivity have bad values, while when the F-measure value =1 precision and sensitivity have the best values.The following equation shows how F-measure is calculated:

$$F - measuring = \frac{2*precision*recall}{precision + recall} \qquad 3.4.6$$

Area Under Curve AUC: The Area under the ROC curves that illustrate the tradeoff between the Recall/sensitivity and the specificity values of the prediction model [46]. The area calculated under the ROC curve between the true-positive rate TPR (i.e.sensitivity) and the false-positive rate FPR (i.e.1-specificity), it appears the ability to classify classes.

3.6 Summary

In this chapter, we have presented the procedures that we have followed in this thesis. Firstly in section 3.1, we have presented the data selection considerations we followed in the DataPal collection and the selection of PIDD datasets. Secondly, in section 3.2 we have presented the pre-processing steps that we took to prepare the datasets before applying them to different models of machine learning. The main three applied models that we have used in this thesis were introduced in section 3.3, including the pseudo code and the flow chart that illustrate the procedures of the PSO-FFNN model. Finally, different performance evaluation metrics were presented in section 3.4 that will be used to measure the efficiency of models applied in the prediction of diabetes.

Chapter 4

Experiments and Results

4 Experiments and Results

The previous chapter described the models applied in this thesis to the hybrid PSO-FFNN, the SVM, and the MLP-BPNN [4]. Before that, the methods used to prepare valuable datasets without any missing values or non-optimal features were presented. and finally, the used metrics to evaluate the performance of the models were presented in the last section of the previous chapter. In this chapter, the PSO-FFNN model applied to both DataPal and PIDD datasets to compare it with the different Matlab models MLP-BPNN [4], SVM. In addition to the three main models mentioned before, we have also applied a set of machine learning algorithms that have been applied in various previous studies to validate the proposed PSO-FFNN model. Support Vector Machine "SVM" with different kernel functions (i.e. Linear, Radial Base [54], Polynomial, and Gaussian) functions, k nearest neighbor "KNN", Discriminant Analysis Classifier "DA", Naive Bayes "NB", decision tree "DT" and Random Forest "RF" algorithms [18][15][8][50][52][53][55][56] , were the algorithms have been used to prove the ability of the PSO-FFNN model in predicting diabetes. The main objectives of this chapter are, first to define if it is possible to apply the PSO-FFNN model on the "PIDD" and the "DataPal" datasets. The second objective is to adjust the parameters with their optimal values. Then to show whether the PSO-FFNN model can outperform the other models in classifying the diabetes mellitus and its types or not. So the experiments in this thesis were applied using two different datasets, the local "DataPal" dataset collected from the Palestinian Institute for diseases of diabetes and the global dataset PIDD. The hybrid PSO-FFNN model, the built-in Matlab SVM tool (i.e. one

of the classification learner tools) and MLP-BPNN model were used to classify the mentioned datasets. Before applying the datasets on the presented models, both datasets were preprocessed as it presents in the next sub-section. Before starting the experiments the datasets were applied to the SVM feature selection method and then parting the datasets into training and testing groups using K-Fold cross-validation that all will be discussed in the next section.

4.1 Implementation Platform

The platform used in this thesis is an off-chip platform that means that the applied ML methods (i.e. PSO-FFNN, SVM and MLP) are executed using a computer without any special ML chips or related devices. Software was used only, the Matlab R2019a tools used to apply the models. The special-purpose software such as C# and Java machine learning programs were suffering from the lack of flexibility in control of the programs compared to the general-purpose software [48]. With the new Matlab versions we have more control over the models we develop. Matlab programs and tools are more flexible and faster to build with additional tools rather than building everything from scratch.

4.2 System Parameters

Each of the SVM and MLP-BPNN models were applied directly using the built-in tools in Matlab. As we have mentioned in chapter 2, to train a feedforward neural network we have to adjust the number of hidden neurons in hidden layers of the network. In this thesis, the hidden neurons were adjusted by trial and error to gain the best pattern recognition results.

Both datasets DataPal and PIDD were applied using the Linear SVM classification learning tool and MLP-BPNN tool in Matlab. The classification accuracy, misclassification rate, sensitivity, specificity, precision, Geometric Mean and F-Measure metrics were used to evaluate the three models' performance. These metrics are used with the models that typically apply the imbalanced datasets in their evaluations as we have mentioned in section 3.4.

Appendix A will present the results of the PSO parameter selections. The optimal PSO parameters were selected according to the overall accuracy produced by applying different parameter values as shown in tables A.1, A.2, A.3, and A.4. Where the optimal values of the PSO algorithm parameters should be selected. Sections 4.4 and 4.5 will show the experimental results using the optimal values of the PSO parameters. By applying both datasets used in this thesis on the three models including the PSO-FFNN model. And therefore the following are the optimal values of the PSO parameters used in the proposed PSO-FFNN model in this thesis:

- **The coefficients of the individual learning component** $c1$**and**$c2$: are the components that allow particles to exploit and explore the search space of the PSO to search for a solution. The optimal values for both components were found to be 2 by increasing their values as shown in tables A.1 and A.2.

- The number of Population or Particles: The number of particles has been adjusted to be 30 particles, the smallest number of particles needed to search the search space for a solution to keep the model within a stable generalization status as shown in tables A.2 and

A.4. The small number of particles make the model get rid of its slow state if a large number of molecules are used [49].

- The inertia weight component: The component that helps the particles to move with the correct direction. The higher the value of the inertia component the higher the chance to explore the whole search space. Therefore, the inertia component value gives the balance between the effect of the individual component in exploiting the search space and the effect of the social component in exploring the whole search space. The value of the inertia component lies between $[0.8: 1.2]$ [39], in our model we have supposed that it can be calculated as $= rand * 0.35$.

- The initial velocity v: The particles start searching the search space with an initial velocity v. in our model we have supposed that the initial velocity v can be calculated as $v = 0.15 * x_0$, where x_0 is the initial position of particles.

- The search space boundaries (i.e the Lower boundary and the upper boundary) LB **and** UB: search spaces should have boundaries to prevent the particles from boundary violations and to stay within the determined boundaries. In this thesis, each of the lower and upper boundaries have been adjusted to be -1.5 and 1.5 respectively. They were selected based on the results obtained by applying different values And choose the optimal among them, as shown in Tables A.5 and A.6.

- The number of input layer neurons in the FFNN: It is flexible and is equal to the number of dataset features.

- Activation Function: As we have mentioned in chapter 3 a lot of activation functions to get the outputs of the hidden and the output layers in the FFNN. In this thesis, we have used the Sigmoidal activation function to calculate the output of the hidden layer neurons and the Binary Step function (Threshold function) to calculate the final output of the output layer. The most successful and used activation functions in multi-class classification.

The following table illustrates the optimal parameter values that have been used in this thesis to train the proposed PSO-FFNN model:

Table 4.1: The PSO-FFNN model optimal parameters.

Parameter	**The optimal value**
$c1$	2
$c2$	2
Population number	30
w	$rand * 0.35$
v	$0.15 * x_0$
LB	-1.5
UB	1.5
Number of input layer neurons	According to the number of input features
Activation function	- The Sigmoidal function for the hidden layer. - The Binary Step function for the output layer.

4.3 DataPal Dataset Experiments

As we have presented before, the main aim of this thesis is to diagnose diabetes mellitus and to classify its types. So, diabetes mellitus patients are aware of the level of seriousness of their illness especially the patients with T2DM. In this chapter, we compare and discuss the results of our experiments using the main three applied models in this thesis (i.e. our proposed PSO-FFNN, Linear-SVM, MLP-BPNN) models. In addition to these models, we have also applied a set of machine learning algorithms from previous studies to prove the ability of our improved model PSO-FFNN to predict diabetes. Support Vector Machine "SVM" with different kernel functions (i.e.Linear, Radial Base [54], Polynomial, and Gaussian) functions, k nearest neighbor "KNN", Discriminant Analysis Classifier "DA", Naive Bayes "NB", decision tree "DT" and Random Forest "RF" algorithms [18][15][8][50][52][53][55][56], were the algorithms have been used to prove the ability of the PSO-FFNN model in predicting diabetes. The best parameters values of our proposed model that made it outperform the other models are proposed as the optimal parameters for building a diabetes prediction model As Appendix A presents. All the results of our experiments are presented in the following subsections:

4.3.1 Linear SVM and MLP-BPNN Experiments on DataPal Dataset

The MLP-BPNN and Linear-SVM models were applied to the DataPal in this section. There are some parameters to adjust when applying MLP-BPNN. The number of neurons in the hidden layer where we apply a single hidden layer in this model. The activation

functions used in the hidden and the output layers are the sigmoidal and binary-step activation functions, respectively as presented in section 4.3. The threshold measure used to evaluate the performance of the MLP-BPNN model is the number of hidden layer neurons. The Linear-SVM model as we have mentioned in chapter 3 is one of the most applied classifiers in medical diagnoses. The kernel function used is linear. The model is validated using both two-fold and four-fold cross-validation methods. The following tests show the results of applying the models mentioned above:

Test 1: The results of applying the two-fold cross-validation method to validate the MLP-BPNN performance are illustrated in this test. Adjusting hidden neurons is important for improving MLP-BPNN network performance. Tables 4.2 and 4.3 illustrate the results of applying the MLP-BPNN model with a different number of hidden neurons. The number of hidden neurons is adjusted by starting with n = 2 and gradually until n = 10. From the following tables, we can notice that both Iterations of this test are somewhat close to each other in the overall accuracy, for both training and testing accuracy. T2DM predictions with both Iterations are good. It is clear that Iteration 1 stably predicts T2DM, but Iteration 2 has perfect T2DM predictions when n=4 by up to 100%. looking at both Iterations results in Iteration 1 and Iteration 2 have almost the same behavior to some extent. Both Iteration 1 and Iteration 2 have overall accuracy up to 97.5 % and specificity up to 88.88% when n=8 and n=10 for Iteration 1 and Iteration 2, respectively. The G-mean values in both Iterations are close as well to each other.

Test 1 results show that the MLP-BPNN was able to classify T1DM and T2DM with an overall accuracy of up to 97.5%. on the one hand, It was able to predict T2DM perfectly when n= 4 in Iteration 2. On the other hand, the Test 1 model has less ability to classify T1DM compared to T2DM, where T2DM is less dangerous than T1DM. The T1DM misdiagnosis may cause death. Figures 4.1 and 4.2 show the ROC curve where the tradeoff between TPR and FPR values is illustrated. It can be noticed from Table 4.4 that Iteration 2 has a stable behavior in the ROC curve while Iteration 1 has curvy and unstable behavior. By applying Iteration 2 an AUC value by up to 0.9185 is obtained. From the above, we can conclude that the MLP-BPNN model was not able to accurately classify the most serious types of diabetes.

Table 4.2: Test1 two-fold cross-validation in Iteration 1 results.

	Two-Fold Cross-Validation Iteration 1				
	n=2	n=4	n=6	n5=8	n=10
Training Accuracy	96.2%	96.8%	96.8%	97.5%	96.8%
Testing Accuracy	95.6%	94.3%	93.7%	94.9%	95.6%
TP	129	129	129	129	129
FP	5	4	4	3	4
TN	22	23	23	24	23
FN	1	1	1	1	1
Error rate	0.038	0.032	0.032	0.032	0.032
Precision	96.27%	96.99%	96.99%	96.27%	96.99%
recall	99.23%	99.23%	99.23%	99.23%	99.23%
Specificity	81.48%	85.15%	85.18%	88.88%	85.18%
Gmean	89.9%	91.9%	91.9%	93.9%	91.9%
TPR	0.9923	0.9923	0.9923	0.9923	0.9923
FPR	0.1851	0.1481	0.1481	0.1111	0.1481
Area	0.0918	-0.0367	0	-0.0367	0.0367

Table 4.3: Test1 two-fold cross-validation in Iteration 2 results.

	Two-Fold Cross-Validation Iteration 2				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	94.3%	96.8%	96.8%	96.2%	97.5%
Testing Accuracy	92.4	93%	96.2%	94.9%	93.7%
TP	122	130	129	129	129
FP	3	5	4	5	3
TN	24	22	23	22	24
FN	8	0	1	1	1
Error rate	0.057	0.032	0.032	0.034	0.025
Precision	97.6%	96.29%	96.99%	96.26%	97.72%
recall	93.84%	100%	99.23%	99.23%	99.23%
Specificity	88.88%	81.48%	85.18%	81.48%	88.88%
Gmean	91.32%	90.26%	91.93%	89.91%	93.91%
TPR	0.9384	1	0.9923	0.9923	0.9923
FPR	0.1111	0.1851	0.1481	0.1851	0.1111
Area	0.0521	0.0717	-0.0369	0.0367	-0.0714

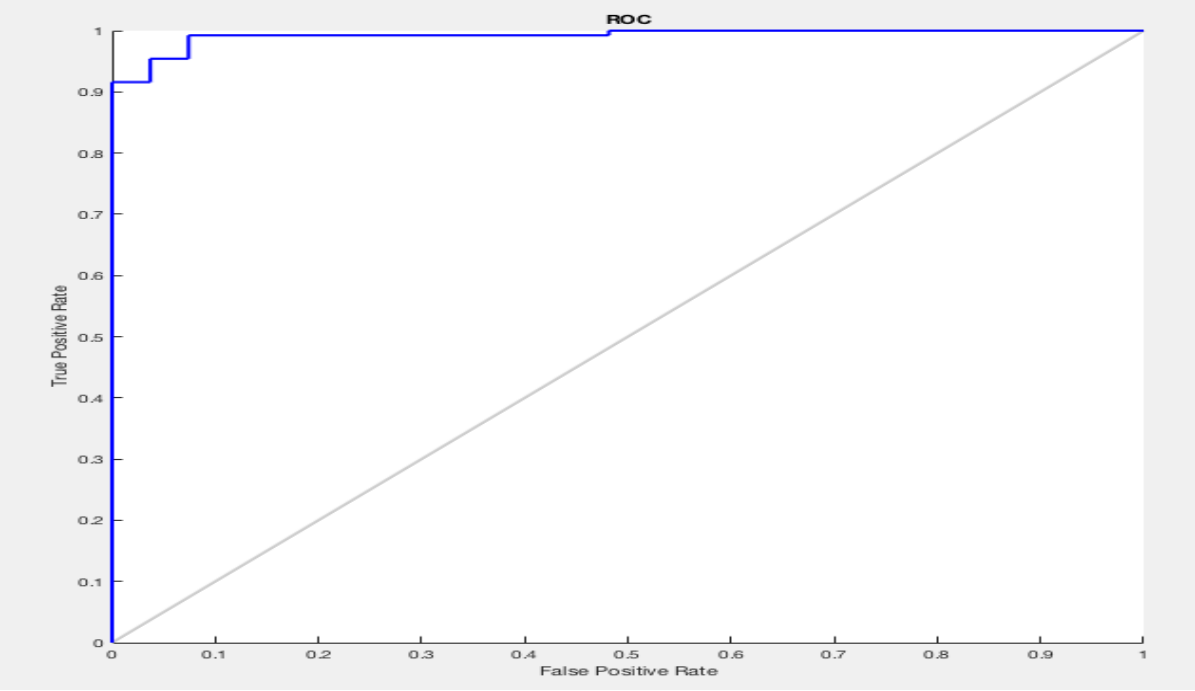


Figure 4.1: The ROC-AUC chart of Test1 two-fold cross-validation in Iteration 1 results.

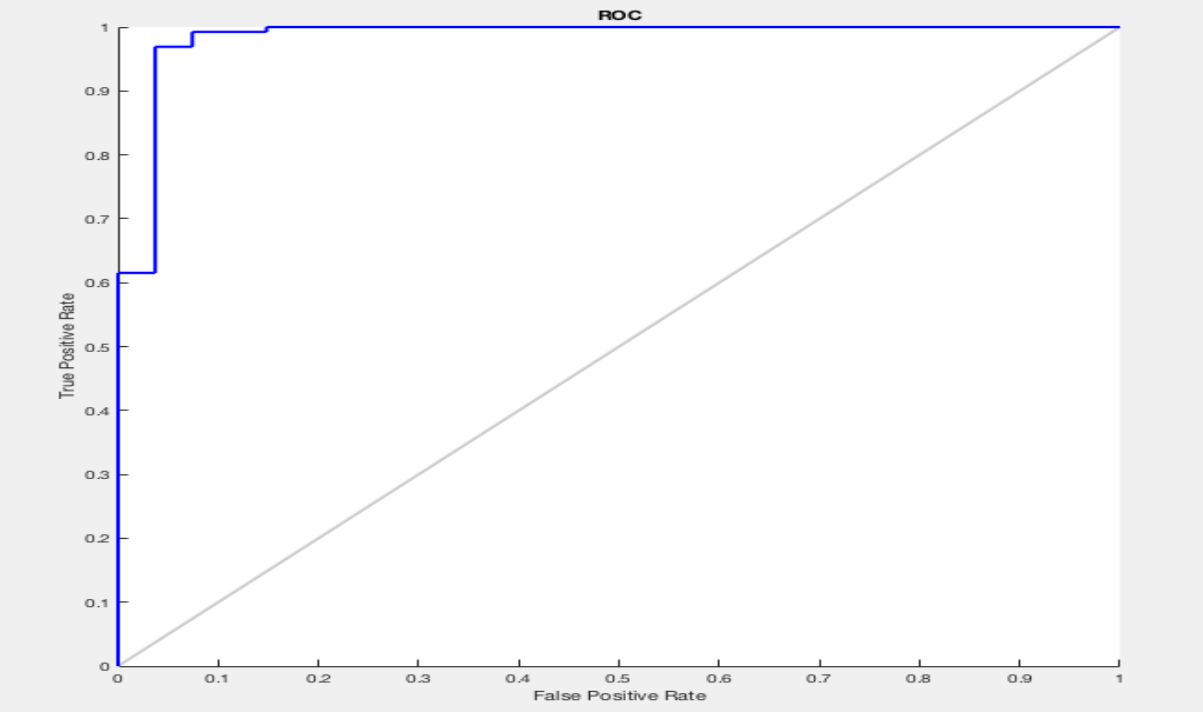


Figure 4.2: The ROC-AUC chart of Test1 two-fold cross-validation in Iteration 2 results.

Table 4.4: Test 1 AUC values.

Iteration	AUC
1	0.9037
2	0.9183

Test 2: In this test, a four-fold cross-validation method has applied to validate the MLP-BPNN model. Four Iterations applied to the MLP-BPNN model used in this thesis. Tables 4.5, 4.6, 4.7 and 4.8 show the results of the four Iterations. The best overall accuracy value obtained at Iteration 1 when n=10. All Iterations got good T2DM predictions. Iteration 1 was the better model in T1DM predictions and the G-mean value compared to others. To distinguish the best model all the above results must be taken into account in addition to the AUC value. Figures 4.3, 4.4, 4.5 and 4.6 illustrate the results of applying the

ROC curve for each of Iteration 1, Iteration 2, Iteration 3 and Iteration 4, respectively. On the one hand, it can be noticed that Iteration 1 and Iteration 4 ROC curves behavior are the closest to each other. But on the other hand, Iteration 1 is more efficient in T1DM and T2DM predictions than others. From Table 4.9, Iteration 1 gives the highest AUC value as well, by up to 0.9348.

From the above, we can conclude that the results of applying four-fold cross-validation to the MLP-BPNN model were better than the results of Test 1. The Iteration in Test 2 was able to predict both T1DM and T2DM better than Test 1. But the results in each of Iteration 2, Iteration 3, and Iteration 4 were worse than the third test, especially in predicting T1DM.

Table 4.5: Test2 four-fold cross-validation in Iteration 1 results.

	Four-Fold Cross-Validation Iteration 1				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	96.2%	97.4%	97.9%	97.9%	98.3%
Testing Accuracy	89.9%	92.4%	92.4%	92.4%	92.4%
TP	193	194	193	193	194
FP	7	5	3	3	3
TN	33	35	37	37	37
FN	2	1	2	2	1
Error rate	0.038	0.026	0.021	0.021	0.017
Precision	96.5%	97.48%	98.46%	98.46%	98.47%
recall	98.97%	99.48%	98.97%	98.97%	99.48%
Specificity	82.5%	87.5%	92.5%	92.5%	92.5%
Gmean	90.36%	93.29%	95.68%	95.68%	95.92%
TPR	0.9897	0.9948	0.9897	0.9897	0.9948
FPR	0.175	0.125	0.075	0.075	0.075
Area	0.0866	-0.0496	-0.0496	0	0

Table 4.6: Test2 four-fold cross-validation in Iteration 2 results.

	Four-Fold Cross-Validation Iteration 2				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	96.2%	96.2%	95.8%	95.8%	97%
Testing Accuracy	97.4%	96.2%	96.2%	93.6%	96.2%
TP	192	192	194	192	193
FP	6	6	9	7	5
TN	35	35	32	34	36
FN	3	3	1	3	2
Error rate	0.038	0.038	0.042	0.042	0.03
Precision	96.96%	96.96%	95.56%	96.48%	97.47%
recall	98.46%	98.46%	99.48%	98.46%	98.97%
Specificity	85.36%	85.36%	78.04%	82.92%	87.80%
Gmean	96.8%	91.68%	88.11%	90.36%	93.22%
TPR	0.9846	0.9846	0.9948	0.9846	0.9897
FPR	0.1463	0.1463	0.2195	0.1707	0.1219
Area	0.0720	0	0.0724	-0.0483	-0.0482

Table 4.7: Test2 four-fold cross-validation in Iteration 3 results.

	Four-Fold Cross-Validation Iteration 3				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	95.7%	93.6%	97%	96.6%	97%
Testing Accuracy	96.2%	92.4%	96.2%	94.9%	96.2%
TP	191	193	192	192	193
FP	6	13	4	5	5
TN	34	27	36	35	35
FN	4	2	3	3	2
Error rate	4.3%	6.4%	3%	3.4%	3%
Precision	96.95%	93.68%	97.95%	97.46%	97.47%
recall	97.94%	98.97%	98.46%	98.46%	98.97%
Specificity	85%	67.5%	90%	87.5%	87.5%
Gmean	91.24%	81.73%	94.13%	92.81%	93.06%
TPR	0.9794	0.9897	0.9846	0.9846	0.9897
FPR	0.15	0.325	0.1	0.125	0.125
Area	0.0735	0.1723	-0.2221	0.0246	0

Table 4.8: Test2 four-fold cross-validation in Iteration 4 results.

	Four-Fold Cross-Validation Iteration 4				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	96.6%	96.2%	97%	97%	94.5%
Testing Accuracy	97.4%	96.2%	96.2%	97.4%	94.9%
TP	192	191	191	192	192
FP	5	5	3	4	10
TN	36	36	38	37	31
FN	3	4	4	3	3
Error rate	0.034	0.038	0.032	0.025	0.032
Precision	97.46%	97.44%	98.45%	97.95%	95.04%
recall	98.46%	97.94%	97.94%	98.46%	98.46%
Specificity	87.80%	87.80%	92.68%	90.24%	75.60%
Gmean	92.98%	92.73%	95.27%	94.26%	86.28%
TPR	0.9846	0.9794	0.9794	0.9846	0.9846
FPR	0.1219	0.1219	0.0731	0.0975	0.2439
Area	0.0600	0	-0.0478	0.0240	0.1441

Table 4.9: Test 2 AUC values.

Iteration	**AUC**
1	0.9348
2	0.9216
3	0.9188
4	0.9306

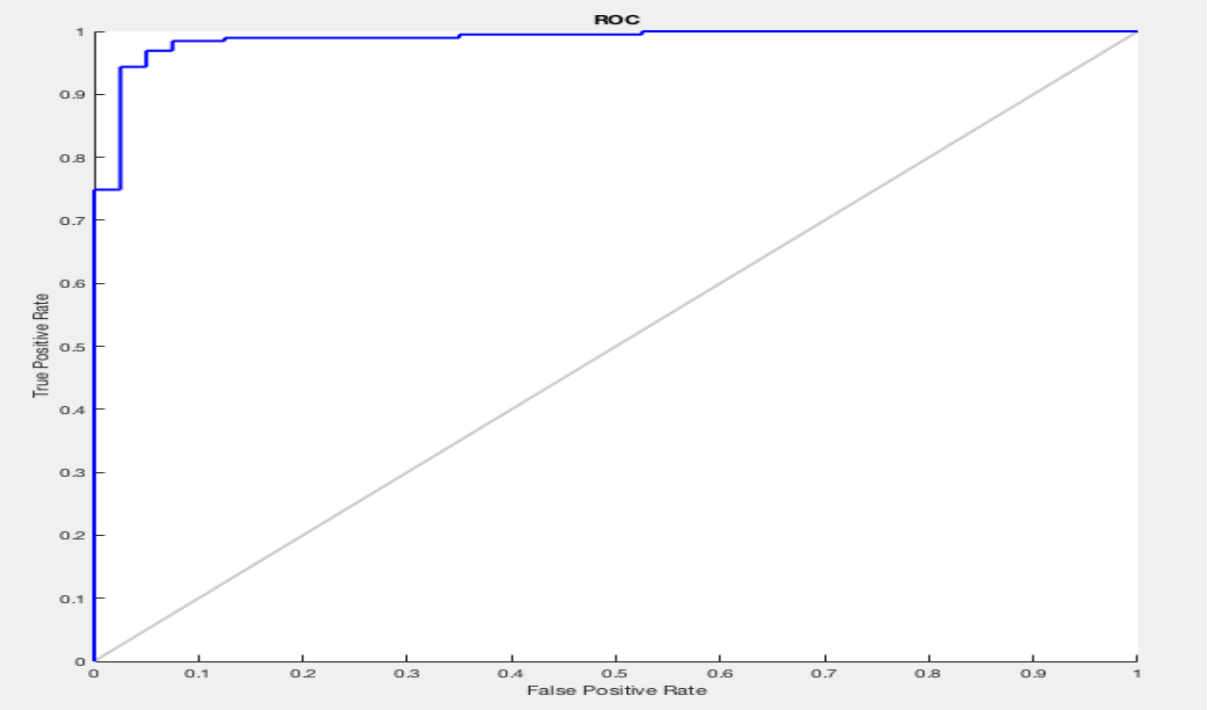


Figure 4.3: The ROC-AUC chart of Test2 four-fold cross-validation in Iteration 1 results.

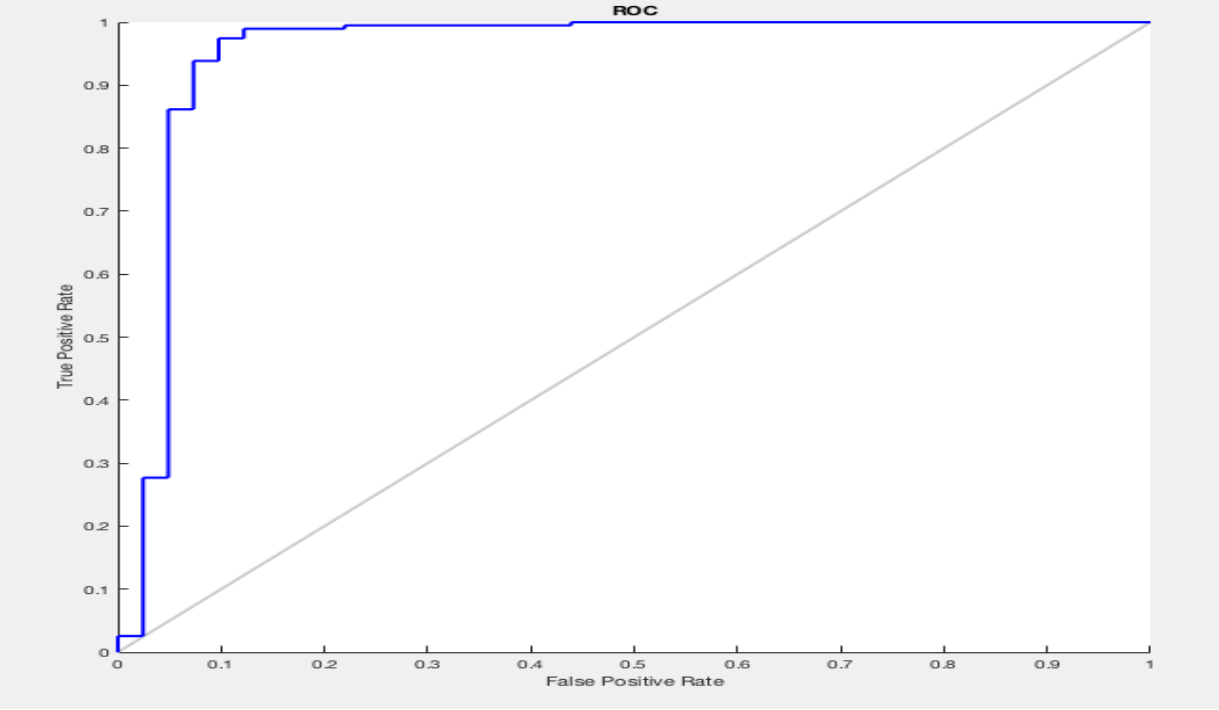


Figure 4.4: The ROC-AUC chart of Test2 four-fold cross-validation in Iteration 2 results.

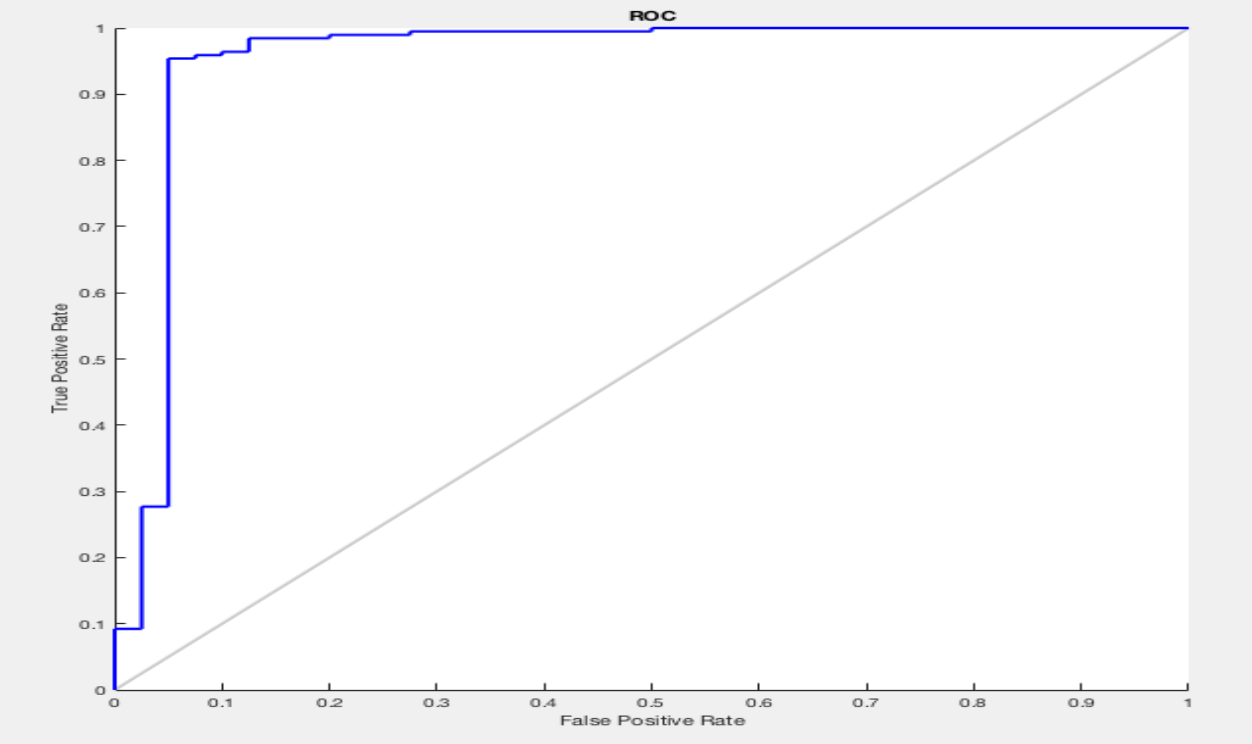


Figure 4.5: The ROC-AUC chart of Test2 four-fold cross-validation in Iteration 3 results.

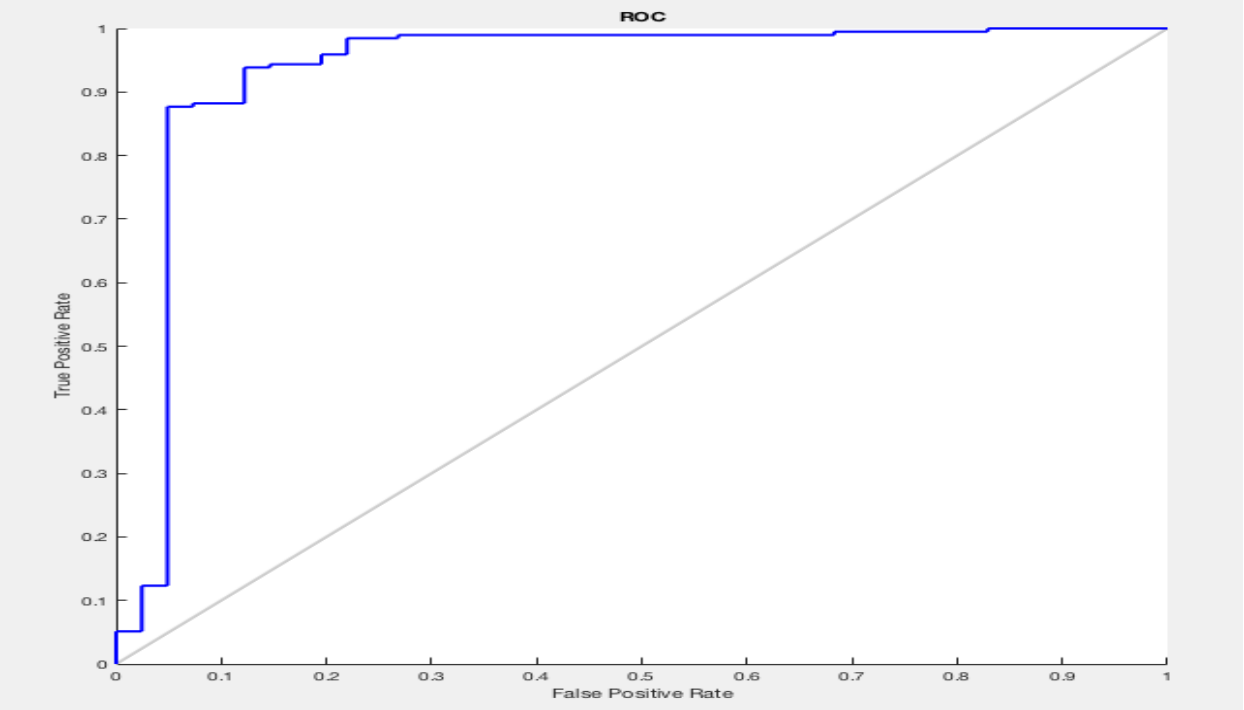


Figure 4.6: The ROC-AUC chart of Test2 four-fold cross-validation in Iteration 4 results.

Test 3: This experiment presents the results of applying a Linear-SVM classifier with a two-fold cross-validation method. A linear kernel used as the hyperplane to separate the samples. It can be noticed that both Iteration 1 and Iteration 2 have the same results. The overall accuracy, the T1DM predictions (i.e.specificity), the T2DM predictions (i.e.recall) and the tradeoff between T1DM and T2DM predictions (i.e.G-mean value) all of them have the same values in both Iterations, as shown table 4.10. While we can conclude that this model can predict both T1DM and T2DM well but not accurately

Table 4.10: Test3 two-fold cross-validation of Iteration 1 and 2 results.

Linear SVM		
	Iteration 1	Iteration 2
Training Accuracy	96.8%	96.8%
Testing Accuracy	96.8%	96.8%
TP	130	130
FP	1	1
TN	23	23
FN	4	4
Error rate	0.032	0.032
Precision	99.23%	99.23%
Recall	97.01%	97.01%
Specificity	95.83%	95.83%
Gmean	96.42%	96.42%

Test 4: This experiment presents the results of applying a Linear-SVM classifier with a four-fold cross-validation method. A linear kernel used as the hyperplane to separate the samples. Table 4.11 illustrates the results of this model. on the one hand, It can be noticed that the performance results of all Iterations are good. But on the other hand, Iteration 1 has

better results than other Iterations. The overall accuracy, the T1DM predictions (i.e.specificity), the T2DM predictions (i.e.recall) and the tradeoff between T1DM and T2DM predictions (i.e.G-mean value) all of them are in their best values in Iteration 1 model compared to other Iterations.

Despite SVM's ability to predict well T1DM and T2DM in both Test 3 and Test 4 models, it is still unable to predict them accurately.

Table 4.11: Test4 four-fold cross-validation of Iteration 1-4 results**.**

Linear SVM				
	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Training Accuracy	98.3%	96.2%	96.2%	96.6%
Testing Accuracy	92.4%	97.4%	98.7%	97.4%
TP	130	194	192	192
FP	1	1	3	3
TN	37	33	34	36
FN	3	8	6	5
Error rate	0.032	0.038	0.038	0.034
Precision	99.23%	99.48%	98.46%	98.46%
recall	97.74%	96.03%	96.96%	97.46%
Specificity	97.36%	97.05	91.89%	92.30%
Gmean	97.55%	96.54%	94.39%	94.84%

4.3.2 PSO-FFNN Experiments on DataPal Dataset

The main idea of the PSO-FFNN model is to use the PSO optimization algorithm to optimize the FFNN weights values. The particles of the PSO search the search space for the optimal weights. We have specified values for some PSO algorithm parameters as their optimal values when applying the PSO algorithm as we have mentioned in section 4.3. The performance evaluation of applying the DataPal dataset on the PSO-FFNN model using K-Fold Cross-Validation methodology. Where the optimal number of neurons in the hidden layer of the FFNN will be adjusted during the following experiments.

Test 5: In this experiment the two fold cross-validation method used to partition the DataPal dataset into training and testing subsets. The results were distributed in terms of the number of neurons in the hidden layer of the FFNN network. The mentioned dataset was applied using the proposed PSO-FFNN model to predict diabetes mellitus. Where some of the PSO algorithm parameters were specified with their optimal values while training the proposed model as we have presented in section 3.4. Tables 4.12 and 4.13 illustrate the overall accuracy of training and testing the dataset where we got high values by up to 98.73%. The high accuracy is not always a good measure, especially with imbalanced datasets. For example, high values of model classification accuracy do not necessarily imply their accuracy in predicting each type of disease present in the DataPal dataset. So, we need more accurate measurements to prove the accurate prediction of the T1DM and T2DM. Where diabetic patients with T1DM are at a more dangerous level of their illness so we must not compromise in the diagnosis of their disease. Recall and Precision are the

metrics that prove the effectiveness of the model in predicting T2DM. The Specificity value proves the effectiveness of the model in predicting T1DM.

Precision is the metric that measures the percentage of the samples correctly classified as T2DM to the all predicted as T2DM. A recall is a metric that measures the percentage of the samples correctly classified as T2DM to all samples that are T2DM actually. So, the recall measures how the model is accurate in predicting T2DM. The specificity is the metric that measures the percentage of the samples correctly classified as T1DM to all samples that are T1DM actually. So, the specificity measures how the model is accurate in predicting T1DM. Tables 4.12 and 4.13 show that the model was effective in predicting both T1DM and T2DM with different numbers of neurons in the hidden layer of the FFNN. Tables 4.12 and 4.13 show the results of Iteration 1 and Iteration 2 respectively of applying the PSO-FFNN model with a two-fold cross-validation method. On the one hand, the results show that the model is effective in predicting diabetes types by looking at the overall accuracy of both Iterations. The overall accuracy of both Iterations looks very close to each other. To judge the final results of the PSO-FFNN model, other measurements should be used such as recall, specificity, and G-mean.

In both Iterations that have been applied in this test, our model got higher efficiency in predicting T1DM and T2DM. When n=6, it can be noticed that Iteration 2 has better values to predict T1DM by up to 100%. Also when n=8, we can notice that the results of Iteration 2 show that the model can predict T2DM perfectly. Both Iterations results show good values for the geometric mean that present the tradeoff between the recall and specificity

predictions. Where the PSO-FFN model got a G-mean value by up to 99.6%, which is perfect.

On the other hand, all the above results show that the performance of the two Iterations is very close to each other. So, we need a final measure to prefer one Iteration over the other to decide the winner between them. Area Under the ROC curve (i.e. AUC), the area under the ROC curve illustrates the tradeoff between the true positive rate and the false positive rate. Where the ROC curve represents the model's ability to separate classes. Table 4.14, figure 4.7 and figure 4.8 show that both Iterations of this test have good AUC values 0.9774 and 0.9591, respectively. We can notice that the Iteration with higher specificity percentages that mean lower FPR values got a higher AUC value (i.e.Iteration 1) the Iteration with the higher recall and specificity values.

So, from the above and taking into account all performance measurements, especially the AUC, the use of a two-fold cross-validation method with the PSO-FFNN model shows that the two-fold cross-validation method was successful in classifying T1DM when n= 6 and T2DM when n= 8 by up to 100%.

Table 4.12: Test5 two-fold cross-validation in Iteration 1 result.

	Two-Fold Cross-Validation Iteration 1				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	97.47%	98.73%	98.73%	97.47%	99.36%
Testing Accuracy	95.54%	96.8%	96.17%	96.8%	96.8%
TP	128	130	130	129	130
FP	1	1	1	2	1
TN	26	26	26	25	26
FN	3	1	1	1	1
Error rate	0.0253	0.0127	0.0127	0.0253	0.0127

Precision	99.22%	99.24%	99.24%	98.47%	99.24%
recall	97.7%	99.24%	99.24%	99.23%	99.24%
Specificity	96.3%	96.3%	96.3%	92.59%	96.3%
Gmean	97%	97.75%	97.75%	95.85%	97.75%
TPR	0.977	0.9923	0.9923	0.9923	0.9923
FPR	0.037	0.037	0.037	0.08	0.037
Area	0.0181	0	0	0.0427	-0.427

Table 4.13: Test5 two-fold cross-validation in Iteration 2 results.

	Two-Fold Cross-Validation Iteration 2				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	98.09%	98.1%	98.73%	98.73%	98.09%
Testing Accuracy	98.73%	98.1%	97.47%	96.83%	94.30%
TP	129	129	130	130	129
FP	2	2	0	2	2
TN	25	25	27	25	25
FN	1	1	1	0	1
Error rate	0.0191	0.0191	0.0127	0.0127	0.0191
Precision	98.47%	98.47%	100%	98.48%	98.47%
recall	99.23%	99.23%	99.23%	100%	99.23%
Specificity	92.59%	92.59%	100%	92.59%	92.59%
Gmean	95.85%	95.85%	99.6%	96.22%	95.85%
TPR	0.9923	0.9923	0.9924	1	0.9923
FPR	0.0740	0.0740	0.0740	0	0.0740
Area	0.0367	0	0	-0.0737	0.0737

Table 4.14: Test 5 AUC values.

Iteration	**AUC**
One	0.9774
Two	0.9591

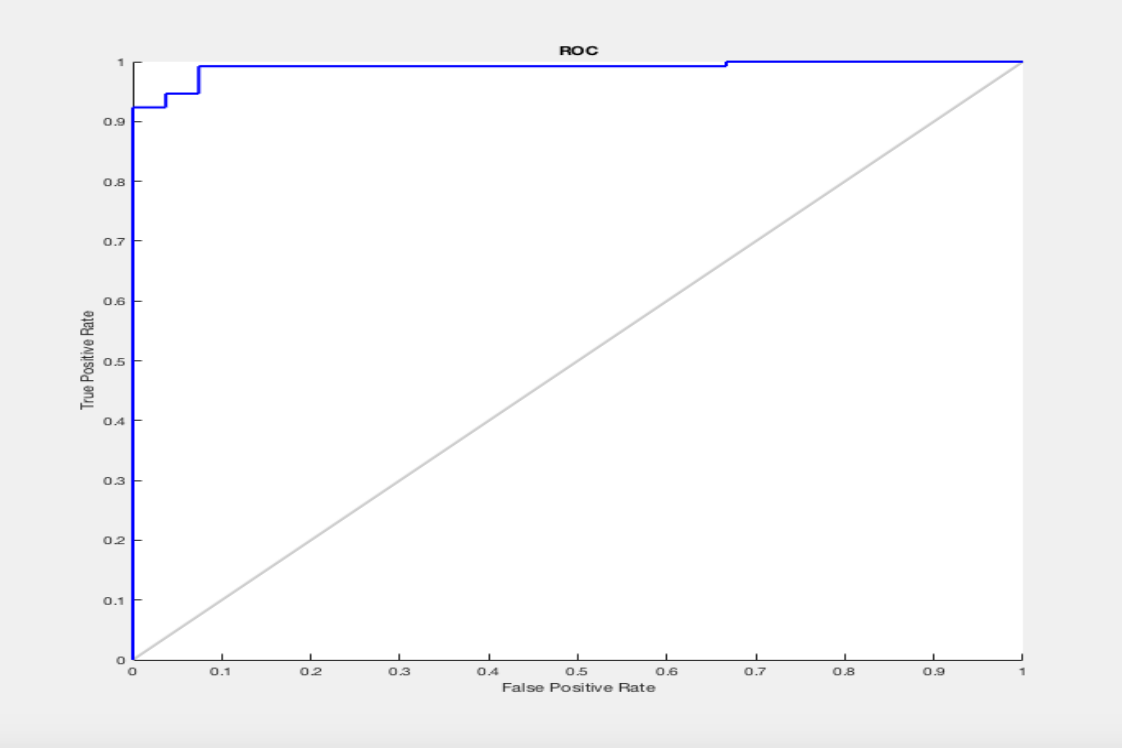


Figure 4.7: The ROC-AUC chart of Test5 two-fold cross-validation in Iteration 1 results.

Test 6: in this experiment, we apply the PSO-FFNN model (i.e. the optimized FFNN)with a four-fold cross-validation method to validate the performance of the model. As we have mentioned before, the optimal number of hidden layer neurons "n" is adjusted by training neural networks with different numbers of hidden layer neurons. So by training the network a different number of hidden neurons to find the optimal "n" and using the K-Fold Cross Validation methodology to cross-validate the datasets, we can measure the performance of our PSO-FFNN model using different performance metrics.

The DataPal dataset was divided into four sub-datasets and validated as shown in the following results through 4 Iterations. Tables 4.15, 4.16, 4.17 and 4.18 present the whole results of applying four Iterations, respectively.

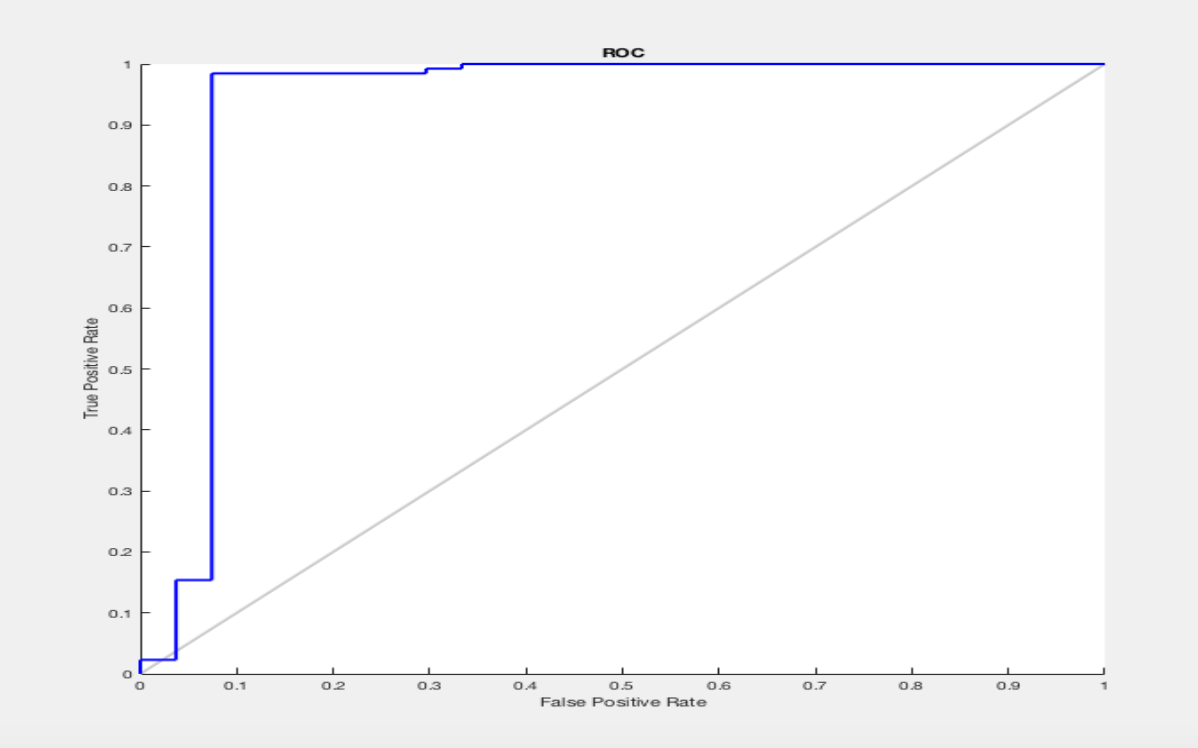


Figure 4.8: The ROC-AUC chart of Test5 two-fold cross-validation in Iteration 2 results.

On the one hand, Immediately, we can notice that Iteration 1 results got higher overall accuracy values compared to the other Iterations, by up to 99.57% training accuracy. But, on the other hand, the overall accuracy results of testing the model trained in Iteration 1 is worse than other Iterations results. The overall accuracy isn't the only measurements to choose the model process methods according to. The precision, recall, and specificity of the metrics used to evaluate the ability of the model to predict T1DM and T2DM correctly are presented to evaluate the better model. The T2DM predictions were perfect as shown in table 4.15, the results of applying the first fold samples as a testing dataset. Table 4.15 also shows that T1DM predictions are better in Iteration 1 than other Iterations. Also, the better

percentages of G-mean metric that describes the tradeoff between predicting T1DM and T2DM are gained by applying the Iteration 1.

Table 4.15: Test6 four-fold cross-validation in Iteration 1 result.

	Four-Fold Cross-Validation Iteration 1				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	98.72%	99.15%	99.15%	98.72%	99.57%
Testing Accuracy	92.4%	91.14%	93.67%	89.8%	91.14%
TP	194	195	195	195	195
FP	2	2	2	3	1
TN	38	38	38	37	39
FN	1	0	0	0	0
Error rate	0.0128	0.0085	0.0085	0.0128	0.0043
Precision	98.98%	98.98%	98.98%	98.84%	99.5%
recall	99.49%	100%	100%	100%	100%
Specificity	95%	95%	95%	92.5%	97.5%
Gmean	97.21%	97.46%	97.46%	96.17%	98.74%
TPR	0.9949	1	1	1	1
FPR	0.05	0.05	0.05	0.075	0.025
Area	-0.05	0.025	0	0	0.0249

To determine the best decision of which model cross-validation to use, the ROC curve must be illustrated to decide the winner according to the previously discussed results and the AUC values. Table 4.19 and figures 4.9, 4.10, 4.11 and 4.12 show that the first Iteration got the most stable state in predicting T2DM compared to other Iterations, with minimum TPR values. The overall accuracy was in its better value at Iteration 1 when n=10 by up to 99.57% training accuracy. T2DM predictions were good at all Iterations but they were perfect and in a stable manner with any number of neurons in Iteration 1 by up to 100%. Iteration 1 got better predictions of T1DM as well. By taking the AUC values of each

Iteration into consideration, we can decide that Iteration 1 has a better area under the ROC curve, by up to 0.9750 which means better predictability in T2DM. So, according to the above, we can decide that the PSO-FFNN model was able to predict each of T1DM and T2DM by up to 97.5% and 100% respectively.

Table 4.16: Test6 four-fold cross-validation in Iteration 2 results.

	Four-Fold Cross-Validation Iteration 2				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	97.46%	97.03%	97.88%	97.88%	97.03%
Testing Accuracy	96.15%	93.59%	94.87%	96.15%	96.15%
TP	193	195	193	193	194
FP	4	5	3	3	6
TN	37	36	38	38	35
FN	2	2	2	2	1
Error rate	0.0254	0.0297	0.0212	0.0212	0.0297
Precision	97.97%	97.5%	98.47%	98.47%	97%
recall	98.97%	98.98%	98.97%	98.97%	99.49%
Specificity	90.24%	87.80%	92.68%	92.68%	85.36%
Gmean	94.5%	93.22%	95.77%	95.77%	92.15%
TPR	0.9897	0.9898	0.9897	0.9897	0.9949
FPR	0.0976	0.1220	0.0732	0.0732	0.1464
Area	0.0483	0.0241	-0.0483	0	0.0726

Table 4.17: Test6 four-fold cross-validation in Iteration 3 results.

	Four-Fold Cross-Validation Iteration 3				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	96.59%	96.59%	96.59%	97.45%	97.45%
Testing Accuracy	94.93%	94.93%	98.73%	93.67%	97.46%
TP	194	193	191	191	193
FP	7	6	4	2	4
TN	33	34	36	38	36
FN	1	2	4	4	2
Error rate	0.0340	0.0340	0.0340	0.0255	0.0255
Precision	96.52%	96.98%	97.95%	98.96%	97.97%

recall	99.49%	98.97%	97.95%	97.95%	98.97%
Specificity	82.5%	85%	90%	95%	90%
Gmean	90.6%	91.71%	93.9%	96.46%	94.37%
TPR	0.9949	0.9897	0.9795	0.9795	0.9897
FPR	0.1750	0.1500	0.1000	0.0500	0.1000
Area	0.0871	-0.0248	-0.0492	-0.0490	0.0492

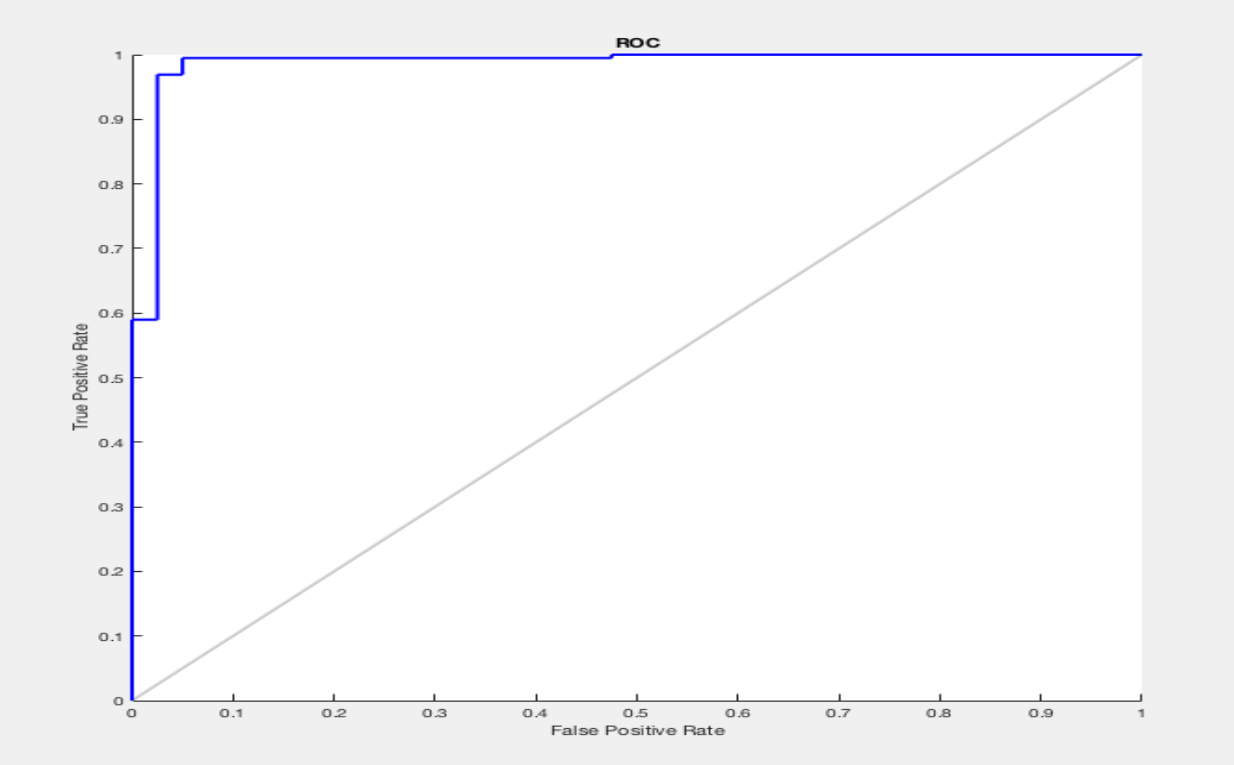


Figure 4.9: The ROC-AUC chart of Test6 four-fold cross-validation in Iteration 1 result.

Table 4.18: Test6 four-fold cross-validation in Iteration 4 results.

	Four-Fold Cross-Validation Iteration 4				
	n=2	n=4	n=6	n=8	n=10
Training Accuracy	97.03%	97.46%	97.88%	97.46%	97.03%
Testing Accuracy	97.43%	96.15%	96.15%	94.87%	96.15%
TP	192	194	192	194	194
FP	4	5	2	5	6
TN	37	36	39	36	35
FN	3	1	3	1	1

Error rate	0.0297	0.0254	0.0212	0.0254	0.0297
Precision	97.96%	97.49%	98.97%	97.49%	97%
recall	98.46%	99.49%	98.46%	99.49%	99.49%
Specificity	90.24%	87.8%	95.12%	87.8%	85.36%
Gmean	94.26%	93.46%	96.77%	93.46%	92.15%
TPR	0.9846	.9949	.9846	.9949	.9949
FPR	0.0976	0.1220	0.0488	0.1220	0.1464
Area	0.0480	0.0241	-0.0728	0.0728	0.0243

Table 4.19: Test 6 AUC values.

Iteration	AUC
1	0.9750
2	0.9482
3	0.9086
4	0.9479

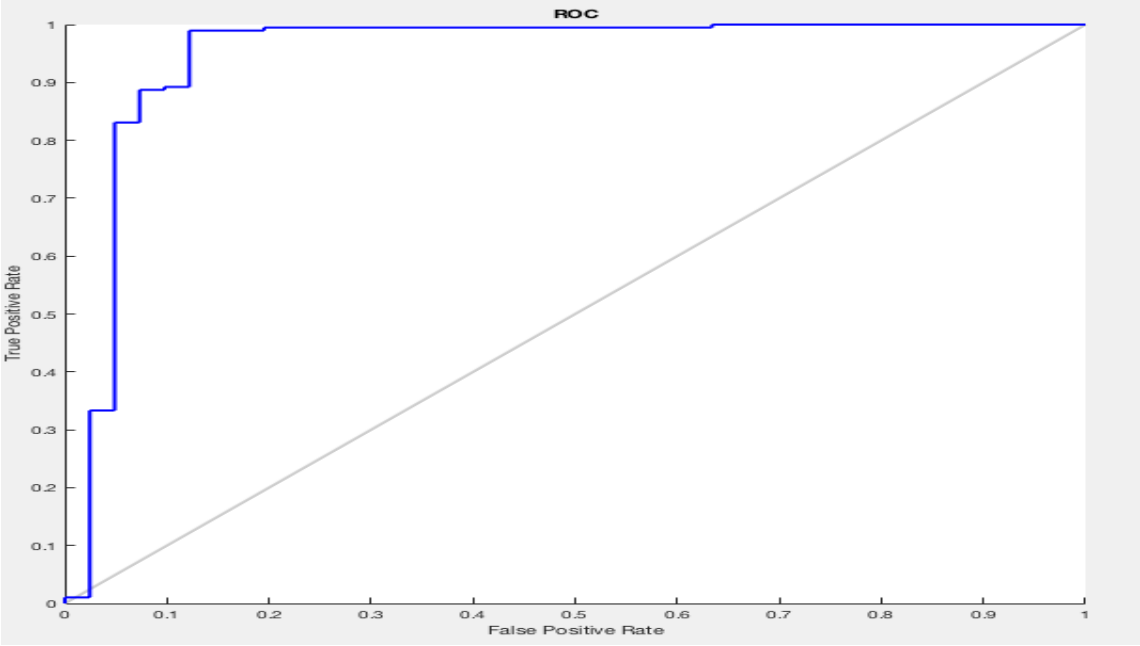


Figure 4.10: The ROC-AUC chart of Test6 four-fold cross-validation in Iteration 2 results.

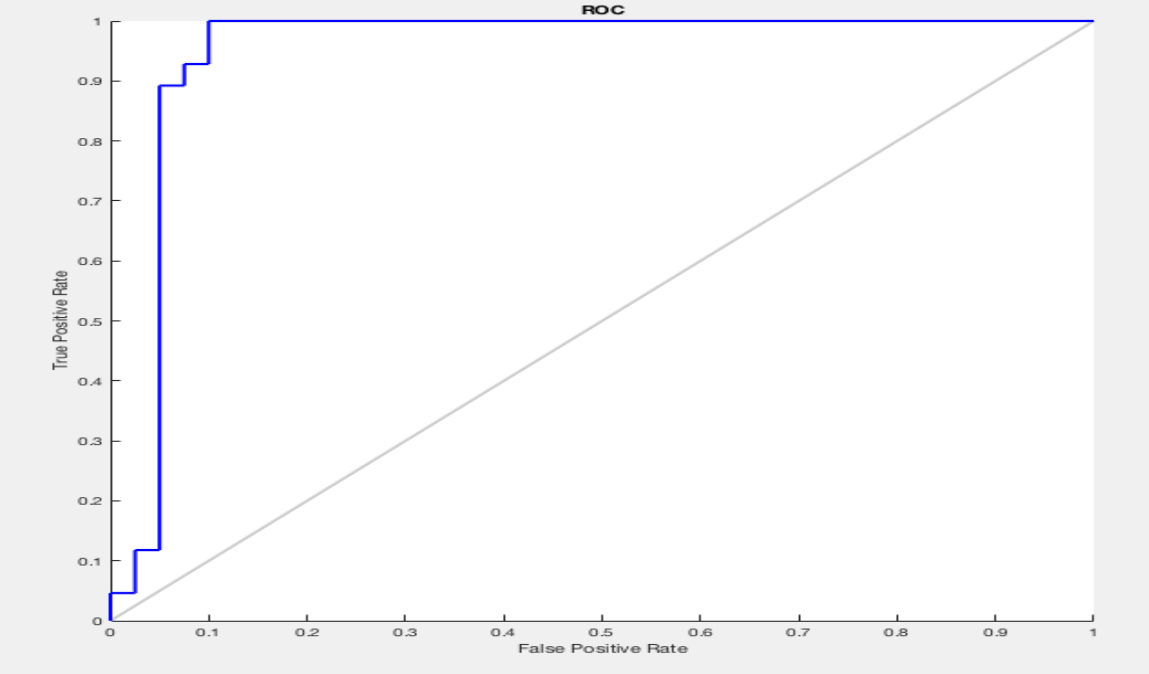


Figure 4.11: The ROC-AUC chart of Test6 four-fold cross-validation in Iteration 3 results.

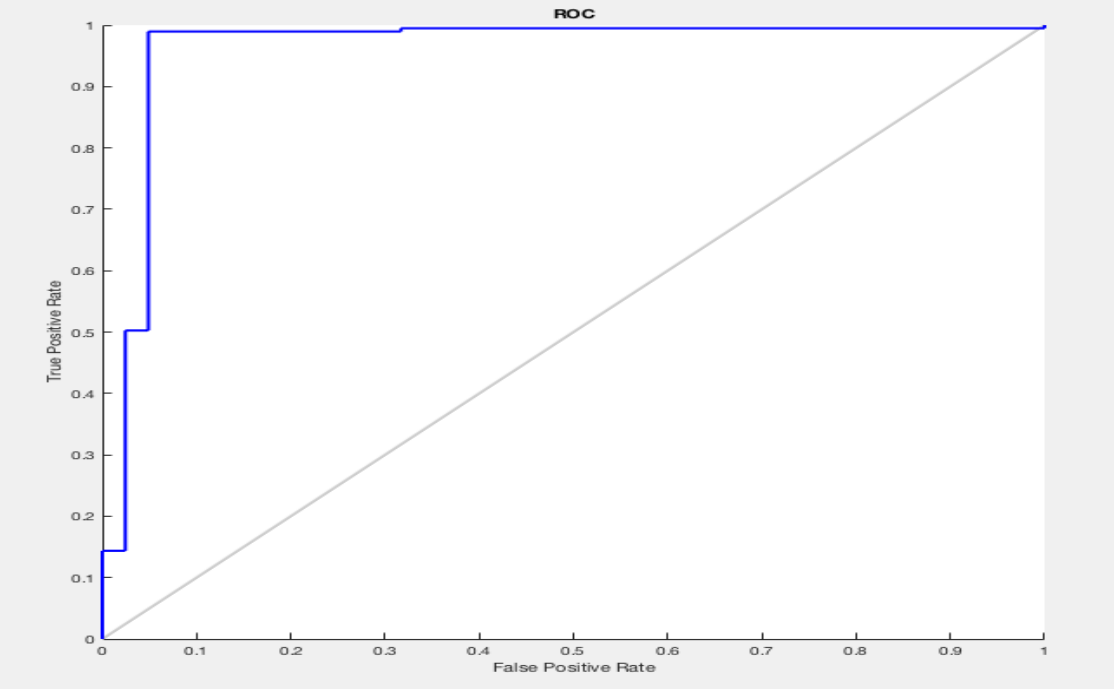


Figure 4.12: The ROC-AUC chart of Test6 four-fold cross-validation in Iteration 4 results.

Summary: Based on the results of PSO-FFNN tests above using two-fold and four-fold cross-validation methods, we can decide that Iteration 1 is the best in each of two-fold and four-fold cross-validation models. In the previous tests, the PSO optimization algorithm has been used to train an FFNN of two layers of network to adjust the FFNN network weights in an optimization manner. On the one hand, the DataPal dataset is an imbalanced dataset where the overall accuracy metric is not enough to determine the optimal model in predictions. On the other hand, tests 5 and 6 shows that both a two-fold and four-fold cross-validation method has proved the ability of the proposed PSO-FFNN model to predict T1DM and T2DM. Test 5 has got an overall accuracy of up to 99.36%, the test model was able to predict T1DM and T2DM perfectly.

Test 6 model was able to predict T1DM and T2DM by up to 97.5% and 100% respectively, with a 99.57% overall accuracy when n=10. It can be noticed that the Test 6 model was unable to compete with the Test 5 model in predicting T1DM, considered the most dangerous type of diabetes. As for their ability to classify the T2DM, both test results were very close to each other by up to 0.9750 and 0.9773 for two-fold and four-fold cross-validation methods, respectively.

4.3.3 Discussion of the Results

The PSO-FFNN presented model, an MLP-BPNN algorithm and a Linear-SVM algorithm were applied on the DataPal dataset. Section 4.3 introduced various parameters that have been adjusted with their optimal values. The three model performance results were better using the determined parameters in section 4.3. The parameters have been adjusted to

determine their optimal values, by training the models with different parameter values. PSO-FFNN model parameters were justified by trying different values to find the optimal parameter values with any number of hidden neurons of the FFNN hidden layer. The K-fold cross-validation method used to validate the performance results of the three models applied in this thesis. Both the two-fold and four-fold cross-validation methods were selected to validate the models applied to the DataPal dataset. The results of the all above tests show that the PSO-FFNN model outperformed the other models in predicting diabetes types. The PSO-FFNN model was the best in T2DM predictions by up to 100% using a four-fold cross-validation method and 10 neurons in the hidden layer. So, the model was able to predict the most dangerous level of diabetes (i.e. T2DM) perfectly. The performance results of the two-fold cross-validation and four-fold cross-validation methods were good. But, it can be noticed that the four-fold cross-validation method outperforms the two-fold cross-validation method in some places. The overall accuracy values of training the Iteration 1 of previous test 2 (i.e. four-fold cross-validation) models were the best values generated by training the DataPal dataset by up to 99.57%. Also, test 2 (i.e. four-fold cross-validation) model was the best in T2DM and T1DM predictions as well, by up to 97.5% and 100% respectively. As for the AUC value, it was better in the mentioned model compared to other models.Therefore, from the above, we can notice that the optimal FFNN weights values (i.e. optimal particle positions) tuned in Iteration 1 of the test 2 model. Thus, the optimal number of neurons in the hidden layer of trained FFNN is 10. Both the

MLP-BPNN and the Linear-SVM classifiers failed to outperform the PSO-FFNN model in predicting the DataPal dataset.

A lot of machine learning techniques have been applied as classifiers for different medical datasets. Diseases predictions should be very accurate because of the sensitivity of medical data and how they relate to human health. Tables 4.20 and 4.21 show the results of applying various machine learning algorithms mentioned in the related works section [18][15][8][50][52][53][55][56] to predict the DataPal dataset. k nearest neighbor "KNN", Discriminant Analysis Classifier "DA", Naive Bayes "NB", decision tree "DT" and Random Forest "RF" were the algorithms that have been used to predict DataPal dataset. In addition to the Support Vector Machine algorithm using different types of kernel functions such as Linear, Radial Base Function "RBF", Polynomial, and Gaussian kernel functions. A two-fold cross-validation method used to define the comparison and validate the PSO-FFNN model. It can be noticed that the Naive Bayes "NB", Discriminant Analysis Classifier "DA" and the Linear-SVM algorithms performance values were close to each other. The overall accuracy, T2DM predictions, and T1DM predictions were in their best values using NB, DA and Linear-SVM algorithms. Figure 4.13 illustrates the comparison between the various previous algorithms and our applied three models. Where the models were compared in terms of total accuracy. Thus, as shown in the figures the PSO-FFNN model outperforms all models in the overall accuracy of predicting DataPal datasets.

Tables 4.22, 4.23, 4.24 and 4.25 present the results of applying different machine learning algorithms to predict the DataPal dataset. A four-fold cross-validation method used to

validate the models as well. Looking at figure 4.15, it can be noticed that the PSO-FFNN model outperforms all models through four Iterations in predicting the DataPal dataset with overall accuracy, by up to 99.57%.

To support our hypothesis, another performance metric can be taken into account such as the Recall metric. Recall percentages represent the ability of models in predicting T2DM that we have mentioned before as the most dangerous level of Diabetes Mellitus. Figure 3.16 shows that the proposed PSO-FFNN model outperforms the other models in T2DM predictions as well. There is no doubt that the PSO-FFNN model outperforms all models in all Iterations with both two-fold and four-fold cross-validation methods. So, PSO-FFNN is the winner.

Table 4.20: Two-fold cross-validation results Iteration previous works - Iteration 1.

Iteration 1	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	92.67%	96.8%	96.8%	95.2%	96.1%	95.2%	94.9%	95.2%	95.2%
TP	247	255	256	255	255	254	252	254	254
FP	13	5	4	5	5	6	8	6	6
TN	44	48	48	44	47	45	46	45	45
FN	10	6	6	10	7	9	8	9	9
Precision	95%	98.07%	98.46%	98.07%	98.07%	97.69%	96.92%	97.69%	97.69%
recall	96.10%	97.7%	97.7%	96.22%	97.32%	96.57%	96.92%	96.57%	96.57%
Specificity	77.19%	90.56%	92.3%	89.79%	90.38%	88.23%	85.18%	88.23%	88.23%
Gmean	86.12%	94.06%	94.97%	92.95%	93.79%	92.31%	90.86%	92.31%	98.74%

Table 4.21: Two-fold cross-validation results Iteration previous works - Iteration 2.

Iteration 2	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	93.63%	96.49%	96.81%	95.22%	96.17%	95.2%	94.9%	95.2%	95.22%
TP	253	255	256	255	255	254	252	254	254
FP	7	5	4	5	5	6	8	6	6
TN	41	48	48	44	47	45	46	45	45
FN	13	6	6	10	7	9	8	9	9
Precision	97.3%	98.07%	98.46%	98.07%	98.07%	97.69%	96.92%	97.69%	97.69%
recall	95.11%	97.7%	97.7%	96.22%	97.32%	96.57%	96.92%	96.57%	96.57%
Specificity	85.41%	90.56%	92.3%	89.79%	90.38%	88.23%	85.18%	88.23%	88.23%
Gmean	90.13%	94.06%	94.97%	92.95%	93.79%	92.31%	90.86%	92.31%	98.74%

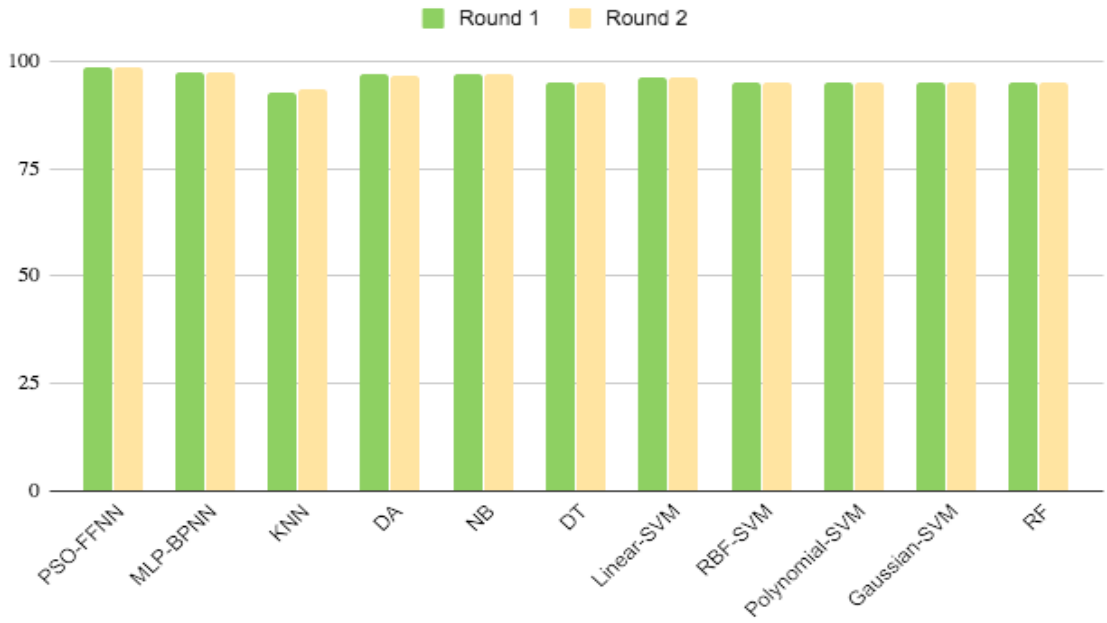


Figure 4.13: The Recall values Iteration applied algorithms on the DataPal dataset using two-fold cross-validation.

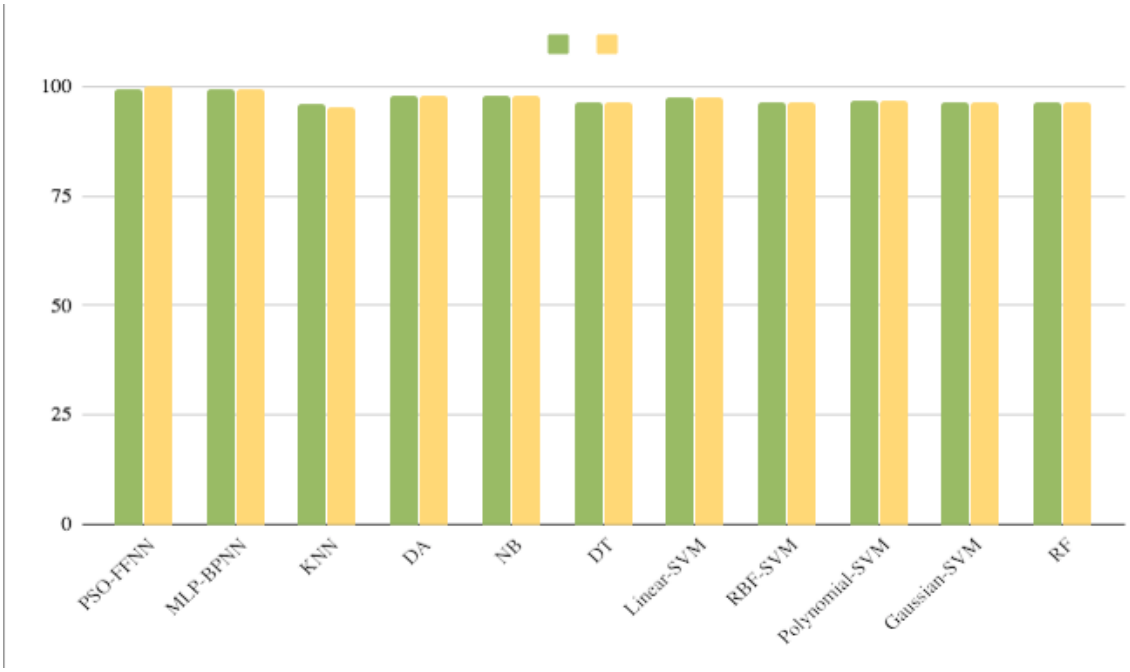


Figure 4.14: The accuracy values Iteration applied algorithms on the DataPal dataset using two-fold cross-validation.

Table 4.22: Four-fold cross-validation results Iteration previous works - Iteration 1.

Iteration 1	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	93.30%	96.81%	96.81%	93.94%	96.18%	95.87%	95.23%	95.87%	96.17
TP	247	256	256	248	254	254	251	254	255
FP	13	4	4	12	6	6	9	6	5
TN	46	48	48	47	48	47	48	47	47
FN	8	6	6	7	6	7	6	7	7
Precision	95%	98.46%	98.46%	95.38%	97.69%	97.69%	96.53%	97.69%	98.07%
recall	96.86%	97.70%	97.70%	97.25%	97.69%	97.31%	97.66%	97.31%	97.32%
Specificity	77.96%	92.30%	92.30%	79.66	88.88%	88.67%	84.21%	88.67%	90.38%
Gmean	86.90%	94.97%	94.97%	88.01%	93.18%	92.89%	90.68%	92.89%	93.79%

Table 4.23: Four-fold cross-validation results Iteration previous works - Iteration 2.

Iteration 2	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	93%	96.81%	96.81%	93.94%	96.18%	95.87%	95.23%	95.87%	96.17
TP	247	256	256	248	254	254	251	254	255
FP	13	4	4	12	6	6	9	6	5

TN	45	48	48	47	48	47	48	47	47
FN	9	6	6	7	6	7	6	7	7
Precision	95%	98.46%	98.46%	95.38%	97.69%	97.69%	96.53%	97.69%	98.07%
recall	96.48%	97.70%	97.70%	97.25%	97.69%	97.31%	97.66%	97.31%	97.32%
Specificity	77.58%	92.30%	92.30%	79.66	88.88%	88.67%	84.21%	88.67%	90.38%
Gmean	86.52%	94.97%	94.97%	88.01%	93.18%	92.89%	90.68%	92.89%	93.79%

Table 4.24: Four-fold cross-validation results Iteration previous works - Iteration 3.

Iteration 3	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	94.9%	96.81	96.81%	93.94%	96.18%	95.87%	95.23%	95.87%	96.17
TP	250	256	256	248	254	254	251	254	255
FP	10	4	4	12	6	6	9	6	5
TN	48	48	48	47	48	47	48	47	47
FN	6	6	6	7	6	7	6	7	7
Precision	96.15%	98.46%	98.46%	95.38%	97.69%	97.69%	96.53%	97.69%	98.07%
recall	97.65%	97.70%	97.70%	97.25%	97.69%	97.31%	97.66%	97.31%	97.32%
Specificity	82.75%	92.30%	92.30%	79.66	88.88%	88.67%	84.21%	88.67%	90.38%
Gmean	89.89%	94.97%	94.97%	88.01%	93.18%	92.89%	90.68%	92.89%	93.79%

Table 4.25: Four-fold cross-validation results Iteration previous works - Iteration 4.

Iteration 4	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	93.3%	96.81%	96.81%	93.94%	96.18%	95.87%	95.23%	95.87%	96.17
TP	247	256	256	248	254	254	251	254	255
FP	13	4	4	12	6	6	9	6	5
TN	46	48	48	47	48	47	48	47	47
FN	8	6	6	7	6	7	6	7	7
Precision	95%	98.46%	98.46%	95.38%	97.69%	97.69%	96.53%	97.69%	98.07%
recall	96.86%	97.70%	97.70%	97.25%	97.69%	97.31%	97.66%	97.31%	97.32%
Specificity	77.96%	92.30%	92.30%	79.66	88.88%	88.67%	84.21%	88.67%	90.38%
Gmean	86.90%	94.97%	94.97%	88.01%	93.18%	92.89%	90.68%	92.89%	93.79%

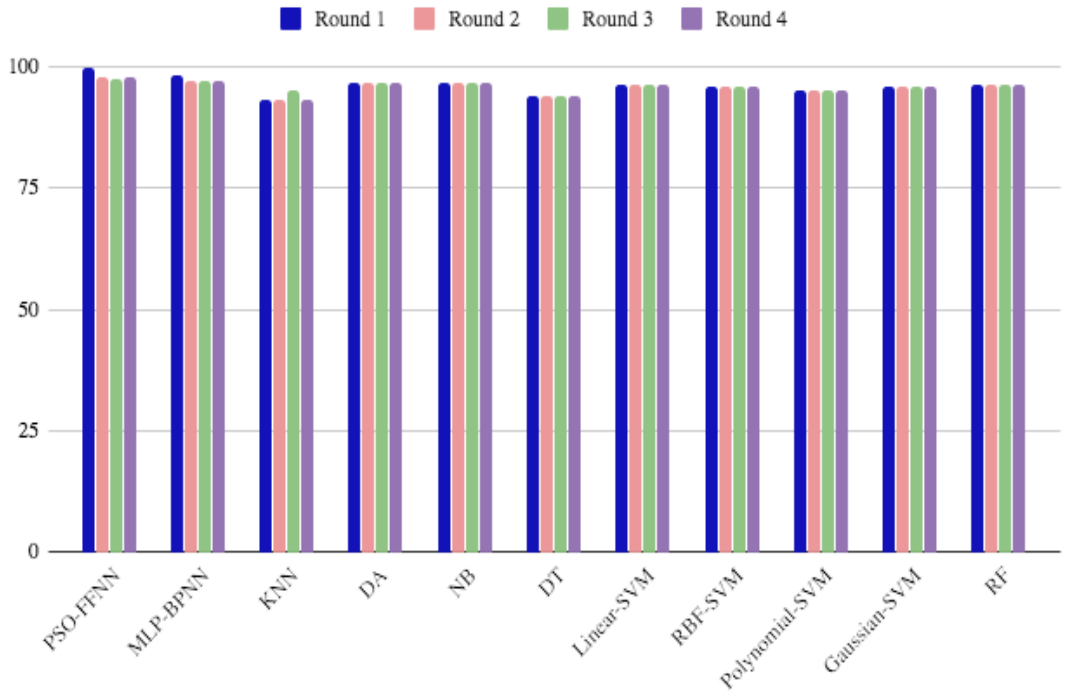


Figure 4.15: The accuracy values Iteration applied algorithms on the DataPal dataset using a four-fold cross-validation method.

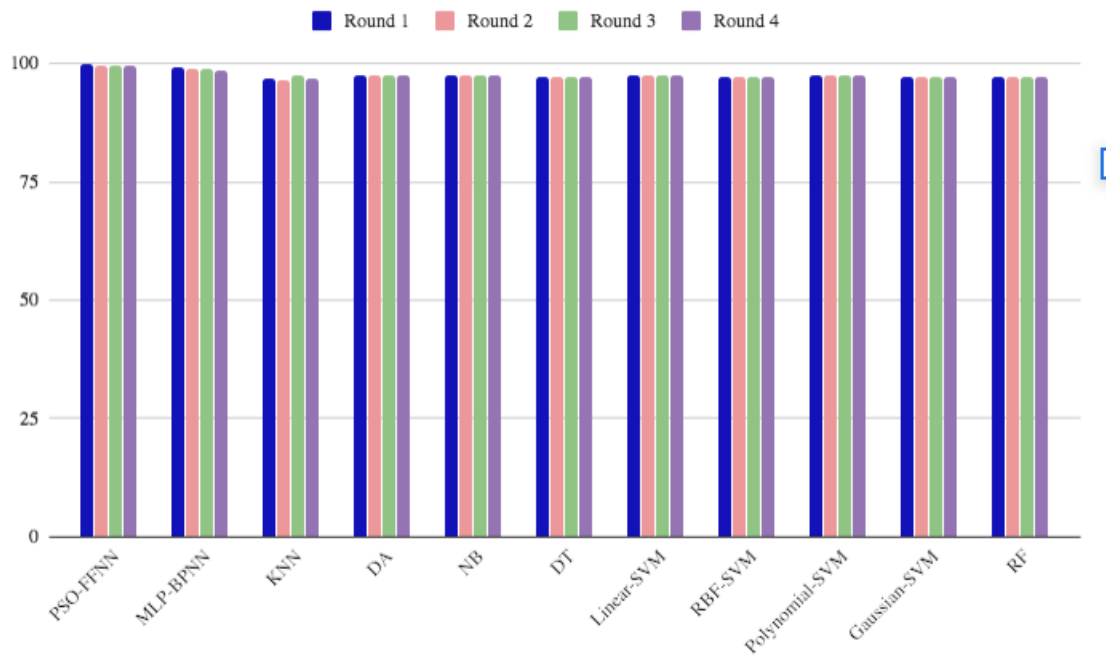


Figure 4.16: The Recall values Iteration applied algorithms on the DataPal dataset using a four-fold cross-validation method.

4.4 The PIDD Dataset Experiments

In these experiments, we apply different machine learning techniques to the PIDD dataset. The dataset that we have presented in section 2.1.2. A dataset consists of 768 records of diabetic and non-diabetic instances. The previous experiments on the DataPal dataset were applied to some machine learning techniques to diagnose diabetes types. While the PIDD dataset used to predict diabetes occurs, where all the Pima citizens who are in diabetes have T2DM diabetes only. So, in these experiments, we apply the PIDD dataset to the proposed PSO-FFNN model and various other machine learning methods in diabetes predictions.

On the one hand, the PIDD dataset consists of 8 predictors used to classify diabetic and non-diabetic people. A linear-SVM classifier has been used to select the most optimal features to predict diabetes. 6 of 8 features were selected to diagnose diabetes. On the other hand, the PIDD dataset includes about 400 records with missing values. The medical diagnoses are the most sensitive predictions where all predictor values are needed to confirm the diagnosis. In this thesis, the K-NN algorithm used to fill in the missing values. Thus, the " K" nearest neighbor instances have been used to estimate the missing values.

Both two-fold and four-fold cross-validation methods have been used to validate the diabetes predictions with the optimal parameters of the models. Where Appendix A presents the selection manner of the optimal parameters for the models applied in this thesis. The following tests show the results of applying the PSO-FFNN model and various machine learning models to the PIDD dataset to predict diabetes and to validate our proposed PSO-FFNN model:

4.4.1 Other Models Experiments on PIDD Dataset

To prove the ability of the PSO-FFNN model in predicting diabetes. We have applied various machine learning techniques to the PIDD dataset. Multiple Layer Perceptron Backpropagation Neural Network “MLP-BPNN”, Support Vector Machine SVM with different kernels, k nearest neighbor "KNN", Discriminant Analysis Classifier "DA", Naive Bayes "NB", decision tree "DT" and Random Forest "RF" were the algorithms that have been used to predict PIDD dataset [18][15][8][50][52][53][55][56].The SVM model was applied with different kernels Linear Function, Radial Base Function "RBF", Polynomial function and Gaussian function kernels. Several previous studies have applied different machine learning techniques to the PIDD to predict diabetic and non-diabetic Mellitus. The following tests present the results of applying the mentioned models to the PIDD dataset:

Test 1: Two-Fold cross-validation methods have been used to validate the models mentioned above. Table 4.26 shows the result of applying the MLP-BPNN with a different number of neurons in the hidden layer. Table 4.27 shows the results of applying the other mentioned algorithms (i.e.SVM, DA, DT, KNN, NB, and RF models) above to the PIDD, with a two-fold cross-validation method. The SVM model was applied with different kernels Linear Function, Radial Base Function "RBF", Polynomial function and Gaussian function kernels. Figure 4.17 illustrates the comparisons between all applied models. It can be noticed that the Polynomial-SVM model was not competitive. The closest results to the

PSO-FFNN model results were achieved by applying the MLP-BPNN model. While the PSO-FFNN model is still outperformed by up to 5% overall accuracy.

Table 4.26: Test 1 Two-fold cross-validation results of applying the MLP-BPNN model to the PIDD.

Iteration 1	n=2	n=4	n=6	n=8	n=10
Training Accuracy	74.7%	78.4%	79.7%	78.9%	80.5%
Testing Accuracy	74%	72.7%	72.7%	74.5%	72.9%
Iteration 2	n=2	n=4	n=6	n=8	n=10
Training Accuracy	75%	76%	74.2%	76.8%	77.3%
Testing Accuracy	72.9%	76%	78.4%	79.2%	79.2%

Table 4.27: Test 1 Two-fold cross-validation results of applying various machine learning methods to the PIDD.

Iteration 1	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	67.44%	78.38%	76.56%	73.69%	77.86%	70.05%	54.94%	70.05%	77.34%
Iteration 2	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	62.5%	71.61%	70.57%	67.96%	72.91%	72.91%	41.14%	72.91%	69.79%

Test 2: Four-Fold cross-validation methods have been used to validate the models mentioned above. Table 4.28 shows the result of applying the MLP-BPNN with a different number of neurons in the hidden layer. Table 4.29 shows the results of applying the other mentioned algorithms (i.e.SVM, DA, DT, KNN, NB, and RF models) above to the PIDD, with a four-fold cross-validation method. The SVM model was applied with different kernels Linear Function, Radial Base Function "RBF", Polynomial function and Gaussian function kernels. Figure 4.18 illustrates the comparisons between all applied models. It can be noticed that the Polynomial-SVM model was not competitive as well as it was in test 1. The closest results to the PSO-FFNN model results were achieved by applying the

MLP-BPNN, Linear-SVM, DA and RF models. While the PSO-FFNN model still outperformed them by up to 11%, 12.5%, 12.5% and 14% overall accuracy, respectively.

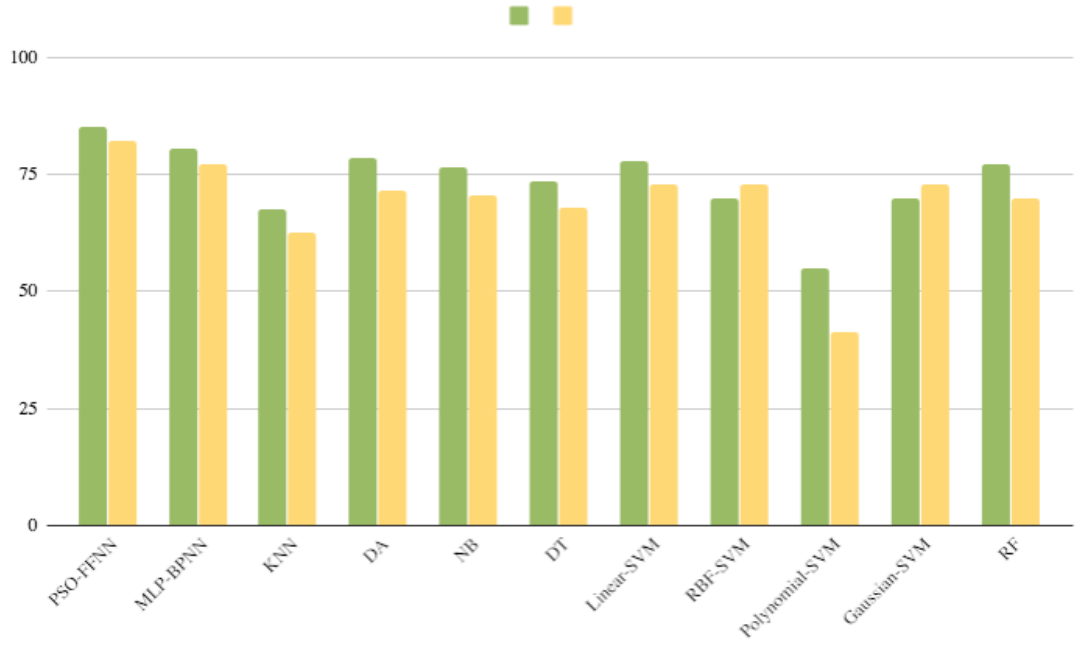


Figure 4.17: The accuracy values Iteration applied algorithms on the PIDD dataset using two-fold cross-validation.

Table 4.28: Test 2 four-fold cross-validation results of applying the MLP-BPNN model to the PIDD.

Iteration 1	n=2	n=4	n=6	n=8	n=10
Training Accuracy	76.4%	77.6%	76.9%	77.3%	78.8%
Testing Accuracy	73.4%	75.5%	73.4%	76.6%	76%
Iteration 2	n=2	n=4	n=6	n=8	n=10
Training Accuracy	76.4%	76.7%	77.3%	77.4%	77.8%
Testing Accuracy	76%	73.4%	72.4%	74.5%	72.9%
Iteration 3	n=2	n=4	n=6	n=8	n=10
Training Accuracy	74.8	77.3%	76.6%	76%	78%
Testing Accuracy	75.5	77.1%	76%	74%	77.1%
Iteration 4	n=2	n=4	n=6	n=8	n=10
Training Accuracy	75%	76.9%	77.3%	78.1%	78%
Testing Accuracy	77.1%	75.5%	76%	78.6%	76%

Table 4.29: Test 2 four-fold cross-validation results of applying various machine learning methods to the PIDD.

Iteration 1	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	68.22%	77.43%	75.69%	71%	77.25%	69.96%	32.11%	69.96%	74.82%
Iteration 2	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	66.84%	75.69%	76.56%	70.83%	76.73%	75 %	44.09%	73.61%	73.61%
Iteration 3	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	65.79%	75.86%	72.56%	70.83%	76.21%	69.27%	36.80%	69.27%	74.82%
Iteration 4	KNN	DA	NB	DT	Linear-SVM	RBF-SVM	Polynomial-SVM	Gaussian-SVM	RF
Accuracy	73.26%	75.86%	73.09%	66.84%	76.04%	72.39%	45.65%	72.39%	75.34%

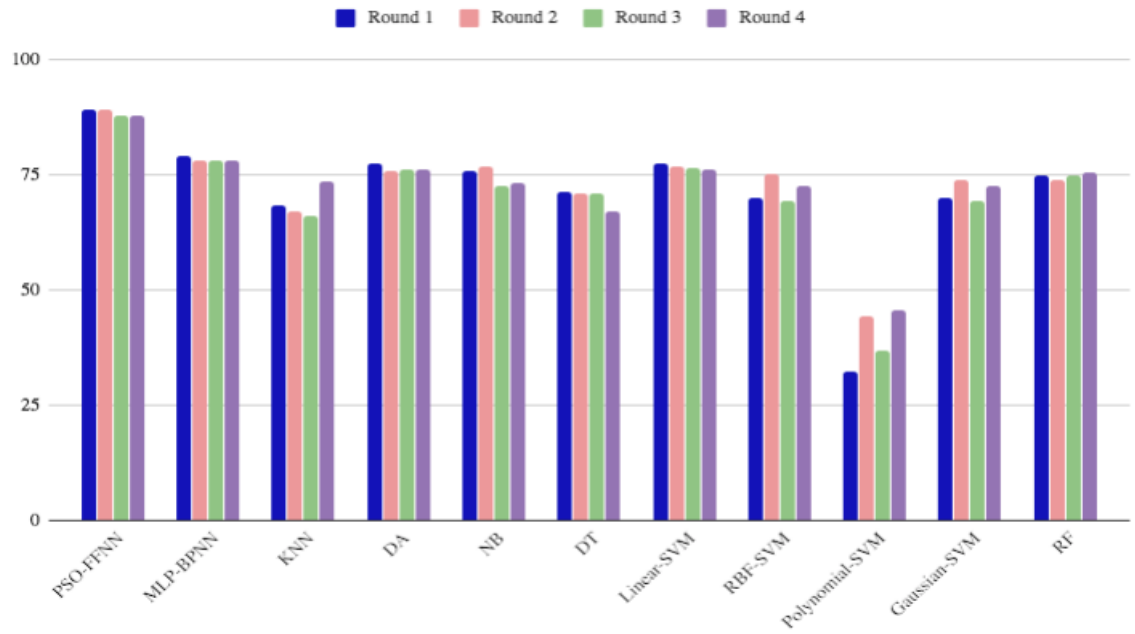


Figure 4.18: The accuracy values Iteration applied algorithms on the PIDD dataset using four-fold cross-validation.

4.4.2 PSO-FFNN Experiments on PIDD Dataset

The main aim of using the PSO evolutionary algorithm is to adjust FFNN weight values that will be used in diabetes predictions. The PSO optimal parameters have been tuned. Appendix A shows the results of tuning the PSO optimal parameters. The following tests present the results of applying the PSO-FFNN model to the PIDD dataset with two-fold and four-fold cross-validation methods:

Test 1: A two-fold cross-validation method used to validate the PSO-FFNN model. The PSO-FFNN model applied to the PIDD model to predict diabetes. The model trained using the optimal parameters determined in Appendix A. Table 4.30 illustrates the results of applying the PSO-FFNN to the PIDD dataset with a two-fold cross-validation method. The results show that the PSO-FFNN model has a very good ability to predict diabetes with a two-fold cross-validation method, by up to 85.16%. Iteration 1 when n= 4,6 or 8 got the best results.

Table 4.30: Test 1 Two-fold cross-validation results of applying the PSO-FFNN model.

Iteration 1	n=2	n=4	n=6	n=8	n=10
Training Accuracy	83.33%	85.16%	85.16%	85.16%	84.9%
Testing Accuracy	81.9%	82.41%	83.92%	82.41%	81.4%
Iteration 2	n=2	n=4	n=6	n=8	n=10
Training Accuracy	82.03%	80.5%	81.77%	82.3%	82.3%
Testing Accuracy	84.9%	81.9%	84.4%	80.4%	86.4%

Test 2: A four-fold cross-validation method used to validate the PSO-FFNN model. The PSO-FFNN model applied to the PIDD model to predict diabetes. The model trained using the optimal parameters determined in Appendix A. Table 4.31 illustrates the results of

applying the PSO-FFNN to the PIDD dataset with a four-fold cross-validation method. The results show that the PSO-FFNN model has a better ability to predict diabetes more than two-fold cross-validation as in Test 1. Iterations 1, 2, 3 and 4 have achieved an overall accuracy up to 89.06%, 88.88%, 87.84%, and 87.67%, respectively.

Table 4.31: Test 2 Four-fold cross-validation results of applying the PSO-FFNN model.

Iteration 1	n=2	n=4	n=6	n=8	n=10
Training Accuracy	87.5%	89.06%	88.71%	86.63%	88.88%
Testing Accuracy	81.77%	83.85%	83.33%	81.77%	82.29%
Iteration 2	n=2	n=4	n=6	n=8	n=10
Training Accuracy	88.88%	88.37%	88.19%	88.37%	87.84%
Testing Accuracy	83.33%	81.77%	80.73%	80.4%	80.21%
Iteration 3	n=2	n=4	n=6	n=8	n=10
Training Accuracy	87.5%	87.67%	86.63%	87.15%	87.84%
Testing Accuracy	84.9%	82.81%	85.41%	84.37%	83.33%
Iteration 4	n=2	n=4	n=6	n=8	n=10
Training Accuracy	87.67%	87.67%	86.8%	86.8%	87.5%
Testing Accuracy	84.37%	83.85%	84.37%	85.93%	84.37%

4.4.3 Discussion of the Results

The above results prove that the PSO-FFNN model outperformed the other models such as SVM with its different kernels, KNN, NB, DA, DT, and the RF models. The PSO-FFNN model with a four-fold cross-validation method has achieved a very good accuracy of up to 89.08% in predicting diabetes and non-diabetes cases. Also, it can be noticed that the model was perfect in predicting diabetes types with a classification accuracy of up to 99.57%.From previous experiments on the PIDD dataset, we have verified the ability of the PSO-FFNN model in predicting diabetes and non-diabetes.

4.5 Summary

In this chapter, we have applied the system procedures that we have introduced in chapter 3. The datasets were preprocessed using both the SVM and the KNN algorithms. Where the KNN algorithm has been used to estimate the missing values and fill them. The SVM algorithm has been used to select the most optimal predictors/attributes that have increased the efficiency of the models in the prediction of diabetes. The platform we have used to implement the models was presented in section 4.2. Section 4.3 has illustrated the optimal PSO parameters that we have obtained by applying them with different values as presented in Appendix A. Both sections 4.4 and 4.5 presented and discussed the results obtained by applying the PSO-FFNN model to both the DataPal and the PIDD datasets. In addition to the results of applying various machine learning algorithms such as SVM, NB, DT, DA and RF to both the DataPal and the PIDD datasets. In short, this chapter showed the ability of the PSO-FFNN model in diabetes predictions. where it was able to outperform other applied models.

Chapter 5

Conclusion and Future Works

Conclusion and Future Works

Taking into account all risks of diabetes, early stages and type of disease must be detected early. Diabetes can be controlled whenever it is detected in the early stages. Also, the correct diagnosis of the type of disease plays a major role in reducing the symptoms of the disease.T1DM is the most dangerous level of diabetes as it is the reason why diabetes is called the silent killer. This type of diabetes cannot be prevented and can only be treated with insulin injections. So, the diagnosis of the type of diabetes must be very accurate.

Using the artificial neural networks model can be designed and implemented for complex medical processes by software. The software systems are more effective and efficient in various medical fields including predict, diagnose, treat and help to the surgeons and physicians and the general population. This research aimed to determine effective variables and their impact on diabetes and estimating a neural network hybrid model with particle swarm optimization to Predict and classify diabetes.

Thus, in this thesis, we have collected a local Palestinian dataset "DataPal" With the assistance of the Palestinian Diabetes Institute. The dataset consists of 314 instances of diabetic females. Where the females are more likely to die due to diabetes compared to males, so we gave them priority in this thesis.

Thus, our samples were for females with both type 1 and 2 diabetes, aged between 5 and 89 years. For the first time, a Palestinian dataset was applied using machine learning algorithms to predict diabetes. The collected dataset was used to train the proposed PSO-FFNN model in addition to a set of machine learning algorithms.

The parameters of the PSO optimization algorithms including the number of the PSO particles, The values of local and global cognitive parameters c1 and c2, and the boundaries of the PSO search space were optimized. Then the optimized PSO algorithm has been used to adjust the FFNN weights values. Both the local DataPal and the Global PIDD datasets were applied using the PSO-FFNN and some of the machine learning algorithms that have been applied in many previous works. In conclusion, The PSO-FFNN model has proved its ability in predicting each of T1DM and T2DM with an accuracy of 99.57%. The PSO-FFNN model has outperformed each of the MLP-BPNN, SVM, K-NN, DT, DA, NB, and RF algorithms in classifying T1DM and T2DM, where each of the MLP-BPNN and the SVM have gained an accuracy of 98.3% for both of them which is the highest value compared to the other models applied in this thesis (i.e. K-NN, DT, DA, NB, and RF). Also, it was best to diagnose if a person has or does not have diabetes as shown in the PIDD experiment results. Given the results obtained, the PSO-FFNN classifier to identify diabetic people is proposed as a useful tool to help the diabetes specialist in the early detection of the disease and to confirm their diagnosis. In particular, the level of success of the system in diagnosis is very high, depending on the patient's symptoms and tests results that are analyzed.

In future work, as the reader could see in the conclusions the challenge for the future is extensive, but promising in this area and could impact the health system in a good way. We plan to develop a medical application that will help people with a family history of diabetes who suspects they are based on linking machine learning algorithms to provide them with

preventive treatments and advice. The Fuzzy-Rule Based system will be used with the PSO-FFNN model to predict diabetes and give preventive treatments automatically. The application will be supervised by doctors and diabetes specialists to build a medical database containing correct preventive treatments. One possibility of future work is to evaluate the possibility of implementing classification and prediction models in the public health system. In addition, developing a prediction and classification system with real-time remote monitoring applications, measures the effective cost of implementing them and the impact they can have on people's lives.

Bibliography

[1] Who.int. (2019). *Diabetes*. [online] Available at: http://www.who.int/en/news-room/fact-sheets/detail/diabetes [Accessed 31 Oct. 2019].

[2] Krasteva, A., Panov, V., Krasteva, A., Kisselova, A., & Krastev, Z. (2011). Oral Cavity and Systemic Diseases—Diabetes Mellitus. *Biotechnology & Biotechnological Equipment*, *25*(1), 2183–2186. doi: 10.5504/bbeq.2011.0022.

[3] Iancu, I., Mota, M., & Iancu, E. (2008). Method for the analysing of blood glucose dynamics in diabetes mellitus patients. *2008 IEEE International Conference on Automation, Quality and Testing, Robotics*. doi: 10.1109/aqtr.2008.4588883.

[4] Cox, M. E., & Edelman, D. (2009). Tests for Screening and Diagnosis of Type 2 Diabetes. *Clinical Diabetes*, *27*(4), 132–138. doi: 10.2337/diaclin.27.4.132.

[5] American Diabetes Association. (2013, January 1). Diagnosis and Classification of Diabetes Mellitus. Retrieved March 14, 2020, from https://care.diabetesjournals.org/content/36/Supplement_1/S67

[6] Robertson, G., Lehmann, E. D., Sandham, W., & Hamilton, D. (2011). Blood Glucose Prediction Using Artificial Neural Networks Trained with the AIDA Diabetes Simulator: A Proof-of-Concept Pilot Study. *Journal of Electrical and Computer Engineering*, *2011*, 1–11. doi: 10.1155/2011/681786

[7] Lee, B. J., & Kim, J. Y. (2016). Identification of Type 2 Diabetes Risk Factors Using Phenotypes Consisting of Anthropometry and Triglycerides based on Machine Learning. *IEEE Journal of Biomedical and Health Informatics*, *20*(1), 39–46. doi: 10.1109/jbhi.2015.2396520.

[8] Alghamdi, M., Al-Mallah, M., Keteyian, S., Brawner, C., Ehrman, J., & Sakr, S. (2017). Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford ExercIse Testing (FIT) project. *Plos One*, *12*(7). doi: 10.1371/journal.pone.0179805.

[9] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine Learning and Data Mining Methods in Diabetes Research. *Computational and Structural Biotechnology Journal*, *15*, 104–116. doi: 10.1016/j.csbj.2016.12.005.

[10] Xiong, X.-L., Zhang, R.-X., Bi, Y., Zhou, W.-H., Yu, Y., & Zhu, D.-L. (2019). Machine Learning Models in Type 2 Diabetes Risk Prediction: Results from a Cross-sectional Retrospective Study in Chinese Adults. *Current Medical Science*, *39*(4), 582–588. doi: 10.1007/s11596-019-2077-4.

[11] Razavian, N., Blecker, S., Schmidt, A. M., Smith-Mclallen, A., Nigam, S., & Sontag, D. (2015). Population-Level Prediction of Type 2 Diabetes From Claims Data and Analysis of Risk Factors. *Big Data*, *3*(4), 277–287. doi: 10.1089/big.2015.0020

[12] Editor Posted on 15th January 2019, & Editor. (2019, April 6). Over 400 people are diagnosed with diabetes in UK every day. Check the risk factors of type 2 diabetes and its links to genes, ethnicity, obesity & diet. Retrieved March 14, 2020, from https://www.diabetes.co.uk/type2-diabetes.html

[13] Diabetes. (2018, August 8). Retrieved March 14, 2020, from https://www.mayoclinic.org/diseases-conditions/diabetes/diagnosis-treatment/drc-20371451

[14] Vaishali, R., Sasikala, R., Ramasubbareddy, S., Remya, S., & Nalluri, S. (2017). Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset. *2017 International Conference on Computing Networking and Informatics (ICCNI)*. doi: 10.1109/iccni.2017.8123815

[15] Choubey, D., Paul, S., Kumar, S., & Kumar, S. (2016). Classification of Pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection. *Communication and Computing Systems*. doi: 10.1201/9781315364094-82

[16] Jayalakshmi, T., & Santhakumaran, A. (2010). A Novel Classification Method for Diagnosis of Diabetes Mellitus Using Artificial Neural Networks. *2010 International Conference on Data Storage and Data Engineering*. doi: 10.1109/dsde.2010.58.

[17] Karegowda, A. G., Manjunath, A., & Jayaram, M. (2011). Application of Genetic Algorithm Optimized Neural Network Connection Weights for Medical Diagnosis of

PIMA Indians Diabetes. *International Journal on Soft Computing*, *2*(2), 15–23. doi: 10.5121/ijsc.2011.2202

[18] Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018). Predicting Diabetes Mellitus With Machine Learning Techniques. *Frontiers in Genetics*, *9*. doi: 10.3389/fgene.2018.00515

[19] Kohavi, R.(1996) Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. Portland, OR: AAAI Press, pp. 202–207.

[20] Salzberg, S.L. Book Review: *C4.5: Programs for Machine Learning* by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning* 16, 235–240 (1994).

[21] Sharma, S., Agrawal, J., & Sharma, S. (2013). Classification Through Machine Learning Technique: C4. 5 Algorithm based on Various Entropies. *International Journal of Computer Applications*, *82*(16), pp.28–32. doi: 10.5120/14249-2444

[22] Marx, V. (2013). The big challenges of big data. *Nature*, 498(7453), pp.255-260.

[23] Mattmann, C. A. (2013). Computing: A vision for data science. Nature, 493, 473-475.

[24] Uci. (2016, October 6). Pima Indians Diabetes Database. Retrieved March 14, 2020, from https://www.kaggle.com/uciml/pima-indians-diabetes-database

[25] Choubey, D. K., Paul, S., & Dhandhania, V. K. (2018). GA_NN: An Intelligent Classification System for Diabetes. *Advances in Intelligent Systems and Computing Soft Computing for Problem Solving*, 11–23. doi: 10.1007/978-981-13-1595-4_2

[26] Roche, M. M., & Wang, P. P. (2013). Sex Differences in All-Cause and Cardiovascular Mortality, Hospitalization for Individuals With and Without Diabetes, and Patients With Diabetes Diagnosed Early and Late. *Diabetes Care*, *36*(9), 2582–2590. doi: 10.2337/dc12-1272

[27] Arnetz, L., Ekberg, N. R., & Alvarsson, M. (2014). Sex differences in type 2 diabetes: focus on disease course and outcomes. *Diabetes, Metabolic Syndrome and Obesity: Targets and Therapy*, 409. doi: 10.2147/dmso.s51301

[28] Simmons, Hannah. "Diabetes in Men versus Women." *News*, 26 Feb. 2019, www.news-medical.net/health/Diabetes-in-Men-versus-Women.aspx.

[29] García, Salvador, et al. *Data Preprocessing in Data Mining*. Vol. 72, Springer, 2015.

[30] Ghaddar, Bissan, and Joe Naoum-Sawaya. "High Dimensional Data Classification and Feature Selection Using Support Vector Machines." *European Journal of Operational Research*, vol. 265, no. 3, 2018, pp. 993–1004., doi:10.1016/j.ejor.2017.08.040.

[31] Chang, Chih-Chung, and Chih-Jen Lin. "Libsvm." *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, Jan. 2011, pp. 1–27., doi:10.1145/1961189.1961199.

[32] Hal Daume III. A course in machine learning. ´ http://ciml.info/dl/v0_8/ciml-v0_8-ch05.pdf, 2012. Accessed: 2020-03-14, p.69.

[33] Tiwari, Sheela, et al. "Comparative Study of Backpropagation Algorithms in Neural Network Based Identification of Power System." *International Journal of Computer Science and Information Technology*, 5(4), 2013, pp. 93–107., doi:10.5121/ijcsit.2013.5407.

[34] Møller, M. F. (1993). *A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks, 6(4), 525–533.* doi:10.1016/s0893-6080(05)80056-5

[35] Hagan, M., & Menhaj, M. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, *5*(6), 989–993. doi: 10.1109/72.329697

[36] Hanan A, R, A., Firas R. M. (2016). Evolutionary Algorithms For Neural Networks Binary And Real Data Classification. International Journal of Scientific & Technology Research, 5(7), 55-60.

[37] Panda, M., & Abraham, A. (2014). Hybrid evolutionary algorithms for classification data mining. *Neural Computing and Applications*, *26*(3), 507–523. doi: 10.1007/s00521-014-1673-2

[38] Mitra, S., Pal, S., & Mitra, P. (2002). Data mining in soft computing framework: a survey. *IEEE Transactions on Neural Networks*, *13*(1), 3–14. doi: 10.1109/72.977258

[39] Ateeq K and Ganapathy G. (2017). The novel hybrid Modified Particle Swarm Optimization - Neural Network (MPSONN) Algorithm for classifying the Diabetes. Comput Intell. 13(4), 595-614.

[40] Sivagowry S, Dr. Durairaj M. (2015). An Intelligent Hybrid Quick Reduct Particle Swarm Optimization Algorithm for Feature Reduction in Cardiac Disease Prediction. International Journal of Emerging Technologies in Computational and Applied Sciences, 12(2), pp 163-173.

[41] Unler, A., Murat, A., & Chinnam, R. B. (2011). mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences*, *181*(20), 4625–4641. doi: 10.1016/j.ins.2010.05.037

[42] Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, *4*(0), 40–79. doi: 10.1214/09-ss054

[43] Burman, P. (1989). A Comparative Study of Ordinary Cross-Validation, v-Fold Cross-Validation and the Repeated Learning-Testing Methods. *Biometrika*, *76*(3), 503. doi: 10.2307/2336116

[44] Geisser, S. (1975). The Predictive Sample Reuse Method with Applications. *Journal of the American Statistical Association*, *70*(350), 320–328. doi: 10.1080/01621459.1975.10479865

[45] S. Wang, L. L. Minku, and X. Yao.(2016) “Dealing with multiple classes in online class imbalance learning,” in Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI), 2016, pp. 2118–2124

[46] J. Akosa, "Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data," in Proceedings of the SAS Global Forum, 2017.

[47] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), pp.861-874.

[48] McCaffrey 12/18/2013, J. (n.d.). Neural Network Training Using Particle Swarm Optimization. Retrieved March 14, 2020, from https://visualstudiomagazine.com/articles/2013/12/01/neural-network-training-using-particle-swarm-optimization.aspx

[49] X. Hu and R. Eberhart, "Solving Constrained Nonlinear Optimization Problems," in Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics, pp. 203-206, Orlando, Florida, USA, 2002.

[50] Krati Saxena, D., Khan, Z., Singh, S.: Diagnosis of diabetes mellitus using K nearest neighbor algorithm. Int. J. Comput. Sci. Trends Technol. (IJCST) 2(4) (2014)

[51] Web.stanford.edu. (2019). [online] Available at: https://web.stanford.edu/~hastie/Papers/LARS/diabetes.data [Accessed 14 Mar. 2020].

[52] P. C. Thirumal and N. Nagarajan, ‘‘Utilization of data mining techniques for diagnosis of diabetes mellitus—A case study,’’ ARPN J. Eng. Appl. Sci., 10(1), pp. 8–13, 2015.

[53] A. Parashar, K. Burse, and K. Rawat, ‘‘A comparative approach for Pima Indians diabetes diagnosis using LDA-support vector machine and feed forward neural network,’’ Int. J. Adv. Res. Comput. Sci. Softw. Eng., 4(11), pp. 378–383, 2014.

[54] V.A. Kumari, R. Chitra, “Classification of diabetes disease using support vector machine,” Int. J. Eng. Res. Appl., 3(2), pp.1797-1801, 2013.

[55] Sarwar, M. A., Kamal, N., Hamid, W., & Shah, M. A. (2018). *Prediction of Diabetes Using Machine Learning Algorithms in Healthcare. 2018 24th International Conference on Automation and Computing (ICAC).* doi:10.23919/iconac.2018.8748992

[56] Vijayan, V. V., & Anjali, C. (2015). Prediction and diagnosis of diabetes mellitus — A machine learning approach. *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. doi: 10.1109/raics.2015.7488400

[57] Minyechil Alehegn, Rahul Joshi and Preeti Mulay. (2018). Analysis and Prediction of Diabetes Mellitus using Machine Learning Algorithm. International Journal of Pure and Applied Mathematics, 118(9), pp. 871-878.

[58] Karegowda, A. G., Manjunath, A., & Jayaram, M. (2011). Application of Genetic Algorithm Optimized Neural Network Connection Weights for Medical Diagnosis of PIMA Indians Diabetes. *International Journal on Soft Computing*, *2*(2), 15–23. doi: 10.5121/ijsc.2011.2202.

[59] Farooqui, N., ., R., & Tyagi, A. (2018). Prediction Model for Diabetes Mellitus Using Machine Learning Techniques. *International Journal of Computer Sciences and Engineering*, *6*(3), 292–296. doi: 10.26438/ijcse/v6i3.292296

[60] G.Kranthi kumar ,K.Swathi. (2017). PERFORMANCE IMPROVEMENT APPROACH FOR DIABETES DISEASE PREDICTION. International Journal of Computer Application, 7(1), 2250-1797.

[61] Alaa Badr Eysa, Asst Prof. Dr. Sefer Kurnaz. (2019). DIABETES DIAGNOSIS USING MACHINE LEARNING. International Journal of Computer Science and Mobile Computing, 8(3), pp. 36-41.

[62] Sejdinović, D., Gurbeta, L., Badnjević, A., Malenica, M., Dujić, T., Čaušević, A., … Mehmedović, L. D. (2017). CLASSIFICATION OF PREDIABETES AND TYPE 2 DIABETES USING ARTIFICIAL NEURAL NETWORK. CMBEBIH 2017, 685–689. doi:10.1007/978-981-10-4166-2_103

[63] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall.(2018). Activation functions: Comparison of trends in practice and research for deep learning, arXiv preprint arXiv:1811.03378.

[64] Punthakee, Z., Goldenberg, R., & Katz, P. (2018). Definition, Classification and Diagnosis of Diabetes, Prediabetes and Metabolic Syndrome. *Canadian Journal of Diabetes*, *42*. doi: 10.1016/j.jcjd.2017.10.003

[65] Dresslerova I., Vojacek J.(2010). Diabetes mellitus and ischemic heart disease. Vnitr Lek. 56(4). pp301–306.

[66] Fadhil J. Al-Tu'ma, Baqer A. Joda and Raghad A. Al-Yassiry (2018). Assessment of Vascular Endothelial Growth Factor-A and Insulin Resistance in Sera of Ischemic Heart Diseases with Type-II Diabetic Patients. Indian Journal of Natural Sciences, 9(51), pp 976-997.

[67] Diabetes. (2018, August 8). Retrieved March 14, 2020, from https://www.mayoclinic.org/diseases-conditions/diabetes/diagnosis-treatment/drc-20371451#targetText=Fasting blood sugar test.&targetText=A fasting blood sugar level,separate tests, you have diabetes.

[68] High Blood Pressure. Retrieved March 14, 2020, from https://www.diabetes.org/diabetes-risk/prevention/high-blood-pressure.

[69] Wako, A., Belay, S., Feleke, Y., & Kebede, T. (2017). Assessment of the Magnitude, Severity and Associated Factors of Hypoglycemia in Diabetic Patients Attending

[70] National Diabetes Referral Clinic at Tikur Anbessa Hospital, Addis Ababa, Ethiopia. *Journal of Diabetes & Metabolism*, *08*(05). doi: 10.4172/2155-6156.1000741.

Home. Retrieved March 14, 2020, from https://www.idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html.

[71] Nuttall, F. Q. (2015). *Body Mass Index. Nutrition Today, 50(3), 117–128.* doi:10.1097/nt.0000000000000092.

[72] Wu, C., & Chau, K. (2013). Prediction of rainfall time series using modular soft computing methods. *Engineering Applications of Artificial Intelligence*, *26*(3), 997–1007. doi: 10.1016/j.engappai.2012.05.023.

[73] Jaiswal, J. K., & Das, R. (2018). Artificial Neural Network Algorithms based Nonlinear Data Analysis for Forecasting in the Finance Sector. *International Journal of Engineering & Technology*, *7*(4.10), 169. doi: 10.14419/ijet.v7i4.10.20829.

[74] Durgalakshmi, B., & Vijayakumar, V. (2020). Feature selection and classification using support vector machine and decision tree. *Computational Intelligence*. doi: 10.1111/coin.12280.

[75] Gestational Diabetes Mellitus. (2003, January 1). Retrieved March 15, 2020, from https://care.diabetesjournals.org/content/26/suppl_1/s103

Appendix

This section includes the results obtained while optimizing the PSO-FFNN model parameters.

Appendix A

This section contains the results obtained during the process of optimizing the parameters of the PSO algorithm that have been used to train an FFNN. The DataPal dataset has been used to generate the following results.

Table A.1: The Accuracy vs. social and cognitive weights results of applying Two-fold cross-validation.

c1 and c2 value	Iteration 1 Accuracy	Iteration 2 Accuracy
1	98.1%	98.1%
1.9	96.83%	97.46%
1.2	96.83%	95.56%
1.3	98.1%	98.1%
1.4	98.73%	97.46%
1.5	96.83%	97.46%
1.6	98.1%	98.1%
1.7	98.73%	98.1%
1.8	97.46%	98.1%
1.9	98.73%	97.46%
2	98.73%	98.1%

Table A.2: The Accuracy vs. social and cognitive weights results of applying Four-fold cross-validation.

c1 and c2 value	Iteration 1 Accuracy	Iteration 2 Accuracy	Iteration 3 Accuracy	Iteration 4 Accuracy
1	98.3%	96.18%	96.18%	95.33%
1.1	95.76%	96.18%	94.91%	95.33%
1.2	99.15%	96.18%	96.18%	95.33%
1.3	98.72%	97.03%	97.45%	96.18%
1.4	98.72%	97.45%	96.6%	96.18%
1.5	98.3%	96.6%	96.6%	97.45%
1.6	98.72%	96.6%	96.6%	96.18%

1.7	98.72%	96.6%	96.6%	97.03%
1.8	99.15%	96.6%	97.03%	97.45%
1.9	98.72%	96.6%	97.03%	97.45%
2	99.58%	97.45%	97.45%	97.03%

Table A.3: The Accuracy vs. # of PSO particles results of applying Two-fold cross-validation.

# of PSO Particles	Iteration 1 Accuracy	Iteration 2 Accuracy
5	98.1%	98.1%
10	97.46%	97.46%
15	97.46%	97.46%
20	97.46%	98.1%
25	97.46%	97.46%
30	98.73%	98.1%
35	98.1%	97.46%
40	99.36%	98.1%
45	98.1%	98.1%
50	98.1%	98.1%

Table A.4: The Accuracy vs. # of PSO particles results of applying Four-fold cross-validation.

# of PSO Particles	Iteration 1 Accuracy	Iteration 2 Accuracy	Iteration 3 Accuracy	Iteration 4 Accuracy
5	97.87%	97%	97.46%	94%
10	99.14%	97.46%	96.18%	97.46%
15	99.14%	97.46%	97.03%	97.46%
20	98.72%	96.6%	96.18%	93.64%
25	99.14%	97.88%	96.6%	94.49%
30	99.57%	97.45%	97.46%	94.49%
35	99.14%	97.88%	97.03%	95.33%
40	99.14%	97.88%	97.46%	94.49%
45	99.14%	97.46%	97.46%	93.22%
50	99.57%	97.46%	97.46%	94.49%

Table A.5: The Accuracy vs. search space boundaries LB and UB results of applying Two-fold cross-validation.

LB and UB values	Iteration 1 Accuracy	Iteration 2 Accuracy	Iteration 3 Accuracy	Iteration 4 Accuracy
0.5 : - 0.5	99.15%	98.3%	97.44%	97.88%
0.1 : - 0.1	99.15%	97.45%	97.02%	97.88%
1.5 : - 1.5	99.58%	97.03%	97.44%	97.03%
2 : - 2	98.73%	97.03%	97.87%	97.88%

Table A.6: The Accuracy vs. search space boundaries LB and UB results of applying Four-fold cross-validation.

LB and UB values	Iteration 1 Accuracy	Iteration 2 Accuracy
0.5 : - 0.5	97.46%	98.1%
0.1 : - 0.1	98.73%	98.1%
1.5 : - 1.5	98.73%	98.1%
2 : - 2	98.1%	97.46%

ذلك قد أثبت النموذج المقترح انه الافضل في تشخيص وجود مرض السكري و في تصنيفه الى النوعين "T1DM " و "T2DM ".

الملخص

أحد أهم القضايا التي يتوجب أخذها في عين الاعتبار عند تطوير حلول للمشكلات الصحية هي قضية التحكم في داء السكري الذي بات يعرف كأحدِ أكثرِ عشرة أمراضٍ فتكا حول العالم. تم اعتبار داء السكري على أنه وباء قاتل لذلك يتوجب الكشف المبكر عنه لمساعدة حامله في التحكم فيه. قامت العديد من الدراسات السابقة بتطبيق عدة نماذج التعلم الآلي "Machine Learning " لتحسين القدرة على الكشف عن وجود أو عدم وجود مرض السكري لدى الأشخاص. في المقابل هناك قلة قاموا باستخدام نماذج التعلم الآلي للتنبؤ بالنوع الثاني من مرض السكري "T2DM". حيث ينقسم حاملو مرض السكر الى عدة أنواع وأكثرها انتشارا هما النوع الأول "T1DM " و النوع الثاني من المرض"T2DM ". حيث يعد النوع الأول هو النوع الأكثر خطورة على حياة الإنسان من ما يسببه من مضاعفات وأمراض نتيجة انخفاض مستويات هرمون الأنسولين في الجسم. حيث يكون البنكرياس غير قادر على إنتاج هرمون الأنسولين الضروري للاستفادة من جلوكوز الدم و تحويله الى طاقة. و يعتبر التشخيص الخاطئ لاحد انواع المرض أمر خطير قد لا يتم تداركه و الذي يتسبب بعدم تزويد المريض بالعلاج الوقائي المناسب. نظرا لذلك، قمنا في هذه الاطروحة بجمع مجموعة بيانات محلية فلسطينية "Data Pal " لاستخدامها في تدريب و تقييم أداء نموذجنا الهجين المقترح بالاضافة الى مجموعة بيانات عالمية اخرى "PIDD " لتصنيف وجود المرض و عدمه. في هذه الاطروحة قدمنا نهج ما قبل المعالجة "Preprocessing " الذي يعتمد على تقدير القيم المفقودة داخل مجموعات البيانات باستخدام خوارزمية "K-nn " بالاضافة الى خوارزمية "SVM " التي تم استخدامها لاختيار اكثر الميزات " Dataset attributes " تأثيرا في نتيجة التنبؤ. كما تم استخدام النموذج الهجين "PSO-FFNN" الذي يعمل على بناء حلول تساعد على تحسين دقة تشخيص مرض السكري وأنواعه. تم تطبيق العديد من خوارزميات التعلم الالى لاثبات قدرة نموذج "PSO-FFNN " على التنبؤ بمرض السكري و انواعه. تم تطبيق قاعدة البيانات "PIDD " لإثبات قدرة النموذج المقترح على تشخيص وجود أو عدم وجود مرض السكري لدى المرضى. كانت خوارزمية "MLP-BPNN " الأكثر منافسة و بالرغم من