MINISTRY OF EDUCATION
IMAM ABDULRAHMAN BIN
FAISAL UNIVERSITY
COLLEGE OF COMPUTER SCIENCE
& INFORMATION TECHNOLOGY

وزارة التعليم
جامعة الإمام
عبدالرحمن بن فيصل
كلية علوم الحاسب
وتقنية المعلومات

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

CS 522: Selected Topics in Computer Science

Title: Predicting Chronic Kidney Disease by using several machine learning techniques.

CS522 Project topic Selection process

Section 1 and Team 4		ID
Project Group		
No.	Name	
1	Rawan Fuad Alessa	2160003634
2	Tasneem Hamdy Dosoqi	2160007430
3	Rahaf Saleh Alzahrani	2160006662
4	Muneera Abdulrahman Alshaalan	2160005828
5	Khawlah Fahad Bajbaa	2160003191

Predicting Chronic Kidney Disease using SVM and KNN Classifiers.

Khawlah F. Bajbaa[1], Muneera A. Alshaalan[2], Rahaf S. Alzahrani[3], Rawan F. Alessa[4], Tasneem H. Dosoqi[5], Nida Aslam[6].
College of Computer Science and Information Technology, Imam Abdulrahman bin Faisal University, Kingdom of Saudi Arabia

[1]2160003191@iau.edu.sa, [2]2160005828@iau.edu.sa, [3]2160006662@iau.edu.sa, [4]2160003634@iau.edu.sa, [5]2160007430@iau.edu.sa, [6]naslam@iau.edu.sa .

Abstract

Chronic Kidney Disease (CKD) is a global health issue that leads to an increase in mortality and morbidity rates, also it produces other kinds of diseases. Early detection of chronic kidney disease helps to reduce the number of patients and the cost that is needed for treatment. This study aims to preemptively predict the presence of chronic kidney disease accurately through using two machine learning techniques which are Support Vector Machines (SVM) and K-nearest neighbor (KNN). The CKD data was obtained from the University of California Irvine (UCI) machine learning repository, which has a huge number of missing values. The data set contained 400 records and 25 attributes included the class label which has two categorical values, namely, CKD and not CKD. The experimental results showed that SVM performed better than KNN with accuracy up to 99.17%, while the KNN has achieved accuracy up to 94.16%.

Keywords: Support Vector Machine, K-nearest neighbors, Machine learning, Chronic Kidney Disease.

1. Introduction

Kidneys are essential organs in the human body where it has many functions to do like, clean the blood from toxins that accumulate in it because of all biological processes that occur in the cells of the body. For example, the digestive system produces many toxins as a result of the digestion process Where the kidneys get rid of these toxins and remove them from the body. Sometimes, Kidneys cannot fully function, and it may fail to remove toxins and extra water from the blood ending up having them in the body. When this fail lasts for two to three months, it is said to be chronic kidney disease (CKD).

Chronic kidney disease and the increased number of its incidence has become a worldwide health issue around the world [1]. One of the characteristics of CKD that it slowly decreases the renal function, and, in the end, it will lead to a significant loss of renal function. At the early stages of CKD, usually the first three months, the symptoms of this disease do not appear clearly. As a result, it will not be detected and diagnosed until significant kidney losses happened. For that reason, it is crucial to predict and diagnose CKD when it is in the early stages [2]. That is to say, if CKD is detected and treated it its early stages, it will prevent from going to later stages. In order to detect CKD, it is essential to have adequate knowledge about the symptoms [1]. In the medical field, Machine Learning has been used for many purposes like diagnosing heart diseases [3], [4], cancer [5], symptoms of the disease [6], and predict various diseases. Machine Learning technology focuses on using different algorithms that can learn for a set of data and make predictions based on what it learns [1]. The prediction of Machine Learning for diseases is usually accurate, which is a promising way of predicting CKD [2].

In this paper, we will develop a machine learning solution to build a classification model using two classifiers: Support vector machines (SVM) and K-Nearest Neighbor algorithm (KNN) classifiers in order to detect CKD and non-CKD patients. By using data from the UCI Machine Learning Repository [7], which consists of 25 attributes with 400 patients. This data suffers from huge missing values in each attribute and also suffer from noisy data. The algorithms used will be compared against each other in an attempt to come up with the best algorithm that yields the highest accuracy for the given dataset. The algorithm that leads to the highest accuracy will be selected as the best model that can predict CKD patients.

The remaining parts of this study organized as follows. Section 2 contains the review of related literatures. Section 3 contains the description of the proposed machine learning techniques. Section 4 contains the empirical study that include the description of dataset, experimental setup, and optimization strategy. Section 5 contains the result and discussion and section 6 contains conclusion and recommendation.

2. Review of Related Literatures

Chronic Kidney Disease (CKD) is a global health issue that leads to an increase in mortality and morbidity rates, also it produces other kinds of diseases. Early detection of chronic kidney disease helps to reduce the number of patients and the cost that is needed for treatment. This study aims to preemptively predict the presence of chronic kidney disease accurately through using two machine learning techniques which are Support Vector Machines (SVM) and K-nearest neighbor (KNN). The CKD data was obtained from the University of California Irvine (UCI) machine learning repository, which has a huge number of missing values. The data set contained 400 records and 25 attributes included the class label which has two categorical values, namely, CKD and not CKD.

In this section we discuss a set of research papers that used various methods and techniques of machine learning, to help in the process of predicting Chronic Kidney Disease (CKD). All researches have provided comparisons between the techniques using some performance measures which are:

Charleonnan et al. [8] discussed several techniques that are used in the process of predicting CKD. These classifiers are Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Logistic Regression (LR). They used UCI database [7], and they converted all nominal attributes into binary. Moreover, greedy hill-climbing was used in the next step which is feature selection. The dataset was divided into 70% for training data and 30% for testing. Different techniques were compared using the same model with the same dataset and same data partition. These classifiers were tested, and each one gave its value as shown in table1:

Table 1: The performance measures for each classifier.

Classifier	Accuracy	Sensitivity	Specificity
SVM	98.3%	99%	98%
KNN	98.1%	96%	99%
LR	96.6%	94%	98%
DT	94.8%	93%	96%

SVM achieved the highest accuracy, so it can be concluded as a suitable classifier for predicting CKD. MATLAB and WEKA were used as tools in this study.

In the research carried by W.H.S.D et al. [1] they have built classification models with diverse classification algorithms which are: Multiclass Decision Jungle, Multiclass Decision Forest Multiclass Neural Network and Multiclass Logistic Regression. The main focus of their work was to predict the CKD and non-CKD patients using CKD dataset taken from UCI machine learning repository [7]. The dataset consists of 400 data records and comprises of 25 features. According to result achieved using Microsoft Azure Machine Learning Studio tool, Decision Forest achieved the highest accuracy of 99.1%, Logistic Regression got the accuracy of 95.0%, Decision Jungle obtained 96.6% accuracy, and Neural Network executes and got the accuracy with 97.5%. Therefore, Decision Forest achieves highest accuracy. However, there are some limitations in their study which are, small size of data, and huge number of missing value attributes.

Polat et al. [9] in this study SVM classification algorithm was applied to determine CKD. Moreover, the dataset was used from UCI machine learning [7]. Filter and wrapper feature selection each of them has two methods which are, Greedy stepwise search and best first were processed to evaluate the accuracy and feature selection methods. Both methods reduced the number of features. First, wrapper approach ClassifierSubsetEval with Greedy stepwise search engine, result in the reduced dataset with 7 features and achieved an accuracy of 98%. Second, wrapper approach, WrapperSubsetEval with Best First search engine, then the result was reduced dataset size to 11 features with the accuracy of 98.25%. Third, filter approach CfsSubsetEval with Greedy stepwise search engine, then the number of features was 16 and accomplished an accuracy of 98.25%. fourth, filter approach FilterSubsetEval with Best First search engine, then the result led to 13 features with the accuracy of 98.5%. Finally, SVM without using feature selection achieved an accuracy of 97.75%. Eventually, FilterSubsetEval with Best First search engine obtained the highest accuracy.

In the previous study in 2017 made by Misir et al. [10] used Correlation-based feature subset selection (CFS) and Levenberg–Marquardt (LM) algorithms for feature evaluation. The authors used only eight factors instead of the whole 25 factors in the Chronic Kidney Disease UCI dataset to predict two binary classifiers. The performance in this system was evaluated based on correct classification accuracy (CCR), sensitivity, accuracy and specificity. The first classifier was incremental back propagation learning networks classifier (IBPLN). Which showed slightly less performance than the second classifier Levenberg–Marquardt (LM) in terms of CCR and specificity. Both algorithms used showed a very high-performance justifying the validity of the reduced set of features. In case of ignoring missing values in the dataset. It may take into consideration in this study the reduction of the features may decrease the accuracy of the decision making.

This paper from the author Aljaaf et al. [11] explores the relationship between data parameters with the target attribute. Initially, the UCI database [7] contained 24 features, and after feature selection process according to its high correlation with the target class, 7 features were selected, and eliminate the redundant features. Several machine learning techniques have been used to perform the prediction process, such as Recursive Partitioning and Regression Trees (RPART), Support vector machines (SVM), Logistic Regression (LR) and Multilayer Perceptron (MLP). The best techniques that yielded higher results were MLP and LR, with an accuracy of 98.1%, sensitivity 98.97% , with a 2% error rate. The only limitation of the study is not considering the possibility that some drugs will affect the predicted outcomes.

In the previous study in 2018 made by Alassaf et al. [12] was about the precautionary diagnosis of Chronic Kidney Disease (CKD) to decrease the patients' need for treatment through precautionary diagnosing the CKD by using some machine learning techniques. The data was used in this study obtained from King Fahad University Hospital (KFUH) that contains 244 records with two class labels. 118 records were classified as CKD while the other 126 records were classified as non-CKD. This study was performed using Weka and Python machine learning library: scikit-learn. Weka was applied for data preprocessing and calculating the correlation of the features. However, the python library was applied for feature selection and for testing and training processes of the four selected classifiers: SVM, K-NN, ANN, and Naïve Bayes. Each classifier was examined by using the performance measures, namely: accuracy, f-measure, recall, and precision as shown in table 2.

Table 2: classifiers performance measures [6]

Classifier	Accuracy	F-measure	Recall	precision
SVM	98%	98.16%	100%	96.4%
KNN	93.9%	94.57%	96.3%	92.9%
ANN	98%	98.16%	100%	96.4%
Naïve Bayes	98%	98.16%,	100%	96.4%

Based on these values it can be deduced that the SVM, ANN, and Naïve Bayes are more suitable than K-NN to be used in the precautionary diagnosis of the CKD.

Radi et al. [13] compared four classification algorithms: Probabilistic Neural Networks (PNN), Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Radial Basis Function (RBF) according to their classification accuracy and execution time. They used DTREG Predictive Modeling System which is a software that read Comma Separated Value (CSV) data files for analysis. They used a dataset of 400 CKD and non-CKD patients with 25 variables that consist of numerical and categorical data and replaced the missing values by medians. The results for the analysis showed that the PNN algorithm yields better results in terms of classification accuracy and prediction for determining CKD compared to other algorithms. On the contrary, MLP requires 3 seconds whereas PNN requires 12 seconds in terms of execution time. The execution time does not affect the accuracy of the algorithms, but it was used by the authors as a comparison element.

Hayashi et al. [14] have used Recursive-Rule extraction (Re-RX) algorithm with J48graft in order to obtain classification rules, which was used to determine the upper limit of Hb levels in predialysis chronic kidney disease patients during anemia treatment. They used a dataset of 400 samples from UCI Machine Learning Repository and used only 243 samples for patients with predialysis and non-predialysis CKD in this study. The results for the analysis showed that the average classification accuracy achieved was 95.18% using three extracted rules from two attributes. They performed k-fold cross-validation to ensure the accuracy of the study. Authors recommend using classification rules approach for predicting the upper limit of Hb levels in predialysis chronic kidney disease patients during anemia treatment since it is inexpensive.

Similarly, another study made by authors [2] proposed a machine learning model for diagnosing Chronic Kidney Disease (CKD) using UCI dataset. The dataset contains large number of missing values and KNN imputation were applied in order to fill these missing values. The proposed model was established by using two machine learning algorithms which are logistic regression (LR) and random forest (FR). The accuracy, specificity, sensitivity, and f-score were applied to examine the performance of the machine learning model as shown in table 3.

Table 3: Proposed model's performance measures [10]

Accuracy	Specificity	Sensitivity	F- score
99.83%	99.80%	99.84%	99.86%

Due to the limitations of conditions and the availability of data samples the performance of generalization might be limited, and the model cannot diagnose severity of CKD. Consequently, the authors decided to collect a large number of complex data in order to improve the generalization performance.

In a recent study in 2020 made by Sobrinho et al.[15], six different algorithms were used on the Chronic Kidney Disease UCI dataset with a different number of features in each algorithm. The best technique with the highest accuracy = 95.00% was J48 decision tree. For the second-best technique, the random forest was used, showed 93.33% accuracy. In addition, more techniques showed 88.33% accuracy for naive Bayes, 76.66% for support vector machine, 75.00% for multilayer perceptron, and 71.67% accuracy for the k-nearest neighbor. According to the authors, they considered the small number of records in the dataset as a limitation. But to overcome this problem, the k-fold cross-validation method was used to improve the accuracy of the study in each algorithm.

In this paper, we will develop a machine learning solution to build a classification model using two classifiers: Support vector machines (SVM) and K-Nearest Neighbor algorithm (KNN) classifiers in order to detect CKD and non-CKD patients. By using data from the UCI Machine Learning Repository [7], which consists of 25 attributes with 400 patients. This data suffers from huge missing values in each attribute and also suffer from noisy data. The algorithms used will be compared against each other in an attempt to come up with the best algorithm that yields the highest accuracy for the given dataset. The algorithm that leads to the highest accuracy will be selected as the best model that can predict CKD patients. Without a doubt, this will help doctors to have fast, accurate predictions about CKD and patients to prevent the disease progression and provide treatment as soon as possible with no time wasted.

3. Description of the Proposed Techniques

3.1. Preprocessing

Pre-processing is the most important process because a lot of real-life data are noisy, incomplete and inconsistent. In this phase, we applied different python libraries to impute the missing values. First, we use “Chronic Kidney Disease” dataset. Then, we specified the percentage of missing values in each column to decide which data imputation method will be used. After that, we imputed the numeric data with the mean value for each column using sklearn. impute library and we imputed the categorical data with the most-frequent value in each column. Finally, we converted the categorical data into numeric data using sklearn. preprocessing library as the SVM and KNN classifiers used with numeric data.

3.2. Features selection

Some features in the dataset were irrelative so we used a feature selection supervised learning algorithm which is Random forests. It tends to be utilized both for classification and regression. It is said that the more trees it has, the more powerful a forest is. Random forests make decision trees on arbitrarily chosen data samples, get an optimal set of features in each iteration and choose the best arrangement by Mean method of voting. It likewise gives a quite decent marker of the feature importance.

We applied random forest technique with a reclusively elimination method that eliminated the features based on its importance at each iteration and to come up with the best optimal features which were used as the final features for the model.

After applying the technique for many iterations, we got the best 8 optimal features sets which are:

18 features: ['htn', 'dm', 'appet', 'pe', 'age', 'bp', 'sg', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'pot','hemo', 'pcv','wc','rc'].

15 features: ['htn', 'dm', 'age', 'bp', 'sg', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'hemo', 'pcv', 'wc','rc'].

14 features-A: ['htn', 'dm', 'age', 'sg', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'hemo', 'pcv', 'wc', 'rc'].

14 features-B: ['htn', 'dm', 'age', 'bp', 'sg', 'al', 'bgr', 'bu', 'sc', 'sod', 'pot', 'hemo', 'pcv','rc'].

11 features: ['sg', 'al', 'rbc', 'bgr', 'bu', 'sc', 'sod', 'hemo', 'pcv','rc' ,'htn'].

10 features: ['htn', 'dm', 'sg', 'al', 'bgr', 'sc', 'sod', 'hemo', 'pcv', 'rc'].

8 features: ['htn', 'sg', 'al', 'bgr', 'sc', 'hemo', 'pcv', 'rc'].

7 features: ['sg', 'al', 'bgr', 'sc', 'hemo', 'pcv', 'rc'].

Finally, the train_test_split filter was used to split the dataset into training and testing sets. Where the test set occupies 30% of the final dataset and the rest 70% is for the training set.

3.3. Classification

3.3.1. SVM

Support Vector Machine (SVM) is a supervised classification method [16] and is a linear model used for solving regression and classification problems and usually used in classification problems. Also, it can solve both linear and non-linear problems. the SVM is an algorithm that creates a line or hyperplane to separate the data into classes [17]. SVM aims to maximize the margin between the hyperplane and the support vector points [18] look at Figure 1.

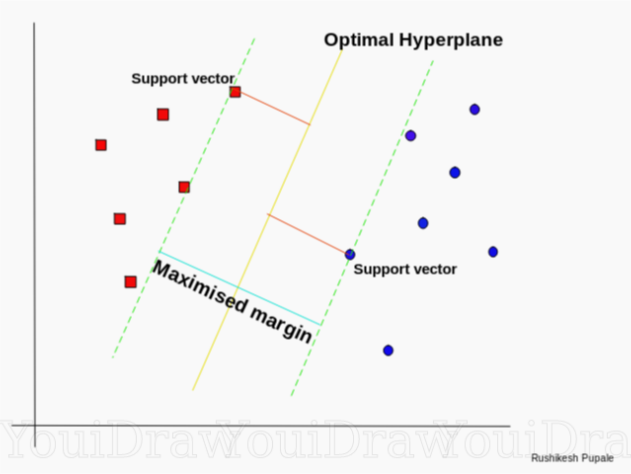

Figure 1:SVM algorithm [17]

There are two cases of SVM: SVM in linear separable clearly, infinite lines exist to separate green and red dots [19] as shown in Figure 2.

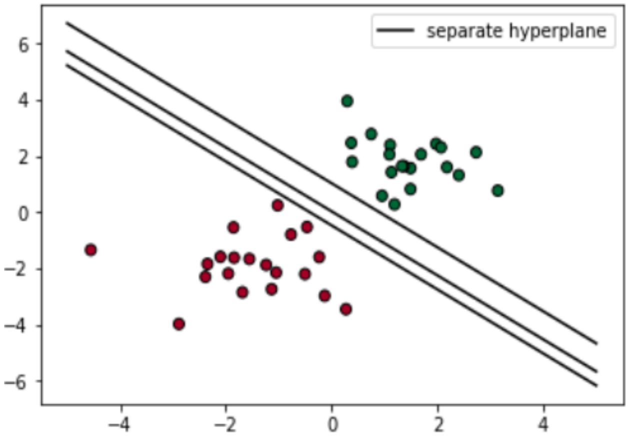

Figure 2: SVM in linear separable [19]

The second case is SVM in non-linear separable, let's clarify with the same previous example. If we add one red dot in the green cluster the data becomes linear non-separable as shown below Figure 3. The solutions for that can be done by applying two concepts: Soft Margin and Kernel Tricks. Soft Margin is trying to detect a line to separate but tolerate one or more than one misclassified dot and considering the degree of tolerance. Therefore, the training dataset can be separated linearly by using the kernel method that is expressed in Equation (1).

$$(1) \quad \varphi(x) \cdot \varphi(y) = (x, x) \cdot (y, y^2\ ) = xy + x^2 y^2$$
$$K(x, y) = xy + x^2\ y^2$$

Kernel Tricks is trying to detect a non-linear option by applying some transformations and creating new features [19].

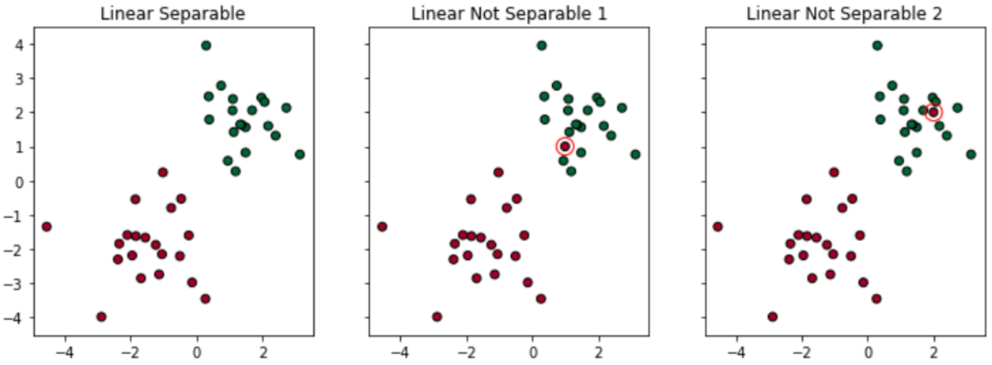

Figure 3: SVM in non-linear separable [19]

Also, There are different types of kernel method that it used in SVM Table 4 shows some of these methods.

Table 4: Kernel types.

Type	Formula
Linear Kernel	$\mathrm{K}(\mathrm{x}, \mathrm{x_i}) = \sum (* \ \mathrm{x_i})$
Polynomial Kernel	$\mathrm{K}(\mathrm{x}, \mathrm{x_i}) = 1 + \sum (\mathrm{x} * \ \mathrm{x_i})^{\mathrm{d}}$
Radial based kernel	$\mathrm{K}(\mathrm{x}, \mathrm{x_i}) = e^{(-\gamma * \lvert \mathrm{x} - \mathrm{x_i} \rvert^2)}$
Sigmoid kernel	$\mathrm{K}(\mathrm{x}, \mathrm{x_i}) = \tan^{-1}(\gamma * \sum (\mathrm{x} * \ \mathrm{x_i}) + \mathrm{c})$
Radial based function	$f(x) = \sum_{i}^{N} a_i y_i \ e^{\left(\frac{-\lvert\lvert x - x_i \rvert\rvert^2}{2s^2}\right) + b}$

In addition, the parameter C controls the trade-off between the smooth decision boundary and classifying training points correctly. and it can have two cases:

- If C is large, then the margin of hyperplane is small and there is a chance of overfitting.
- If C is small, then the margin of hyperplane is large there is a chance of underfitting.

A slack variable is represented by ξ and this definition is referred to determine the point (new record) is belong to which class if the value is 0 that means the point was classified correctly if it is in range of [0-1] that means the point was classified correctly but by less of margin compare with SVM wanted. The last possible value is one or more than 1 the point was classified incorrectly.

3.3.2. KNN

K-nearest neighbors (KNN) is a supervised machine learning algorithm that is widely used for classification like pattern recognition, to classify unknown examples based on how its neighbors are classified [8]. KNN is considered as a lazy learner algorithm where training data is stored, and the function is approximated only locally [20]. That means it does not do any generalization with the training data point. To make it clearer, the training phase is not explicitly made from the beginning and it is a fast phase that does not consume much time. So, all the training data will be needed in the test phase. KNN uses the similarity methods like Euclidean distance [8], Manhattan distance, Makowski distance for continuous values and hamming distance for categorical values to calculate the similarity between the training set and testing set and find out the closest k training samples to the test sample. After that it will assign the dominant category class label from k training samples to that test sample.

The idea of KNN is illustrated in Figure 4 below where Xu is the test sample that we want to predict its class label value, (w1,w2,w3) are the values in the class label for the training samples, and the arrows indicate the closest samples neighbors to Xu based on the value of k [21].

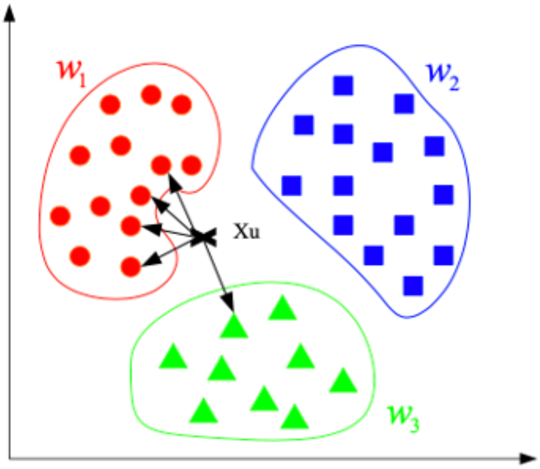

Figure 4:KNN Algorithm [21].

The steps for KNN classifier are quite simple:

- First you need to choose the number of k, where K is the number of nearest neighbors. The optimal number of neighbors(K) in KNN is based on the data set requirements. Generally, it is always a wise choice to choose K as an odd number if the number of classes is even to avoid ambiguous results.
- Second, for the given test set you need to find the k closest training data point using some similarity measures and predict the class for it.
- Third, classify the distance and find the nearest neighbors K based on the minimum distance.

4. Empirical Studies

4.1. Description of dataset

We used the Chronic Kidney Disease Data Set, which is an open-source dataset, and we found it in the machine learning repository on the UCI website. The dataset contains 400 rows and 25 attributes. The class label shows whether or not the patient has CKD.

4.1.1. Statistical Analysis of the Dataset

1) Feature description

This section will explain each attribute in the dataset which is 25 attributes and the type of data it contains, as well as the unit of measurement.

#	Feature Name	Data Type	Unit of Measure
1	Age	Numerical	years
2	Blood Pressure	Numerical	mm/Hg
3	Specific Gravity	Categorical	1.005,1.010,1.015, 1.020,1.025
4	Albumin	Categorical	0,1,2,3,4,5
5	Sugar	Categorical	0,1,2,3,4,5
6	Red Blood Cells	Categorical	normal, abnormal
7	Pus Cell	Categorical	normal, abnormal
8	Pus Cell Clumps	Categorical	present, not present
9	Bacteria	Categorical	present, not present
10	Blood Glucose Random	Numerical	mgs/dl
11	Blood Urea	Numerical	mgs/dl
12	Serum Creatinine	Numerical	mgs/dl
13	Sodium	Numerical	mEq/L
14	Potassium	Numerical	mEq/L
15	Hemoglobin	Numerical	gms
16	Packed Cell Volume	Numerical	percentage
17	White Blood Cell Count	Numerical	cells/cumm
18	Red Blood Cell Count	Numerical	millions/cmm
19	Hypertension	Categorical	yes, no
20	Diabetes Mellitus	Categorical	yes, no

21	Coronary Artery Disease	Categorical	yes, no
22	Appetite	Categorical	good,poor
23	Pedal Edema	Categorical	yes, no
24	Anemia	Categorical	yes, no
25	Class	Categorical	ckd, notckd

2) Numerical Features

In this section, numerical data has been classified into a separate table, and some important values have been calculated to analyze numerical data: the minimum value, the maximum value and the average value for each attribute.

#	Feature Name	Min	Max	Mean
1	Age	2	90	51.483
2	Blood Pressure	50	180	76.469
3	Blood Glucose Random	22	490	148.037
4	Blood Urea	1.5	391	57.426
5	Serum Creatinine	0.4	76	3.072
6	Sodium	4.5	163	137.529
7	Potassium	2.5	47	4.627
8	Hemoglobin	3.1	17.8	12.526
9	Packed Cell Volume	9	54	38.884
10	White Blood Cell Count	2200	26400	8406.122
11	Red Blood Cell Count	2.1	8	4.707

3) Categorical Features

In this section, the categorical data is classified in a separate table, and the number of having each label is calculated for all attributes.

#	Feature Name	Label	Count
1	Specific Gravity	1.005 1.010 1.015 1.020 1.025	7 84 75 106 81
2	Albumin	0 1 2 3 4 5	199 44 43 43 24 1
3	Sugar	0 1 2 3 4 5	290 13 18 14 13 3
4	Red Blood Cells	Normal Abnormal	201 47
5	Pus Cell	Normal Abnormal	259 76
6	Pus Cell Clumps	Present Not present	42 354
7	Bacteria	Present Not present	22 374
8	Diabetes Mellitus	Yes No	137 261
9	Coronary Artery Disease	Yes No	34 364
10	Appetite	Good Poor	317 82
11	Pedal Edema	Yes No	76 323
12	Anemia	Yes No	60 339

13	Hypertension	Yes No	147 251

4.1.2. Performance Criteria

The quality of the analysis will be calculated using four measures, and they were chosen according to the most used measures in the project field, and they are:

1. Accuracy = (TP+TN) / (P + N).
2. Sensitivity = Recall = TP / (TP+FN).
3. Precision = TP/(TP+FP).
4. F-measure = (2 * precision * recall) / (precision * recall).

We will focus on F-measure because we have an imbalance dataset.

4.2. Experimental Setup

The experiment was implemented using the Python language through the Anaconda program. Python was chosen due to that it has good support in data analysis and machine learning. In addition to the presence of many libraries supported by Python, which help in the process of forecasting and analyzing data.

4.2.1. SVM

We implemented the SVM classifier using Anaconda IDE with different number of features starting with the whole features of data set (24 features) in order to monitor the classifier's performance when the features number is changed. Table 4 shows the results of each number of features.

# Features	Accuracy	Precision	Recall	F- measure
24 Features	63.33%	0%	0	0%
18 Features	63.33%	0%	0	0%
15 Features	63.33%	0%	0	0%

14-1 Features	63.33%	0%	0	0%
14-2 Features	88.33%	76.79%	0.98	86%
11 Features	90.83%	8%	1	88.89%
10 Features	88.33%	76.79%	0.98	86%
8 Features	89.17%	78.18%	0.98	86.87%
7 Features	89.17%	78.18%	0.98	86.87%

Table 5: SVM classification

After applying the SVM classifier with different numbers of features we got the highest values when the number of features equal to 11 as it reached the highest accuracy up to 90.83%, precision equals to 8%, recall equals to 1, and F-measure equals to 88.89%.

4.2.2. KNN

KNN was implemented using Anaconda IDE. First the data was trained using different values of K that chosen randomly on the original dataset (24 features without the class), The KNN model predicts a data test set that has never been trained before. The table below explains the values of the evaluation measures with each value of K. Table 5 shows the results of each number of K values.

K value	Accuracy	Precision	Recall	F-measure
9	72.5 %	55.9 %	0.825	66.6 %
11	72.5 %	55.7 %	0.85	67.3 %
15	71.6 %	54.6 %	0.875	67.3 %

Table 6: KNN classification.

After applying the KNN classifier with different values of K we found that when the value of K is 11, we got an F-measure up to 67.3 % which was the highest one and accuracy 72.5%. For that we implemented the KNN classifier using a different number of features in the same value of

K, which is 11, in order to observe the classifier's performance on feature selection. Table 6 shows the results of each number of features.

# Features	Accuracy	Precision	Recall	F- measure
24 Features	72.50%	55.73%	0.85	67.32%
18 Features	72.50%	55.73%	0.85	67.32%
15 Features	72.50%	55.73%	0.85	67.32%
14-1 Features	07.00%	53.44%	0.77	63.26%
14-2 Features	85.00%	71.15%	0.92	80.43%
11 Features	85.83%	72.54%	0.92	81.31%
10 Features	91.66%	82.60%	0.95	88.37%
8 Features	90.83%	80.85%	0.95	87.35%
7 Features	90.83%	80.85%	0.95	87.35%

Table 7: KNN classification with different numbers of features.

After applying the KNN classifier with different numbers of features, we found that when the number of features is 8 and 7, we got an accuracy of 90.83%, precision equals to 80.85 %, recall 0.95, and F-measure up to 87.35 % which was the highest one. Consequently, we concluded that when the number of features is less the classifier has high F-measure and accuracy.

4.3. Optimization strategy

4.3.1. SVM

In our project, we used the grid-search which was used in the machine learning models to find the optimal hyperparameter which made the best result in the predictions. We used several parameters, and we substituted them with different values in order to find the best possible value for these parameters, including:

- C: [0.1, 1, 10, 100, 1000]
- Gamma: [1, 0.1, 0.01, 0.001, 0.0001]
- Kernel: [rbf]

Table 7 shows the value of each performance measure with considering the best parameter for each features.

# Features	Accuracy	Precision	Recall	F- measure
24 Features	84.17%	75.51%	0.84	79.57%
18 Features	84.17%	75.51%	0.84	79.57%
15 Features	84.17%	75.51%	0.84	79.57%
14-1 Features	85.00%	76.00%	0.86	80.85%
14-2 Features	95.83%	93.33%	0.95	94.38%
11 Features	96.67%	95.45%	0.95	95.45%
10 Features	97.5%	95.56%	0.98	96.62%
8 Features	99.17%	97.78%	1	98.88%
7 Features	99.17%	97.78%	1	98.88%

Table 8: SVM optimizing.

After applying the grid-search, we found that it gave us the highest value when the number of features are 8 and 7. The best accuracy was 99.17% , precision equals to 97.78%, recall equals to 1 and f-measure equals to 98.88% . The number of features equal to 7 has been chosen because it is considered to be a smaller number and therefore less computation power.

4.3.2. KNN

We used the K-Fold Cross Validation with KNN algorithm for parameter tuning which made the best result in the predictions. with K being our parameter, we apply K-Fold Cross Validation to find the optimal value of k(neighbors) in KNN that leads to increase the KNN performance.

In the beginning, we applied 10-fold cross validation on our dataset with the different number of features based on the previous classification and feature selection technique. And using a generated list of k(neighbors) from (1 - 50) where k(neighbors) is chosen to be odd number only because it is better to choose the value of k to be odd if the class label has even number of values which in our case (CKD, nonCKD). After that, we got the optimal value of k(neighbors) in KNN which was (k=3) by calculating the misclassification error.

After that, we applied KNN with k=3 as the optimal value of k on the dataset. Table 9 shows the value of parameter k(neighbors) with the value of the performance measure.

# Features	Accuracy	Precision	Recall	F- measure
24 Features	79.16%	63.15%	0.90	04.22%
18 Features	79.16%	63.15%	0.90	04.22%
15 Features	79.16%	63.15%	0.90	04.22%
14-1 Features	78.33%	62.96%	0.85	72.34%
14-2 Features	89.16%	77.55%	0.95	85.39%
11 Features	89.16%	77.55%	0.95	85.39%
10 Features	93.33%	86.36%	0.95	90.47%
8 Features	94.16%	90.24%	0.92	91.35%
7 Features	94.16%	90.24%	0.92	91.35%

Table 9: KNN optimizing.

After applying the cross validation, we found that it gave us the highest values when the number of features is 8 and 7, where k=3. The best accuracy was 94.16 %, with precision equals to 90.24%, recall equals to 0.92, and F-measure up to 91.35 %. Consequently, we choose the number of features to be 7 because we conclude when the features number is less with K=3, the classifier has high performance measures due to less computation power.

5. Result and discussion

Presented below are the results and discussions of various experimental options.

Table 10: Benchmark table

Reference	Year	Techniques	Features	Findings

Aljaaf et al.[11]	2018	RPART, SVM, LOGR, MLP	7 features	The accuracy for each technique:
				RPART = 95.6%
				SVM = 95.0%
				LOGR = 98.1%
				MLP = 98.1%

Hayashi et al.[14]	2019	Algorithm with J48graft	22 features	Accuracy = 95.18%
W.H.S.D et al. [1]	2017	Neural networks Cross Multiclass (Decision Forest, Decision Jungle , Logistic Regression , Neural Network)	25 features	The accuracy for each technique: Decision forest = 99.1% Decision jungle = 96.6% Logistic regression = 95.0% Neural network = 97.5%
Polat et al.[9]	2017	SVM, fold cross validation	7 features	Accuracy = 98.5%
QIN et al.[2]	2019	Logistic regression, Random Forest, LOG with RF	Not fixed	The accuracy for each technique: - RF = 99.75% - LOG = 98.75% Integrated = 99.83%
Rady et al.[13]	2019	Probabilistic Neural Networks (PNN), Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Radial Basis Function (RBF)	25 features	The accuracy for each technique: - PNN = 98.06% - SVM = 90.58% - RBF = 98.06% MLP = 91.69%

Bajbaa et al.	2020	SVM, KNN	7 features	Accuracy = 99.17%
				F-measure = 98.88%

We apply many experimental options with SVM and KNN classifiers on UCI CKD dataset. The discussions and results are as below.

First, for the comparison matter between SVM and KNN, we focused on the accuracy because we have a small imbalance dataset where CKD patients have higher samples than nonCKD patients. In the feature selection phase, we got the best 3 optimal features sets. The first one includes 11 features which are: ['sg', 'al', 'rbc', 'bgr', 'bu', 'sc', 'sod', 'hemo', 'pcv','rc' ,'htn'], and the other set includes 8 features which are : ['htn', 'sg', 'al', 'bgr', 'sc', 'hemo', 'pcv', 'rc']., and the last set includes 7 features which are : ['sg', 'al', 'bgr', 'sc', 'hemo', 'pcv', 'rc']. The number of optimal features influenced the classifier performance.

In SVM, the classification was applied on a different number of features. It was found that the best number of attributes to implement the classification is 7 features. After that, the optimization process was applied using grid search with the following parameters:
C: [0.1, 1, 10, 100, 1000], Gamma : [1, 0.1, 0.01, 0.001, 0.0001], Kernel : [rbf]. The highest value that was reached using the SVM classifier was 99.17% with C = 1000 and Gamma = 0.0001. The following graph shown in Figure 5 compares the SVM classifier result before optimization and after optimization depends on feature selection.

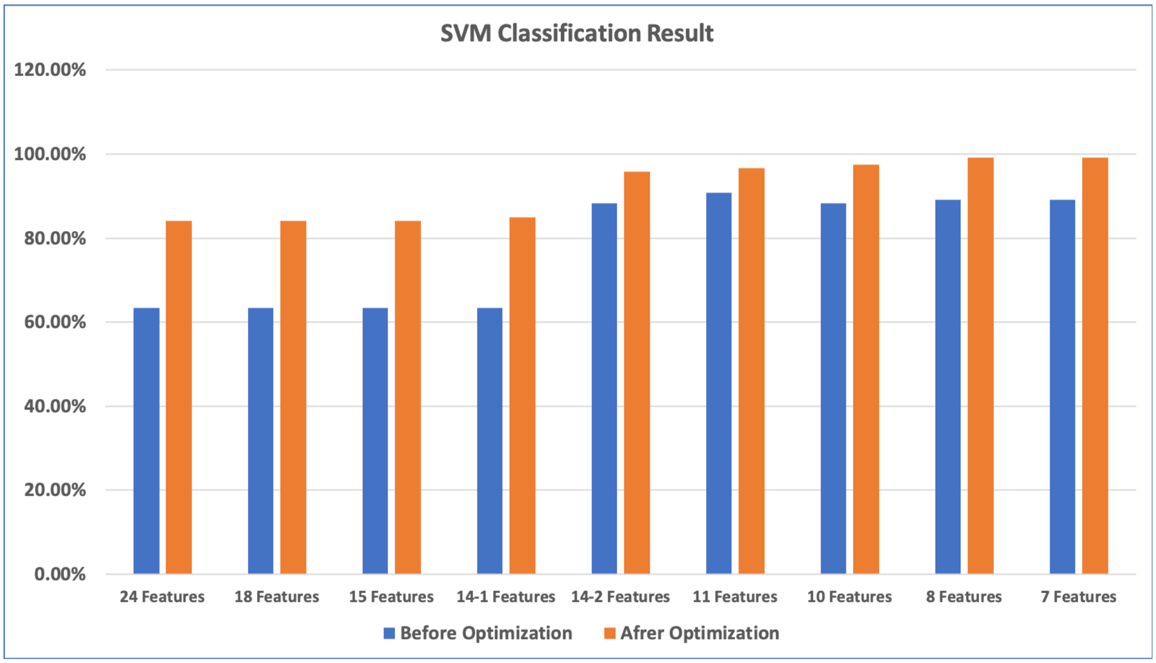

Figure 5: SVM classification result

On the other hand, for the KNN classifier, we applied different numbers of k randomly on the original dataset: 9,11, and 15 to show the best performance and the best k was when k=11. For classification, it was applied on different numbers of features and we found that the best number of features to implement the classification is 7 features. After that, in the optimization phase, we use 10-fold cross validation to find the optimal number of k and it was when k=3. The highest accuracy that reached using KNN classifier with 10-fold cross validation and k=3 for KNN was 94.16% with F-measure up to 91.35%. The below graph shown in Figure 6 compares the KNN classifier result before optimization and after optimization depends on feature selection.

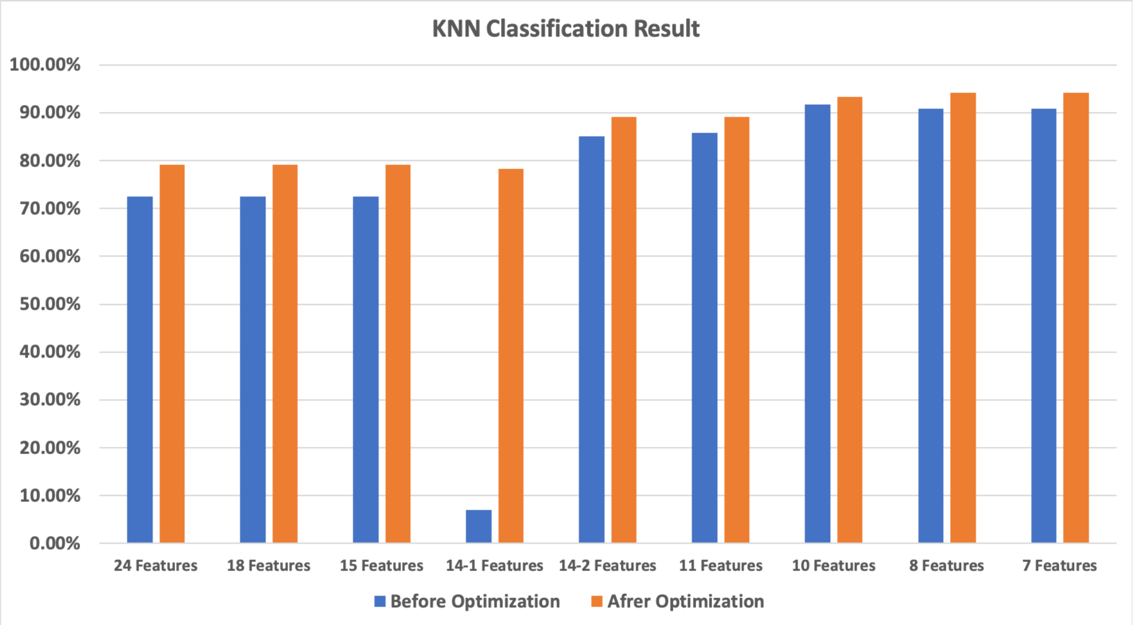

Figure 6: KNN classification result

Based on the benchmark that shown in Table 10 we found that our proposed methodology reached F-measure equals to 98.88%, precision equals to 97.78% and recall up to 1 which indicates the largest value of the recall range. In addition, our proposed methodology got the highest accuracy than all paper that listed in table 9 with accuracy up to 99.17% with 7 features, except QIN et al. paper that reached accuracy up to 99.83% with 8 features.

6. Conclusion and recommendation

CKD become a worldwide health issue around the world. In medical field, to detect CKD, it is important to have an appropriate knowledge about the symptoms. Machine learning can be used to detect CKD where it helps doctors to have fast, accurate predictions about CKD and patients to prevent the disease progression and provide treatment as soon as possible with no time wasted. Furthermore, the proposed models in this paper to predict CKD disease were KNN and SVM where the optimal model that perform more accurate results for CKD diagnosis is the model that constructed with SVM with an accuracy equals to 99.17 %, precision equals to 97.78 %, recall equals to 1 and f-measure equals to 98.88 %. This model reached the highest accuracy with minimum number of features. In addition, we only tested the dataset on two models which are KNN and SVM. As a recommendation, SVM model is good technique to predict CKD since it has reached an adequate accuracy, recall, precision and f-measure values. Also, it will be good to try the same dataset on other models as well other than the proposed model because it may result in better evaluation measures.

References

[1] G. W.H.S.D, “Performance Evaluation on Machine Learning Classification Techniques for Disease (CKD),” *Ieee*, pp. 291–296, 2017.

[2] J. Qin, L. Chen, Y. Liu, C. Liu, C. Feng, and B. Chen, “A Machine Learning Methodology for Diagnosing Chronic Kidney Disease,” *IEEE Access*, 2019.

[3] E. Alickovic and A. Subasi, “Medical Decision Support System for Diagnosis of Heart Arrhythmia using DWT and Random Forests Classifier,” *J. Med. Syst.*, 2016.

[4] Z. Masetic and A. Subasi, “Congestive heart failure detection using random forest classifier,” *Comput. Methods Programs Biomed.*, 2016.

[5] M. Patrício *et al.*, “Using Resistin, glucose, age and BMI to predict the presence of breast cancer,” *BMC Cancer*, 2018.

[6] M. Mahyoub, M. Randles, T. Baker, and P. Yang, “Comparison analysis of machine learning algorithms to rank Alzheimer’s disease risk factors by importance,” in *Proceedings - International Conference on Developments in eSystems Engineering, DeSE*, 2019.

[7] “UCI Machine Learning Repository. Chronic kidney disease dataset.” .

[8] A. Charleonnan, T. Fufaung, T. Niyomwong, W. Chokchueypattanakit, S. Suwannawach, and N. Ninchawee, “Predictive analytics for chronic kidney disease using machine learning techniques,” in *2016 Management and Innovation Technology International Conference, MITiCON 2016*, 2017.

[9] H. Polat, H. Danaei Mehr, and A. Cetin, “Diagnosis of Chronic Kidney Disease Based on Support Vector Machine by Feature Selection Methods,” *J. Med. Syst.*, vol. 41, no. 4, 2017.

[10] R. Misir, M. Mitra, and R. Samanta, “A reduced set of features for chronic kidney disease prediction,” *J. Pathol. Inform.*, 2017.

[11] A. J. Aljaaf *et al.*, “Early Prediction of Chronic Kidney Disease Using Machine Learning Supported by Predictive Analytics,” in *2018 IEEE Congress on Evolutionary Computation, CEC 2018 - Proceedings*, 2018.

[12] R. A. Alassaf *et al.*, “Preemptive Diagnosis of Diabetes Mellitus Using Machine Learning,” in *21st Saudi Computer Society National Computer Conference, NCC 2018*, 2018.

[13] E. H. A. Rady and A. S. Anwar, “Prediction of kidney disease stages using data mining

algorithms,” *Informatics in Medicine Unlocked*. 2019.

[14] Y. Hayashi, K. Nakajima, and K. Nakajima, “A rule extraction approach to explore the upper limit of hemoglobin during anemia treatment in patients with predialysis chronic kidney disease,” *Informatics Med. Unlocked*, vol. 17, no. September, p. 100262, 2019.

[15] D. B. Costa and E. Pinheiro, “Computer-aided diagnosis of chronic kidney disease in developing countries : A comparative analysis of machine IEEE Access Original Manuscript ID : Access-2019-58139 Original Article Title : “ Computer-aided diagnosis of chronic kidney disease in developi,” 2020.

[16] D. Subramanian, “No Title,” *medium*, 2019. .

[17] R. Pupale, “Support Vector Machines(SVM) — An Overview,” 2018. .

[18] J. Faytong and R. Gove, *Chapter 4 - Machine Learning and Event-Based Software Testing: Classifiers for Identifying Infeasible GUI Event Sequences*. ScienceDirect, 2012.

[19] L. Chen, “No TitleSupport Vector Machine — Simply Explained,” *towardsdatascience*, 2019. .

[20] S. B. Imandoust and M. Bolandraftar, “Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events : Theoretical Background,” *Int. J. Eng. Res. Appl.*, 2013.

[21] G. F. Fan, Y. H. Guo, J. M. Zheng, and W. C. Hong, “Application of the weighted k-nearest neighbor algorithm for short-term load forecasting,” *Energies*, 2019.