Arab American University

Faculty of Graduate Studies

Classification of Chronic Kidney Disease Using Hybrid Models of Neural Networks and Evolutionary Algorithms

By

Sherin Anan "Hussein Ali"

Supervisor

Prof. Dr. Mohammed Awad

This thesis was submitted in partial fulfillment of the requirements for the Master's degree in Computer Science

6 / 2024

I

Thesis Approval

Classification of Chronic Kidney Disease Using Hybrid Models of Neural Networks and Evolutionary Algorithms

By
Sherin Anan "Hussein Ali"

This thesis was defended successfully on 6/27/2024 and approved by:

Committee members	Signature
1. Prof. Mohammed Awad: Supervisor	
2. Dr. Ahmad Ewais: Internal Examiner	
3. Dr. Thaer Samar: External Examiner	

.

Declaration

I declare that, except where explicit reference is made to the contribution of others, this thesis is substantially my own work and has not been submitted for any other degree at the Arab American University or any other institution.

The Name of The Student: Sherin Anan Abd Al Karim “Hussein Ali”

ID: 201912780

Signature: Sherin Hussein 

Date: 9/23/2024

Dedication

To my husband Sami, daughters Rima and Razan who give my life another meaning.

To my father, mother, sisters and brothers for their love, support that gave to me and for standing with me at all times.

Also, I dedicate this work to my supervisor, Prof. Mohammed Awad for all the support.

Acknowledgments

I would like to take this opportunity to express my deep regards to Prof. Mohammed Awad for his advice, support, and time that he spent reviewing my work. Prof. Mohammed provided valuable suggestions that have had a significant impact and helped in overcoming many obstacles in writing this thesis in the best way.

Abstract

Chronic Kidney Disease (CKD) is a risk factor for cardiovascular disease and has a significant economic impact on healthcare systems. Early detection of chronic kidney disease can save a person's life from a heart attack. Artificial Intelligence (AI) has emerged as a new tool that helps in the early detection of disease and predicting its occurrence. AI has a good impact on saving lives, providing a treatment plan for the disease, and conducting more developmental research.

Various artificial intelligence techniques can be used to classify and predict chronic kidney diseases by applying them to medical data. This thesis presents hybrid models combining evolutionary algorithms, neural networks, and machine learning techniques to classify chronic kidney disease. In this thesis, global and local datasets were used as well. In the first stage, several different machine learning models were applied to datasets, including Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Multi-Layer Neural Networks (MLPNNs). In the second stage, several hybrid models of evolutionary algorithms, including biogeography-based optimization (BBO), particle swarm optimization (PSO), and genetic algorithms (GAs) were trained on multi-layer neural networks (MLPNNs) to obtain the best results in CKD classification.

The DT, SVM, KNN, and MLPNNs models were applied to the global dataset for chronic kidney disease classification, and revealed accuracy results of 97%, 99.5%, 98.2%, and 99.8%, respectively. Furthermore, the MLPNNs and SVM models showed the highest accuracy and the best models in classification with close accuracy rates. These models were also applied to the Palestinian local dataset to classify chronic kidney diseases, and the accuracy results obtained were: 96.4%, 96.2%, 93.6%, and 98.1%, respectively, in the same order mentioned previously. The MLPNNs model revealed the highest accuracy and was the best model in classification followed by the DT model.

The experimental results in applying the hybrid models to the global dataset showed that both GAs-MLPNNs and BBO-MLPNNs were almost similar in performance, with the results for GAs-MLPNNs being: 99.5%, 99.6%, 99.3%, 99.6%, and 99.6% accurate, sensitive, specific, precise, and F-scored, respectively. In addition, the results in applying the hybrid models to the local dataset of both the GAs-MLPNNs and BBO-MLPNNs were also similar in performance, as they were: 99%, 99%, 99.1%, 99.2%, and 99.1% accurate, sensitive, specific, precise, and F-scored, respectively.

Table of Contents

List of Tables

List of Figures

Lists of Abbreviations

CKD	Chronic kidney disease
DT	Decision Tree
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
NN	Neural Networks
MLPNNs	Multi-Layer Perceptron Neural Networks
ANNs	Artificial neural network
ML	Machine Learning
EAs	Evolutionary algorithms
BBO	Biogeography-Based Optimization
PSO	Particle Swarm Optimization
GAs	Genetic Algorithms
MI	Mutual information
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ROC	Receiver Operating Characteristic Curve
AUC	Area under the curve
ALT	Alanine Transaminase
AST	Aspartate Amino Transferase
BG	Blood Glucose
BUN	Blood Urea Nitrogen
HGB	Hemoglobin
SIVs	Suitability Index Variables
HSI	Habitat Suitability Index
BBO-MLPNNs	Biogeography Based Optimization-Multi-Layer Perceptron Neural Networks
PSO-MLPNNs	Particle Swarm Optimization-Multi-Layer Perceptron Neural Networks
GAs-MLPNNs	Genetic Algorithms- Multi-Layer Perceptron Neural Networks
CDSS	clinical decision support system

Chapter 1

1.1 Introduction

Chronic kidney disease (CKD) is a common Systemic disease that poses a threat to human life and poses a major health care challenge, it affects about 8–16% of the world population [1][2]. And considered as the ninth cause of death in Palestine, with a rate of 3.2% [3], and the eighteenth cause of death in the world [4]. Chronic kidney disease is a disease prevalent in society, with one out of every seven people in the United States suffering from chronic kidney disease at a rate of 15% of adults in the United States, and this percentage is more common in women 14% than men 12% [5]. Chronic kidney disease is a progressive disease with no cure. It causes high morbidity and mortality rates that occur commonly in people with diabetes and high blood pressure. Patients with this pathology remain asymptomatic most of the time, presenting the complications of renal dysfunction only in more advanced stages. It may be that a person with chronic kidney disease does not know that he has the disease, and this is considered one of the challenges in detecting this disease because of its slow development over months or years, meaning that symptoms appear on a person when the disease has reached advanced stages or that kidney functions have been significantly affected. Therefore, it is necessary to do a periodic examination, whether the person has symptoms or not.

Chronic kidney disease (CKD) is a deterioration in the ability of the kidneys to perform their functions as a result of a malfunction, and the wastes and fluids accumulate inside the kidneys and are excreted in the blood, and from here the symptoms of kidney failure begin. In addition to high blood pressure, the presence of stones and cancerous tumors in the kidneys are reasons that lead to kidney failure. Kidney failure results in decreased urine output, nausea, and vomiting. Decreased appetite, tiredness and weakness, difficulty sleeping, muscle cramps, and swollen ankles and feet [6]. The kidney is one of the most important parts of the body, as any malfunction in its function leads to many health problems. The appearance of some symptoms

such as fluid retention and decreased urine output is the beginning of a diagnosis of the disease, but the best diagnosis is after laboratory tests such as urine test, blood analysis, sodium, and potassium analysis, and many other tests [7] [8][9].

Physicians deduce the disease through medical tests and the patient's medical history manually, which takes a long time and sometimes causes errors in diagnosis. Late diagnosis of the disease or delay in treatment leads to complications, including fluid retention that may lead to body swelling, heart disease, anemia, and irreversible kidney damage, which eventually requires either dialysis or a kidney transplant to survive. But there are some cases in which kidney function is completely lost and eventually leads to death. Therefore, there is a need for accurate diagnosis and early detection of this disease, so artificial intelligence can help doctors and researchers detect chronic kidney disease before it increases the severity of the disease and affects other parts of the body.

The emergence of artificial intelligence (AI) at present has contributed to the development in many fields, including medicine and health. Thus, the development of computing has facilitated the discovery of patterns using biostatistics through the analysis of big data quickly and with high accuracy using algorithms [10]. The medical sector was able to apply AI to the huge data in this sector and helped in diagnosis, disease prevention, decision-making, medical consultations, providing treatment plans, and developing them through that patient data. Intelligence was also used in the manufacture of medical devices that intervene in surgical operations [11]. AI has now become a means of developing aspects of life, including the field of medicine, where the computer has become capable of thinking, making decisions, and the ability to collect digital medical data, medical statistics, health care, personalized medicine, patient care, improving the level of accuracy and efficiency of diagnosis and treatment decisions [12][13], and also use robots to perform medical operations [14].

Machine learning (ML) techniques focuses on developing algorithms and models to teach computers to perform tasks such as prediction and decision-making by learning patterns and relationships of datasets to solve problems and identify patterns [15][16][17]. ML helped in analyzing and processing large datasets and image-based data processing, which reduced human effort and time. ML has become important for artificial intelligence, it is present in many applications and scientific fields, including the medical field, such as algorithms that aim to classify diseases, predict them, and suggest a treatment plan [18]. That is what this thesis aims to achieve. This thesis proposes to develop some hybrid models that combine evolutionary algorithms (EAs) and neural networks (NNs) to obtain the best results in classifying CKD. We applied these models to two datasets, which are a global dataset and a Palestinian local dataset. The datasets included data for laboratory medical examinations of patients.

In the first phase of this research, ML techniques have also been used to classify and predict KCD in the early stages to take immediate action for treatment, and these techniques have been applied to the same datasets. These techniques are: Decision Tree (DT) [19], Support Vector Machine (SVM) [20], k-nearest Neighbor (KNN) [21], and artificial neural network model (ANNs) [22]. In the second phase, Evolutionary algorithms (EAs) are combined with multi-layer perceptron neural network (MLPNNs) with the aim of improving the classification result of CKD in a global and local dataset. EAs used to optimize weights of MLPNNs, which leads to better performance accuracy of the CKD classification. In this research three different methods of EAs have been used, biogeography-based optimization (BBO) which was developed based on the biogeography and distribution of biological species in nature across time and space [23], particle swarm optimization (PSO)which was developed based on swarm behavior, such as the behavior of swarms of birds and fish in learning in nature [24] and genetic algorithms (GAs) which developed to find solutions to research problems, it is inspired by genetics, the process of natural selection, and biology such as mutation and inheritance. [25].

1.2 Aims and Objectives

The main objective of this thesis is to improve the accuracy in diagnosing CKD applying different ML techniques and hybrid techniques of EAs and MLPNNs using a global and local dataset. Feature selection techniques was applied to identify variables that are effective in disease classification. Different ML models which are DT, SVM, KNN, and MLPNN were applied. On other hand, hybrid models of NNs and EAs, such that BBO, PSO, and GAs have been applied aiming to improve classification accuracy. The use of these models will contribute to early detection and prediction of CKD, and assist doctors in diagnosis, which reduces the development and exacerbation of diseases in patients.

Specific Objectives of the Study

- Use feature selection techniques to find out which feature in the dataset has the most effective effect on the classification.
- Applying some machine learning techniques to classify chronic kidney disease on local and global datasets to find out the most appropriate techniques in this classification.
- Applying the hybrid models to classify chronic kidney disease on local and global datasets.
- Comparing the results of all the models that are used in terms of classification accuracy and identifying the most appropriate model in classifying chronic kidney disease.

1.3 Contribution

ML techniques were used and several algorithms were proposed to classify and predict CKD. This thesis presents several models to classify and predict CKD using the local dataset and the global dataset, as well as comparing results. Hybrid models of NNs and EAs were used, to

obtain higher diagnostic accuracy, which are genetic algorithms (GAs), biogeography-based optimization (BBO), and particle swarm optimization (PSO). There are no previous studies that applied hybrid models to the Palestinian local dataset to classify and predict chronic kidney diseases. Therefore, the beginning of the thesis was to collect the local dataset that related to CKD then using ML techniques and developed hybrid models to classify and predict CKD. As well as applying these techniques and models to global data, to classify and predict CKD and compare results with each other.

1.4 Overview

The thesis will be arranged and organized as follows: Chapter 1 presents an introduction to the thesis, the importance of the research study, the goals and objectives, the contribution, and a brief overview of the thesis topic. Chapter 2 presents background on work in general and discusses the literature review of related works in CKD diagnosis and some of the techniques used to diagnose, classify, and predict CKD. Chapter 3 presents the methodologies used as follows: First, the proposed method. Secondly, describe the study area and data collection in detail Thirdly, a description of the pre-processing stages, which are feature selection and dealing with missing values. Fourthly, explain the ML models which are DT, SVM, KNN, and MLPNNs. Fifth, explain the hybrid models of NNs with EAs. These algorithms are GAs, BBO, and PSO. Sixthly, different performance measurements were presented to evaluate the models. Seventh, explain the cross-validation. Finally, explain the software that was used in this thesis. Chapter 4 discusses the results of data experiments on different models (ML models and hybrid models), both for local datasets and global datasets. Comparing the results of all the models that apply to the global and local datasets and show the best model in the classification of chronic kidney disease. Chapter 5 presents the conclusion and future work.

Chapter 2

2.1. Background

The health sector relied on health information technology and its development, which in turn led to progress in diagnosing and treating many diseases and limiting their spread in society. Technology also helped in creating big data for patients and systematically analyzing them using the best technologies, which opened the way for new ways of modern health care. [26][27]. The kidney is considered one of the important organs of the body, any defect in one of its functions leads to some health problems and the emergence of some symptoms such as fluid retention and decreased urine production, which is the beginning of the diagnosis of the disease, but the best diagnosis is after laboratory tests [28]. The delay in diagnosis increases the incidence of CKD, which developing countries consider a serious and costly health problem. The possibility of diagnosing CKD using the latest technological methods and artificial intelligence contributes to early detection of the disease and giving a treatment plan for the disease before complications occur or reach an advanced stage of the disease, which is often costly to the patient and society [29][30].

Many scientific studies suggest the use of ML techniques in the classification of chronic diseases, especially CKD. In this work, several ML models that were used in classification were applied to the local dataset and the global datasets of CKD such as, DT, SVM, KNN, and NNs. Also, hybrid models were applied, and the evolutionary algorithms methods such as (GAs, BBO, and PSO) trained a MLPNNs to perform the best result of the accuracy in classification of the disease.

2.2 Related Work

Most of the previous studies that using machine learning in terms of classification and prediction of chronic kidney disease were applied to a global dataset with different variables, most of which were based on medical laboratory tests, and some researchers presented a comparison study of

different types of machine learning algorithms to prove the accuracy of a technique on the dataset, this chapter will present the most relevant studies.

Machine learning algorithms like decision tree (DT), k-nearest neighbors' algorithm (KNN), support vector machines (SVM), naive Bayes (NB), and other algorithms were used in the medical field, especially in classifying and predicting chronic diseases. For example, Poonia et al [31] proposed in their study to use these and other algorithms, in addition to redundant feature removal (RFE). Also, techniques of Chi-Square test feature selection were used to analyze different prediction models on the dataset of healthy and kidney disease patients. The results showed that the prediction model based on logistic regression is the most accurate, with a rate of 98.75 %. Jongbo et al [32] used two ensemble approaches which are the Bagging and Random Subspace methods and applied them to k Nearest Neighbors, Naïve Bayes, and Decision Tree to improve the model's performance in classification. The result showed that the random subspace ensemble on the KNN classifier achieved 100% accuracy of prediction. and the model is appropriate for the diagnosis of chronic kidney disease. Singh et al [33] presented a novel deep learning model for the detection and prediction of CKD that aims to create a deep neural network, they compared the model's performance with other machine learning techniques. The results showed that the proposed model was better than the other classifiers like Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and other classifiers, the accuracy was 100% and it could be a useful technique for detecting CKD. While Gunarathne et al [41] proposed applying some classification algorithms to predict the CKD and non-CKD status of a patient on a dataset obtained from the UCI repository. The results showed that the Multiclass Decision Forest algorithm is the best in classification with an accuracy of 99.1%

Ravindra et al [7] used SVM- neural networks to classify chronic and non-chronic kidney disease with a radial basis kernel function. They considered four cases including the nominal

and numerical values to train the attributes. The result of the maximum classification accuracy was about 93.75% which makes the SVM a good choice for classification of CKD and NCKD. while Polat, H. et al [6] used a Support Vector Machine with feature selection methods to predict chronic kidney disease, they used two methods to reduce the dataset dimension of chronic kidney disease which are wrapper and filter approaches, the result of the accuracy in the diagnosis of chronic kidney disease was 98.5% compared to other selected methods. Tekale et al. [34] in their research, an accuracy prediction for some machine learning models, including Decision Trees (DT), Random Forest (RF), and Support Vector Machines (SVM), were analyzed. The results of the predictions show that the precision rate is 93.08% for SVM and 85.02% for Decision Tree (DT) algorithms. Muntasir Nishat et al. [35] developed eight machine-learning models using the Python language to detect and classify chronic kidney disease. They were applied to a dataset of patients from the University of California's machine learning repository. Model results were compared by evaluating different performance parameters such as accuracy, precision, sensitivity, F1 score, and ROC-AUC. The Random Forest model had the highest accuracy of 99.75%. While Al-Moman et al [36] applied machine learning methods including ANNs, SVMs, and k-Nearest Neighbors (KNN) to provide early diagnosis of CKD. They tested the models on a dataset consisting of 400 samples and 13 variables. The results showed that the ANN model was the best in classification with 99.2% accuracy.

Hybrid models were developed to classify and predict chronic diseases. many researchers suggest hybrid models using the neural network to classify chronic kidney disease, for example, Sankhadeep Chatterjee et al [3] proposed a Neural Network-modified Cuckoo search-based model (NN-MCS) to detect chronic kidney disease and solve the problem of using local search-based learning algorithms to train the NNs. The model was compared with other classifiers which are Multilayer Perceptron Feed-forward Networks (MLP-FFN) and Neural Network

based on Particle Swarm Optimization (PSO-NN). The proposed hybrid model provided the best results in comparison between classifiers in the ability to detect chronic kidney disease, the results of the accuracy were about 99.6% for the hybrid model, 98.5% for PSO-NN, and 96.33% for MLP-FFN.

Mohamed Elhoseny et al [4] presented an intelligent system for predicting chronic kidney disease. They used a density-based Feature Selection (DFS) model to get rid of the irrelevant features then used the Ant Colony based Optimization (D-ACO) algorithm for the selected features to predict chronic kidney disease. The result of the D-ACO algorithm outperformed the other methodologies with improved performance of classification in various aspects. The support vector machine (SVM)has been applied to predict chronic kidney disease. Hore et al [37] proposed a genetic algorithm (GA) trained neural network (NN)-based model to detect chronic kidney disease (CKD). The model solves the issue of using local search-based learning algorithms by using GA to train the NN to optimize the input weight vector of the NN. They compared the model's performance with the performance of other classifiers and the results showed that the proposed model can detect CKD more efficiently compared to other models.

Yadav et al [38] developed a hybrid model that uses a neural network as an ensemble model with different features and techniques to increase the accuracy of the classification of chronic kidney disease. The feature techniques used were Chi-square, Pearson correlation, Extra Tree, and lasso regularization. The model was applied to datasets consisting of 400 samples with 26 features. The neural network ensemble with the Lasso model has the highest accuracy rating (99.98%). Arvind Kumar et al [39] developed a model using classification techniques that combine KNN with particle swarm optimization (PSO). In addition to developing a new model of the fitness function that depends on the distance in the genetic code to deal with the imbalanced dataset that affects classification. The results of the techniques showed that the

proposed fitness function is the best, as its accuracy rate was 99.33% with an AUC value of 0.99. The KNN technique obtained an accuracy of 83.54% with an AUC value of 0.69. The new technique, PSO-KNN, obtained an accuracy of 96.79% with an AUC value of 0.94.

Manonmani and Sarojini [40] in their research, applied the Improved Teacher Learner Based Optimization (ITLBO) algorithm and the original TLBO algorithm to the chronic kidney disease dataset. The ITLBO algorithm aims to select the best subset of features in the dataset. The ITLBO algorithm achieved a feature reduction of 36%. While the percentage of the original TLBO algorithm was 25%. They applied the feature selection algorithms (ITLBO) and (TLBO) to the dataset on Support Vector Machine (SVM), Convolution Neural Networks (CNN), and Gradient Boosting classification algorithms. Notice from the results that the classification accuracy of the algorithms for the feature subset has improved. Medical Images can be used in classifying and diagnosing diseases. Kim et al [42] collected images of patients and used a total of 741 images (251 images of normal kidneys, 328 images of moderate chronic kidney disease, and 162 images of chronic kidney disease). Using the GLCM algorithm in analyzing ultrasound images to determine the important parameters in the image, in their research they used 58 parameters. They used an artificial neural network model (ANN), where the network consisted of 10 hidden layers. 58 input parameters and 3 output layers. The final classification result was 95.4%.

In our work, we used a global dataset and a Palestinian local dataset for chronic kidney disease in applying some machine learning models (DT, SVM, KNN) and hybrid models that combine neural networks with evolutionary algorithms (GA, PSO, BBO) to obtain higher accuracy in classification and prediction of chronic kidney disease. The plan is to create a local dataset from Palestinian hospitals, and then work to generalize the results. So, it can be used for local or international communities and to provide solutions to medical staff and doctors. Most studies

have been applied to predict and classify chronic kidney diseases using some machine learning techniques such as support vector machines and applied to an international dataset, and some researchers have compared the accuracy of the results of the algorithms. There are no previous studies on applying the ML models and hybrid models of NN and EAs which proposed in this thesis to local Palestinian dataset for the diagnosis and classification of CKD. Therefore, this research was presented to develop the diagnosis and classification process for CKD disease through advanced technological.

Chapter 3

3.1 Proposed Method

This chapter describes the methodologies and models that are used to improve the classification accuracy of CKD. This thesis aims to develop some hybrid models that combine EAs and NNs to obtain the best results in CKD classification. In addition to applying different ML techniques to datasets, At the beginning of the work, the first goal was to collect data on CKD, where a local Palestinian dataset and a global dataset were collected, explored, preprocessed, and analyzed. These two datasets were preprocessed and handled with missing data using appropriate methods in both the local and global datasets so that they were ready to be used by the selected models to obtain accurate results. In addition, a process of selecting the appropriate features has been applied to the datasets, which will have an impact on the results. The data was divided into training and testing sets, then ML models (DT, SVM, KNN, and MLPNN) were applied to both datasets, and the results were collected. The proposed hybrid models (BBO-MLPNNs, PSO-MLPNNs, and GAs-MLPNNs), which combine developmental algorithms and NNs, were applied to both datasets, and the model parameters were changed to obtain the best results. Finally, the results of the models will be evaluated and discussed, and then the models will be deployed. The following figure (3.1) shows the system structure.

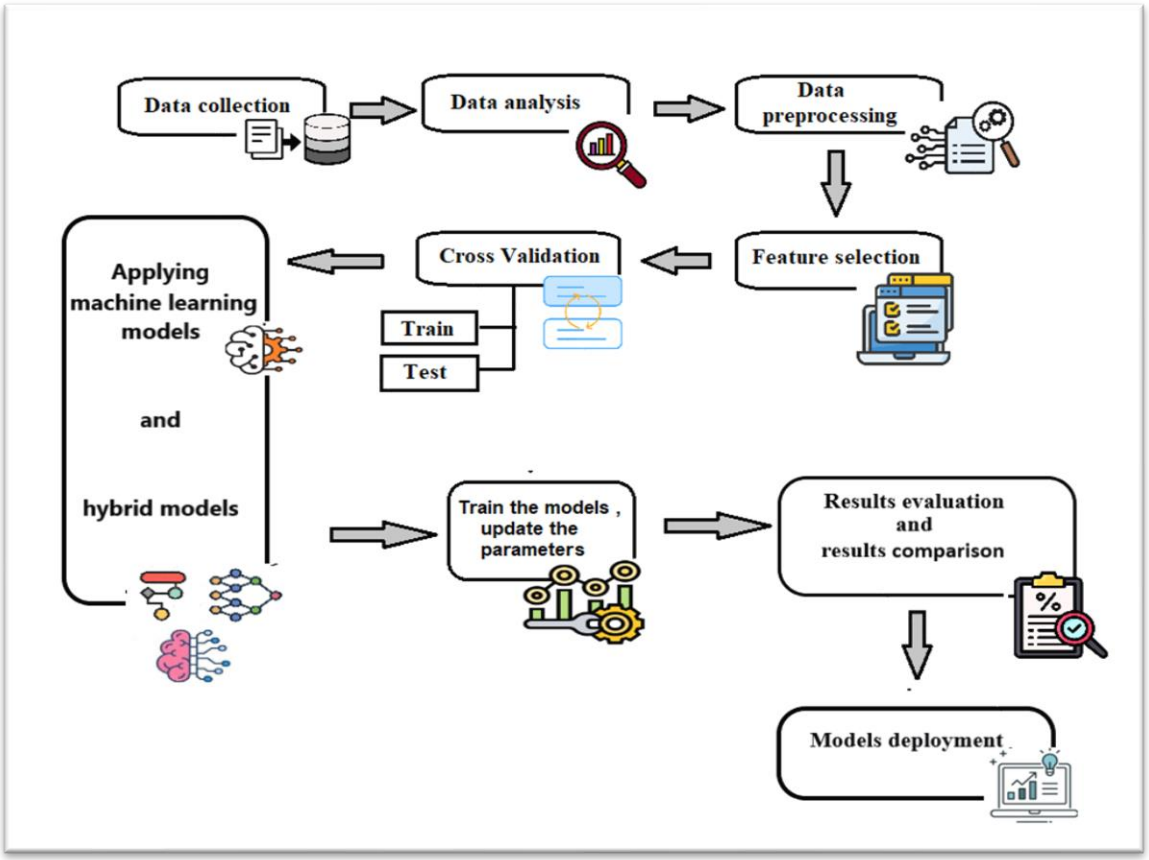

Figure 3-1: system structure of the proposed models

3.2 Datasets

In our work, we used two datasets: The global dataset was obtained from the UCI's machine learning repository [43], and the Palestinian local datasets were obtained from the Palestinian Ministry of Health, which was collected from patients' data in a Palestinian hospital.

3.2.1 Global Dataset

The global dataset was obtained from the UCI's machine learning repository [43], which was donated by Soundarapandian et al, and collected from Apollo Hospital patient data in 2015. In this dataset there are 400 samples, each sample contains 25 features, 24 features of which include laboratory and physical tests, and the remaining feature is the target attribute. Of these

400 samples, 250 samples are patients with CKD and the remaining 150 are not. The following table (3.1) shows the details of the global dataset.

Table 3-1:Description for each feature in a global dataset

	Attribute	Abbreviation	Value
1	age	age	years
2	Blood Pressure	bp	mm/Hg
3	Specific Gravity	sg	1.005, 1.010, 1.015, 1.020, 1.025
4	Albumin	al	0, 1, 2, 3, 4, 5
5	Sugar	su	0, 1, 2, 3, 4, 5
6	Red Blood Cells	rbc	normal, abnormal
7	Pus Cell	pc	normal, abnormal
8	Pus Cell clumps	pcc	present, not present
9	Bacteria	ba	present, not present
10	Blood Glucose Random	bgr	mgs/dl
11	Blood Urea	bu	mgs/dl
12	Serum Creatinine	sc	mgs/dl
13	Sodium	sod	mEq/L
14	Potassium	pot	mEq/L
15	Hemoglobin	hemo	gms
16	Packed Cell Volume	pcv	mEq/L
17	White Blood Cell Count	wc	cells/cumm
18	Red Blood Cell Count	rc	millions/cmm
19	Hypertension	htn	yes, no
20	Diabetes Mellitus	dm	yes, no
21	Coronary Artery Disease	cad	yes, no
22	Appetite	appet	good, poor
23	Pedal Edema	pe	yes, no
24	Anemia	ane	yes, no
25	Class	class	ckd, not ckd

3.2.2 Local Dataset

The local dataset was obtained from the Palestinian Ministry of Health [44], which was collected from patients' data in a Palestinian hospital. In this dataset there are 731 samples, each sample contains 17 features, 16 features of which include laboratory and physical tests, and the remaining feature is the target attribute. Of these 731 samples, 398 samples are patients with CKD and the remaining 333 are not. The following table (3.2) shows the details of the local dataset.

Table 3-2: Description of each feature in local dataset

	Attribute	Abbreviation	Value
1	age	age	years
2	Alanine Transaminase	ALT	normal, abnormal
3	Albumin	al	0, 1, 2, 3, 4, 5
4	The aspartate aminotransferase	AST	normal, abnormal
5	Blood Glucose Random (BG)	BG	mgs/dl
6	Blood Urea Nitrogen (BUN)	BUN	mgs/dl
7	Calcium Serum	cs	mEq/L
8	Serum Creatinine	sc	mgs/dl
9	Gender	g	Male/ Female
10	Hemoglobin	hemo	gms
11	phosphorous	p	mgs/dl
12	Potassium	pot	mEq/L
13	Province	p	0,1,2,…,9
14	Sodium Serum	sod	mEq/L
15	Uric Acid	UA	mgs/dl
16	Diabetes Mellitus	dm	yes, no
17	Class	class	ckd, not ckd

Both datasets contain features including laboratory and physical tests. Some of these tests that are common to local and global datasets which are age, Albumin, Blood Glucose Random (BG), Blood Urea Nitrogen (BUN), Serum Creatinine, Hemoglobin, Potassium, Sodium Serum and Diabetes. The definitions of some of these tests are:

1. **Alanine Transaminase (ALT):** It is a test that measures the amount of ALT in the blood. ALT is an enzyme present in the liver, and its level helps in assessing the health of the liver. If its level is high in the blood, this indicates the presence of a diseased condition in the liver or liver damage.
2. **Albumin**: test that measures the amount of albumin in blood. Albumin is a protein that is made by the liver and is loaded with enzymes and hormones. It enters the bloodstream and prevents fluids from leaking out of the blood vessels. A low albumin level is a sign of a medical condition or disease of the liver or kidneys.
3. **The aspartate aminotransferase (AST)**: A blood test to check whether there is a problem with the liver (damage) or not. AST is an enzyme that the liver makes and is also called glutamic-oxaloacetic transaminase (GOT). A high AST enzyme level indicates a diseased condition in the liver (damage) or in another organ such as the kidneys.

4. **Blood glucose (BG):** test that measures the glucose levels in blood. Low or high blood glucose is a sign of a medical condition. For example, high glucose levels lead to diabetes. Diabetic patients have an increased risk of developing chronic kidney disease.
5. **Blood Urea Nitrogen (BUN)**: The test measures the amount of urea nitrogen in the blood. Urea nitrogen is a non-beneficial product that the kidneys remove from the blood. A high level of urea nitrogen indicates a health problem for the kidneys, and it is also one of the most common tests that detect kidney problems at an early stage.
6. **Calcium Serum :**The test measures the amount of calcium in the blood. Calcium is considered one of the important minerals in the human body. An increase or decrease in the percentage of calcium in the body expresses the presence of a disease such as diseases of the bones, thyroid gland, kidneys, and other conditions.
7. **Creatinine Serum**: The test measures creatinine levels in the blood or urine. Creatinine is a non-beneficial product that is filtered by the kidneys from the blood and out of the body through the urine. Any defect in the level of creatinine in the blood indicates a problem in the kidneys
8. **Hemoglobin (HGB) :**A test that measures the level of hemoglobin in the blood. It is a protein found in red blood cells; its function is to carry oxygen from the lungs to the rest of the body. Any defect in the level of hemoglobin indicates a problem in the blood.
9. **Phosphorous:** A test that measures the amount of phosphate in the blood. The kidneys filter excess phosphate from the body through the blood. Any abnormality in the level of phosphate in the blood indicates the presence of kidney disease or another serious disorder.
10. **Potassium Serum**: A test that measures the amount of potassium in the blood. The body needs a percentage of potassium to function normally and properly, and a high or low level of potassium is a sign of a medical problem.
11. **Sodium Serum :** A test that measures the amount of sodium in the blood. The kidneys excrete excess sodium in the urine. An increase or decrease in the percentage of sodium in the body indicates the presence of a pathological condition such as a problem in the kidneys, the presence of dehydration in the body, or other conditions.
12. **Uric Acid:** A test that measures the amount of uric acid in the blood or urine. Uric acid is a useless product produced by the body. Part of this acid dissolves in the blood. The kidneys filter uric acid from the blood and excrete it in the urine. A high or low uric acid level indicates a medical condition, especially a kidney problem.

3.3 Data Preprocessing

In this research, we used in the preprocessing step a feature selection technique and a technique for dealing with missing values. These steps will be described in detail as follows:

3.3.1 Feature Selection (FS)

Mutual information (MI) is a technique that is used in many fields such as statistics and machine learning. Mutual information is a measure to quantify how much information is shared by two random variables and determines the interdependence between the variables [45][46]. also help in selecting informative features. The mutual information is defined as

$$MI(x;y) = \int \int p(x,y).log\left(\frac{p\,(x,y)}{p\,(x).p\,(y)}\right)dx\,dy \qquad (3.1)$$

Where p (x, y) is the joint probability density function of X and Y, p(x) and p(y) are the marginal probability density functions of X and Y. We applied this technique in selecting features for the local and global data that we have, and the results were as follows in table (3.3) and table (3.4).

Table 3-3: Feature selection for local dataset

Tests (Variables)	Weight
Creatinine Serum	0.470603
Uric Acid	0.442342
Province	0.440660
Phosphorous	0.334812
Blood Urea Nitrogen	0.267673
Potassium Serum	0.216770
Alanine Transaminase	0.170766
Albumin	0.163966
Aspartate Amino transferase	0.162822
Calcium Serum	0.120352
Hemoglobin	0.061387
Age	0.048768
BG	0.047578
Sodium Serum	0.024696
diabetes	0.000000
Gender	0.000000

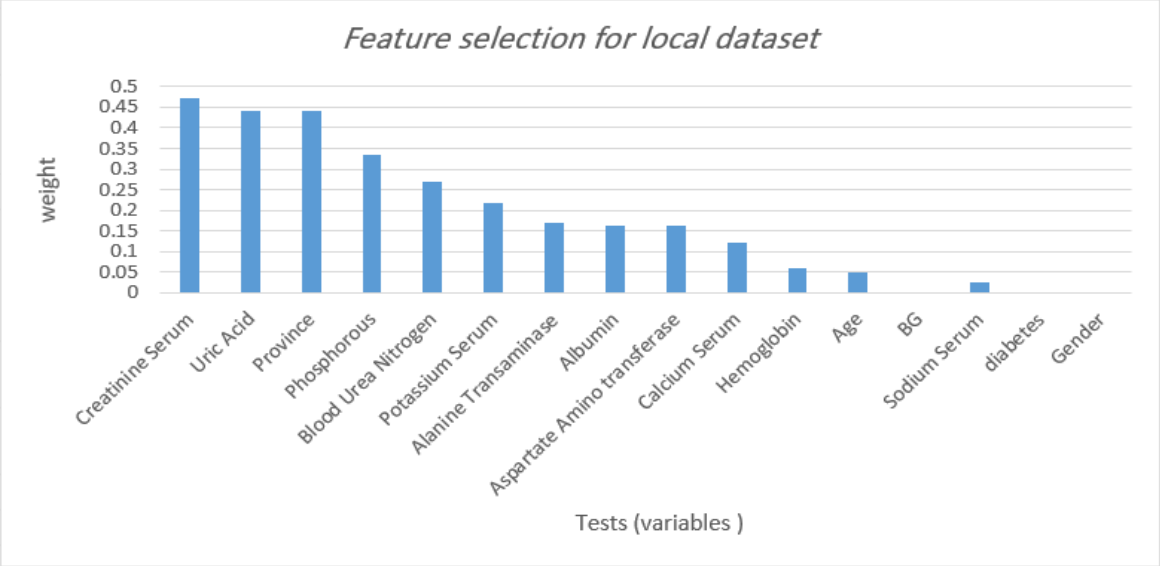

Figure 3-2: The chart show the feature selection for the local dataset

Table 3-4: Feature selection for global dataset

Test (variable)	Weight
Hemoglobin	0.445844
Packed Cell Volume	0.426203
Red Blood Cell Count	0.413973
Specific Gravity	0.406754
Serum Creatinine	0.387319
Albumin	0.316592
Sodium	0.284916
Red Blood Cells	0.269714
Hypertension	0.254486
Diabetes Mellitus	0.227864
Potassium	0.212240
Blood Urea	0.175296
Blood Glucose Random	0.172676
Blood Pressure	0.139558
White Blood Cell Count	0.127307
Pus Cell	0.126028
Sugar	0.117928
Appetite	0.112028
Pedal Edema	0.111922
Pus Cell clumps	0.072422
Coronary Artery Disease	0.067847
Age	0.065865
Anemia	0.053611
Bacteria	0.046326

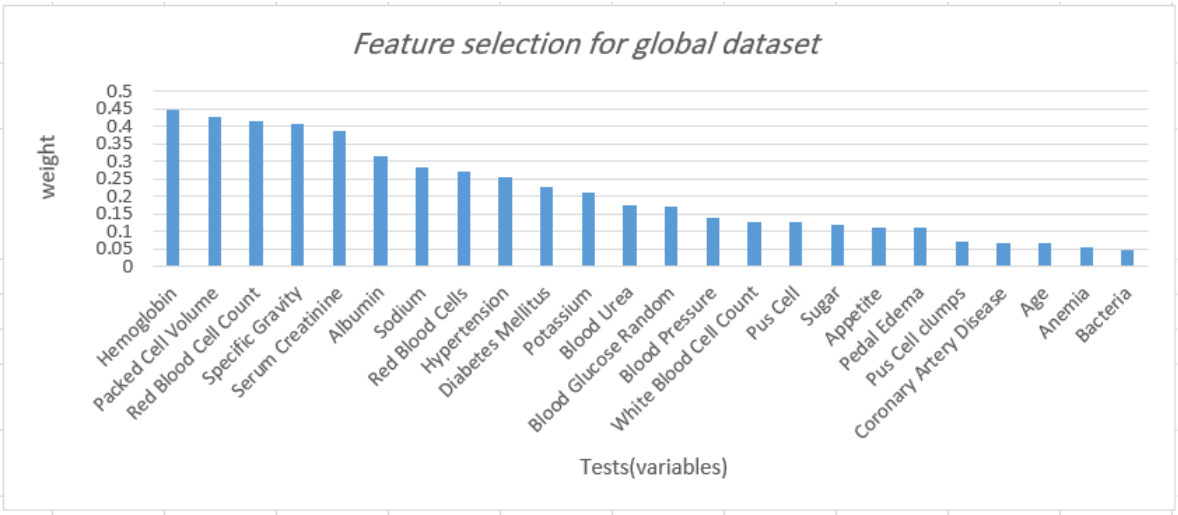

Figure 3-3: The chart show the feature selection for a global dataset

3.3.2 Dealing with Missing Values

The success of machine learning algorithms is based on the quality of the dataset to obtain a general predictive model for the classification problem [47]. Data normalization is useful to improve the data quality and the performance of machine learning algorithms. After collecting the datasets that we had, whether local or global data, we faced a challenge in how to deal with missing values. Many techniques have been developed to solve this problem [77]. For example, Imputation is a method of replacing missing data with alternative values by estimating and predicting [48], deleting records that contain missing data also replacing missing values with other values using the mean, median, and mode. We chose the method of replacing missing values using the mean, median, and mode. So, the best method is chosen according to the variable to find the best value.

3.4 Applied Models

In this section, we will present the different ML models that have been applied to the datasets (global and local) that were used in detail. Several machine learning models were applied to

the datasets such as Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Multi-layer perceptron neural network (MLPNN), then compare the results with each other in terms of classification accuracy.

3.4.1 Decision Tree (DT)

DT is one of the simple and popular machine learning algorithms, which uses supervised machine learning to solve regression and classification problems [16][49]. It works like a tree structure, starting from a single node called the root node (the starting point in the decision-making process and representing data) and then branching repeatedly in two or more directions, either branching into decision nodes or into leaf nodes as shown in figure (3.4). Decision nodes will branch again and represent the feature. As for leaf nodes, there is no branching after them, and are represented by a value. The tree ends when the final result is achieved. DT maps observations about a particular item and concludes with the item's target value.

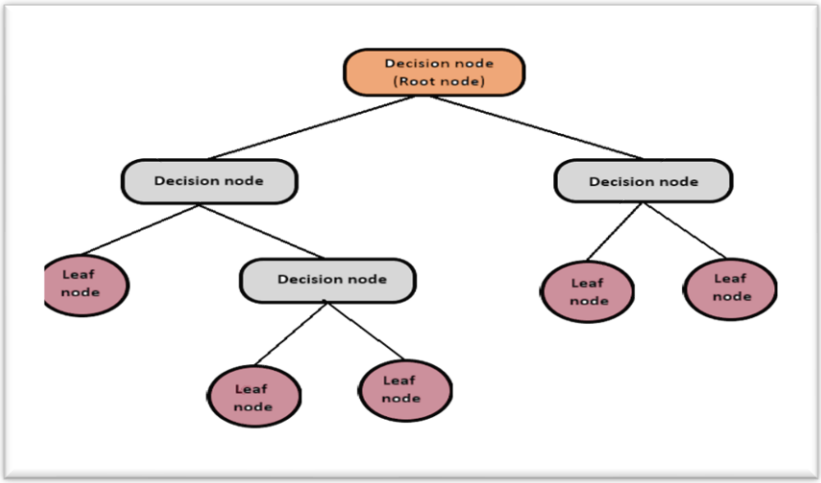

Figure 3-4 : Diagram of decision tree (DT) [73]

The root node is the beginning of the tree, which contains the dataset. Branching is then done based on Attribute Selection Measures (ASM), which are used as a criterion to determine which feature will create subsets. The process of branching into subsets containing the possible values

of the best Attribute will be repeated, which is called recursive partitioning [50][51]. There are two criteria for division: Gini impurity and information gain (entropy). Entropy is considered a measure of the degree of randomness in a dataset, as it measures randomness based on the distribution of class labels in the dataset. The value of entropy is between 0 and 1. The entropy equation is as follows:

$$E(s) = \sum_{i=1}^{c} - pi \log_2 pi \qquad (3.2)$$

Where S is the Total number of samples and pi is the frequentist probability of an element i in the data. Gini Impurity is considered an evaluation measure used in the decision tree model to evaluate the accuracy of the split between the classified groups, using the range between 0 and 1 as an evaluation of the result.

$$\text{Gini Impurity} = 1 - \sum {p_i}^2 \qquad (3.3)$$

where p_i is the proportion of the elements which belong to the classes. After that, in the pruning stage, branches that contain data that are not useful to the model's performance are removed to increase accuracy and for the tree to have a lower depth during branching.

3.4.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the machine learning algorithms that is supervised by machine learning and used to solve complex regression and outlier detection and classification problems. Mainly it is used for classification problems in the machine learning sector. This algorithm was developed in 1990 by Vladimir Vapnik and his colleagues [16] [52]. The target behind using SVM is to generate the best decision boundary and line which is used to separate n-dimensional space into categories and this helps new data points to be set in the correct category. Hyperplanes are called the best decision boundary. So, the main principle

of SVM is to find the best hyperplane in high dimension which helps to separate data points in an effective way to different classes [49][53].

The meaning of "support vector " represents the data points that are located close to the hyperplane or decision boundary. So, the target is to maximize the margin, the distance between the data point of each class, and the hyperplane as shown in Figure (3.5).

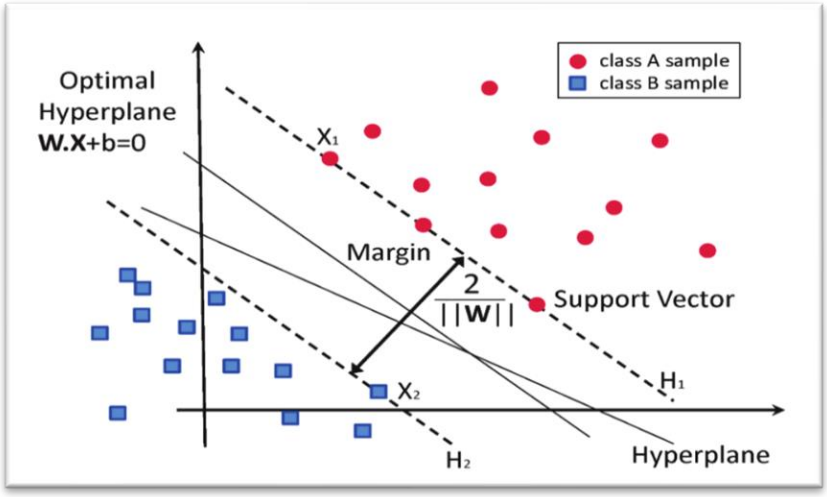

Figure 3-5: Support Vector Machine algorithm [74]

There is an equation that describes the optimal hyperplane of SVM and this equation is:

$$w.x + b = 0 \qquad (3.4)$$

where w is the weight vector perpendicular to the hyperplane, x is the input vector and b is the bias. Moreover, there are two types of Support Vector Machines:

1. Linear SVM: the data is separated by line so the dataset can be classified by a single line.
2. Non-linear SVM: the data is separated by non-line so the dataset cannot be classified by single line.

Many factors of SVM make it strong to use such as the algorithm doing well in High dimensional spaces and less sensitive to outliers because SVM focuses on nearest data points. Its effective algorithm compared to another algorithm can find global optimum. However, the

main challenges are SVM complexity in time and memory usage especially when SVM deals with large datasets. SVM depends on selecting the right tuning parameter and kernel which require domain knowledge and this can affect the model's performance. SVM algorithm can be applied in many applications such as It is used in image classification such as object detection and facial recognition and used in the analysis of biological data such as gene expression profiling.

3.4.3 K-Nearest Neighbor (KNN)

The k-Nearest Neighbors' algorithm (KNN) is one of the machine learning algorithms that is simple and supervised machine learning. It's used to solve regression and classification problems. Also, it is used widely for predictive modeling and pattern recognition [16]. This algorithm is easy to understand and implement but one of the problems that this algorithm faces is the slowness when Data is become large. There is another name for KNN which is called lazy learner algorithm because this algorithm doesn't learn from the training data, it stores the dataset and through time of classification, it does an action for it. KNN algorithm compares the similarity between available data and new data and puts new data into the right category which is similar to the available category. KNN is used to classify the new data point based on similarity and it stores the available data. so, any new data shown then is classified [49]. The process of the KNN Algorithm is to calculate the distance between all of the existing data points and new data points as shown in Figure (3.6).

And different distance metrics can be applied based on the data's nature such as Manhattan distance or Euclidean distance. The following equation is the Euclidean distance equation

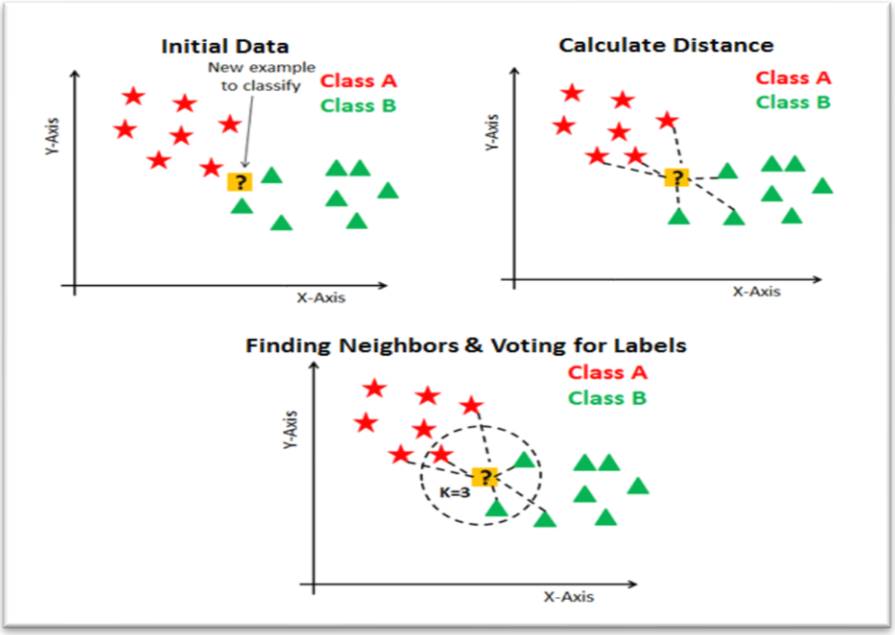

Figure 3-6: KNN Algorithm [75]

$$D(x,y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \tag{3.5}$$

Where x and y are the points.

Then Identify Neighbors by selecting 'K' points which have short distances with new points. Then algorithm can be applied to regression tasks or classification tasks. The algorithm in the Regression task predicts the target variable by averaging or taking the majority values of the k-nearest neighbor. Also, the algorithm in the classification task classifies the classes that show the relationship through new data points and k-nearest neighbors [54][55]. There are many advantages to using the KNN algorithm as this algorithm is simple to implement and when training data is large it can be more effective, and it is robust to the noisy training data. So, it can be used for data that is complex and needs clear structure. Despite the benefits, there are many challenges and considerations of KNN such as high cost especially when calculating the distance between existing points and new data points especially large datasets. Another

challenge, the importance of distance becomes less when there are high-dimensional spaces, in this case, reduction techniques are important. KNN algorithm is used in many applications such as speech and image recognition systems for feature extraction and pattern matching. Also, it's used in recommendation systems and medical diagnosis by predicting the likelihood of disease based on historical cases of patient data.

3.4.4 Multi-Layer Perceptron Neural Network (MLPNNs)

Neural Networks (NN) are a subset of artificial intelligence [56], inspired by the human brain and how biological neural networks work in it. It has another name, which is artificial neural networks (ANNs). NN consists of interconnected nodes (artificial neurons) that are in the form of layers. It works to transfer information as it can learn it and link it to the outputs. The most widely used type is multi-layer perceptron neural networks (MLPNNs) [57][58]. The general structure of these networks consists of several layers: the input, one or more hidden layers, and the output, as shown in Figure (3.7). Each layer is composed of one or more neurons. The network is connected by linking the neurons to each other between the layers and with specific weights, meaning that each connection has a specific weight value that is determined through the learning process.

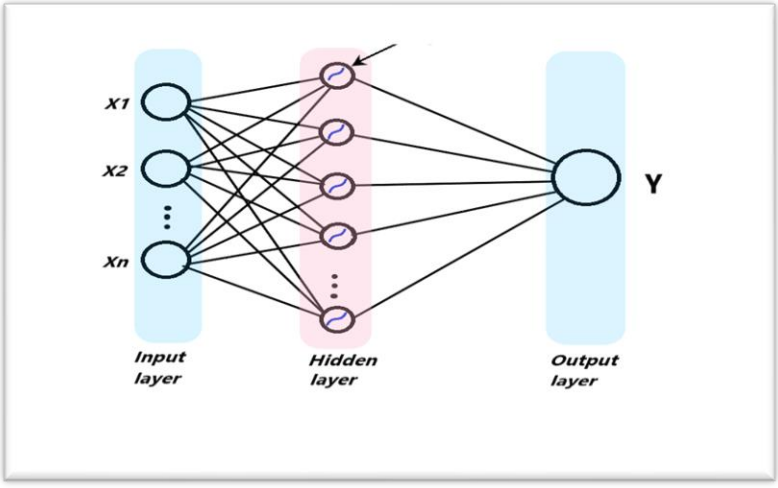

Figure 3-7: The structure of multilayer perceptron neural network (MLPNN) [76]

The training process for MLPNNs involves two main stages:

- Forward Propagation: In this stage, the network is fed with the input data, performing mathematical operations through the network, calculating the sum of the weights for each neuron, and applying the activation function to produce outputs for each neuron until the process ends with the output layer (predictions), calculating the error and sending it to the delay stage.
- Back Propagation: At this stage, the expected results are compared with the actual ones, the error is calculated and disseminated across the network, and the weights of the neurons are updated to reduce the error value in the output layer.

We will explain how MLPNN works step by step:

1. The network is fed with input data, performing mathematical operations through the network and giving it random weight values. Each neuron in the input layer will be connected to one or more neurons in the hidden layer.
2. The total weights will be calculated by multiplying the input values by their corresponding weights as in the following equation

$$\text{Yij} = \sum \text{wij} \, . \, x_i \tag{3.6}$$

Where x_i is the i^{th} input and w_{ij} is the weight between the i (neuron in the input layer) and the j (neuron in the hidden layer).

3. Apply an activation function to produce outputs for each neuron in the hidden layer using the following equation

$$S_k = \text{sigmoid(Sj)} = \frac{1}{(1 + \exp -(\sum \text{Yij})} \tag{3.7}$$

4. until the process ends with the output layer (predictions) by using the previous equations, the difference value (Δj) in the output layer will be computed by the equation (3.8)

$$\Delta_k = t_k - S_k \quad (3.8)$$

Where tj is target value and the Sj is the output values.

5. After that, the error signal of the output layer will be calculated using the equation (3.9), and the weights will be updating using equation (3.10) and (3.11).

$$\delta_k = \Delta_k \cdot S_k \cdot (1 - S_k) \quad (3.9)$$

where δk is the error signal in output layer, and S_k is the output of node.

$$\Delta w_{jk} = \alpha \cdot \ \delta_k \cdot x_i \quad (3.10)$$

$$w_{jk} = \ w_{jk} + \Delta w_{jk} \quad (3.11)$$

where w_{jk} is the weight between node j in hidden layer and node k in output layer, and α is the learning rate.

6. Calculate the error signal for the neurons in the hidden layer using equation (3.12)

$$\delta_j = \Delta_j \cdot S_j \cdot (1 - S_j) \ . \sum \delta_k . \ w_{jk} \quad (3.12)$$

where δj is the error signal in hidden layer

7. the weights will be updating using equations (3.13) and (3.14)

$$\Delta w_{ij} = \alpha \cdot \ \delta_j \cdot x_i \quad (3.13)$$

$$w_{ij} = \ w_{ij} + \Delta w_{ij} \quad (3.14)$$

where w_{ij} is the weight between node i in the input layer and node j in hidden layer, and α is the learning rate.

8. Repeat the forward and backward propagation process until the network performance improves and the error rate decreases, then the model performance is evaluated.

3.5 Developed Models

In this part of the chapter, we will present the hybrid models that we have applied to the datasets (global and local), we developed the evolutionary algorithms GAs, BBO, and PSO, used to find a set of optimal weights for the neural networks while the MLPNN training and classifying the dataset for high classification accuracy. Then we made comparisons between the results of the hybrid models to get the best result of classification accuracy the chronic kidney disease.

3.5.1 Biogeography-Based Optimization (BBO)

Biogeography-based optimization (BBO) is one of the evolutionary algorithms used in optimization. It was inspired by nature, the geographical distribution of biological organisms, and migration. This algorithm was proposed in 2008 [59]. The solution in this algorithm is similar to the habitat, the solution component is similar to the suitability index variables (SIVs), and the suitability and quality of the solution are similar to the habitat suitability index (HSI). So, if the HSI index is high, it means that the habitat has a good solution (good habitat), but if the HSI index is low, it has a not good solution (not good habitat) [60][61][62].

The optimization in this algorithm is done by iteratively improving candidate solutions concerning a certain quality measure or fitness function to evaluate the solutions. This algorithm mainly consists of two basic processes: the migration process and the mutation process. Each habitat has two parameters which are migration rate and emigration rate, both are related to the (HSI). There are more species in high HSI habitat, so the emigration rate is high and the immigration rate is low, due to the high competition for resources from other species. But few species are in low HSI habitats, so the emigration rate is low and the immigration rate is high because there is a lot of room for additional species. The following equations explain this

$$\lambda_i = I \times \left(1 - \frac{S_i}{S_{max}}\right) \qquad (3.15)$$

$$\mu_i = E \times \frac{s_i}{S_{max}} \quad (3.16)$$

Where habitat H_i has species S_i, S_{max} has a maximum number of species. λ_i is the immigration rate of H_i, μ_i is the emigration rate of H_i, I is the maximum immigration rate and E is the maximum emigration rate.

In the mutation process, there are some properties of the habitat that may change for example the number of species and HSI. species probability determines the Mutation rate, if the number of species in the habitat is large or small, the species probability is low. But if the number of species is moderate, the species probability is high. The following equation explains this:

$$m_i = m_{max} \times \left(1 - \frac{P_i}{p_{max}}\right) \quad (3.17)$$

Where Mutation rate m_i, species probability P_i which is decided by several species S_i, m_{max} is the maximum mutation rate and P_{max} is the maximum species probability.

The BBO algorithm can be summarized as the following steps:

1- Preparing a set of habitats.
2- Find the HSI value for each habitat to find the rate of emigration, immigration, and mutation
3- In the migration process, habitats are modified (emigration, immigration) according to the rates of emigration, and immigration.
4- In the mutation process, the mutation rate is determined and the suitability of each habitat is evaluated.
5- Repeat the process until convergence criteria are met or an acceptable solution is obtained.

Figure (3.8) shows the general flowchart of the BBO algorithm.

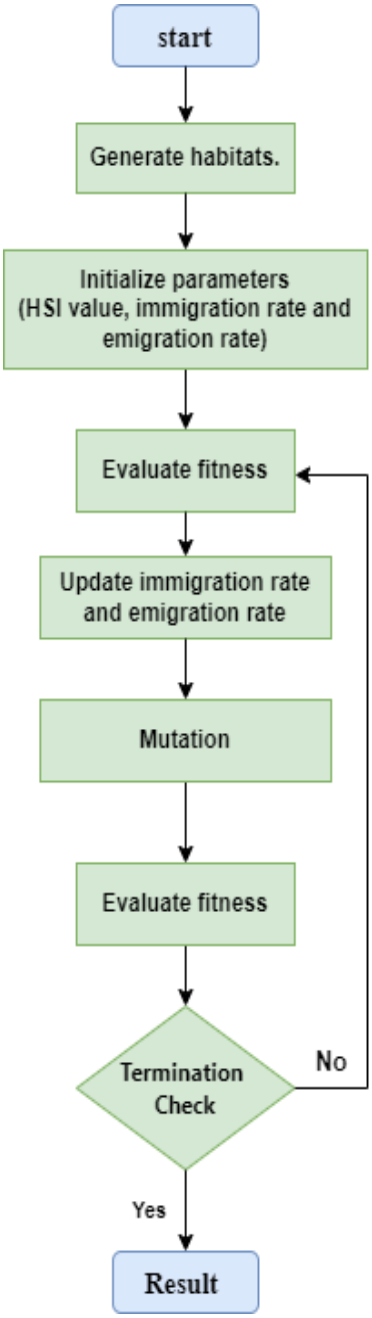

Figure 3-8: Flowchart of the BBO algorithm

3.5.2 Particle Swarm Optimization (PSO)

One of the evolutionary algorithms used in optimization, it was developed based on swarm behavior, such as the behavior of swarms of birds and fish in learning in nature [63], which

gave scope for artificial intelligence to benefit from the behavior and teach the machine and train it on the same behavior in the field of classification by adjusting the paths of the individual agents (particles) that move in a multi-dimensional space for research. A single particle represents a solution to the problem and the space represents the problem. The particles move collectively and explore the search space together, giving them the ability to iteratively adjust their positions based on experience. this algorithm was proposed by Kennedy and Eberhart (1995) [64][65]. The algorithm consists of particles and some parameters: position and velocity, in addition to rules for updating the parameter values. We can calculate the update velocity (v) and the position (p) by using the following equations [66]:

$$v_i(t+1) = w.v_i\,(t) + c_1.r_1\,.\big(\,pbest_i - p_i(t)\big) + c_2.r_2\,.\big(\,gbest_i - p_i(t)\big) \qquad (3.18)$$

$$p_i(t+1) = p_i\,(t) + v_i(t+1) \qquad (3.19)$$

Where v_i is the velocity of the particle, p_i is the position of the particle, w is the inertia weight, pbest is the personal best position for the particle, gbest is the global best position for the particle, c_1 and c_2 are the acceleration coefficients, r_1 and r_2 are random variables between 0 and 1. The steps of how the PSO algorithm works:

- The algorithm begins by initializing a random swarm of particles, each with its position and speed, in the search space.
- The fitness function of each particle in the swarm is evaluated, and its performance in solving the optimization problem is determined.
- After that, each particle updates its velocity(v)and position(p) if it finds a better location based on its experience so that it chooses the best location for itself (pbest) and the best location for its neighbors(gbest).
- The update is done repeatedly until a suitable solution for the particle is found that is the best. Figure (3.9) shows the general flowchart of the PSO algorithm.

Figure 3-9: Flowchart of the PSO algorithm.

3.5.3 Genetic Algorithm (GA)

Genetic algorithm (GA) is one of the evolutionary algorithms used to find solutions to search and optimization problems, inspired by genetics and the process of natural selection. It aims to improve the fitness of populations by developing a range of solutions over generations. each one of the Candidate solutions (individuals) contains a set of characteristics that can be changed and mutated to reach the best solution to the optimization problem [67][68].

The steps of how the GA algorithm works [69]:

- A set of candidate solutions (individuals) is generated in the chromosome population, each solution representing a set of genes (parameters) in the chromosome's population (search space).
- A fitness function is applied to each chromosome to evaluate and determine how well the chromosome performs (the solution).
- select between the chromosomes according to their fitness. The highest opportunity is for those with high fitness.
- In the process of crossover, the chromosomes are paired and recombined to form new chromosomes that combine the characteristics present in the parents.
- In the mutation process, changes are introduced to some new chromosomes, which adds diversity and leads to the solution.
- After that, the chromosomes are replaced with new chromosomes, keeping the population size constant, and the processes are repeated for a specified number of generations or until the solution is reached.

Figure (3.10) shows the general flowchart of the GA algorithm.

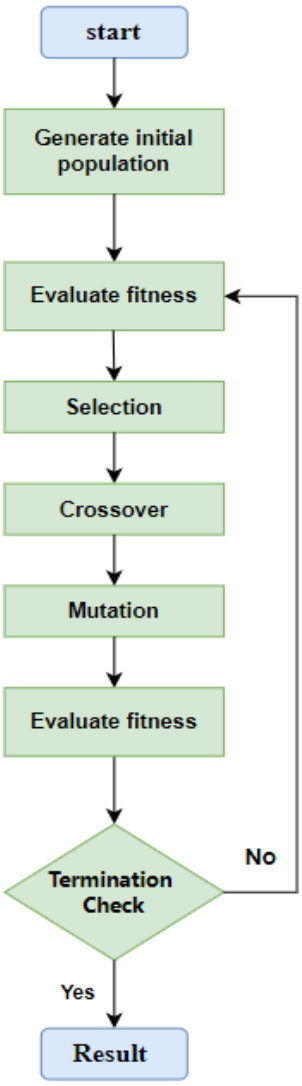

Figure 3-10: Flowchart of the GA algorithm.

3.5.4 Hybrid Models (BBO-MLPNNs, PSO-MLPNNs, and GAs-MLPNNs Algorithms)

Many optimization algorithms are applied to machine learning to train neural networks and classify datasets. In this thesis, some of these algorithms were used, including BBO, PSO, and GA. Each of these algorithms works with a multilayer perceptron neural network to classify and diagnose chronic kidney disease. The hybrid models (BBO-MLPNNs, PSO-MLPNNs, and GAs-MLPNNs algorithms) train the multilayer perceptron neural network and adjust the network weights to obtain the best results. Figures (3.11) show the flowchart for the hybrid

models BBO-MLPNNs, PSO-MLPNNs, and GAs-MLPNNs algorithms. At the beginning of the work, data on CKD was collected, where a local Palestinian dataset and a global dataset were collected, explored, processed, and analyzed. These two datasets were pre-processed. The structure of the neural network is determined, and the initial values for weight and bias are given. Also, determine the number of generations required to calculate the best and mean of each generation, through which accuracy and performance will be calculated.

The BBO algorithm was initially applied, starting with generating habitats and configuring the parameters (HSI value, migration rate, migration rate), then simulating the network and adjusting the parameters and the network weights to obtain the best results. After that, the PSO algorithm will be applied so that it begins by configuring the parameters of the algorithm and then simulating the neural network and modifying the parameters of the algorithm, such as updating personal best and neighborhood best, velocity and position, and also modifying the network weights to get the best results. After that, the GA algorithm will be applied, starting with configuring the parameters of the algorithm, then simulating the neural network, modifying the parameters, and going through the three stages (selection, crossover, and mutation) and also modifying the network weights to obtain the best results. The results were shown for all the algorithms, identifying the best hybrid model that was applied to the global and local datasets.

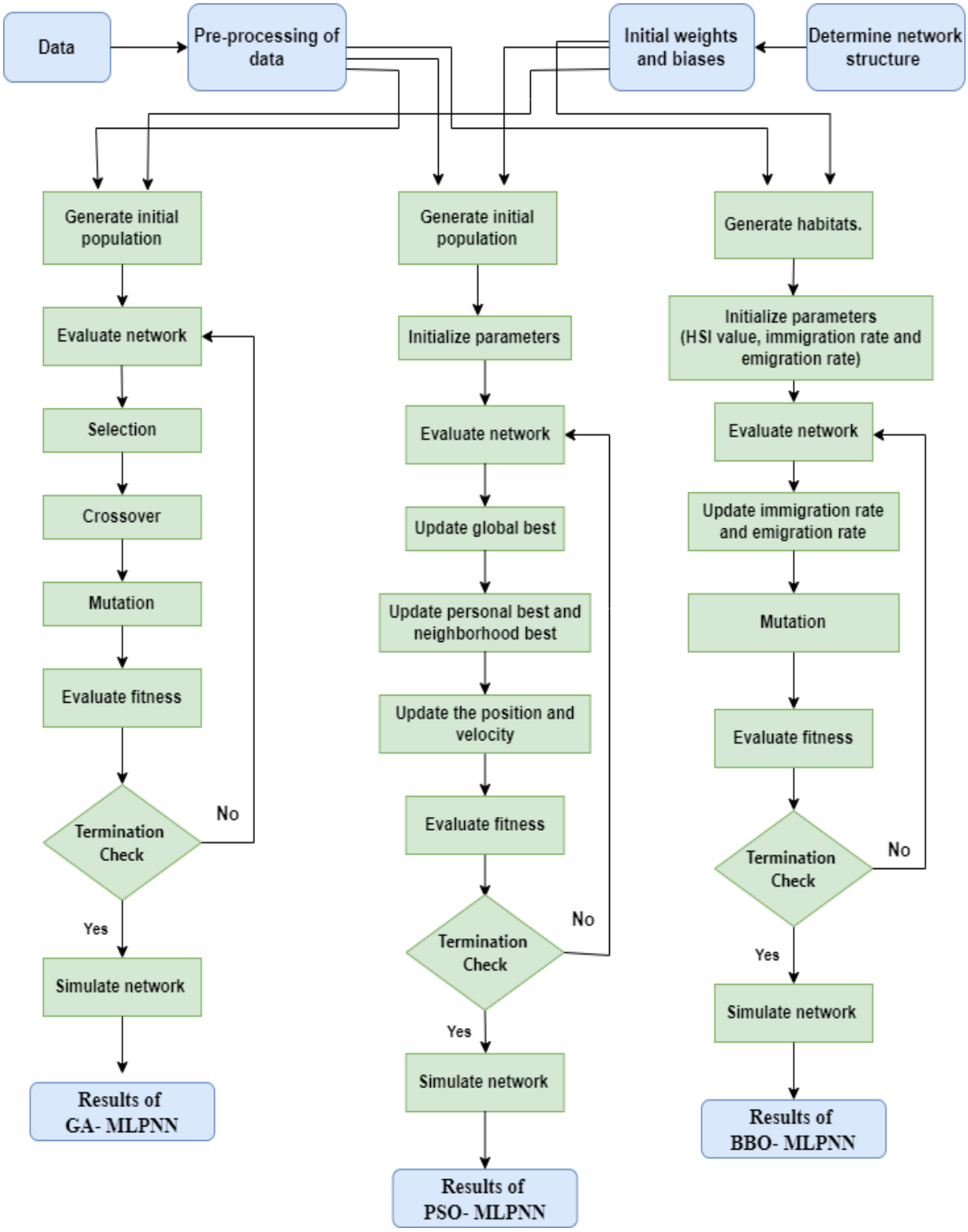

Figure 3-11: Flowchart for the hybrid models

3.6 Performance Metrics Selection

Different evaluation matrices were used to evaluate the performance of the classifier. After applying the previous classification models to global and local datasets, a set of metrics was used and displayed, such as the confusion matrix, accuracy, sensitivity, specificity, precision, Receiver Operating Characteristic Curve (ROC), and the area under the curve (AUC) for all models. Each measure will be explained as follows:

- **Confusion matrix**: It is a table used to evaluate the performance of classification algorithms, and through it, most performance measures such as accuracy, precision, sensitivity, and others can be calculated by knowing the values of the matrix consisting of rows representing the predicted values and columns representing the actual values. The entries of the matrix are:

1- True Positive (TP): The number of cases that were correctly predicted to be positive by the model, and the true samples were positive.
2- True Negative (TN): The number of cases that were correctly predicted as negative by the model, and the true samples were negative.
3- False positive (FP): The number of cases that were incorrectly predicted as positive by the model and the true samples were negative.
4- False negative (FN): The number of cases that were incorrectly predicted as negative by the model and the true samples were positive.

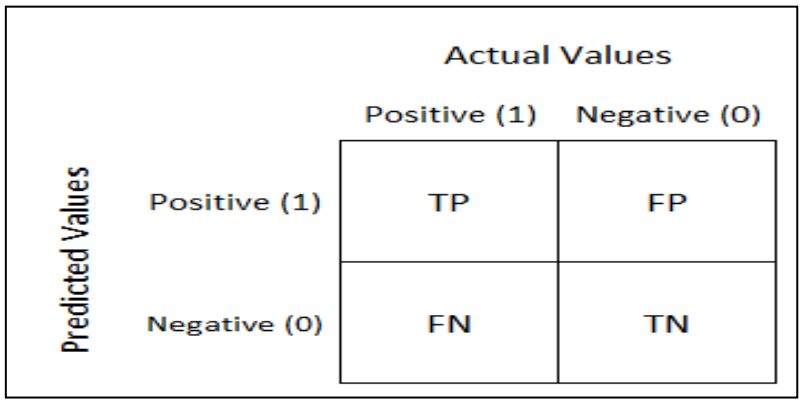

Figure 3-12: Confusion matrix

- **Accuracy:** Classification accuracy is defined as the percentage of samples that the learner of the classification system classifies correctly. It is calculated using the following equation

$$Accuracy = \frac{TP + TN}{(\mathrm{TN} + \mathrm{TP} + \mathrm{FN} + \mathrm{F}P)} \quad (3.20)$$

- **Sensitivity:** A measure to evaluate the model's ability to correctly identify positive cases, it can be determined by dividing the true positives (TP) by the sum of true positives (TP) and false negative (FN) as in the following equation:

$$Sensitivity = \frac{TP}{(\mathrm{FN} + \mathrm{TP})} \quad (3.21)$$

- **Specificity:** A measure to evaluate the model's ability to correctly identify negative cases, it can be determined by dividing the true negative (TN) by the sum of true negative (TN) and false positive (FP) as in the following equation:

$$Specificity = \frac{TN}{(\mathrm{FP} + \mathrm{TN})} \quad (3.22)$$

- **Precision:** or positive predictive value, it can be determined by dividing the true positives (TP) by the sum of true positives (TP) and false positives (FP). It is calculated using the following equation

$$Precision = \frac{TP}{(\mathrm{FP} + \mathrm{TP})} \quad (3.23)$$

- **Receiver Operating Characteristic (ROC) curve**: is a graphical representation that illustrates the performance metric for classification problems. the curve plots the true positive rate (TPR) against the false positive rate (FPR) across different threshold values. The line of the curve represents the performance of the classifier and the results

of the classifier are considered good whenever the points are in the upper corner of the curve.

- **Area under the curve (AUC):** which is the area under the ROC curve. This area can be a good measure to evaluate the performance of the classification model. When the AUC value is close to 1 (the area under the curve is high) it means that the results are good and the model can classify.

3.7 Cross-Validation

The K-fold cross-validation method, which is an effective and reliable technique in machine learning used to build models and evaluate their performance in prediction by dividing the data into folds, also helps to prevent overfitting. The model trains several folds, tests the rest of the folds, and repeats the process with several folds (K), and each time the test set used is a different fold. The average performance metrics from each fold are then calculated [70][71][72]. In our experiments, we used 5-fold cross-validation, 4 times for training and 1 time for testing randomly as shown in figure (3.13).

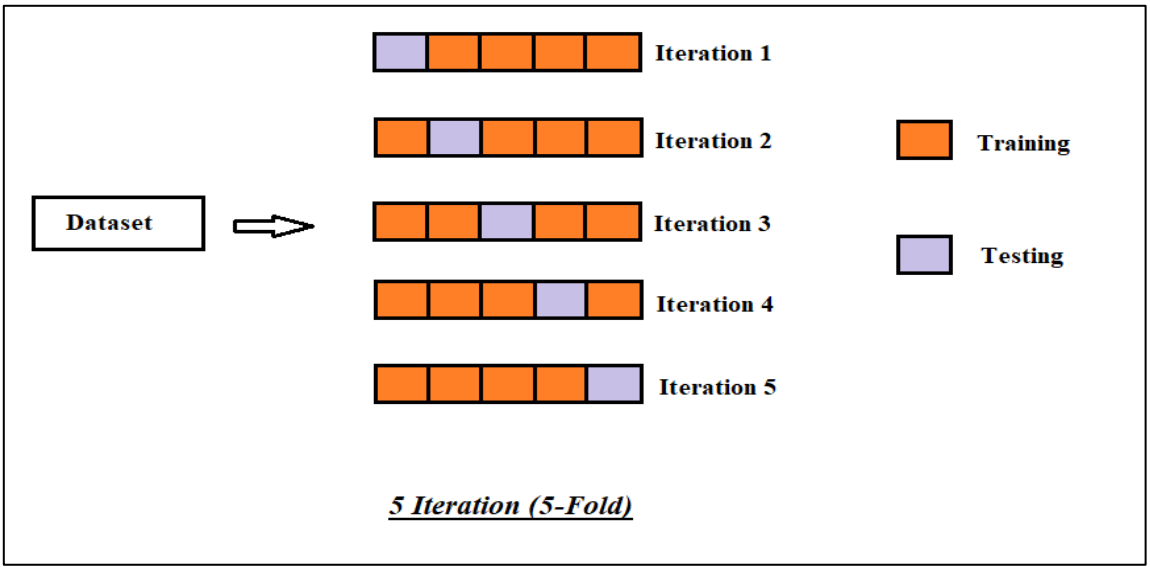

Figure 3-13**:** 5-Fold Cross Validation Graphical

Chapter 4

4.1 Experiments and Results

In this chapter, we will present the results of applying all the previously explained classification models to the datasets, discuss these results, and evaluate the proposal. We applied classification models (DT, SVM, KNN, and MLPNN) in two stages. In the first stage, the models were applied to the global dataset, and the accuracy, specificity, sensitivity, and precision of each model were measured. In addition, the receiver operating curves (ROC) and the area under the curve (AUC) were calculated for the best classification model, and determining the best classification model for the global dataset from among the applied models. In the second stage, we applied the models to the local dataset and measured the accuracy, specificity, sensitivity, and precision of each model. In addition, the receiver operating curves (ROC) and the area under the curve (AUC) were calculated for the best classification model, and determining the best classification model for the local dataset from among the applied models. After that, we discussed and compared the application of classification models (DT, SVM, KNN, and MLPNN) to both global and local datasets in the two stages and identified the best classification model.

We applied the hybrid classification models (BBO, PSO, and GAs) in two stages : In the first stage, the models were applied to global datasets, the accuracy, specificity, sensitivity, and precision of each model were measured and the best classification model for the global dataset was determined from among the applied models. In the second stage, we applied the models to the local dataset, the accuracy, specificity, sensitivity, and precision of each model were measured and determined the best classification model for the local datasets from among the applied models. After that, we discussed and compared the application of hybrid classification models (BBO, PSO, and GAs) on both global and local datasets in the two stages and determined the best hybrid model in classification.

4.2 Software

In our work, MATLAB was used for applying all the machine-learning procedures and to develop the hybrid models (MATLAB R2021a). The Jupyter Notebook was used for data preprocessing. The basic hardware configuration for the work done by ASUS X556 SnonicMaster, Intel(R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz, RAM: 8GB,64-bit and the operating system, x64-based processor Windows 10 Pro.

4.3 Classification Models Experiment

4.3.1 The Results of Classification Models for the Global Dataset

- **Application of DT, SVM, and KNN Techniques**

In this part, we applied classification algorithms to the global dataset. They are decision tree (DT), support vector machine (SVM), and K-nearest neighbor (KNN). In experiment, we used 5-fold cross-validation, 4 times for training and 1 time for testing randomly. The global dataset contained 400 records and each time 80 records were obtained. Table (4.1) shows the results of applying the models to the global dataset and performance metrics for each model.

Table 4-1 : The classification results for the models on the global dataset

Model	Accuracy	Specificity	Sensitivity	precision
Decision Tree	97%	95.3%	98%	97.2%
SVM	99.5%	100%	99.2%	100%
KNN	98.2%	100%	97.2%	100%

Figure 4-1: The classification results for the models on the global dataset

The results showed that the best model for classifying the global dataset among the three models is the SVM, which had an accuracy rate of 99.5%, followed by the KNN model with an accuracy rate of 98.2% and the DT model was the least accurate among the three models with an accuracy rate 97%. The best accuracy was 99.5%, the best sensitivity was 99.2%, and the best Specificity was 100%. The 5-fold cross-validation was used here, where 80% of the data is for training and 20% of the data is for testing. Figure (4.1) shows the comparison between all classification models used in terms of the four-performance metrics. We conclude that the SVM model obtained the best results and is the best model in terms of accuracy at 99.5%. In the figures (4.2), (4.3) and (4.4). the lines represent the ROC curves, and the areas in blue represent the AUC curves for the three models DT, SVM, and KNN respectively. In each figure, there are two classes: class 0 which does not have chronic kidney disease (not ckd), and class 1 which has CKD. In this curve, the results are considered good whenever the points are in the upper corner of the curve. In the DT model, the area under the curve occupied 98% for both classes and the current classifier was equal (0.02,0.95) (0.05,0.98) for the classes 0,1 respectively. In the SVM model, the area under the curve occupied 100% for both classes and the current classifier was equal (0.01,1.00) (0.00,0.99) for the class 0,1 respectively In the KNN model, the area under the curve occupied 99% for both classes and the current classifier was

equal (0.03,1.00) (0.00,0.97) for the class 0,1 respectively. We conclude that the SVM model obtained the best results in terms of AUC and ROC curves.

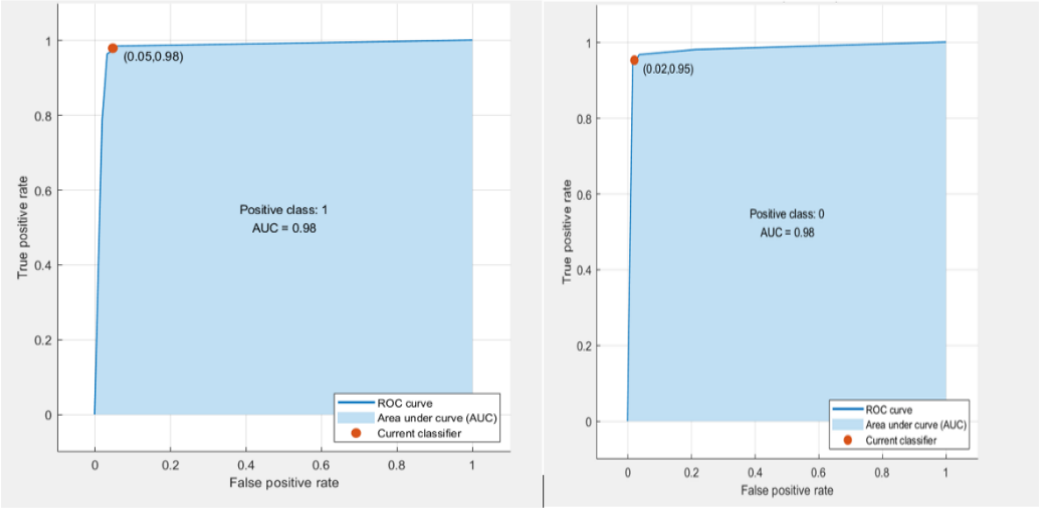

Figure 4-2 :AUC & ROC curves by DT model for global dataset (Class 0 and 1)

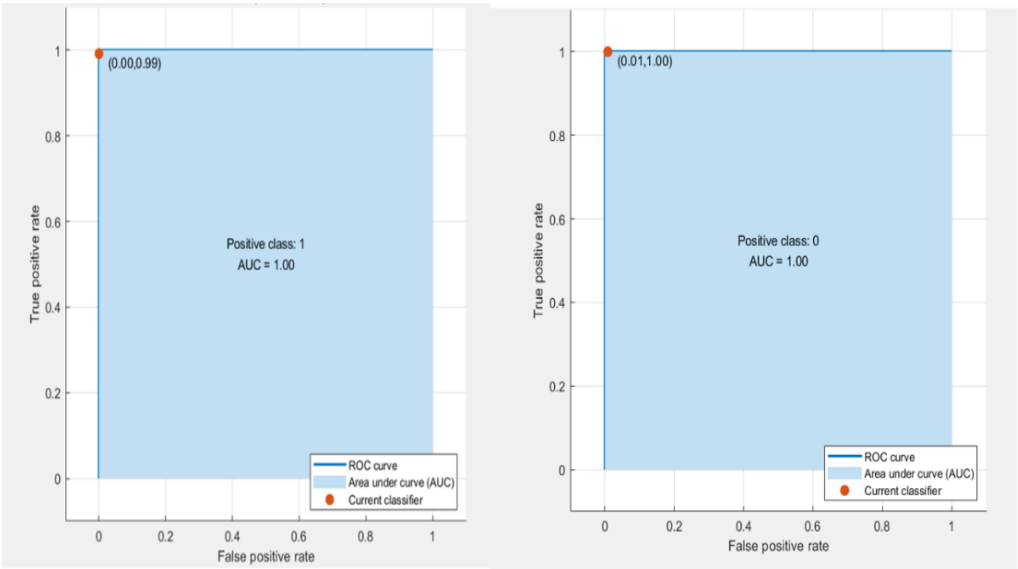

Figure 4-3: AUC & ROC curves by SVM model for global dataset (Class 0 and 1)

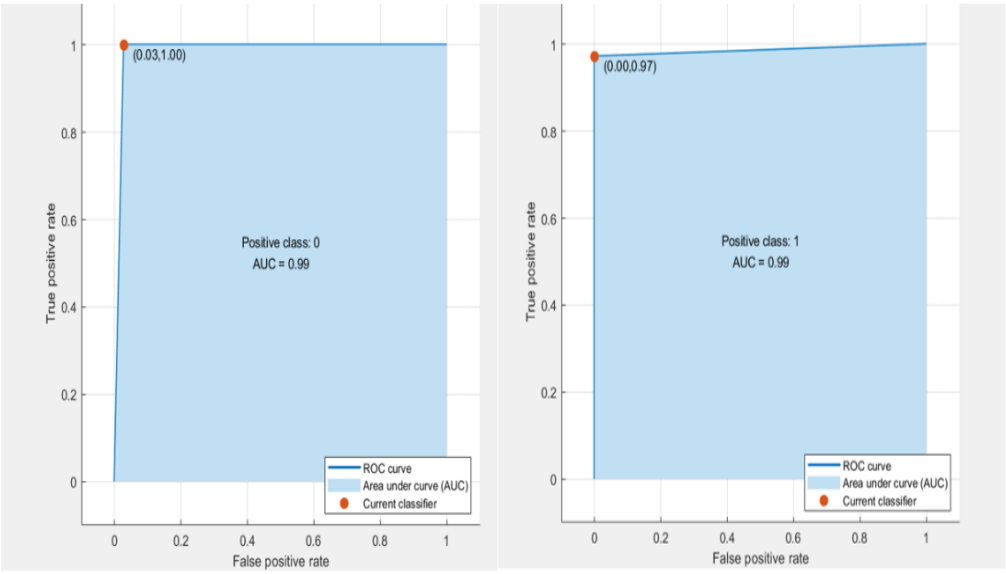

Figure 4-4: AUC & ROC curves by KNN model for global dataset (Class 0 and 1)

❖ Application MLPNNs Model

In this part, we applied the **MLPNN** model to the global dataset. We used NNs as one of the powerful classification methods and built the network with a two-layer feed-forward .The data is divided into three parts: the training part, which was 70% (280 records), the validation part, which was 15% (80 records), and the testing part, 15% (80 records). Several tests were conducted, using 10 and 50 neurons in the hidden layer for each test.

Scaled conjugate gradient backpropagation was used for training the global data. Table (4.2) shows the accuracy results of changing the number of neurons in the hidden layer after training the data. Figure (4.5) shows the result of accuracy for the model when changing the number of neurons in the hidden layer.

Table 4-2: The Classification Results for MLPNNs on the global dataset

Number of neurons	Test accuracy	All accuracy	Specificity	Sensitivity	precision
5	98.3%	99.8%	99.3%	100%	99.6%
10	96.7%	99.5%	99.3%	99.6%	99.6%
15	96.7%	98.8%	98.0%	99.2%	98.8%
20	98.3%	99.8%	99.3%	100%	99.6%
25	98.3%	99.8%	99.3%	100%	99.6%
30	98.3%	99.5%	98.7%	100%	99.2%
35	98.3%	99.8%	99.3%	100%	99.6%
40	98.3%	99.5%	98.7%	100%	99.2%
45	100	99.8%	99.3%	100%	99.6%
50	100	99.8%	99.3%	100%	99.6%

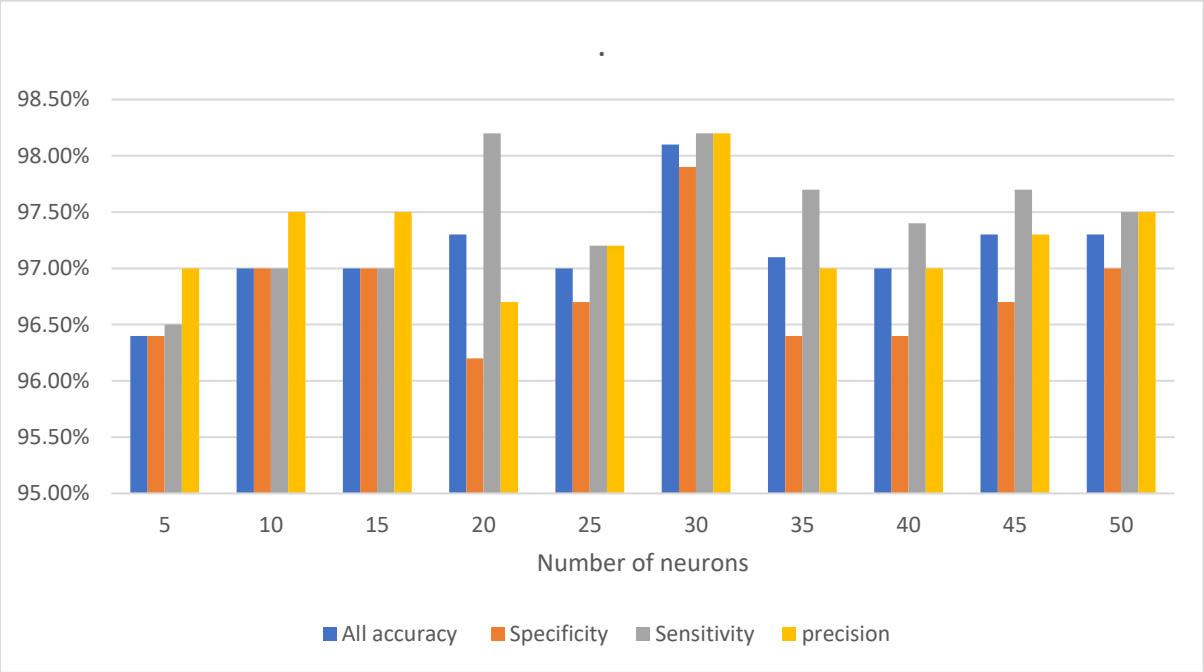

Figure 4-5: Chart for classification results for MLPNN model on global dataset.

The MLPNNs model achieved the best performance when the number of neurons in the hidden layer equal 50 (N=50) with an accuracy of 100% for testing and the result of all accuracy, Specificity, Sensitivity, and precision was 99.8%,99.3%,100%, and 99.6% respectively, while the performance of the model when the number of neurons in the hidden layer equal 10 (N=10) was lower as the accuracy rate was 96.7% for testing and the result of the all accuracy, Specificity, Sensitivity and precision were 99.5%,99.3%,99.6% and 99.6% respectively. We

conclude that the more neurons in the hidden layer, the better the performance and classification. The confusion matrices for the MLPNNs model for both N=10 and N=50 is shown in figures (4.6) & (4.7) respectively. The figures illustrate the confusion matrices with all the values of the accuracy.

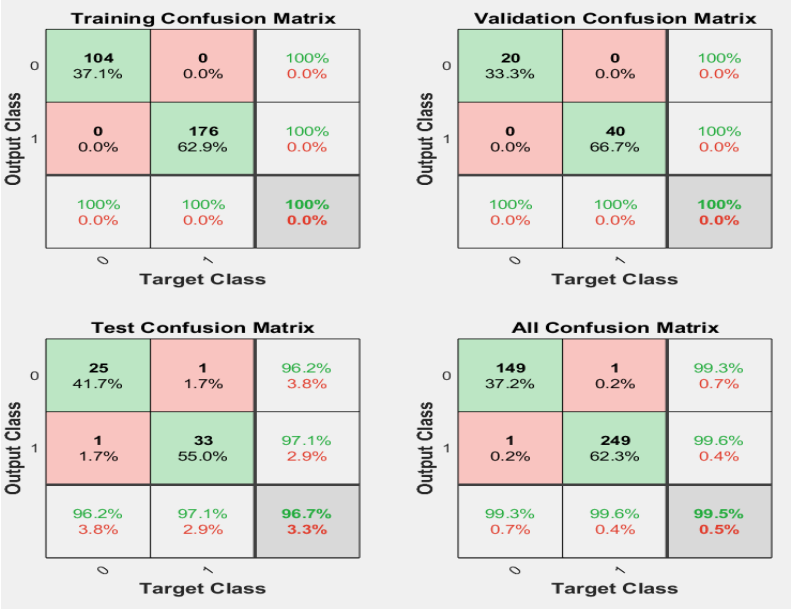

Figure 4-6: Confusion Matrices for MLPNNs on global dataset when N=10.

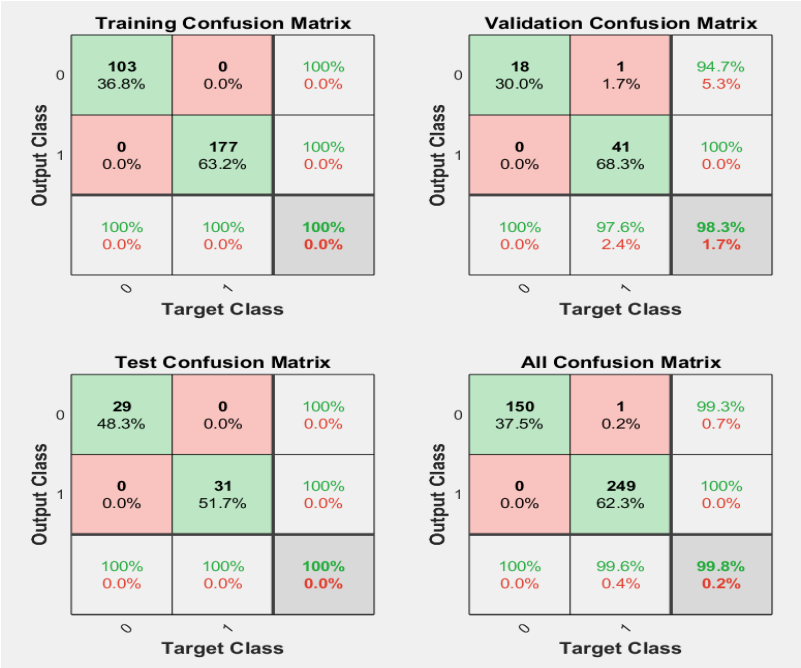

Figure 4-7: Confusion Matrices for MLPNNs on global dataset when N=50.

The ROC curve for the MLPNNs model for both N=10 and N=50 is shown in figures (4.8) & (4.9) respectively. the points of the line are in the upper-left corner which means that the performance is good.

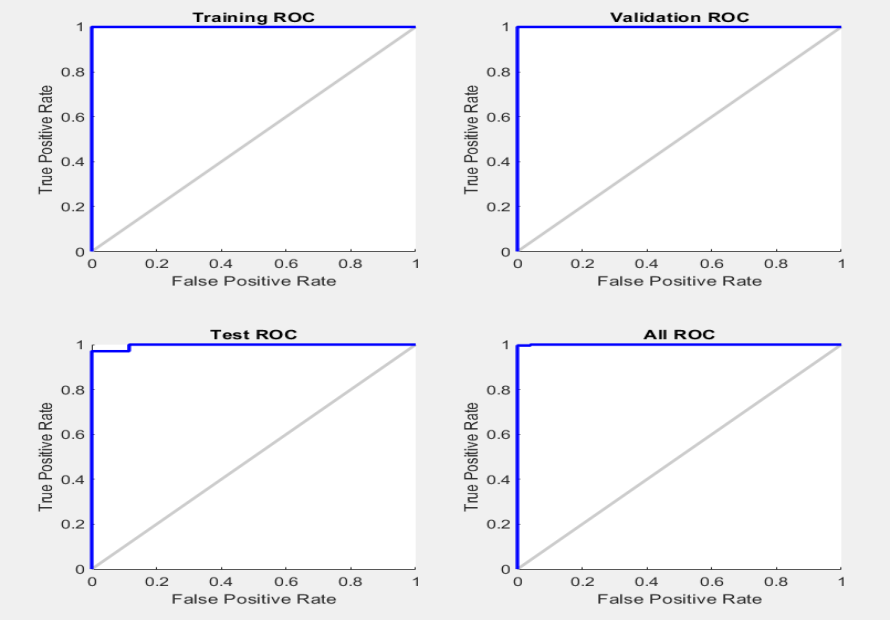

Figure 4-8: The ROC curves for MLPNNs on global dataset when N=10.

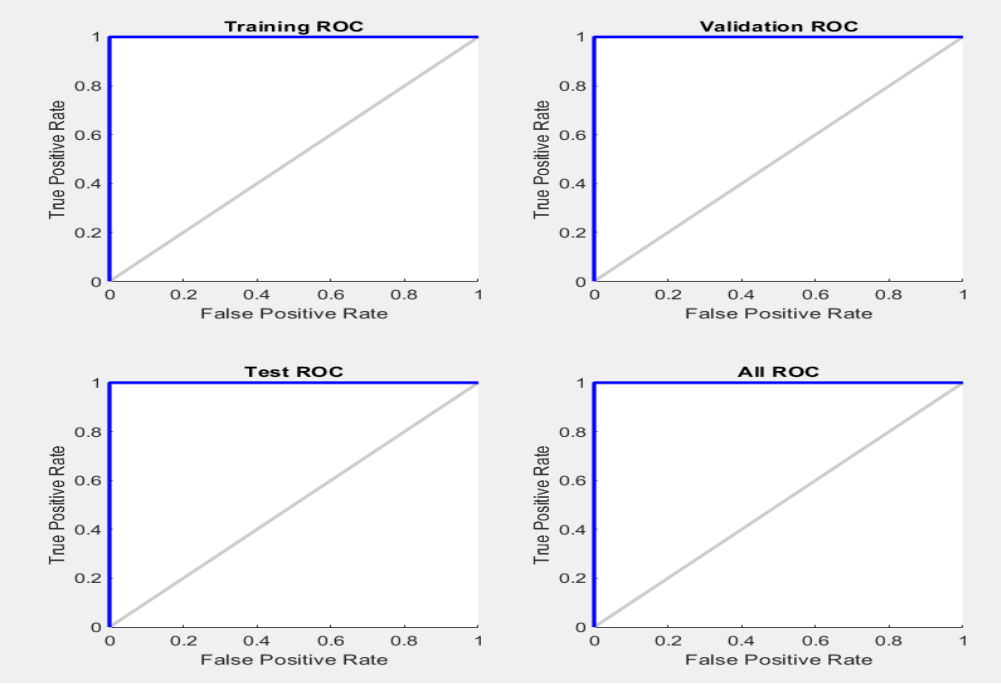

Figure 4-9: The ROC curves for MLPNNs on global dataset when N=50.

Summary of the results of the experiments in applying the classification models (DT, SVM, KNN, and MLPNN) to the global dataset. The SVM model was the best in the classification with an accuracy rate of 99%. In addition, the MLPNN with an accuracy of 99.8% when N=50. Figure (4.10) shows each model and its accuracy rate on test data.

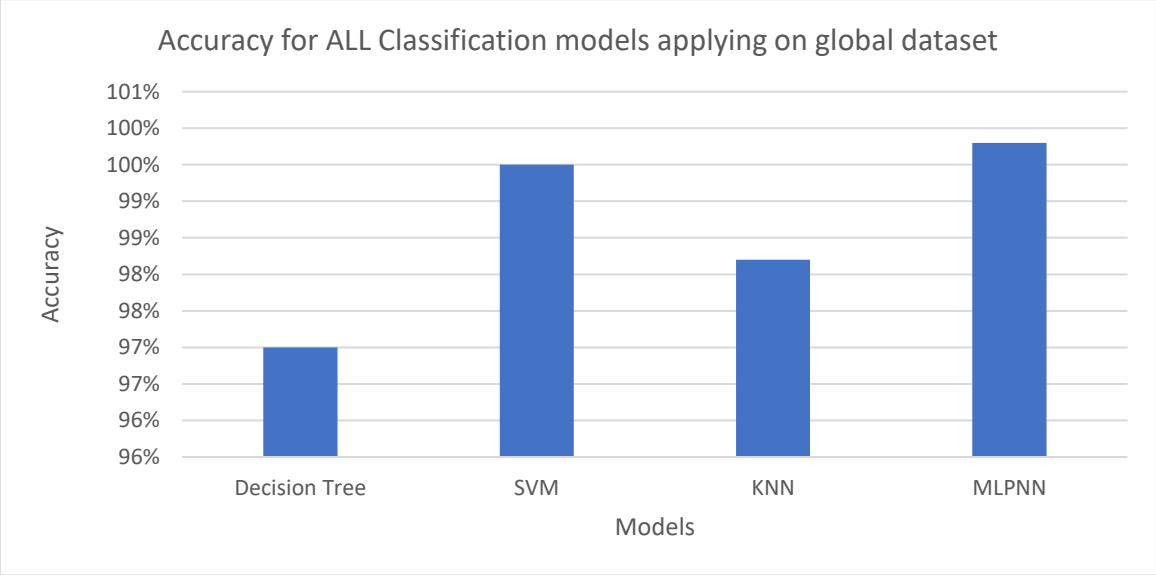

Figure 4-10: The summary of the accuracy results for all the models on the global dataset

4.3.2 The Results of Classification Models for the Local Dataset

- **Application of DT, SVM, and KNN algorithms**

In this part, we applied classification algorithms to the local dataset. They are decision tree (DT), support vector machine (SVM), and K-nearest neighbor (KNN). In our experiment, we used 5-fold cross-validation, 4 times for training and 1 time for testing randomly. The local dataset contained 731 records and each time 146 records were obtained. Table (4.3) shows the results of applying the models to the local dataset and performance metrics for each model.

Table 4-3: The classification results for the models on the local dataset

Model	Accuracy	Specificity	Sensitivity	precision
Decision Tree	96.4%	96.4%	96.5%	96.9%
SVM	96.2%	95.5%	96.7%	96%
KNN	93.6%	91.9%	94.9%	93.3%

The results showed that the best model for classifying the local dataset among the three models is the DT, which had an accuracy rate of 96.4%, followed by the SVM model with an accuracy rate of 96.2% and the KNN model was the least accurate among the three models with an accuracy rate 93.6%. The best accuracy was 96.4%, the best sensitivity was 96.7%, and the best Specificity was 96.4%. The 5-fold cross-validation was used here, where 80% of the data is for training and 20% of the data is for testing. Figure (4.11) shows the comparison between all classification models used in terms of the four-performance metrics. We conclude that the DT model obtained the best results and is the best model in terms of accuracy at 96.4%

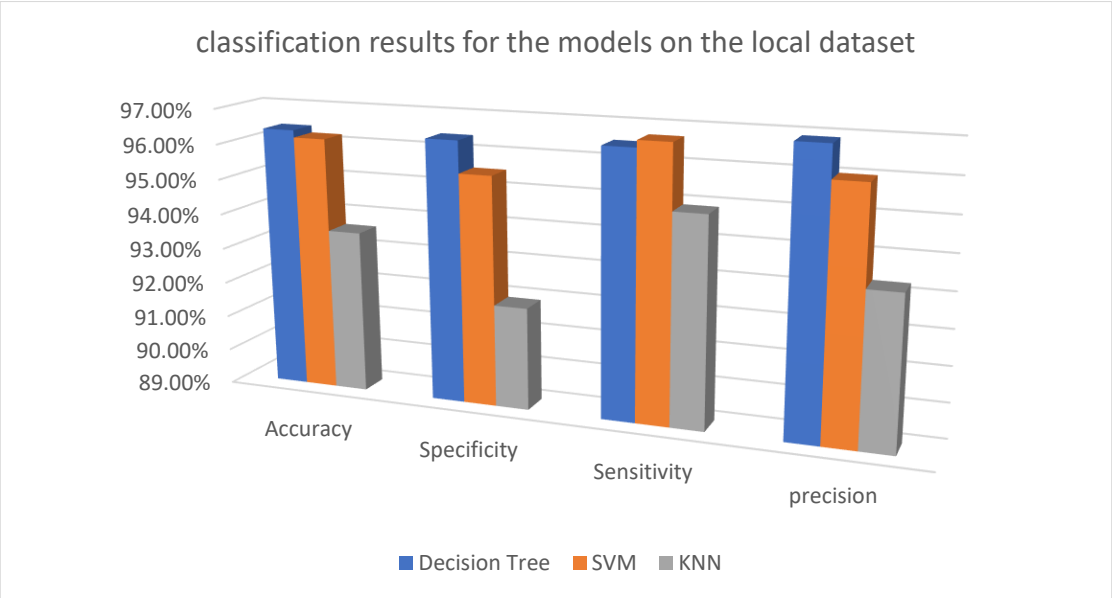

Figure 4-11: The classification results for the models on the local dataset

In Figures (4.12), (4.13), and (4.14) the lines represent the ROC curves, and the areas in blue represent the AUC curves for the three models DT, SVM, and KNN respectively. In each figure, there are two classes: class 0 which does not have chronic kidney disease (not ckd), and class 1 which has chronic kidney disease(ckd). In this curve, the results are considered good whenever the points are in the upper corner of the curve. In the DT model, the area under the curve occupied 97% for both classes and the current classifier was equal (0.04,0.96) (0.04,0.96) for the class 0,1 respectively. In the SVM model, the area under the curve occupied 99% for both classes and the current classifier was equal (0.03,0.95) (0.05,0.97) for the class 0,1 respectively In the KNN model, the area under the curve occupied 93% for both classes and the current classifier was equal (0.05,0.92) (0.08,0.95) for the class 0,1 respectively. We conclude that the SVM model obtained the best results in terms of AUC and ROC curves.

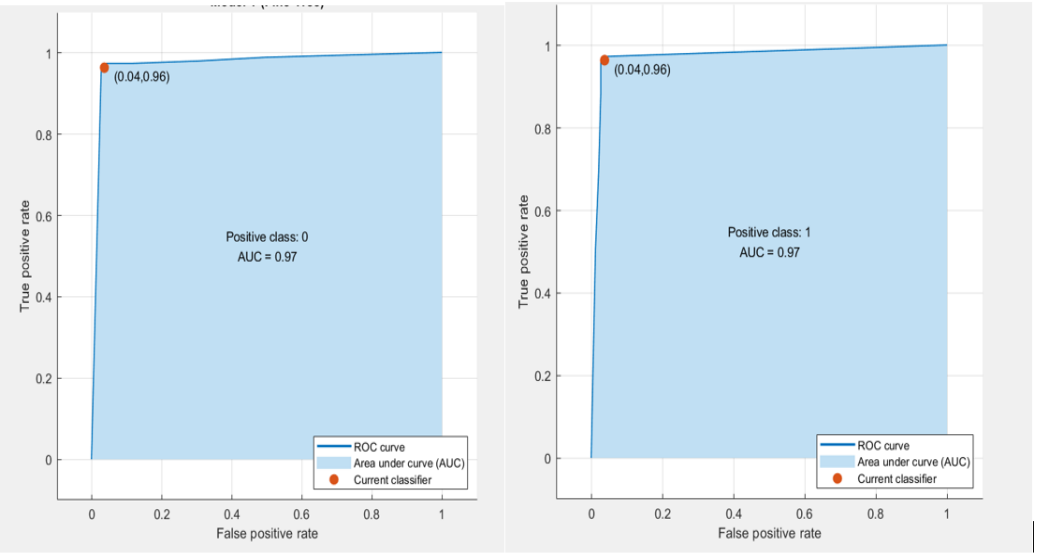

Figure 4-12: AUC & ROC curves by DT model for local dataset (Class 0 & 1)

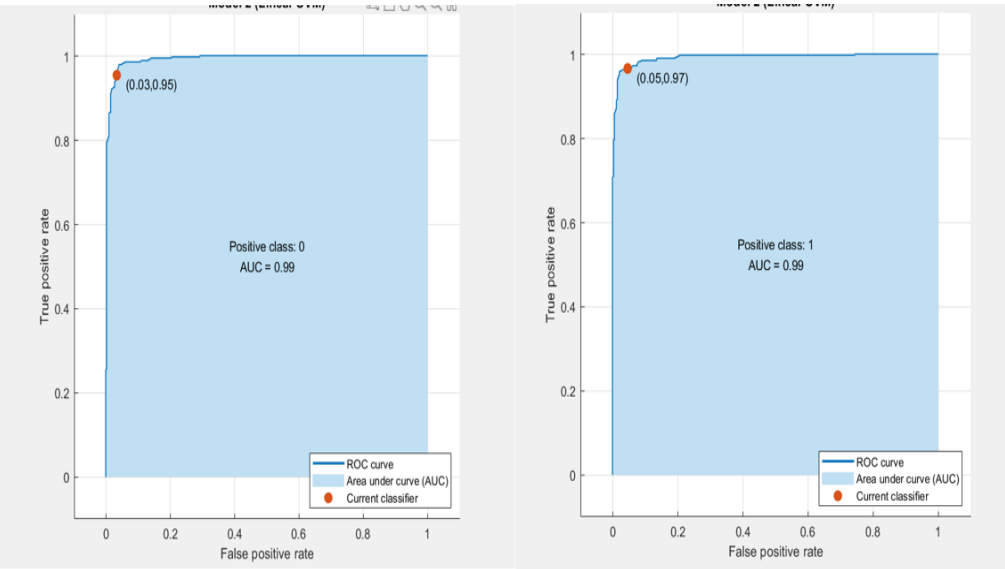

Figure 4-13: AUC & ROC curves by SVM model for local dataset (Class 0 & 1)

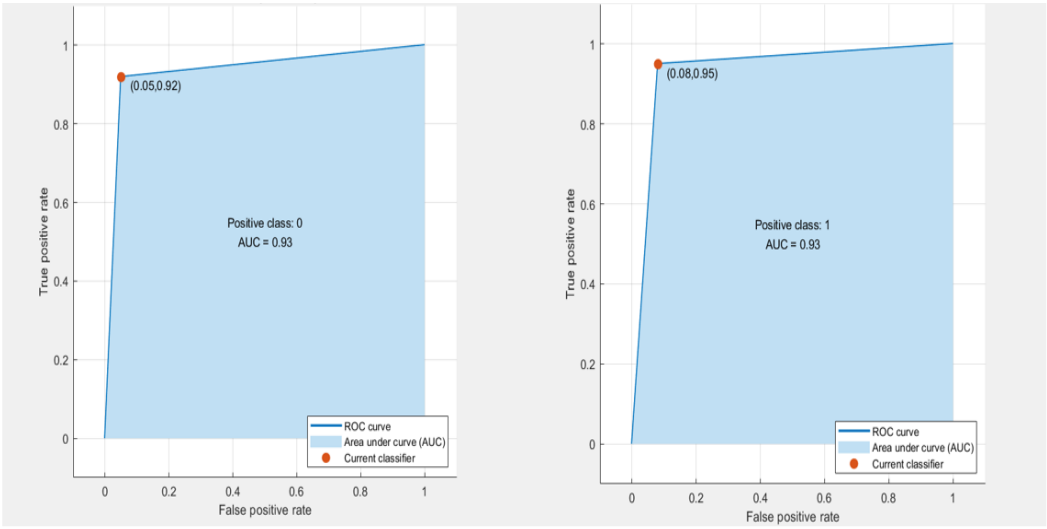

Figure 4-14: AUC & ROC curves by KNN model for local dataset (Class 0 & 1)

❖ Application MLPNNs Model

In this part, we applied the MLPNN model to the local dataset. We used NNs as one of the powerful classification methods and built the network with a two-layer feed-forward .The data is divided into three parts: the training part, which was 70% (511 records), the validation part,

which was 15% (110 records), and the testing part, 15% (110 records). Several tests were conducted, using 10 and 50 neurons in the hidden layer for each test.

Scaled conjugate gradient backpropagation was used for training the local dataset. Table (4.4) shows the accuracy results of changing the number of neurons in the hidden layer after training the data. Figure (4.15) shows the result of accuracy for the model when changing the number of neurons in the hidden layer.

Table 4-4: The Classification Results for MLPNNs on the local dataset

Number of neurons	Test accuracy	All accuracy	Specificity	Sensitivity	Precision
5	90.9%	96.4%	96.4%	96.5%	97%
10	96.4%	97%	97%	97%	97.5%
15	97.3%	97%	97%	97%	97.5%
20	95.5%	97.3%	96.2%	98.2%	96.7%
25	99.1%	97%	96.7%	97.2%	97.2%
30	99.1%	98.1%	97.9%	98.2%	98.2%
35	98.2%	97.1%	96.4%	97.7%	97%
40	94.5%	97%	96.4%	97.4%	97%
45	99.1%	97.3%	96.7%	97.7%	97.3%
50	93.6%	97.3%	97%	97.5%	97.5%

The MLPNNs model achieved the best performance when the number of neurons in the hidden layer equal 30 (N=30) with an accuracy of 99.1% for testing and the result of all accuracy, Specificity, Sensitivity, and precision were 98.1%,97.9%,98.2%, and 98.2% respectively, while the performance of the model when the number of neurons in the hidden layer equal 5 (N=5) was lower as the accuracy rate was 90.9% for testing and the result of the all accuracy, Specificity, Sensitivity and precision were 96.4%,96.4%,96.5% and 97% respectively. We conclude that the more neurons in the hidden layer, the better the performance and classification.

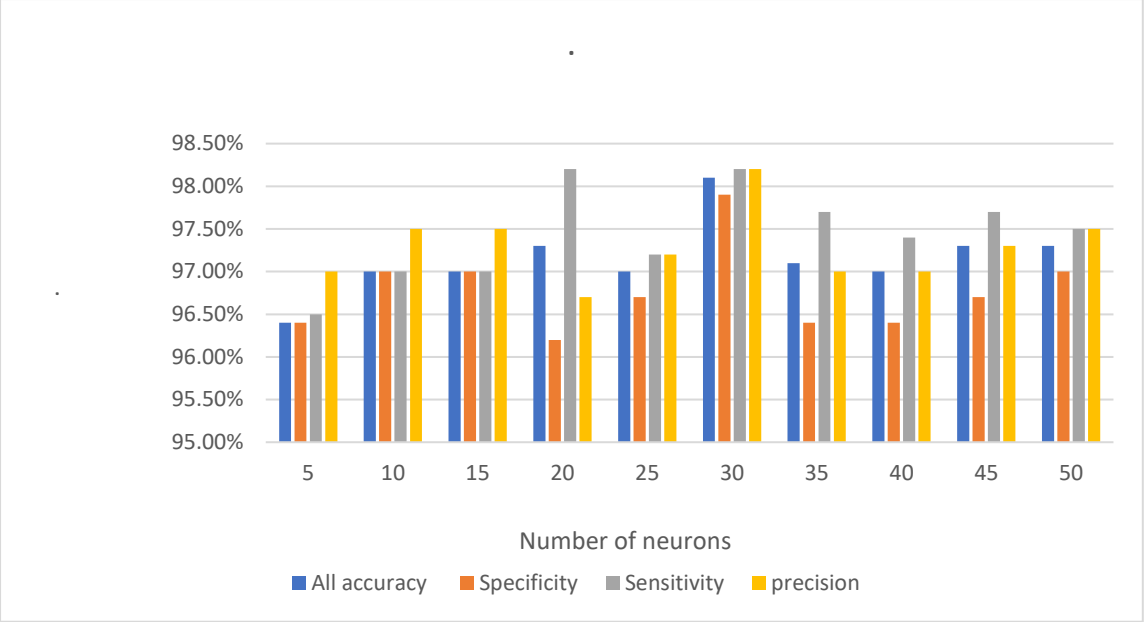

Figure 4-15: Chart for classification results for MLPNN model on local dataset.

The confusion matrices for the MLPNNs model for both N=5 and N=30 is shown in figures (4.16) and (4.17) respectively. The figures illustrate the confusion matrices with all the values of the accuracy for training and testing and all accuracy.

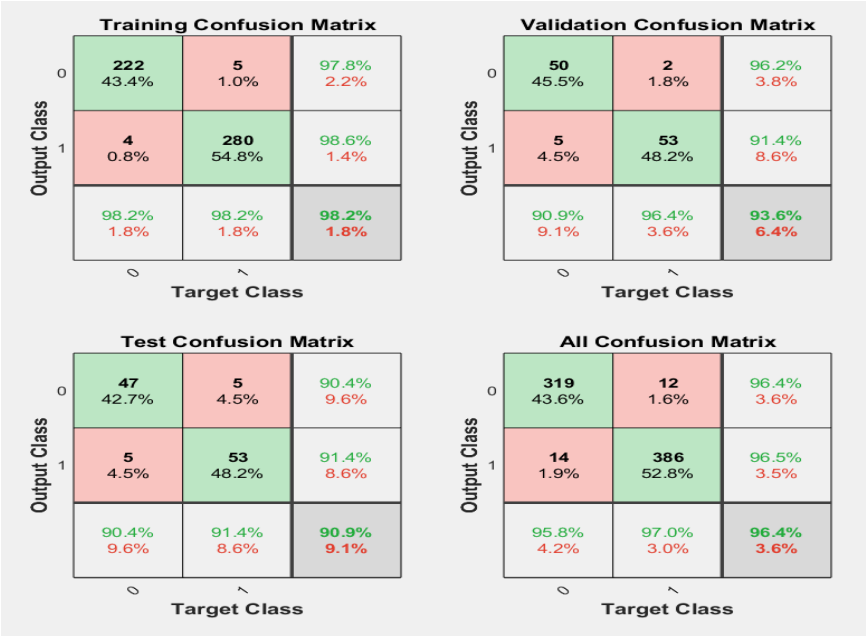

Figure 4-16: Confusion Matrices for MLPNNs on local dataset when N=5.

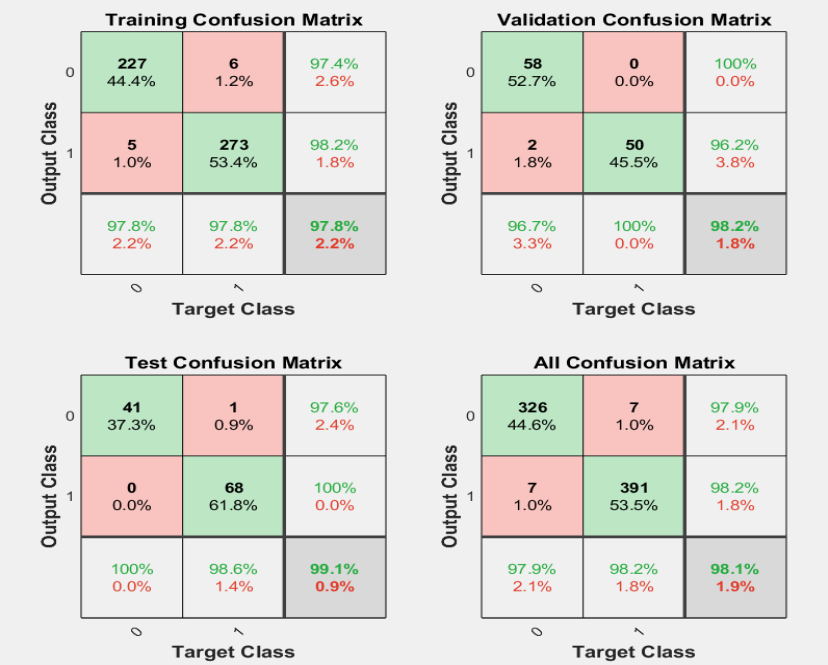

Figure 4-17: Confusion Matrices for MLPNNs on local dataset when N=30

The ROC curve for the MLPNNs model for both N=5 and N=30 is shown in figures (4.18) & (4.19) respectively. the points of the line are in the upper-left corner which means that the performance is good.

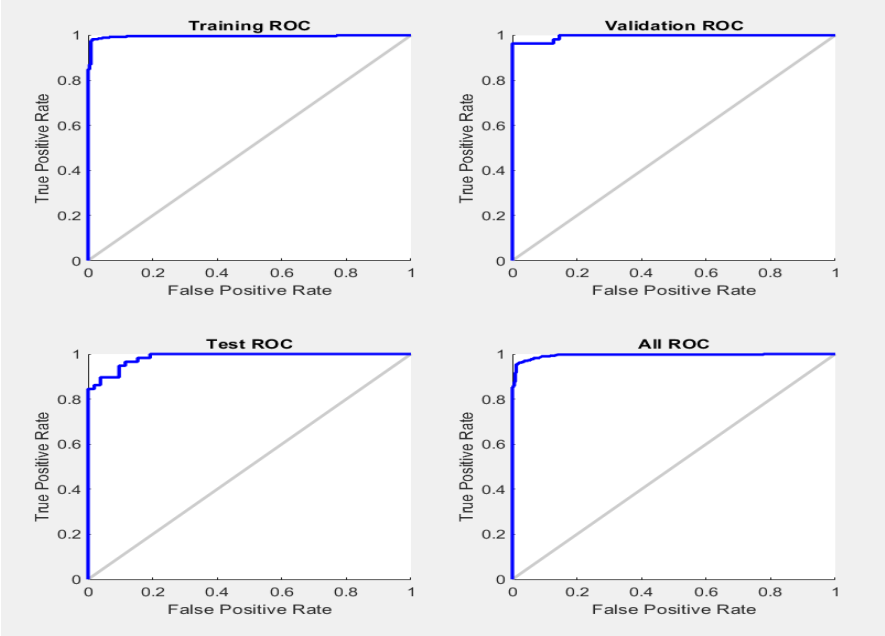

Figure 4-18: The ROC curves for MLPNNs on local dataset when N=5.

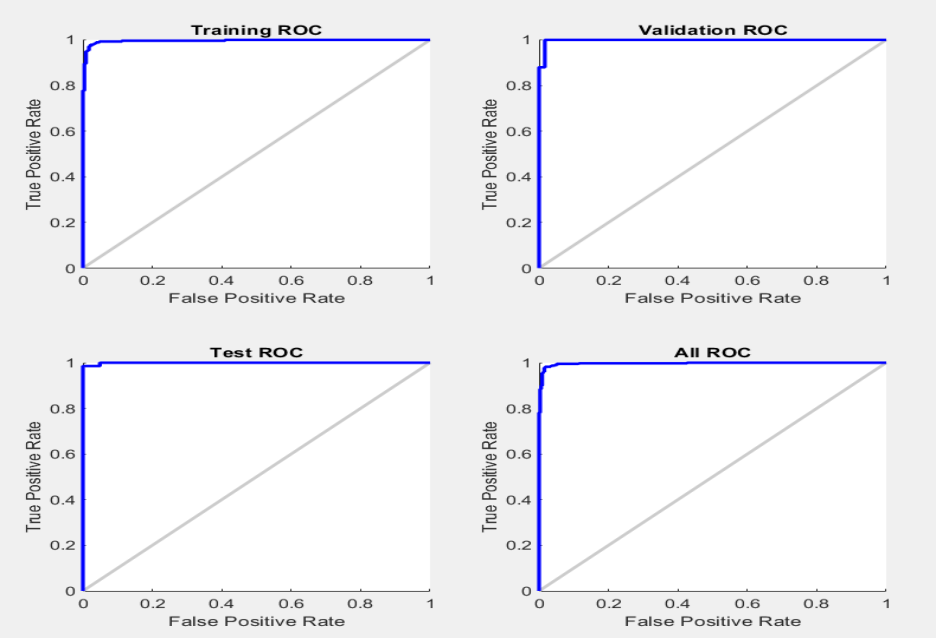

Figure 4-19: The ROC curves for MLPNNs on local dataset when N=30.

Summary of the results of the experiments in applying the classification models (DT, SVM, KNN, and MLPNN) to the local dataset. The DT model was the best in the classification with an accuracy rate of 96.4%. In addition to the MLPNN with an accuracy of 98.1% when N=30. Figure (4.20) shows each model and its accuracy rate on test data.

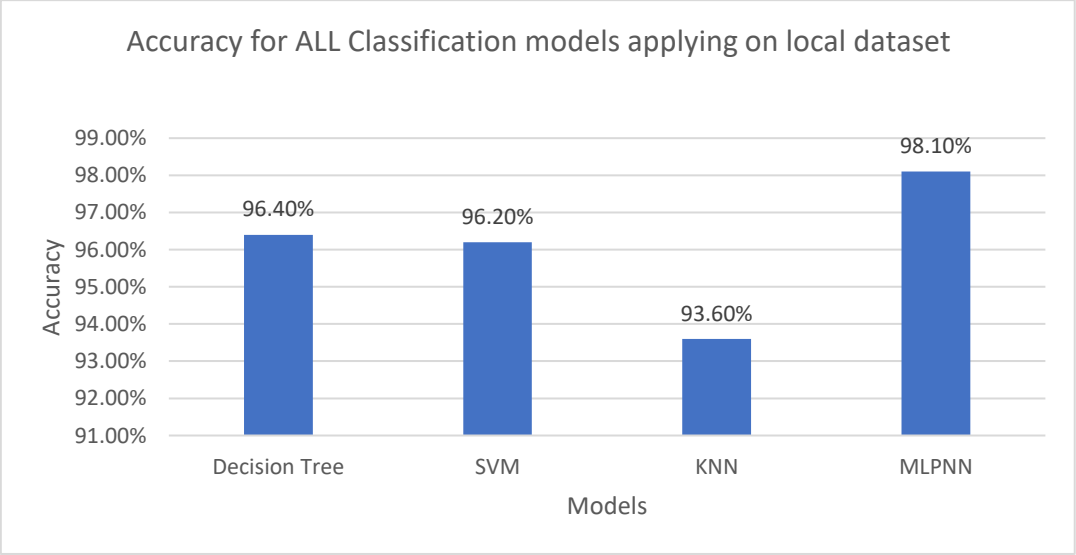

Figure 4-20: The summary of the accuracy results for all the models on the local dataset

4.4 Hybrid Classification Models Experiment

The hybrid models were evaluated by applying them to both the global dataset and the local dataset. The hybrid models which are BBO-MLPNNs, PSO-MLPNNs, and GAs-MLPNNs algorithms were used to create to classification of KCD. The performance of the hybrid models depends on the number of neurons in the hidden layers, the number of iterations, the activation function of the hidden layers which used the sigmoidal activation function, and population size. The performance of the proposed models was evaluated and measured using the confusion matrix, accuracy, Sensitivity, Specificity, precision, and F-score.

4.4.1 The Results of Hybrid Models for the Global Dataset

In this section, the hybrid models (BBO-MLPNNs, PSO-MLPNNs, and GAs-MLPNNs) were used for CKD detection and prediction on the global dataset. The hybrid models with certain numbers of neurons in the hidden layers and certain numbers of iterations were proposed.

- **BBO-MLPNNs Experiments on the Global Dataset**

The BBO-MLPNNs algorithm was applied to a global dataset and experiments were conducted by modifying parameters, namely the number of neurons in the hidden layer and the number of iterations, to obtain the best values for the parameters and improve the performance of the algorithm in detecting and predicting chronic kidney disease. These experiments are shown in the following Table (4.5).

Several experiments were conducted to obtain the best values of the number of neurons in the hidden layer and the number of iterations that achieved the optimal model, and a population number equal to 50 was used for all experiments. The results showed that the best model for BBO-MLPNNs when the number of neurons in the hidden layer was 20, and the accuracy result

was 99.4%, and the rest of the performance parameters were represented by 99.6%, 99.4%, 99.3%, and 99.8% Sensitivity, specificity, precision, and F-score respectively

Table 4-5: BBO-MLPNNs Experiments results on a global dataset

Number of iterations	Number of neurons	Accuracy	Sensitivity	Specificity	precision	F-score
50	10	98.4%	98.4%	98.7%	99.2%	98.8%
100		98.5%	98.4%	98.7%	99.2%	98.8%
150		98.6%	98.4%	98.7%	99.2%	98.8%
200		98.6%	98.4%	98.7%	99.2%	98.8%
250		98.6%	98.8%	98.7%	99.2%	99%
50	15	99.2%	99.2%	99.3%	99.6%	99.4%
100		99.3%	99.2%	99.3%	99.6%	99.4%
150		99.3%	99.2%	99.3%	99.6%	99.4%
200		99.3%	99.2%	99.3%	99.6%	99.4%
250		99.3%	99.2%	99.3%	99.6%	99.4%
50	20	99.3%	99.2%	99.3%	99.6%	99.4%
100		99.4%	99.6%	99.3%	99.6%	99.6%
150		99.4%	99.6%	99.3%	99.6%	99.6%
200		99.4%	99.6%	99.3%	99.6%	99.6%
250		99.4%	99.6%	99.4%	99.3%	99.8%

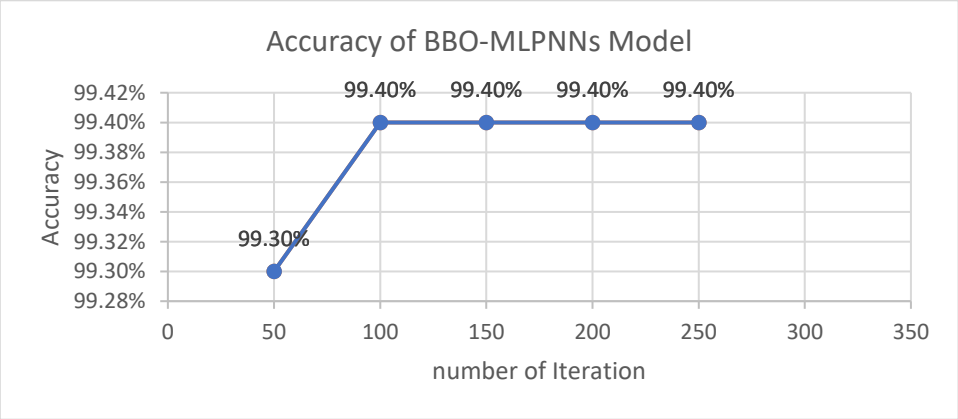

Figure 4-21: The accuracy of BBO-MLPNNs model related to the number of iterations.

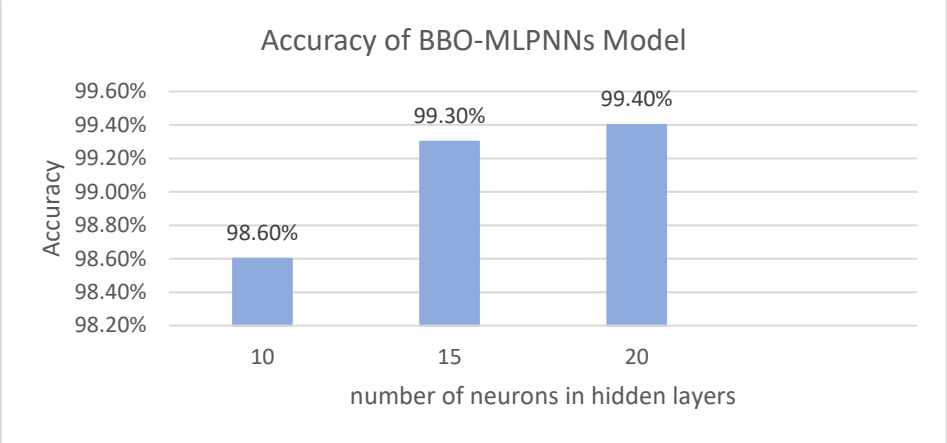

Figure 4-22: The accuracy of BBO-MLPNNs model related to the number of neurons.

Figure (4.21) shows the accuracy of the BBO-MLPNNs model when the number of neurons in the hidden layers equals 20 and the numbers of the iteration's values are 50,100,150, 200, 250. The accuracy was 99.3% when the number of the iterations was 50 then the accuracy increased to 99.4% when the number of the iterations was 100, then it remained constant as the number of iterations increased. Figure (4.22) shows a comparison between the accuracy of the BBO-MLPNNs model when the number of iterations equal 250 and number of neurons in the hidden layer's values are 10,15, 20. The best accuracy of 99.4 % when the number of neurons =20, while the accuracy of 99.3% when several neurons = is 15, and 98.6% when the number of neurons =10.Figure (4.23) show the ROC curve for BBO-MLPNNs model when the number of iterations equal 250 and number of neurons in the hidden layer's is equal 20 . the points of the line are in the upper-left corner which means that the performance is good.

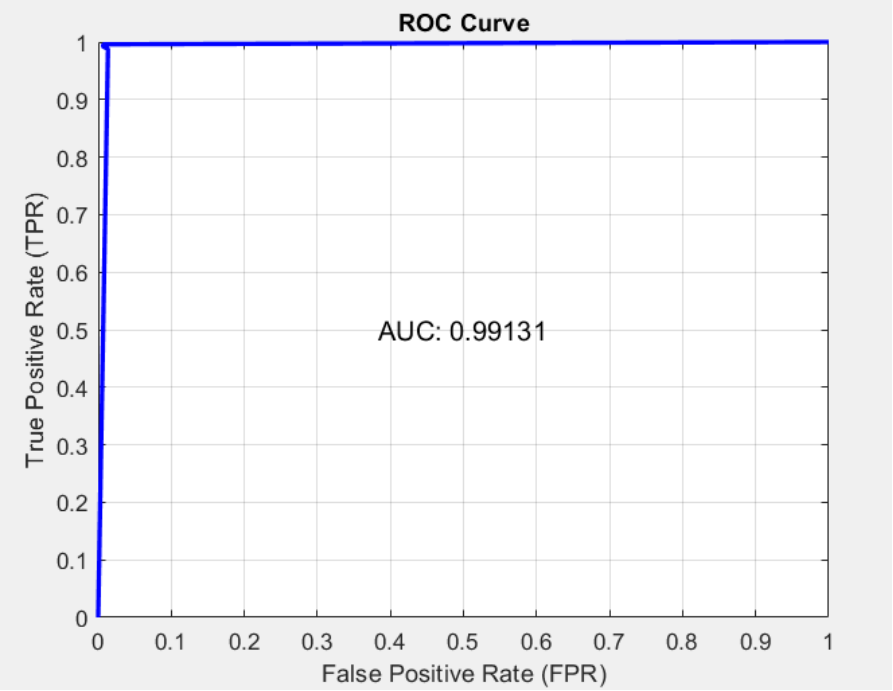

Figure 4-23: The ROC curves for BBO-MLPNNs model

❖ PSO-MLPNNs Experiments on the Global Dataset

The PSO-MLPNNs algorithm was applied to a global dataset and experiments were conducted by modifying parameters, namely the number of neurons in the hidden layer and the number of iterations, to obtain the best values for the parameters and improve the performance of the algorithm in detecting and predicting chronic kidney disease. These experiments are shown in the following Table (4.6).

Table 4-6: PSO-MLPNNs Experiments results on a global dataset

Number of iterations	Number neurons	Accuracy	Sensitivity	Specificity	precision	F-score
50	10	93.1%	93.2%	93.3%	95.9%	94.5%
100		93.3%	93.2%	93.3%	95.9%	94.5%
150		94.6%	94.4%	94.7%	96.7%	95.5%
200		94.7%	94.8%	94.7%	96.7%	95.8%
250		94.7 %	94.8%	94.7%	96.7%	95.8%
50	15	94.1%	94%	94%	96.3%	95.1%
100		94.6%	94.4%	94.7%	96.7%	95.5%
150		96.7%	96.8%	96.7%	98%	97.4%
200		96.8%	96.8%	96.7%	98%	97.4%

250		96.8%	96.8%	96.7%	98%	97.4%
50	20	95.3%	95.2%	95.3%	97.1%	96.2%
100		97%	97.2%	97.3%	98.4%	97.8%
150		97%	97.2%	97.3%	98.4%	97.8%
200		97%	97.2%	97.3%	98.4%	97.8%
250		97%	97.2%	97.3%	98.4%	97.8%

Several experiments were conducted to obtain the best values of the number of neurons in the hidden layer and the number of iterations that achieved the optimal model, and a population number equal to 50 was used for all experiments. The results showed that the best model for PSO-MLPNNs when the number of neurons in the hidden layer was 20, and the accuracy result was 97%, and the rest of the performance parameters were represented by 97.2%, 97.3%, 98.4%, and 97.8% Sensitivity, specificity, precision, and F-score respectively.

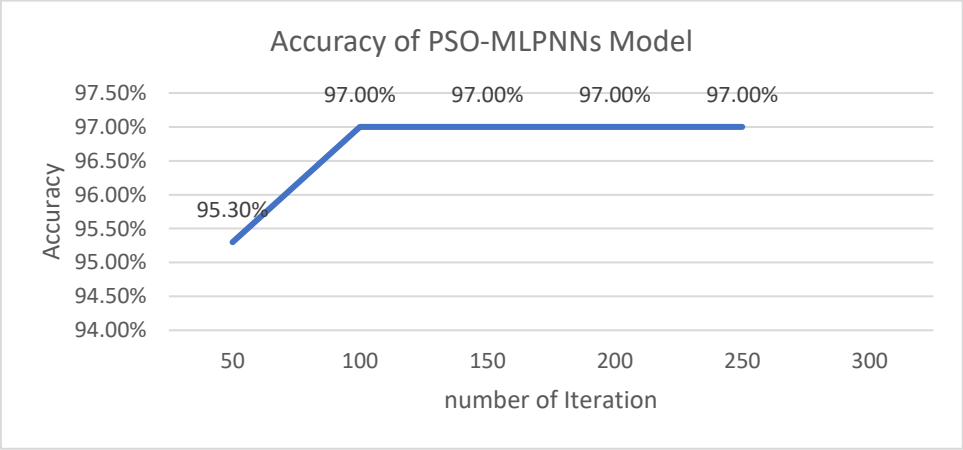

Figure 4-24: The accuracy of PSO-MLPNNs model related to the number of iterations.

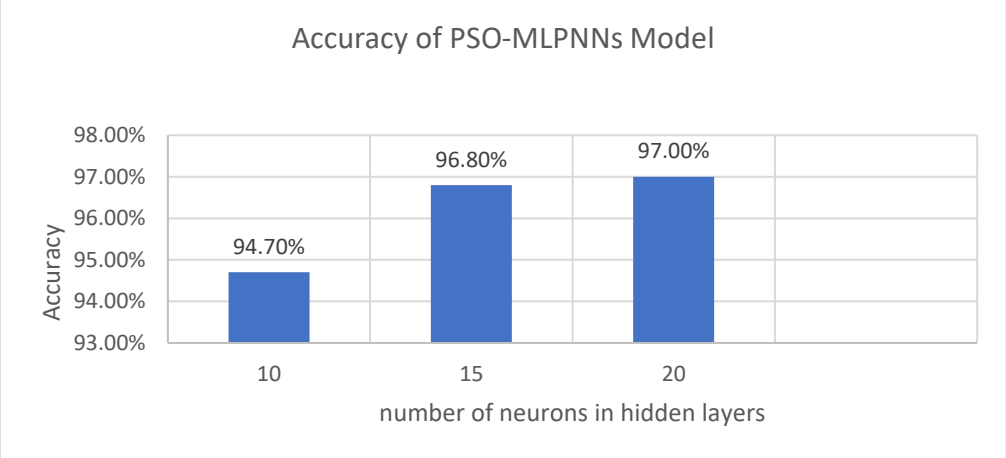

Figure 4-25: The accuracy of PSO-MLPNNs model related to the number of neurons.

Figure (4.24) shows the accuracy of the PSO-MLPNNs model when the number of neurons in the hidden layers equals 20 and the numbers of the iteration's values are 50,100,150, 200, and 250. The accuracy was 95.3% when the number of the iterations was 50 then the accuracy increased to 97% when the number of the iterations was 100, then it remained constant as the number of iterations increased. Figure (4.25) shows a comparison between the accuracy of the PSO-MLPNNs model when the number of iterations equals 250 and the number of neurons in the hidden layer's values are 10,15 and 20. The best accuracy of 97 % when several neurons = is 20, while the accuracy of 96.8% when the number of neurons = is 15, and 94.7% when the number of neurons =10. Figure (4.26) show the ROC curve for PSO -MLPNNs model when the number of iterations equal 250 and number of neurons in the hidden layer's is equal 20. the points of the line are in the upper-left corner which means that the performance is good.

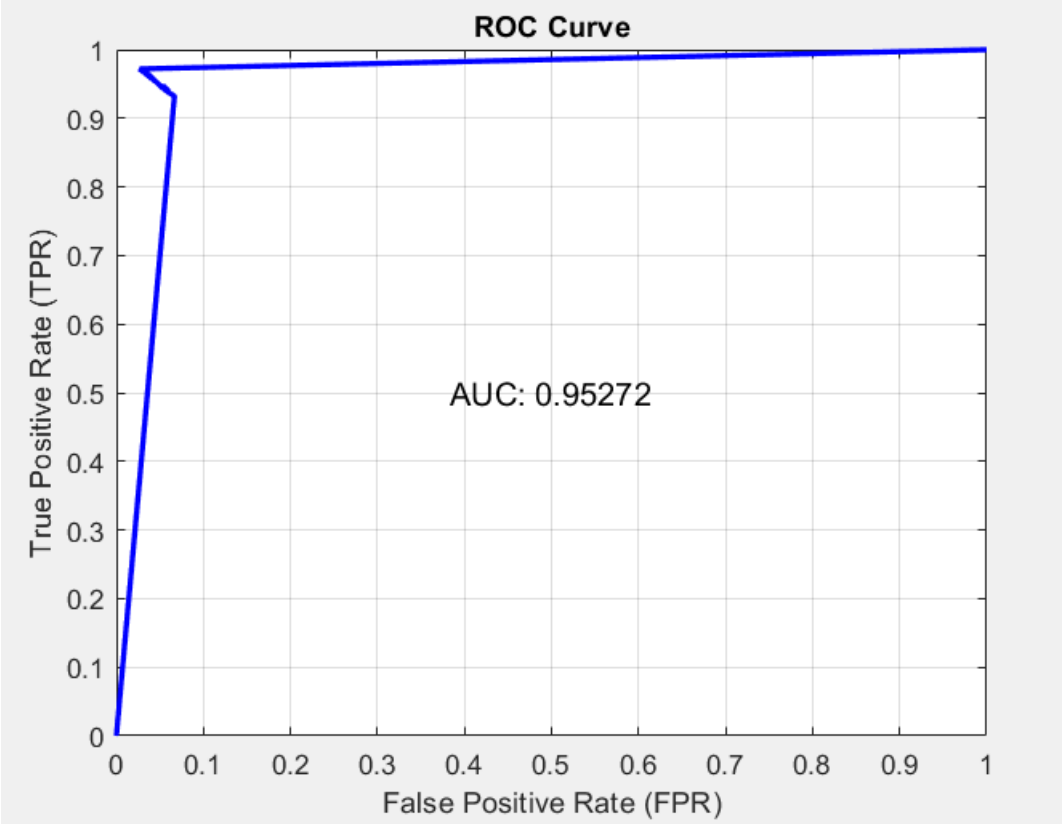

Figure 4-26: The ROC curves for PSO -MLPNNs model

❖ GA-MLPNNs Experiments on the Global Dataset

The GA-MLPNNs algorithm was applied to a global dataset and experiments were conducted by modifying parameters, namely the number of neurons in the hidden layer and the number of iterations, to obtain the best values for the parameters and improve the performance of the algorithm in detecting and predicting chronic kidney disease. These experiments are shown in the following Table (4.7).

Several experiments were conducted to obtain the best values of the number of neurons in the hidden layer and the number of iterations that achieved the optimal model, and a population number equal to 50 was used for all experiments. The results showed that the best model for GA-MLPNNs when the number of neurons in the hidden layer was 20, and the accuracy result was 99.5%, and the rest of the performance parameters were represented by 99.6%, 99.3%, 99.6%, and 99.6% Sensitivity, specificity, precision, and F-score respectively

Table 4-7: GA-MLPNNs Experiments Results on a global dataset

Number of Iterations	Number Neurons	Accuracy	Sensitivity	Specificity	precision	F-score
50	10	98.3%	98.4%	98%	98.8%	98.6%
100		98.4%	98.4%	98.7%	99.2%	98.8%
150		98.5%	98.4%	98.7%	99.2%	98.8%
200		98.6%	98.4%	98.7%	99.2%	98.8%
250		98.6%	98.4%	98.7%	99.2%	98.8%
50	15	98.9%	98.8%	98.7%	99.2%	99%
100		99.2%	99.2%	99.3%	99.6%	99.4%
150		99.2%	99.2%	99.3%	99.6%	99.4%
200		99.3%	99.2%	99.3%	99.6%	99.4%
250		99.3%	99.2%	99.3%	99.6%	99.4%
50	20	99.2%	99.2%	99.3%	99.6%	99.4%
100		99.3%	99.2%	99.3%	99.6%	99.4%
150		99.4%	99.2%	99.3%	99.6%	99.4%
200		99.4%	99.2%	99.3%	99.6%	99.4%
250		99.5%	99.6%	99.3%	99.6%	99.6%

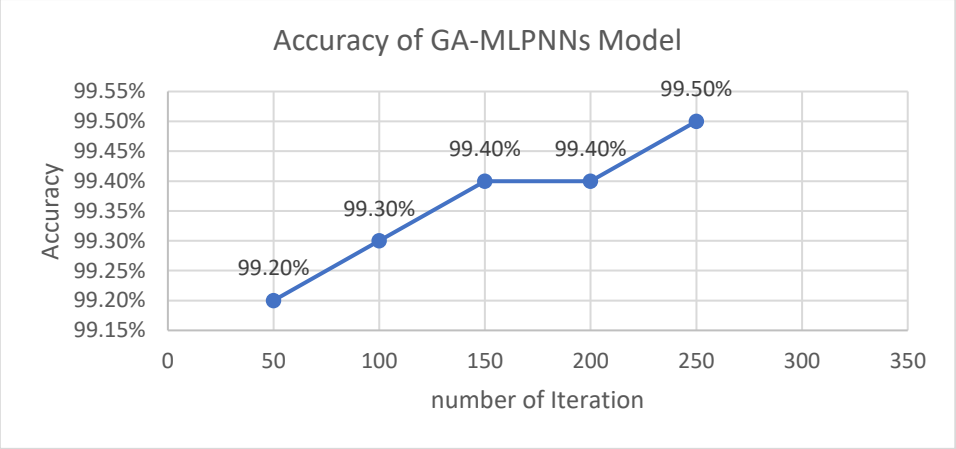

Figure 4-27: The accuracy of GA-MLPNNs model related to the number of iterations.

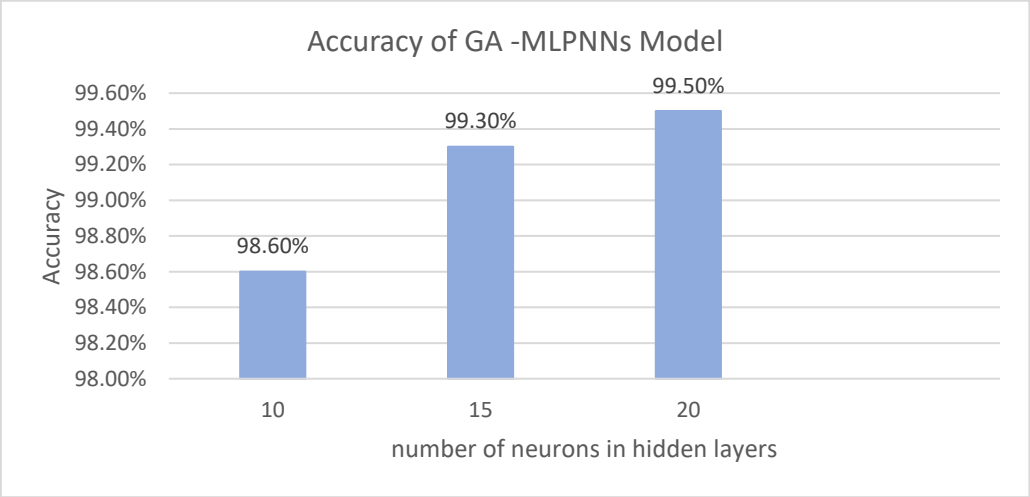

Figure 4-28: The accuracy of GA-MLPNNs model related to the number of neurons.

Figure (4.27) shows the accuracy of the GA-MLPNNs model when the number of neurons in the hidden layers equals 20 and the numbers of the iteration's values are 50,100,150, 200, 250. The accuracy increases as the number of iterations increases, the highest accuracy value for the model is 99.5% when the numbers of the iterations equal 250. Figure (4.28) shows a comparison between the accuracy of the GA-MLPNNs model when the number of iterations equals 250 and the number of neurons in the hidden layer's values is 10,15, 20. The best accuracy of 99.5 % when the number of neurons =20, while the accuracy of 99.3% when the number of neurons =15, and 98.6% when the number of neurons =10. Figure (4.29) show the ROC curve for GA-MLPNNs model when the number of iterations equal 250 and number of

neurons in the hidden layer's is equal 20. the points of the line are in the upper-left corner which means that the performance is good.

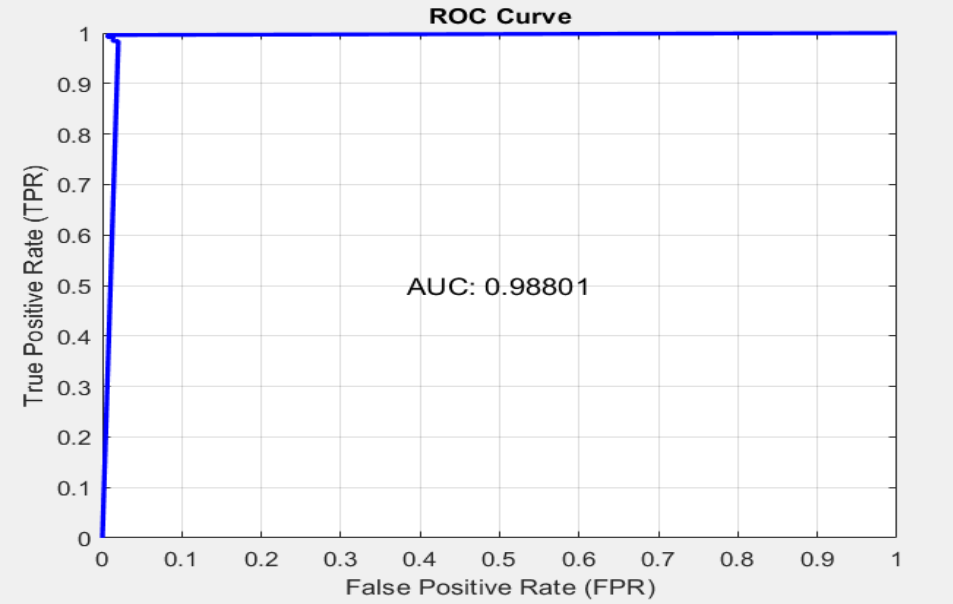

Figure 4-29: The ROC curves for GA-MLPNNs model

4.4.2 The results of hybrid models for the Local dataset

In this section, the hybrid models (BBO-MLPNNs, PSO-MLPNNs, and GAs-MLPNNs) were used for chronic kidney disease detection and prediction on the Palestinian local dataset. The hybrid models with certain numbers of neurons in the hidden layers and certain numbers of iterations were proposed.

- **BBO-MLPNNs Experiments on the Local Dataset**

The BBO-MLPNNs algorithm was applied to the local dataset and experiments were conducted by modifying parameters, namely the number of neurons in the hidden layer and the number of iterations, to obtain the best values for the parameters and improve the performance of the algorithm in detecting and predicting chronic kidney disease. These experiments are shown in the following Table (4.8).

Table 4-8: BBO-MLPNNs Experiments results on a local dataset

Number of iterations	Number of neurons	Accuracy	Sensitivity	Specificity	precision	F-score
50	10	98.4%	98.5%	98.5%	98.7%	98.6%
100		98.5%	98.5%	98.5%	98.7%	98.6%
150		98.5%	98.5%	98.5%	98.7%	98.6%
200		98.5%	98.5%	98.5%	98.7%	98.6%
250		98.5%	98.5%	98.5%	98.7%	98.6%
50	15	98.9%	98.7%	98.8%	99%	98.9%
100		98.9%	99%	98.8%	99%	99%
150		98.9%	99%	98.8%	99%	99%
200		98.9%	99%	98.8%	99%	99%
250		98.9%	99%	98.8%	99%	99%
50	20	99%	99%	99.1%	99.2%	99.1%
100		99%	99%	99.1%	99.2%	99.1%
150		99%	99%	99.1%	99.2%	99.1%
200		99%	99%	99.1%	99.2%	99.1%
250		99%	99%	99.1%	99.2%	99.1%

Several experiments were conducted to obtain the best values of the number of neurons in the hidden layer and the number of iterations that achieved the optimal model, and a population number equal to 50 was used for all experiments. The results showed that the best model for BBO-MLPNNs when the number of neurons in the hidden layer was 20, the accuracy result was 99 %, and the rest of the performance parameters were represented by 99%, 99.1%, 99.2%, and 99.1% Sensitivity, specificity, precision, and F-score respectively.

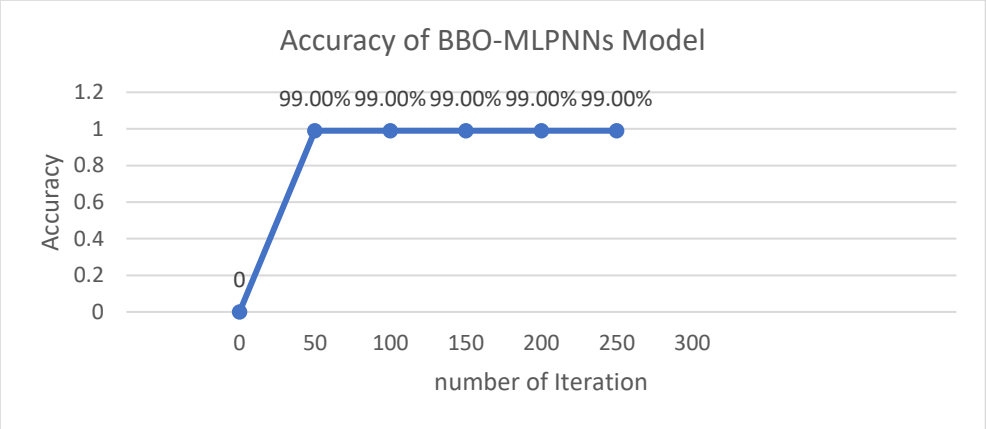

Figure 4-30: The accuracy of BBO-MLPNNs model related to the number of iterations.

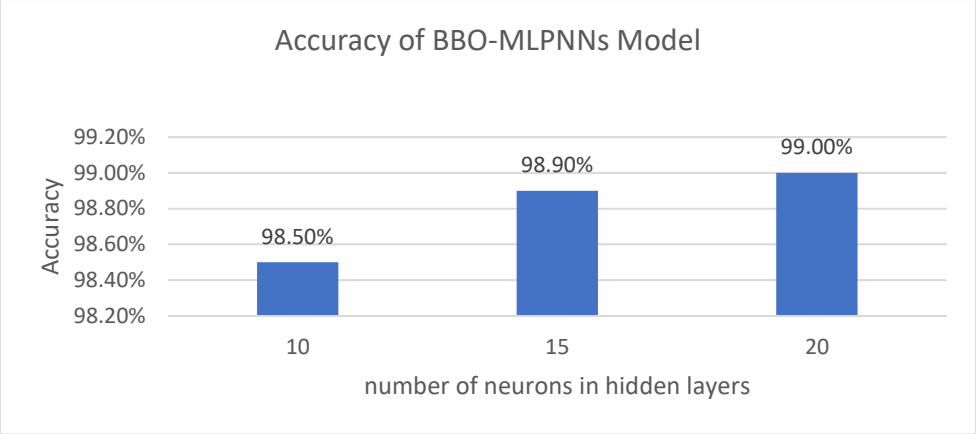

Figure 4-31:The accuracy of BBO-MLPNNs model related to the number of neurons.

Figure (4.30) shows the accuracy of the BBO-MLPNNs model when the number of neurons in the hidden layers equals 20 and the numbers of the iteration's values are 50,100,150, 200, 250. The accuracy remains at a constant value equal to 99% as the number of iterations increases. Figure (4.31) shows a comparison between the accuracy of the BBO-MLPNNs model when the number of iterations equals 250 and the number of neurons in the hidden layer's values are 10,15 and 20. The best accuracy of 99.0 % when the number of neurons = is 20, while the accuracy of 98.9 % when the number of neurons =15, and 98.5% when the number of neurons =10. Figure (4.32) show the ROC curve for BBO-MLPNNs model when the number of iterations equal 250 and number of neurons in the hidden layer's is equal 20. The points of the line are in the upper-left corner which means that the performance is good.

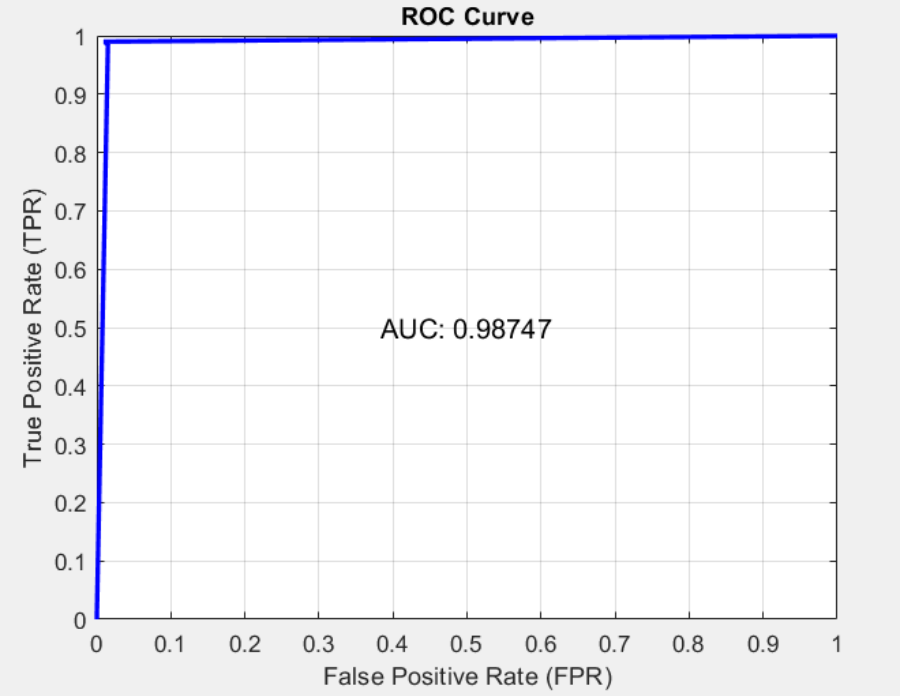

Figure 4-32: The ROC curves for BBO-MLPNNs model

❖ **PSO-MLPNNs Experiments on the Local Dataset**

The PSO-MLPNNs algorithm was applied to the local dataset and experiments were conducted by modifying parameters, namely the number of neurons in the hidden layer and the number of iterations, to obtain the best values for the parameters and improve the performance of the algorithm in detecting and predicting chronic kidney disease. These experiments are shown in the following Table (4.9).

Several experiments were conducted to obtain the best values of the number of neurons in the hidden layer and the number of iterations that achieved the optimal model, and a population number equal to 50 was used for all experiments. The results showed that the best model for PSO-MLPNNs when the number of neurons in the hidden layer was 20, and the accuracy result was 96.7 %, and the rest of the performance parameters were represented by 97%, 97%, 97.5%, and 97.2% Sensitivity, specificity, precision, and F-score respectively.

Table 4-9: PSO-MLPNNs Experiments results on a local dataset

Number of iterations	Number of neurons	Accuracy	Sensitivity	Specificity	precision	F-score
50	10	95.4%	95.5%	95.5%	96.2%	95.8%
100		95.4%	95.5%	95.5%	96.2%	95.8%
150		95.4%	95.5%	95.5%	96.2%	95.8%
200		95.6%	95.5%	95.5%	96.2%	95.8%
250		95.6%	95.5%	95.5%	96.2%	95.8%
50	15	94.6%	94.7%	94.6%	95.4%	95.1%
100		95.9%	96%	95.8%	96.5%	96.2%
150		95.9%	96%	95.8%	96.5%	96.2%
200		95.9%	96%	95.8%	96.5%	96.2%
250		95.9%	96%	95.8%	96.5%	96.2%
50	20	95.3%	95.2%	95.2%	95.9%	95.6%
100		96.4%	96.5%	96.4%	97%	96.7%
150		96.6%	96.5%	96.7%	97.2%	96.8%
200		96.6%	96.5%	96.7%	97.2%	96.8%
250		96.9%	97%	97%	97.5%	97.2%

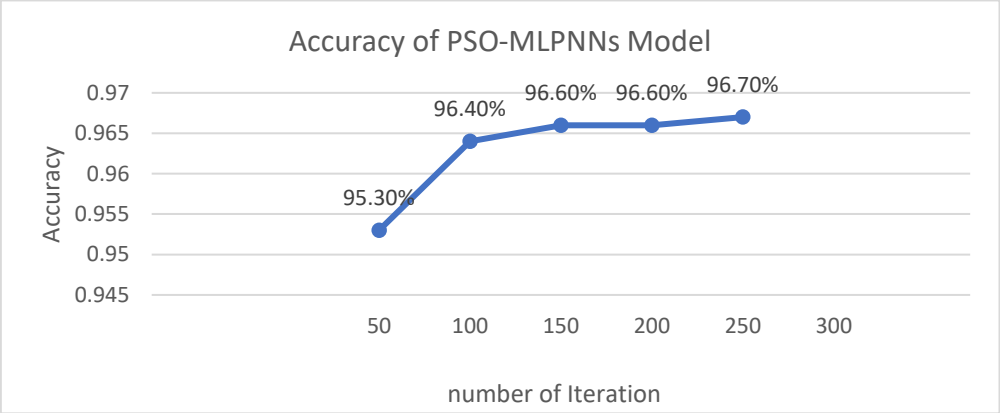

Figure 4-33: The accuracy of PSO-MLPNNs model related to the number of iterations.

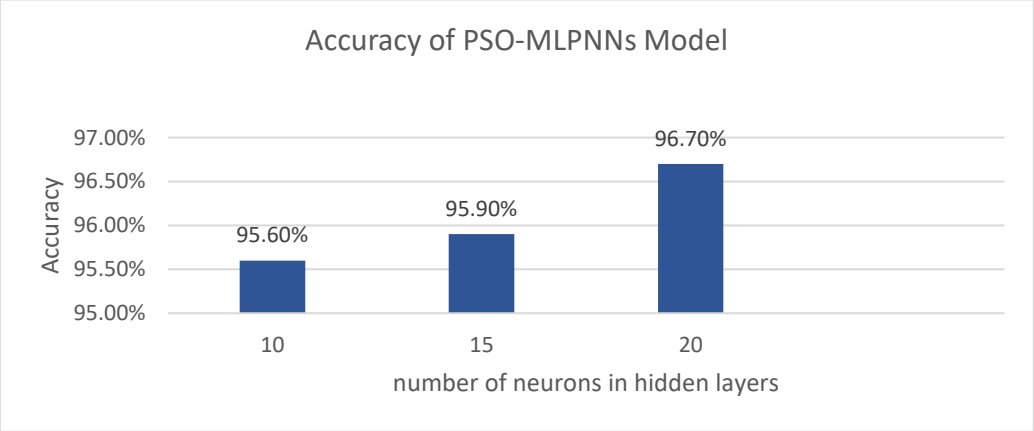

Figure 4-34: The accuracy of PSO-MLPNNs model related to the number of neurons.

Figure (4.33) shows the accuracy of the PSO-MLPNNs model when the number of neurons in the hidden layers equals 20 and the numbers of the iteration's values are 50,100,150, 200, 250. The accuracy was 95.3% when the number of the iterations was 50 then the accuracy increased as the number of iterations increased, the highest accuracy value for the model was 96.7% when the number of the iterations equaled 250. Figure (4.34) shows a comparison between the accuracy of the PSO-MLPNNs model when the number of iterations equals 250 and the number of neurons in the hidden layer's values are 10,15 and 20. The best accuracy of 96.7% when the number of neurons =20, while the accuracy of 95.9% when the number of neurons =15, and 95.6% when the number of neurons =10. Figure (4.35) show the ROC curve for PSO-MLPNNs model when the number of iterations equal 250 and number of neurons in the hidden layer's is equal 20. the points of the line are in the upper-left corner which means that the performance is good.

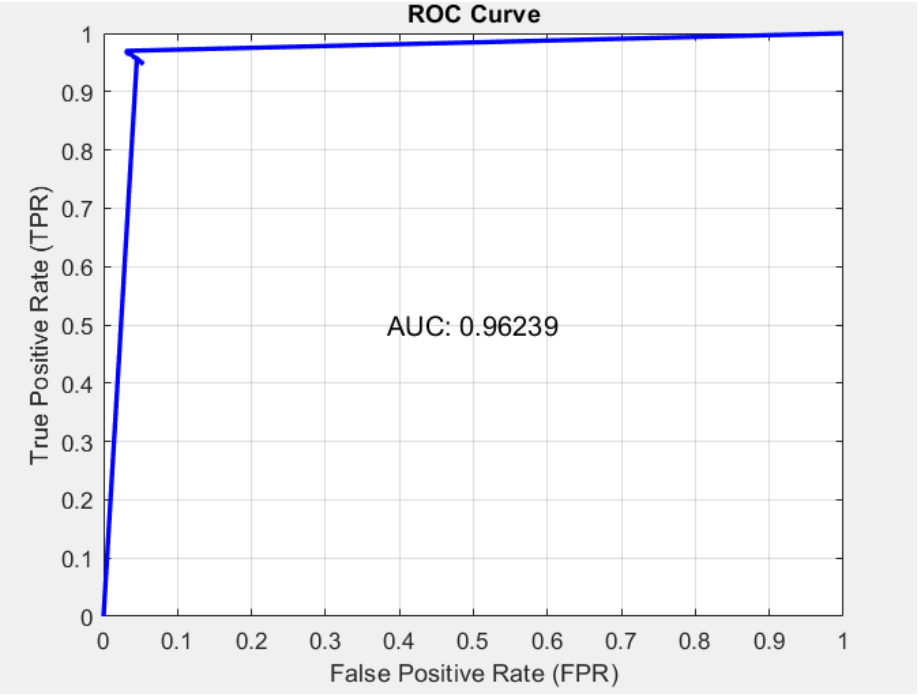

Figure 4-35: The ROC curves for PSO-MLPNNs model

❖ GA-MLPNNs Experiments on the Local Dataset

The GA-MLPNNs algorithm was applied to the local dataset and experiments were conducted by modifying parameters, namely the number of neurons in the hidden layer and the number of iterations, to obtain the best values for the parameters and improve the performance of the algorithm in detecting and predicting chronic kidney disease. These experiments are shown in the following Table (4.10).

Table 4-10: GA-MLPNNs Experiments results on a local dataset

Number of iterations	Number of neurons	Accuracy	Sensitivity	Specificity	Precision	F-score
50	10	97.6%	97.5%	97.6%	98%	97.7%
100		98.2%	98.2%	98.2%	98.5%	98.4%
150		98.4%	98.5%	98.5%	98.7%	98.6%
200		98.4%	98.5%	98.5%	98.7%	98.6%
250		98.5%	98.5%	98.5%	98.7%	98.6%
50	15	98.4%	98.5%	98.5%	98.7%	98.6%
100		98.8%	98.7%	98.8%	99%	98.9%
150		98.8%	98.7%	98.8%	99%	98.9%
200		98.9%	99%	98.8%	99%	99%
250		98.9%	99%	98.8%	99%	99%
50	20	98.8%	98.7%	98.8%	99%	98.9%
100		98.9%	99%	98.8%	99%	99%
150		99%	99%	99.1%	99.2%	99.1%
200		99%	99%	99.1%	99.2%	99.1%
250		99%	99%	99.1%	99.2%	99.1%

Several experiments were conducted to obtain the best values of the number of neurons in the hidden layer and the number of iterations that achieved the optimal model, and a population number equal to 50 was used for all experiments. The results showed that the best model for GA-MLPNNs when the number of neurons in the hidden layer was 20, the accuracy result was 99%, and the rest of the performance parameters were represented by 99%, 99.1%, 99.2%, and 99.1% Sensitivity, specificity, precision, and F-score respectively.

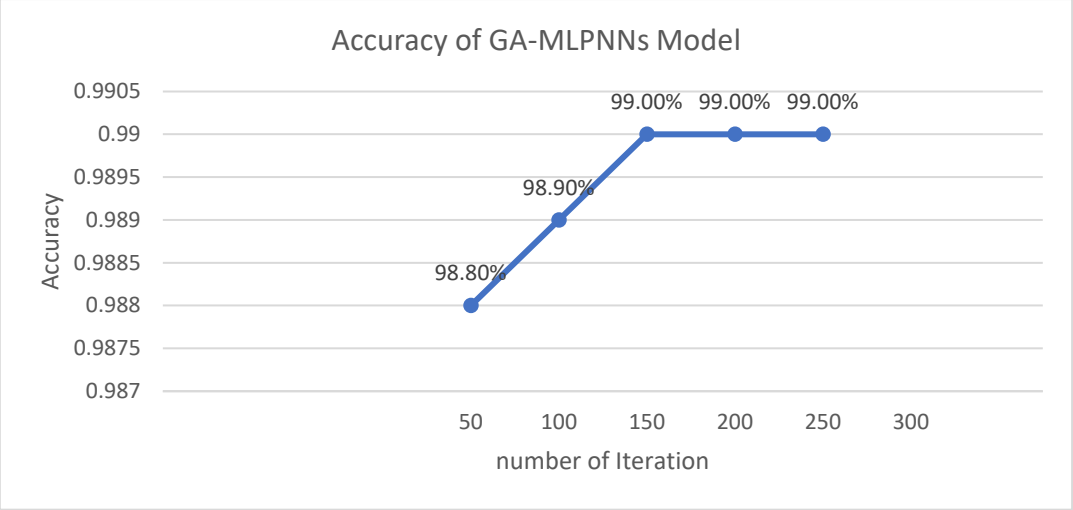

Figure 4-36:The accuracy of GA-MLPNNs model related to the number of iterations.

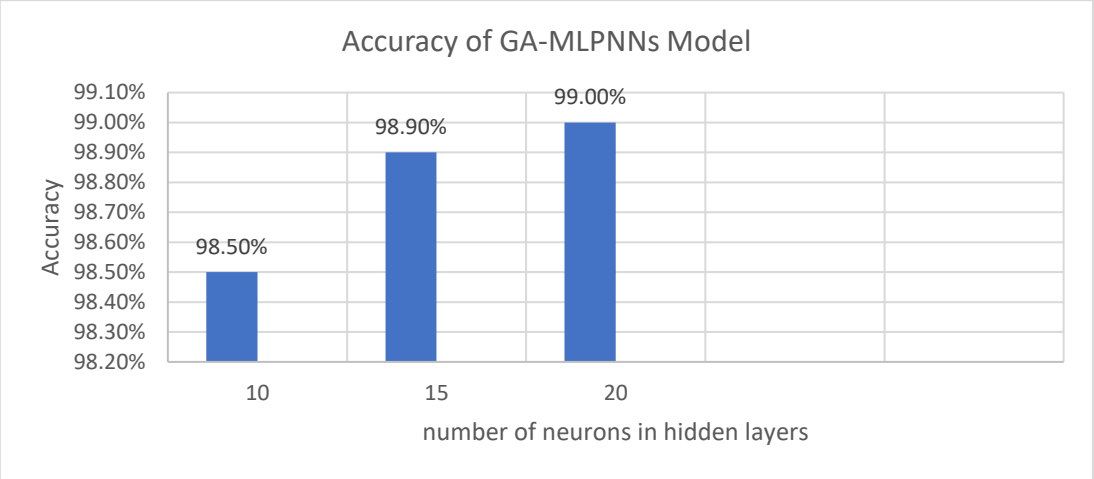

Figure 4-37:The accuracy of GA-MLPNNs model related to the number of neurons.

Figure (4.36) shows the accuracy of the GA-MLPNNs model when the number of neurons in the hidden layers equals 20 and the numbers of the iteration's values are 50,100,150, 200, 250. The accuracy was 98.8 % when the number of the iterations was 50, The accuracy was 98.9 % when the number of the iterations was 100 then the accuracy increased to 99% when the number of the iterations was 150, and it remained constant as the number of iterations increases. Figure (4.37) shows a comparison between the accuracy of the GA-MLPNNs model when the number of iterations equals 250 and the number of neurons in the hidden layer's values is 10,15, 20. The best accuracy of 99 % when the number of neurons =20, while the accuracy of 98.9%

when the number of neurons =15 and 98.5% when the number of neurons =10. Figure (4.38) show the ROC curve for GA-MLPNNs model when the number of iterations equal 250 and number of neurons in the hidden layer's is equal 20. the points of the line are in the upper-left corner which means that the performance is good.

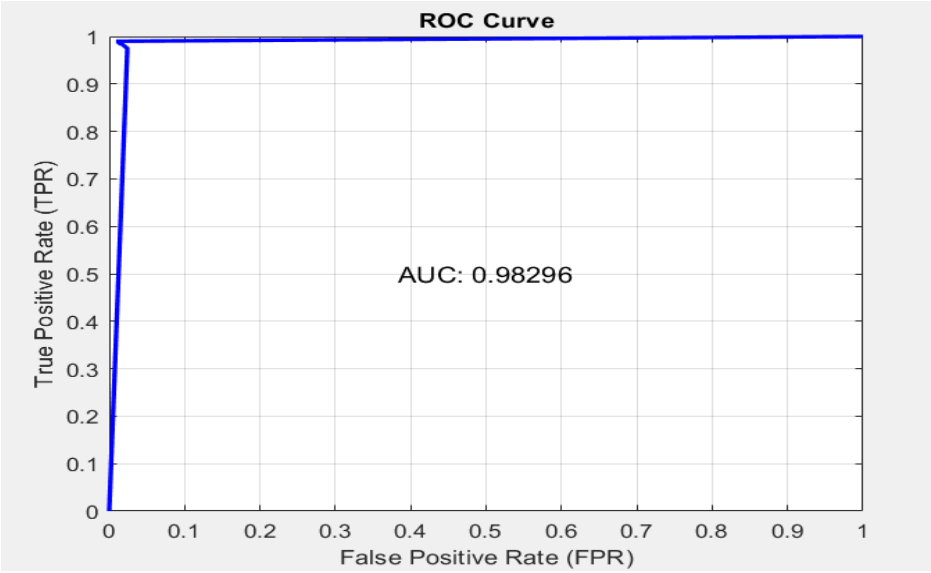

Figure *4-38*: The ROC curves for GA-MLPNNs model

4.5 Comparison and Discussion

We observe the results of experiments in applying classification models (DT, SVM, KNN, and MLPNN) to the global dataset and that the best classification model with 99% accuracy is the SVM model. In addition, MLPNN with an accuracy of 99.8% when the number of neurons in the hidden layer is equal to 50. While we observe the results of the experiments in applying the same models but on the local Palestinian dataset, the best model in classification with an accuracy rate of 96.4% is the DT model. In addition, MLPNN with an accuracy of 98.1% when the number of neurons in the hidden layer is equal to 30. The hybrid models (BBO-MLPNNs, PSO-MLPNNs, and GAs-MLPNNs) have been applied to both the global dataset and the Palestinian local dataset to classify and predict chronic kidney disease. The parameters were

changed and improved to obtain the best results so that they were the number of iterations (50,100,150,200,250) and the number of neurons in the hidden layer (10,15,20). Table (4.11) shows the accuracy results for these hybrid models with the optimized parameters as the number of iterations (50,100,150,200,250) and the number of neurons in the hidden layer equal to 20. Figure (4.39) show the chart of the summary results of the hybrid models on the global dataset and figure (4.40) show the chart of the summary results of the hybrid models on the local dataset.

Table 4-11:The summary results of the hybrid models on both global and local datasets

Dataset	**Algorithms**	**Parameters**		**Accuracy**
		Number of iterations	**Number of Hidden Neurons**	
Global dataset.	BBO-MLPNNs	50	20	99.3%
		100		99.4%
		150		99.4%
		200		99.4%
		250		99.4%
	PSO-MLPNNs	50	20	95.30%
		100		97.00%
		150		97.00%
		200		97.00%
		250		97.00%
	GA-MLPNNs	50	20	99.20%
		100		99.30%
		150		99.40%
		200		99.40%
		250		99.50%
Palestinian local dataset	BBO-MLPNNs	50	20	99.00%
		100		99.00%
		150		99.00%
		200		99.00%
		250		99.00%
	PSO-MLPNNs	50	20	95.30%
		100		96.40%
		150		96.60%
		200		96.60%
		250		96.70%
	GA-MLPNNs	50	20	98.80%
		100		98.90%
		150		99.00%
		200		99.00%
		250		99.00%

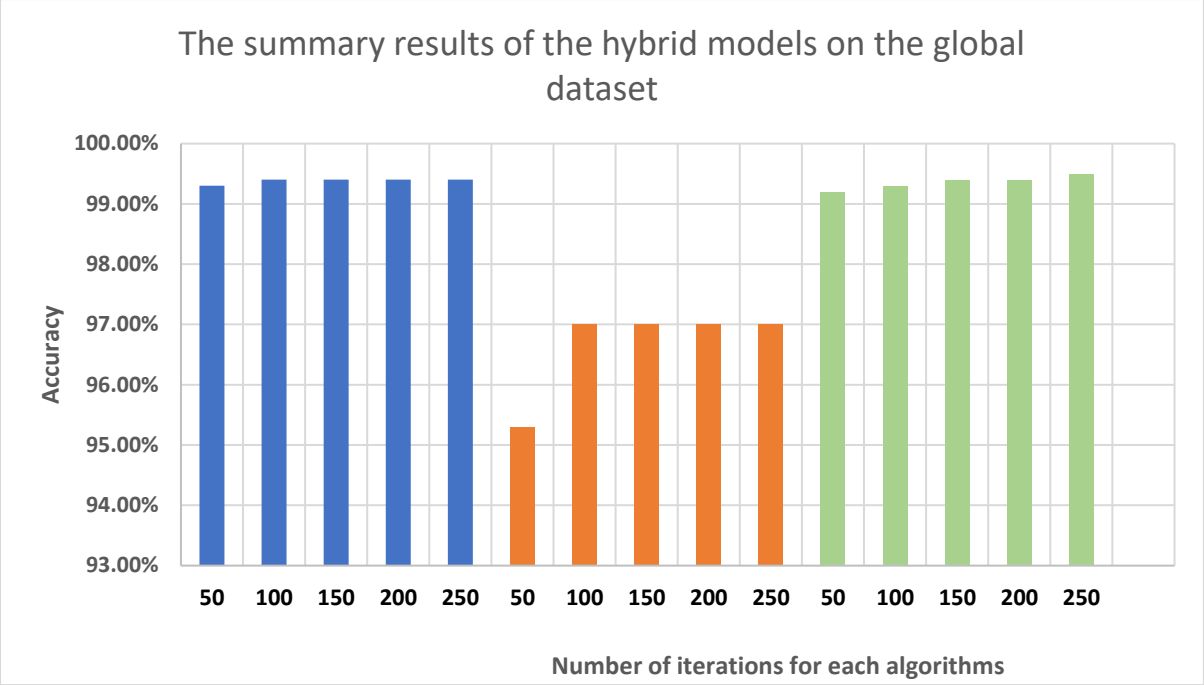

Figure 4-39: The summary results of the hybrid models on the global dataset

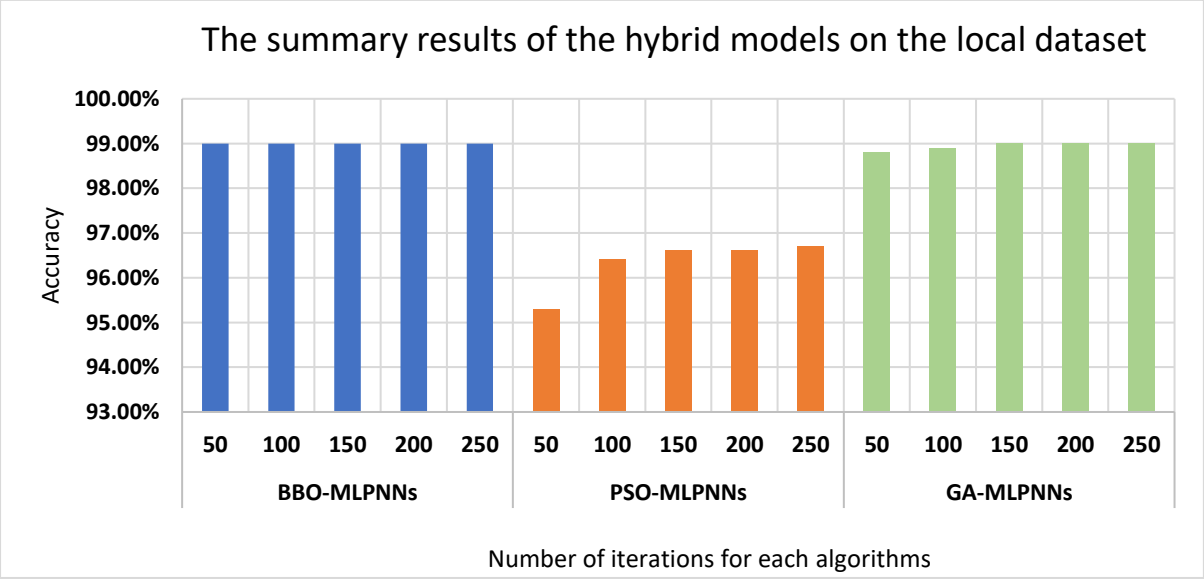

Figure 4-40: The summary results of the hybrid models on the local dataset

We observe the results of experiments in applying hybrid models to the global dataset. Both GAs-MLPNNs and BBO-MLPNNs were similar in performance in terms of accuracy with a rate of (99.5%, and 99.4%) respectively. while the PSO-MLPNNs model had a lower performance with an accuracy rate of 97% . While the results of the experiments in applying hybrid models to the local Palestinian dataset both the GAs-MLPNNs and BBO-MLPNNs were similar in performance in terms of accuracy at 99. %, while the PSO-MLPNNs model had a lower performance with an accuracy rate of 96.7%.

By comparing our results with the results of previous research in applying different ML models to the global dataset for CKD in terms of accuracy, we conclude that there is a difference in the results. For example, Jongbo et al [32] applied the global dataset to a set of ML models, including K Nearest Neighbors, Naïve Bayes and Decision Tree. The result was 100% for KNN. Also, Polat, H. et al [6] applied SVM and the result was 98.5%. While we applied the models to the global dataset, we obtained a result of 99% for the SVM model and 98.2% for the KNN model. The difference in results between research in applying different ML models to the global dataset for CKD is due to the method of preparing the data, the difference in the number of the feature selection of the dataset that used when applying the models and the difference in tools used in the application. For example, some researchers used Python and others used MATLAB. The following table (4.12) shows a summary of the results of other previous work in applying ML models to global dataset.

Table 4-12: summary of the results of other previous work in applying ML models to global dataset

Researchers	Year	Models	Accuracy
Poonia et al	**2022**	**Redundant feature removal (RFE)** **Chi-Square** **logistic regression**	**98.75 %.**
Jongbo et al	**2020**	**k Nearest Neighbors** **Naïve Bayes** **Decision Tree**	**100%**
Singh et al	**2022**	**novel deep learning model** **Support Vector Machine (SVM)** **K-Nearest Neighbor (KNN)**	**100%**
Ravindra et al	**2017**	**SVM- neural networks**	**93.75%**
Polat, H. et al	**2017**	**SVM**	**98.5%**
Tekale et al.	**2018**	**Decision Trees (DT)** **Random Forest (RF)** **Support Vector Machines (SVM),**	**85.02%** **93.08%**
Muntasir et al.	**2021**	**eight machine-learning models using the Python language**	**99.75% for Random Forest**
Al-Moman et al	**2022**	**ANNs** **SVMs** **k-Nearest Neighbors (KNN)**	**99.2%**

4.6 Limitation

We faced some challenges, especially with the local data used in the models. Since not all of the patient's information was recorded in many records by the medical staff, we had to use mathematical methods to complete these values. Also, there was not much agreement between the features found in the local and global datasets, so it was difficult to make a comparison of the results we obtained from applying the models to all the features found between the local and global datasets.

Chapter 5

5.1 Conclusion

The increase in diseases in society, especially chronic diseases among the elderly, makes it a burden on society. Errors in diagnosis or delays in diagnosing diseases often occur, which makes treatment costs high. Artificial intelligence in the medical field is very useful, as it helps in making many medical decisions, making early diagnoses, and starting the treatment phase faster and earlier. Medical data is collected from patients and used to train neural networks, apply machine learning techniques, and create advanced medical systems for classification and diagnosis. Chronic kidney disease (CKD) is a health condition that requires early detection and treatment to minimize harm. Patients do not show any noticeable symptoms in the early stages, which makes diagnosis take a long time and results in high treatment costs. Therefore, the presence of medical and technical development has helped in the possibility of making an early diagnosis of this disease in addition to classification using machine learning techniques and algorithms.

In this thesis, hybrid models were proposed, and several machine-learning techniques were used to classify, diagnose, and predict chronic kidney disease by applying these techniques and models to the global dataset and the local Palestinian dataset. The proposed hybrid models, BBO-MLPNNs, PSO-MLPNNs, and GAs-MLPNNs, are systems that combine algorithms with neural networks (NNs) to improve the performance of the neural network. We also applied classification algorithms, which are: decision tree (DT), support vector machine (SVM), and K-nearest neighbor (KNN). We observed from the experimental results in applying the hybrid models to the global dataset that both GAs-MLPNNs and BBO-MLPNNs are almost similar in performance, as the results for GAs-MLPNNs were (99.5%, 99.6%, 99.3%, 99.6%, and 99.6%) accuracy, sensitivity, specificity, precision, and F-score, respectively. While for BBO-MLPNNs, they were (99.4%, 99.6%, 99.4%, 99.3%, and 99.8%) accuracy, sensitivity,

specificity, precision, and F-score, respectively. It was also observed from the results of applying the hybrid models to the local dataset that both GAs-MLPNNs and BBO-MLPNNs are similar in performance, as the results for GAs-MLPNNs were (99%, 99%, 99.1%, 99.2%, and 99.1%) accuracy, sensitivity, specificity, precision, and F-score, respectively. While for BBO-MLPNNs, they were (99%, 99%, 99.1%, 99.2%, and 99.1%) accuracy, sensitivity, specificity, precision, and F-score, respectively. We note from the results of experiments in applying classification models (DT, SVM, KNN, and MLPNN) that the SVM model is the best in classification with an accuracy rate of 99%. In addition to the MLPNN with an accuracy of 99.8% for the global dataset, the DT model was the best in the classification with an accuracy of 96.4%. In addition, MLPNN has an accuracy of 98.1% for the Palestinian local dataset.

5.2 Future Work

In future work, other algorithms with other input parameters will be applied to chronic kidney disease datasets to improve classification, accuracy, and performance efficiency. In addition to applying other machine learning techniques to the Palestinian local dataset, collect disease data from various hospitals to obtain the largest possible amount of different data and use it in developing research and models. As future work that serves the health sector, we propose a clinical decision support system (CDSS) for chronic kidney disease (CKD). The system helps health care professionals provide the best medical services promptly, such as assessing the risk of disease progression, supporting the correct diagnosis of the disease, and providing an advanced treatment plan based on the stage of disease progression and suggesting lifestyle modifications, CDSS can alert healthcare professionals to drug management in terms of medication timing, drug alternatives, or any dosage-related discrepancies depending on the patient's health condition. Work on developing the model and implementing it as a user interface for the application in real life, especially in health centers, by medical profession.

References

GBD Chronic Kidney Disease Collaboration. Global, regional, and national burden of chronic kidney disease, 1990-2017: a systematic analysis for the Global Burden of Disease Study 2017. Lancet 395, 709–733 (2020).

Levin, A. et al. Global kidney health 2017 and beyond: a roadmap for closing gaps in care, research, and policy. Lancet 390, 1888–1917 (2017).

S. Chatterjee, S. Dzitac; S. Sen, N.C. Rohatinovici, N. Dey, A.S. Ashour, V.E. Balas. "Hybrid modified Cuckoo Search-Neural Network in chronic kidney disease classification"". 14th International Conference on Engineering of Modern Electric Systems (EMES), 2017, pp. 164 – 167.

Elhoseny, M., Shankar, K. & Uthayakumar, J. Intelligent Diagnostic Prediction and Classification System for Chronic Kidney Disease. Sci Rep 9, 9583 (2019).

https://www.cdc.gov/kidneydisease/pdf/CKD-Factsheet-H.pdf

Polat, Hüseyin & Danaei-mehr, Homay & Çetin, Aydın. (2017). Diagnosis of Chronic Kidney Disease Based on Support Vector Machine by Feature Selection Methods. Journal of Medical Systems

Balija, Ravindra & Sriraam, Natarajan & Geetha, M. (2017). Classification of non-chronic and chronic kidney disease using SVM neural networks. International Journal of Engineering & Technology.

Bird, L., & Walker, D. (2015). Treatment of chronic kidney disease. Companion Animal, 20(2), 104–111.

Chronic Kidney Disease Testing Among Primary Care Patients with Type 2 Diabetes Across 24 U.S. Health Care Organizations

Miller, D. D., & Brown, E. W. (2018). Artificial Intelligence in Medical Practice: The Question to the Answer? The American Journal of Medicine, 131(2), 129–133.

He, J., Baxter, S.L., Xu, J. et al. The practical implementation of artificial intelligence technologies in medicine. Nat Med 25, 30–36 (2019).

Pavel Hamet, Johanne Tremblay. Artificial intelligence in medicine.

Mirbabaie, M., Stieglitz, S., & Frick, N. R. J. (2021). Artificial intelligence in disease diagnostics: A critical review and classification on the current state of research guiding future direction. Health and Technology, 11(4), 693–731.

Loh, E. (2018). Medicine and the rise of the robots: a qualitative review of recent advances of artificial intelligence in health. BMJ Leader, leader–2018

Badillo, S., Banfai, B., Birzele, F., Davydov, I.I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B. and Zhang, J.D. (2020), An Introduction to Machine Learning. Clin. Pharmacol. Ther., 107: 871-885.

Mahesh, Batta. (2019). Machine Learning Algorithms -A Review.

Bashir, S., Qamar, U., Khan, F. H., & Naseem, L. (2016). HMV: A medical decision support framework using multi-layer classifiers for disease prediction. Journal of Computational Science, 13, 10–25

Ekanayake, I. U., & Herath, D. (2020). Chronic Kidney Disease Prediction Using Machine Learning Methods. 2020 Moratuwa Engineering Research Conference (MERCon)

Chao, C.-M., Yu, Y.-W., Cheng, B.-W., & Kuo, Y.-L. (2014). Construction the Model on the Breast Cancer Survival Analysis Use Support Vector Machine, Logistic Regression and Decision Tree. Journal of Medical Systems, 38(10).

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: applications, challenges and trends. Neurocomputing.

Moldagulova, A., & Sulaiman, R. B. (2017). Using KNN algorithm for classification of textual documents. 2017 8th International Conference on Information Technology (ICIT).

D.S. Huang, J.X. Du, A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks, IEEE Trans. Neural Networks 19 (Dec 2008) 2099–2115.

Imon, D. (2008). Biogeography-based optimization. IEEE transactions on evolutionary computation, 12(6), 702-713.

Marini, F., & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. Chemometrics and Intelligent Laboratory Systems, 149, 153-165.

Gen, M., & Lin, L. (2007). Genetic algorithms. Wiley Encyclopedia of Computer Science and Engineering, 1-15.

Dash, S., Shakyawar, S. K., Sharma, M., & Kaushik, S. (2019). Big data in healthcare: management, analysis and future prospects. Journal of Big Data, 6(1).

Alotaibi YK, Federico F. The impact of health information technology on patient safety. Saudi Med J. 2017

Chen TK, Knicely DH, Grams ME. Chronic Kidney Disease Diagnosis and Management: A Review. JAMA. 2019.

A. Sobrinho, A. C. M. D. S. Queiroz, L. Dias Da Silva, E. De Barros Costa, M. Eliete Pinheiro and A. Perkusich, "Computer-Aided Diagnosis of Chronic Kidney Disease in Developing Countries: A Comparative Analysis of Machine Learning Techniques," in IEEE Access, vol. 8, pp. 25407-25419, 2020.

Gulmez H. Detection of chronic disease in Primary Care Using Artificial Intelligence Techniques. In: Computational Intelligence and Soft Computing Applications in Healthcare Management Science. 2020.

Poonia, R.C.; Gupta, M.K.; Abunadi, I.; Albraikan, A.A.; Al-Wesabi, F.N.; Hamza, M.A.; B, T. Intelligent Diagnostic Prediction and Classification Models for Detection of Kidney Disease. Healthcare 2022, 10, 371.

Jongbo, O. A., Adetunmbi, A. O., Ogunrinde, R. B., & Badeji-Ajisafe, B. (2020). Development of an Ensemble Approach to Chronic Kidney Disease Diagnosis. Scientific African, e00456.

Singh, V.; Asari, V.K.; Rajasekaran, R. A Deep Neural Network for Early Detection and Prediction of Chronic Kidney Disease. Diagnostics 2022, 12, 116.

Siddheshwar Tekale, Pranjal Shingavi, Sukanya Wandhekar, Ankit Chatorikar. Prediction of Chronic Kidney Disease Using Machine Learning Algorithm. International Journal of Advanced Research in Computer and Communication Engineering. 2018 Oct;7(10).

Muntasir Nishat M, Faisal F, Rahman Dip R, Nasrullah SM, Ahsan R, Shikder F, Ar-Raihan Asif MA-, Hoque MA. A Comprehensive Analysis on Detecting Chronic Kidney Disease by Employing Machine Learning Algorithms. EAI Endorsed Trans Perv Health Tech [Internet]. 2021 Aug.

Al-Momani R, Al-Mustafa G, Zeidan R, Alquran H, Mustafa WA, Alkhayyat A. Chronic Kidney Disease Detection Using Machine Learning Technique. In: 2022 5th International Conference on Engineering Technology and its Applications (IICETA). 2022. p. 153–8.

Hore, S., Chatterjee, S., Shaw, R. K., Dey, N., & Virmani, J. (2017). Detection of Chronic Kidney Disease: A NN-GA-Based Approach. Advances in Intelligent Systems and Computing, 109–115.

Yadav D. C, Pal S. Performance based Evaluation of Algorithms on Chronic Kidney Disease using Hybrid Ensemble Model in Machine Learning. Biomed Pharmacol J 2021;14(3).

Arvind Kumar, Nishant Sinha, Arpit Bhardwaj & Shivani Goel (2022) Clinical risk assessment of chronic kidney disease patients using genetic programming, Computer Methods in Biomechanics and Biomedical Engineering, 25:8, 887-895.

M, M., & Balakrishnan, S. (2020). Feature Selection Using Improved Teaching Learning Based Algorithm on Chronic Kidney Disease Dataset. Procedia Computer Science, 171, 1660–1669.

Gunarathne, W. H. S.., Perera, K. D.., & Kahandawaarachchi, K. A. D. C.. (2017). Performance Evaluation on Machine Learning Classification Techniques for Disease Classification and Forecasting through Data Analytics for Chronic Kidney Disease (CKD). 2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE).

Kim, D.-H., & Ye, S.-Y. (2021). Classification of Chronic Kidney Disease in Sonography Using the GLCM and Artificial Neural Network. Diagnostics, 11(5), 864.

https://archive.ics.uci.edu/dataset/336/chronic+kidney+disease51

https://site.moh.ps/

Liu, H., Sun, J., Liu, L., & Zhang, H. (2009). Feature selection with dynamic mutual information. Pattern Recognition, 42(7), 1330–1339.

Pascoal, C., Oliveira, M. R., Pacheco, A., & Valadas, R. (2017). Theoretical evaluation of feature selection methods based on mutual information. Neurocomputing, 226, 168–181.

Singh, D., & Singh, B. (2019). Investigating the impact of data normalization on classification performance. Applied Soft Computing, 105524

Aljuaid, T., & Sasi, S. (2016). Proper imputation techniques for missing values in data sets. 2016 International Conference on Data Science and Engineering (ICDSE).

Akinsola, J E T. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. International Journal of Computer Trends and Technology (IJCTT). 48. 128 - 138.

Mienye, I. D., Sun, Y., & Wang, Z. (2019). Prediction performance of improved decision tree-based algorithms: a review. Procedia Manufacturing, 35, 698–703.

Utgoff, P. E. (1989). Machine Learning, 4(2), 161–186.

Awad, M., & Khanna, R. (2015). Support Vector Machines for Classification. Efficient Learning Machines, 39–66.

Wang, Z., & Xue, X. (2014). Multi-Class Support Vector Machine. Support Vector Machines Applications, 23–48.

Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. Annals of Translational Medicine, 4(11), 218–218.

Moldagulova, A., & Sulaiman, R. B. (2017). Using KNN algorithm for classification of textual documents. 2017 8th International Conference on Information Technology (ICIT).

Wang, J., Lu, S., Wang, S.-H., & Zhang, Y.-D. (2021). A review on extreme learning machine. Multimedia Tools and Applications

Gardner, M.., & Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric Environment, 32(14-15), 2627–2636.

Chittora, P., Chaurasia, S., Chakrabarti, P., Kumawat, G., Chakrabarti, T., Leonowicz, Z., … Bolshev, V. (2021). Prediction of Chronic Kidney Disease - A Machine Learning Perspective. IEEE Access, 9, 17312–17334

Simon, D. (2008). "Biogeography-based optimization" (PDF). IEEE Transactions on Evolutionary Computation. 12 (6): 702–713.

Zheng, Y.-J., Ling, H.-F., & Xue, J.-Y. (2014). Ecogeography-based optimization: Enhancing biogeography-based optimization with ecogeographic barriers and differentiations. Computers & Operations Research, 50, 115–127

Zhao, X., Ji, Y., & Hao, J. (2020). A Novel Biogeography-Based Optimization Algorithm with Momentum Migration and Taxonomic Mutation. Advances in Swarm Intelligence: 11th International Conference, ICSI 2020, Belgrade, Serbia, July 14–20, 2020, Proceedings, 12145, 83–93.

Ma, H., Simon, D., Siarry, P., Yang, Z., & Fei, M. (2017). Biogeography-Based Optimization: A 10-Year Review. IEEE Transactions on Emerging Topics in Computational Intelligence, 1(5), 391–407.

Bonyadi, M. R., & Michalewicz, Z. (2017). Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. Evolutionary Computation, 25(1), 1–54.

Bratton, D., & Kennedy, J. (2007). Defining a Standard for Particle Swarm Optimization. 2007 IEEE Swarm Intelligence Symposium

Eberhart, & Yuhui Shi. (n.d.). Particle swarm optimization: developments, applications and resources. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546).

Zhang, Y., Wang, S., & Ji, G. (2015). A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. Mathematical Problems in Engineering, 2015, 1–38.

Lambora, A., Gupta, K., & Chopra, K. (2019). Genetic Algorithm- A Literature Review. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon).

Hamdia, K. M., Zhuang, X., & Rabczuk, T. (2020). An efficient optimization approach for designing machine learning models based on genetic algorithm. Neural Computing and Applications.

Wang, L. (2005). A hybrid genetic algorithm–neural network strategy for simulation optimization. Applied Mathematics and Computation, 170(2), 1329–1343.

Pal, K., & Patel, B. V. (2020). Data Classification with k-fold Cross Validation and Holdout Accuracy Estimation Methods with 5 Different Machine Learning Techniques. 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC).

T. -T. Wong and P. -Y. Yeh, "Reliable Accuracy Estimates from k-Fold Cross Validation," in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 8, pp. 1586-1594, 1 Aug. 2020

K. Pal and B. V. Patel, "Data Classification with k-fold Cross Validation and Holdout Accuracy Estimation Methods with 5 Different Machine Learning Techniques," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 83-*87*

Akhtar, S., Shahzad, S., Zaheer, A., Ullah, H. S., Kilic, H., Gono, R., Jasiński, M., & Leonowicz, Z. (2023). Short-Term Load Forecasting Models: A Review of Challenges, Progress, and the Road Ahead. Energies, 16(10), Article 4060.

Hichri, A., Hajji, M., Mansouri, M., Harkat, M.-F., Kouadri, A., Nounou, H., & Nounou, M. (2020). Fault detection and diagnosis in grid-connected photovoltaic systems. 2020 17th International Multi-Conference on Systems, Signals & Devices (SSD).

Sharma*, S., & Parmar, M. (2020). Heart Diseases Prediction using Deep Learning Neural Network Model. In International Journal of Innovative Technology and Exploring Engineering (Vol. 9, Issue 3, pp. 2244–2248).

A. Elsalamony, H. (2014). Bank Direct Marketing Analysis of Data Mining Techniques. In International Journal of Computer Applications (Vol. 85, Issue 7, pp. 12–22). Foundation of Computer Science.

Makaba, T., & Dogo, E. (2019). A Comparison of Strategies for Missing Values in Data on Machine Learning Classification Algorithms. 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC).

الملخص

يعد مرض الكلى المزمن (CKD) عامل خطر للإصابة بأمراض القلب والأوعية الدموية وله تأثير اقتصادي كبير على أنظمة الرعاية الصحية. الكشف المبكر عن مرض الكلى المزمن يمكن أن ينقذ حياة الشخص من نوبة قلبية. ظهر الذكاء الاصطناعي (AI) كأداة جديدة تساعد في الكشف المبكر عن الأمراض والتنبؤ بحدوثها. وللذكاء الاصطناعي الأثر الجيد في إنقاذ الأرواح وتقديم خطة علاجية للمرض وإجراء المزيد من الأبحاث التطويرية.

يمكن استخدام تقنيات الذكاء الاصطناعي المختلفة لتصنيف أمراض الكلى المزمنة والتنبؤ بها من خلال تطبيقها على البيانات الطبية. تقدم هذه الأطروحة نماذج هجينة تجمع بين الخوارزميات التطورية والشبكات العصبية وبعض تقنيات التعلم الآلي لتصنيف أمراض الكلى المزمنة. في هذه الأطروحة، تم استخدام مجموعة البيانات العالمية والمحلية أيضا. في المرحلة الأولى، تم تطبيق العديد من نماذج التعلم الآلي المختلفة على مجموعات البيانات، بما في ذلك شجرة القرار (DT)، وآلة ناقل الدعم (SVM)، و(K-Nearest Neighbor (KNN، والشبكات العصبية متعددة الطبقات (MLPNNs). في المرحلة الثانية، تم تدريب عدة نماذج هجينة من الخوارزميات التطورية، بما في ذلك التحسين القائم على الجغرافيا الحيوية (BBO)، وتحسين سرب الجسيمات (PSO)، والخوارزميات الجينية (GAs) على الشبكات العصبية متعددة الطبقات (MLPNNs) بهدف الحصول على أفضل النتائج في تصنيف CKD.

تم تطبيق نماذج DT، وSVM، وKNN، وMLPNNs على مجموعة البيانات العالمية لتصنيف أمراض الكلى المزمنة، وكشفت عن نتائج دقة تبلغ: 97%، و99.5%، و98.2%، و99.8% على التوالي. علاوة على ذلك، أظهر نموذجي MLPNN وSVM أعلى دقة وأفضل النماذج في التصنيف بمعدلات دقة متقاربة. كما تم تطبيق هذه النماذج على مجموعة البيانات المحلية الفلسطينية لتصنيف أمراض الكلى المزمنة، وكانت نتائج الدقة التي تم الحصول عليها هي: 96.4%، 96.2%، 93.6%، و98.1% على التوالي، وبنفس الترتيب المذكور سابقاً. وقد أظهر نموذج MLPNNs أعلى دقة وكان النموذج الأفضل في التصنيف يليه نموذج DT.

أظهرت النتائج التجريبية في تطبيق النماذج الهجينة على مجموعة البيانات العالمية أن كلا من GAs-MLPNNs وBBO-MLPNNs كانت متشابهة تقريبًا في الأداء، وكانت نتائج GAs-MLPNNs هي: 99.5%، 99.6%، 99.3%، 99.6%، و99.6% دقة وحساسة ومحددة ودقيقة ودرجة F على التوالي. بالإضافة إلى ذلك، كانت النتائج في تطبيق النماذج الهجينة على مجموعة البيانات المحلية لكل من GAs-MLPNNs وBBO-MLPNNs متشابهة أيضًا في الأداء، حيث كانت دقيقة بنسبة 99% و99% و99.1% و99.2% و99.1%. وحساسة ومحددة ودقيقة وF على التوالي.