Computers in Biology and Medicine 109 (2019) 101–111

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study

Njoud Abdullah Almansour, Hajra Fahim Syed, Nuha Radwan Khayat, Rawan Kanaan Altheeb, Renad Emad Juri, Jamal Alhiyafi*, Saleh Alrashed, Sunday O. Olatunji

Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, PO Box 1982, Dammam, Saudi Arabia

ARTICLE INFO

Keywords:
Machine learning
Artificial Neural Network (ANN)
Support Vector Machine (SVM)
Chronic Kidney Disease (CKD)

ABSTRACT

This paper aims to assist in the prevention of Chronic Kidney Disease (CKD) by utilizing machine learning techniques to diagnose CKD at an early stage. Kidney diseases are disorders that disrupt the normal function of the kidney. As the percentage of patients affected by CKD is significantly increasing, effective prediction procedures should be considered. In this paper, we focus on applying different machine learning classification algorithms to a dataset of 400 patients and 24 attributes related to diagnosis of chronic kidney disease. The classification techniques used in this study include Artificial Neural Network (ANN) and Support Vector Machine (SVM). To perform experiments, all missing values in the dataset were replaced by the mean of the corresponding attributes. Then, the optimized parameters for the Artificial Neural Network (ANN) and Support Vector Machine (SVM) techniques were determined by tuning the parameters and performing several experiments. The final models of the two proposed techniques were developed using the best-obtained parameters and features. The empirical results from the experiments indicated that ANN performed better than SVM, with accuracies of 99.75% and 97.75%, respectively, indicating that the outcome of this study is very promising.

1. Introduction

Kidney diseases are disorders that affect the functions of the kidney. During the late stages, kidney diseases can cause kidney failure. According to the data most recently released by the Saudi Center of Organ Transplantation Registry, 10,203 patients diagnosed with kidney disease receive hemodialysis [1]. Diabetes, high blood pressure, and unhealthy lifestyles have led to an increase in the number of patients with CKD [2]. Patients with CKD suffer from various side effects. These complications include damage to the nervous and immune systems that disrupt daily activities. To assist in the prevention of CKD, machine learning techniques can be utilized to diagnose CKD at an early stage.

Artificial Neural Network (ANN) and Support Vector Machine (SVM) are widely used machine learning techniques. Both ANN and SVM have advantages and have been proven to perform excellently in several fields, including medical diagnosis, weather prediction, stock market analysis, and image processing [3]. ANN is a mathematical model that acts similar to human neurons. ANN has the following two major features: the ability to learn how to perform its functions once it is properly trained and the ability to generalize and produce a reasonable solution to unobserved data [4]. In contrast, SVM is a computational algorithm that can learn from experience and examples to allocate labels to objects. The basic function of SVM is to separate binary labeled data based on a line achieving the maximum distance between the labeled data [5]. SVM performs well and has good accuracy even with limited examples. Due to its fruitful advantages, SVM is distinct from other machine learning techniques [6–8]. This paper applies both ANN and SVM techniques to a CKD dataset to measure their accuracy and compare their performance. The WEKA was used to help in preprocessing and conducting the experiment using the dataset. The experimental results indicated that ANN outperformed SVM in almost all models of this dataset. ANN showed a better performance with an accuracy of 99.75% using the optimized features, while SVM had a performance accuracy of 97.75%. The ANN classifier had a considerably longer runtime than SVM, but no large difference in accuracy was observed.

The remainder of this work is organized as follows. Section 2 provides a review of the related literature. Section 3 discusses the proposed

* Corresponding author.
E-mail addresses: 2130008151@iau.edu.sa (N.A. Almansour), 2130009210@iau.edu.sa (H.F. Syed), 2130005966@iau.edu.sa (N.R. Khayat), 2130007114@iau.edu.sa (R.K. Altheeb), 2130009362@iau.edu.sa (R.E. Juri), jalhiyafi@iau.edu.sa (J. Alhiyafi), saalrashed@iau.edu.sa (S. Alrashed), osunday@iau.edu.sa (S.O. Olatunji).

https://doi.org/10.1016/j.compbiomed.2019.04.017
Received 6 October 2018; Received in revised form 19 April 2019; Accepted 20 April 2019

machine learning techniques, including the neural network and support vector classifiers. In section 4, we present the empirical studies, including the dataset description, experimental setup, methodology, adopted optimization strategy and parameter search strategy. Section 5 presents the results and discussion, section 6 presents the conclusion, and section 7 presents the recommendations emanating from this work.

2. Review of related literature

Z. Chen, X. Zhang, and Z. Zhang presented the applicability of clinical data and the robustness of three multivariate models in clinical risk assessment of CKD patients [2]. The UCI repository dataset for Chronic Renal Failure (CFR) was used in this study. The three multivariate models used included the K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Soft Independent Modeling of Class Analogy (SIMCA). The authors were able to label the patients as CKD or non-CKD with an accuracy of over than 93%. Another important finding of this paper was that SVM processed the noise disturbance of the composite dataset better and had the highest accuracy in predicting CKD compared to the other two models as its accuracy reached 99%.

In Ref. [9], the authors compared three different models for supervised feature selection. In their study, the authors used a supervised rule mining algorithm to explore the feature space and predict CKD stages based on proton Nuclear Magnetic Resonance (1H NMR) data. The first model used by the authors involved global data mining, which is a dataset wide analysis using fisher's exact test in which features with a very small probability (p-value ≤ 0.05) are chosen, and normalized mutual information is used. The second model involved a local approach based on supervised rule mining that the authors used to lift and z-score. The final model consisted of the output of the first model, which served as the input of the second model. After examining the three models, the authors found that the local approach based on the supervised rule mining model was the best model since it had the highest F1 score and mean and the lowest standard deviation.

In Ref. [10], the authors developed a decision support system to help doctors predict the occurrence of CRF in patients. This decision was based on various machine learning techniques, including Naive Bayes, LDA, K-Nearest Neighbor (KNN), tree-based decision, and random subspace classification algorithms. These techniques were applied to a UCI repository dataset of CRF. It was found that the K-Nearest Neighbor (KNN) with random subspace classifier had 94% accuracy in prediction. The authors suggested using kernel or neural based classifiers instead of KNN with a random subspace classifier.

In a research study [11], a C4.5 decision tree algorithm was performed using the UCI repository dataset to diagnose CRF in patients. After applying the algorithm, eight if-then rules were developed to predict CRF. The accuracy of the algorithm was found to be 98.25%, which is higher than the accuracy reported in a study conducted by Al-Taee, Al-Hyari, & Al-Taee [10], who used the same dataset. The experiment was performed using 3-fold cross-validation. The researchers also found that serum creatinine, pedal edema, diabetes mellitus, hemoglobin, and specific gravity were the most important features in the dataset. Finally, the authors suggested using a dataset with more diversity in age, gender, and location of patients to allow the algorithm used for diagnosis to be universally applicable.

In another study [12], the authors also developed a decision support system similar to that in the study performed by Ref. [10]. However, the machine learning techniques used for classification included Artificial Neural Network (ANN), Naïve Bayes, and a decision tree. This system was applied to a dataset from Prince Hamza Hospital in Jordan. The performance of the machine learning techniques was analyzed by calculating the sensitivity, specificity, and accuracy. The authors found that the decision tree was the most accurate technique in predicting CRF. Finally, the authors suggested implementing more machine learning techniques, including Support Vector Machine (SVM), to predict CRF.

Salekin and J. Stankovic introduced two methods in their study to reduce overfitting and identify the most important attributes for CKD prediction (Salekin & Stankovic, 2016). The two methods were LASSO regularization and wrapper. The authors used three different classification methods and compared their accuracy using the F1 measure and root mean square error. The three classifiers included random forest, k-nearest neighbors and neural networks. It was found that the random forest algorithm can detect CKD with the highest accuracy of 0.998 using the F1-measure with a 0.107 root mean square error.

In Ref. [13], the authors examined the feasibility of the fuzzy based expert system (FuRES) and fuzzy optimal associative memory (FOAM) techniques for the diagnosis of CKD. The UCI repository dataset of CRF was used in this study. A partial least squares discriminant analysis (PLS-DA) and linear classifier methods were used for comparison. The accuracy, sensitivity, and specificity were calculated to evaluate the performance of the two fuzzy classifiers. Both FuRES and FOAM exhibited similar accuracies with average prediction rates exceeding 98%. FuRES exhibited better specificity, while FOAM had better sensitivity. In the comparison with PLS-DA, PLS-DA exhibited the lowest accuracy and specificity but achieved a sensitivity of 100%. To measure the robustness of the two fuzzy classifiers, 11 levels of noise, i.e., 0%–10%, were added to the dataset similar to the study conducted in Ref. [2]. Finally, the study proved that both FuRES and FOAM are feasible for the classification of biomedical data, including CKD prediction.

In a study performed by Ref. [14], machine learning techniques, including SVM, decision tree, Naïve Bayes, and KNN, were used to detect whether a patient has CKD using the UCI repository dataset of CRF. The selection of features in the dataset was performed by implementing a ranking algorithm. The algorithm yielded an output of 10 features that were important for CKD prediction. Then, different machine learning techniques were applied to the dataset, and their accuracies were calculated using the root mean square error and mean absolute error and by plotting a receiver operating characteristic curve. The decision tree algorithm was found to be most accurate with an accuracy of 99.75%, while SVM was the second most accurate with an accuracy of 97.75%. Finally, the authors suggested implementing ANN and fuzzy logic to the dataset and analyzing their accuracies.

In Ref. [15], the author proposed a new method to automate the diagnosis of CKD by using data mining methods. K-means was applied to the data for preprocessing. The classification methods KNN, SVM, and Naïve Bayes were applied to the preprocessed data as the data mining methods. An accuracy rate of 97.8% was obtained, which was the highest rate, using the proposed method with urine test attributes. The accuracy was increased in the age group of 35 and above using the same attributes. This study also revealed that different combinations of the dataset attributes resulted in different accuracy rates ranging from 83.75% to 97.8%.

In Ref. [16], the SVM method was applied to the UCI repository dataset of CRF to diagnose CKD. The study compared the results of applying SVM to all features and applying SVM to selected features. The features were selected using wrapper and filter techniques, while the study in Ref. [17] used regularization and the wrapper approach. Each technique was applied using two different evaluators. The accuracy of the results was determined by calculating several parameters, including the confusion matrix, recall, and precision. It was found that reducing the number of features increased the accuracy of the results. The study concluded that the filtered subset evaluator with the best first search engine technique to select features in the dataset was the most accurate technique with an accuracy of 98.5%.

In Ref. [18], the authors proposed a hierarchical meta-classification method. The baseline classifiers used in the study were Naïve Bayes, a logistic regression, a decision tree, and random forests. Then, the proposed method was used to form two subtasks of the multiclass classification step. The first subtask involved the division of the majority and minority classes, and the second subtask involved the separation of the records in the minority class. The results of this paper showed that the

proposed method can identify patients with severe cases with high sensitivity, specificity, and F-measure even with a reduction in the dataset size.

Z. Chen, X. Zhang, and Z. Zhang utilized K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Soft Independent Modeling of Class Analogy (SIMCA) in their study and were able to reach an accuracy of 93%. However [10], found that K-Nearest Neighbor (KNN) with a random subspace classifier had 94% accuracy in prediction. In another study [12], the authors found that the decision tree was the most accurate technique for predicting CRF. Furthermore, Salekin and J. Stankovic found that the random forest algorithm can detect CKD with the highest accuracy of 0.998. In another study [16], it was found that reducing the number of features increased the accuracy of the results with an accuracy of 98.5%. Therefore, this study used a recursive feature elimination algorithm that uses the correlation between the attributes and target variable to rank the features.

The recommendations of authors from several literature reviews were considered in this study. In Ref. [10], the author suggested using neural based classifiers instead of KNN. This study used ANN as a neural based classifier and achieved a high accuracy of 99.75%. In Ref. [12], the authors suggested implementing more machine learning techniques, including Support Vector Machine (SVM). This study utilizes the SVM machine learning technique, reaching performance accuracy of 97.75%. The ANN machine learning technique has been shown to have the highest accuracy compared with the other utilized techniques in all previous studies.

3. Description of the proposed techniques

Artificial Neural Network (ANN) and Support Vector Machine (SVM) are the two techniques used in this study. The following section provides an overview and a technical description of both techniques.

3.1. Artificial Neural Network (ANN)

3.1.1. Overview

Regarding the performance of complicated tasks, such as recognition and prediction, the human brain performs remarkably better than digital computers due to its fascinating ability to learn. The human brain consists of hundreds and thousands of billions of nerve cells called neurons. Axons connect these neurons to other cells. The human brain performs computations in an interconnected network of neurons that communicate through electric pulses. Stimuli from the external environment or inputs from sensory organs are received by dendrites. An electric impulse is created by these inputs that travels through the neural network. A neuron can either send the message to another neuron to address the issue or not send the message and solve the issue itself. The idea of ANN was derived from how the brain can form accurate connections [19].

In the 1940s, the neurophysiologist and mathematician Warren McCulloch and Walter Pitts modeled a neuron. The model was a simple switch with two inputs and one binary output. The neuron receives its input from other neurons, and its output is either zero or one, i.e., active or inactive, based on the total weighted input [20]. Subsequently, in the 1960s, Rosenblatt discovered that a network of such modeled neurons could perform recognition and function with some destroyed neurons performing similarly to the human brain [20]. ANN is a mathematical model that acts as neurons in the human brain. ANN has many usages in real life, such as predicting the stock market [21]. ANN has two major features, i.e., the ability to learn how to perform its functions once it is properly trained and the ability to generalize, which is the ability to produce a reasonable solution to unobserved data. ANN is mainly used to solve three problems, including classification, noise reduction, and extrapolation [4].

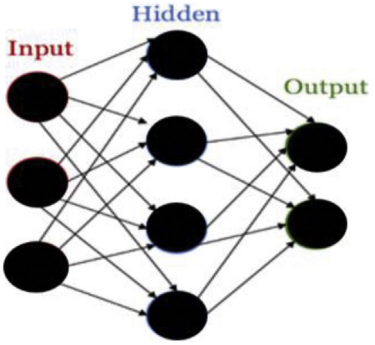

Fig. 1. ANN layers.

3.1.2. Technical description

In general, an Artificial Neural Network (ANN) can be divided into the following three layers [22] as shown in Fig. 1:

1. Input layer:

This layer is the first layer where the data/features are received. Some normalization techniques are applied to these inputs to limit the inputs to a certain range. The normalized inputs facilitate the work of the neural network, resulting in better precision.

2. Hidden (intermediate or invisible) layer:

This layer can be a collection of layers depending on the application of the network. The responsibility of these layers is to recognize the pattern of a process or system. Most neural network activities are performed in these layers.

3. Output layer:

This layer also contains neurons, which are used to represent the outputs of the final network produced from the previous neuron-processing layers. The fundamentals of the composition of ANN involve how the neurons interconnect with each other and how their layers are formed; in addition, their disposition can be categorized into four different types. These types include single-layer feedforward, mesh, recurrent, and multilayer feedforward networks [22].

3.1.2.1. Single-layer feedforward network. This network has a single input layer and an output layer, which is also known as a neural layer without the hidden layer. The data typically flow in one direction (unidirectional) from the input layer to the output layer as shown in Fig. 2. These types of networks are usually used to solve pattern classification and linear filtering problems [22].

3.1.2.2. Multilayer feedforward networks. In contrast to the previous

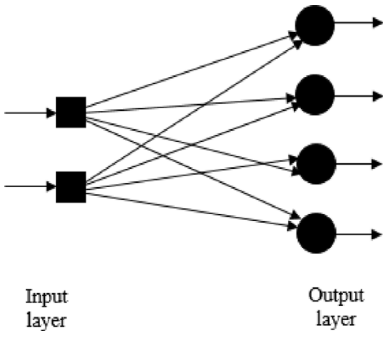

Fig. 2. Single layer network.

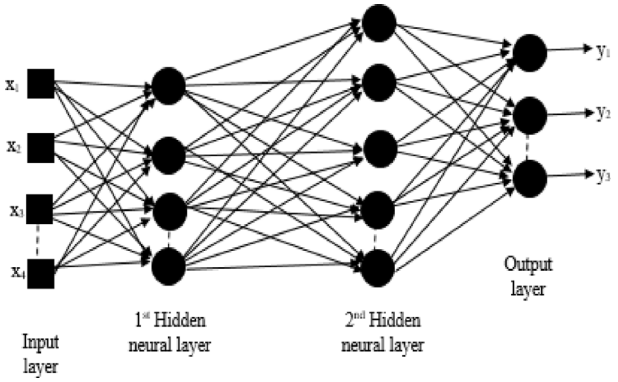

Fig. 3. Multilayer network.

feedforward network, these feedforward networks have multiple layers that contain one or more hidden neural layers as shown in Fig. 3. These layers are used as solutions to diverse problems, including problems related to function approximation, process control, system identification, optimization, and other applications. These types of networks use Multilayer Perceptron (MLP) and the Radial Basis Function (RBF), and their training processes use the generalized delta rule and competitive/delta rule, respectively [22].

3.1.2.3. Recurrent (feedback) network. In a recurrent network, the output of the neurons is used as input in other nodes. Due to this feature, these types of networks are good candidates for dynamic information processing, suggesting that they can be applied to time-variant systems, such as time series prediction, system identification and optimization, process control, and other problems, by using the feedback process. The previous output is considered the product of the current output.

3.1.2.4. Mesh networks. Mesh networks have a structured topology in which all nodes can cooperate in the distribution of data in the network. These networks are used in a wide range of applications, including problems involving data clustering, pattern recognition, system optimization, graphs, and other applications. By using the formula $\frac{N*(N-1)}{2}$, where N is the number of nodes in the network, we can calculate the connections in the mesh network.

3.2. Support Vector Machines (SVM)

3.2.1. Overview

In the late 1990s, Vapnik, Cortes, and Boser introduced a classifier obtained from statistical learning theory [3]. This classifier is Support Vector Machine (SVM), which subsequently in the same year became widely known in the machine learning community. SVM is a supervised machine learning technique that is used for both classification and regression problems. SVM is most widely used for binary classification problems. SVM continued to increasingly grow and became one of the most popular machine learning techniques. SVM has been utilized in several fields, including bioinformatics and handwritten recognition [3]. SVM is also used in other applications, including medical diagnosis, weather prediction, stock market analysis, and image processing. Similar to all other machine learning techniques, SVM is a computational algorithm that learns from experience and examples to allocate labels to objects. For example, for SVM to distinguish between real and non-real credit cards, SVM needs to inspect a large set of real and non-real credit card images. The basic function of SVM is to separate binary labeled data based on a line achieving the maximum distance between the labeled data [5].

Most machine learning techniques suffer from the so-called curse of dimensionality. The curse of dimensionality occurs when a model inspects a limited number of examples and has limited experience in the presence of several features. Such limitations could reduce a model's performance. It has been proven that the SVM model is prone to the curse of dimensionality [6–8]. SVM performs well and has good accuracy even with limited examples. Due to these fruitful advantages, SVM is distinct from other machine learning techniques [6–8]. SVM uses the so-called kernels function to help separate labeled data. One of the advantages associated with using kernels in SVM is that SVM applies kernel definitions to non-vector inputs, i.e., inputs without size or direction, which are particularly important in the medical field and critical to biological applications. Such an advantage allows SVM to label DNA and protein sequences. In addition, kernels can be defined based on a combination of different data types [23].

3.2.2. Technical description

SVM is an algorithm whose goal is to increase the accuracy of the hypothesis function. The following four basic concepts must be introduced to fully understand SVM [23]:

3.2.2.1. Separation hyperplane separating labels. The separation hyperplane basically corresponds to the line that separates the labeled data as shown in Fig. 4. The usage of the separation hyperplane eases the prediction process by simply determining whether a new example is a part of the labeled data above or below the line. However, if the points are not separable by a straight line, the hyperplane is transformed into a higher dimensional hyperplane to separate the points. In one dimension, the separation hyperplane can be a point, while in two dimensions, the separation hyperplane can be a line. In three dimensions, the separation hyperplane must be a plane [23].

3.2.2.2. Maximum margin hyperplane. The notion of classifying points by dividing them using a line or hyperplane in three dimensions is also used in other machine learning techniques. However, in SVM, the technique by which the hyperplane is selected is different. Multiple lines can separate the data points as shown in Fig. 5. However, the optimal line is selected by using support vectors to develop a maximum margin hyperplane as shown in Fig. 6 [23].

To maximize the margin hyperplane, the following equation is used: $f(\vec{x}) = sgn(\sum_{i=1}^{n} y_i \alpha_i \cdot K(\vec{x}, \vec{x_i}) + b)$ (Cortes & Vapnik, 1995), where y is the output variable, b is the bias and α_i is obtained from solving the following equation on each input: $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \cdot y_i y_j \cdot K(\vec{x_i}, \vec{x_j})$, where x_j are the named support vectors if their corresponding α_i is greater than 0 [24].

3.2.2.3. Soft margin. The SVM classifier is used to separate data points using a hyperplane. However, occasionally, the data points may lie on the wrong side of the line due to an anomaly in the dataset. For such points, a soft margin must be created to allow some points to be on the wrong side of the hyperplane without affecting the percentage accuracy of the classifier. The soft margin can be set by changing certain parameters of the SVM algorithm. These parameters monitor the

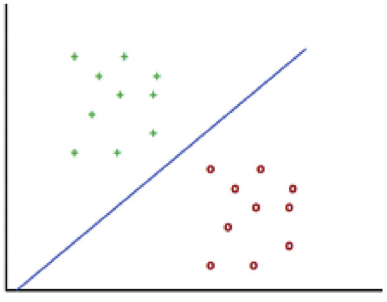

Fig. 4. Separation hyperplane.

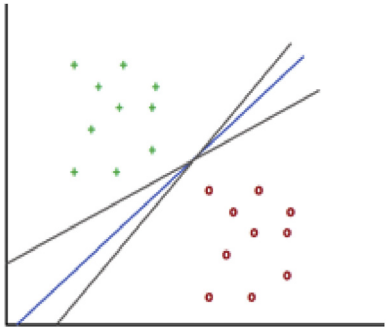

Fig. 5. Multiple hyperplane options.

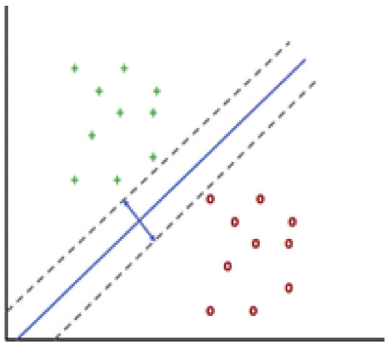

Fig. 6. Maximum margin hyperplane.

number of points that can be on the wrong side of the hyperplane. These parameters also control the distance between the points and the hyperplane [23].

3.2.2.4. Kernel function. The kernel function assists in separating the data points in a higher dimension space if they are not separable in a lower dimension (see Fig. 7). However, projecting the points in a higher dimension may lead to the curse of dimensionality. According to the curse of dimensionality, to obtain an adequate level of accuracy, the number of training samples must exponentially increase as the number of features increases. Projecting points to a higher dimension may also lead to the problem of overfitting. Overfitting implies that the classifier works well with the training set but does not have good levels of accuracy with new examples. Therefore, there is a need to choose the right number of dimensions for the kernel, which can be achieved by trial and error to develop the optimal kernel function. Then, cross-validation can be applied to determine the accuracy levels of the kernels (Noble, 2006).

The kernel function is represented using mathematical equations. Different equations are used to represent different types of kernel functions, which is illustrated in the following examples [25]:

1. Polynomial kernel: $K(\vec{x_i}, \vec{x_j}) = (\vec{x_i} . \vec{x_j} + 1)^d$

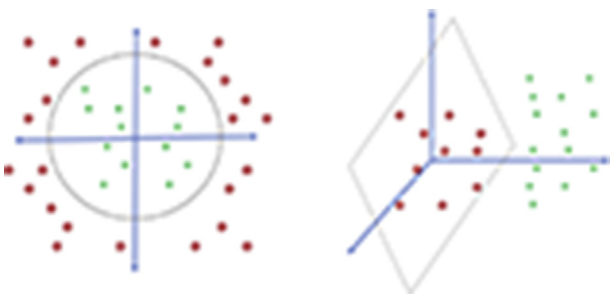

Fig. 7. Using the kernels function to map the data points to a higher dimension.

2. Linear kernel: $K(\vec{x_i}, \vec{x_j}) = x_i^T x_j$
3. Gaussian kernel: $K(\vec{x_i}, \vec{x_j}) = exp(-\gamma \|\vec{x_i} - \vec{x_j}\|^d)$

4. Empirical studies

This section describes the experiments performed in this study. This section first provides a detailed description of the experiments and statistical analysis of the dataset. This section describes how the experiment was carried out. Finally, a description of how the search for the parameter that achieves the best possible performance was carried out is provided.

4.1. Description of the dataset

This experiment was performed using the CKD dataset available in the UCI machine learning repository [26]. The CKD dataset was released in July 2015 by Apollo Hospitals, Tamilnadu, India. The dataset consists of 24 attributes, including 11 numerical and 13 nominal attributes, collected from 400 instances over a two-month period. These 400 instances consist of 250 CKD patients and 150 non-CKD patients. As a result, this dataset is used for binary classification of either CKD or non-CKD. Table 1 presents a list of the attributes, their types and values.

4.1.1. Statistical analysis of the dataset

The statistical analysis of the dataset is introduced in Table 2. The mean, median, standard deviation, minimum and maximum of the numerical attributes in the dataset are introduced.

The nominal attributes, their labels and their counts are presented in Table 3. The dataset was preprocessed before performing the statistical analysis, and the missing values were replaced with the mean of the numerical distribution.

Table 4 presents the correlation coefficients between each numerical attribute and the targeted class attribute. Correlation coefficients are used to determine how much each numerical attribute contributes to the target attribute. Correlation coefficients help in the process of feature selection. As shown in Table 4, the largest correlation is approximately 0.40 (blood glucose random attribute). Overall, the correlation between the attributes and the target class is considered weak, and the correlation values ranged from −0.7 to 0.4.

Table 1
Dataset description (adapted from Ref. [26]).

Attribute	Type	Value
Age	Numerical	Years
Blood Pressure	Numerical	mm/Hg
Specific Gravity	Nominal	1.005, 1.010, 1.015, 1.020, 1.025
Albumin	Nominal	0, 1, 2, 3, 4, 5
Sugar	Nominal	0, 1, 2, 3, 4, 5
Red Blood Cells	Nominal	Normal, Abnormal
Pus Cell	Nominal	Normal, Abnormal
Pus Cell Clumps	Nominal	Present, Not present
Bacteria	Nominal	Present, Not present
Blood Glucose Random	Numerical	mgs/dl
Blood Urea	Numerical	mgs/dl
Serum Creatinine	Numerical	mgs/dl
Sodium	Numerical	mEq/L
Potassium	Numerical	mEq/L
Hemoglobin	Numerical	gms
Packed Cell Volume	Numerical	____
White Blood Cell Count	Numerical	cells/cumm
Red Blood Cell Count	Numerical	millions/cmm
Hypertension	Nominal	Yes, No
Diabetes Mellitus	Nominal	Yes, No
Coronary Artery Disease	Nominal	Yes, No
Appetite	Nominal	Good, Poor
Pedal Edema	Nominal	Yes, No
Anemia	Nominal	Yes, No

Table 2
Statistical Analysis of the dataset.

Attribute	Mean	Median	Standard Deviation	Maximum	Minimum
Age	51.483	54	16.975	90	2
Blood Pressure	76.469	80.235	13.476	180	50
Blood Glucose Random	148.037	126	74.783	490	22
Blood Urea	57.426	44	49.286	391	1.5
Serum Creatinine	3.072	1.4	5.617	76	0.4
Sodium	137.529	137.529	9.204	163	4.5
Potassium	4.627	4.627	2.82	47	2.5
Hemoglobin	12.526	12.526	2.716	17.8	3.1
Packed Cell Volume	38.884	38.884	8.151	54	9
White Blood Cell Count	8406.12	8406.12	2523.22	26400	2200
Red Blood Cell Count	4.707	4.707	0.84	8	2.1

Table 3
Analysis of the nominal attributes.

Attribute	Label	Count
Specific Gravity	1.005	7
	1.01	84
	1.015	75
	1.02	153
	1.025	81
Albumin	0	245
	1	44
	2	43
Albumin	3	43
	4	24
	5	1
Sugar	0	339
	1	13
	2	18
	3	14
	4	13
	5	3
Red Blood Cells	Normal	353
	Abnormal	47
Pus Cell	Normal	324
	Abnormal	76
Pus Cell Clumps	Present	42
	Not present	358
Bacteria	Present	22
	Not present	378
Hypertension	Yes	147
	No	253
Diabetes Mellitus	Yes	137
	No	263
Coronary Artery Disease	Yes	34
	No	366
Appetite	Good	318
	Poor	82
Pedal Edema	Yes	76
	No	324
Anemia	Yes	60
	No	340

4.2. Experimental setup

The experiment was performed to predict and compare the accuracies of two machine learning techniques, i.e., Artificial Neural Network and Support Vector Machine, using the CKD dataset. The machine learning algorithms were performed using the dataset via WEKA (Waikato Environment for Knowledge Analysis). WEKA consists of various machine learning algorithms that can be applied to datasets. WEKA also contains options for preprocessing data, selecting features and applying techniques to partition the dataset for training and testing.

Table 4
Correlation coefficients between each attribute and the target attribute.

Attribute	Target attribute	Correlation coefficient
Age	Class	0.22541
Blood Pressure	Class	0.2906
Blood Glucose Random	Class	0.40137
Blood Urea	Class	0.37203
Serum Creatinine	Class	0.29408
Sodium	Class	−0.3423
Potassium	Class	0.07692
Hemoglobin	Class	−0.7296
Packed Cell Volume	Class	−0.6901
White Blood Cell Count	Class	0.20527
Red Blood Cell Count	Class	−0.5909

The dataset is first preprocessed and prepared for the experiment. The missing values in the dataset were replaced with the mean of the numerical distribution of the attributes. This step was performed via WEKA using an unsupervised attribute filter (to replace the missing values). Then, the correlation coefficients of each numerical attribute were calculated to rank the features for the feature selection.

Subsequently, the optimized parameters for Artificial Neural Network and Support Vector Machine were identified. This step was performed by tuning the seed, hidden layers and learning rate parameters of ANN, whereas the kernel type and cost parameters of SVM were used to determine the optimized parameters. Then, the values of the parameters of both techniques were changed to perform several experiments.

After changing the values of the parameters of both techniques to perform several experiments, it was found that ANN performed better, ~~performs better~~ with an accuracy of 99.75%, using the optimized features, while SVM had a performance accuracy of 97.75%. A 10-fold cross-validation was used to partition the dataset (90% for the training data and 10% for the testing data following 10-fold cross-validation principles).

Finally, an investigation of the effect of selecting features from the dataset was conducted. This step was performed using 10-fold cross-validation. The number of features was minimized by half each time, and the time required and accuracy achieved were recorded for each technique based on the divided dataset. This process was repeated until one feature remained. The process of eliminating the features was consistent with the results obtained in the feature correlation with the target variable. In each iteration, we chose the first half that had the highest correlation. Hence, the remaining feature was the feature that had the highest correlation with the target variable. Finally, the accuracy of each classifier was calculated using a confusion matrix, and the result was enhanced by a Receiver Operating Characteristic curve (or ROC curve) area plot. ROC is a plot that illustrates the false positive rate against the true positive rate with different possible cut-off points of a diagnostic test.

4.3. Optimization strategy

4.3.1. ANN optimized parameters

After changing three parameters of the ANN algorithm, the prediction accuracy of ANN was compared. The three parameters varied and their default values are as follows:

1. Seed – 0.
2. Hidden Layers – a.
3. Learning Rate - 0.3.

The accuracy of the default model was 97.75%. To obtain higher accuracies, the parameters were tuned, and their accuracies were analyzed. The first parameter that was tuned was the hidden layer

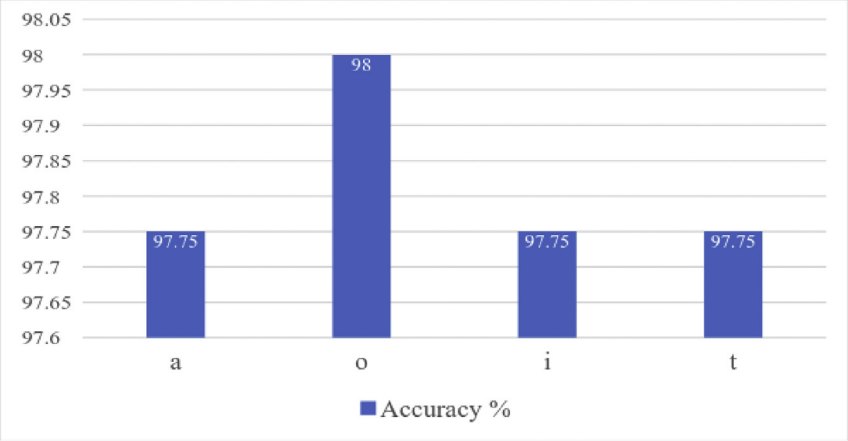

Fig. 8. No. of nodes in the hidden layers and their accuracies.

parameter, representing the number of nodes in each layer, which can be a list of natural numbers separated by commas or letters as follows:

1. 'a' represents the number of attributes and classes in the dataset divided by 2.
2. "' represents the number of attributes in the dataset.
3. 'o' represents the number of classes in the dataset.
4. 't' represents the number of attributes and classes in the dataset.

The seed parameter is used to seed the random number generator, and its value can range from one to infinity. Random numbers are used to set the initial weights of the connections between the nodes. Finally, the learning rate defines the learning rate of the backpropagation algorithm, and its allowed values range from 0 to 1.

First, the number of nodes in each hidden layer parameter was changed, and the effect on the model's accuracy was observed. All parameter changes used 10-fold cross-validation for the partitioning of the training and test datasets with the ratio (90: 10). Fig. 8 shows the changes in the hidden layer parameter and the corresponding accuracies in a bar chart.

Then, the seed parameter of each value of the previously specified hidden layer was changed, and the accuracies were compared. A random exhaustive search method was used to choose the values. Fig. 9 shows the seed values and their prediction accuracies after changing the values. It was found that among all hidden layer node values, the highest accuracy was observed when the seed value was set to 150.

Finally, the learning rate parameter was tuned to each value of the hidden layer nodes, and the accuracies were calculated. The default

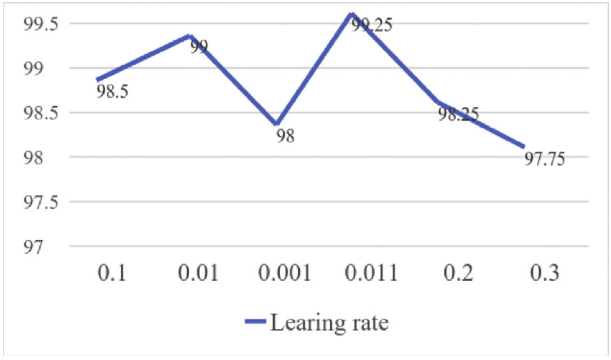

Fig. 10. Learning rate values and their accuracies.

value of this parameter is 0.3. The values were altered in a range from 0 to 0.3 since values above 0.3 always exhibit lower accuracy. In addition, the seed value was set to 150 as this value yielded the highest accuracy of all values of nodes in the previously selected hidden layers. Fig. 10 shows the results of the analysis. It was found that the trend of the graph of all hidden layer node numbers with a seed value of 150 was the same. As a result, only one line is drawn to represent the trend of all numbers of hidden layer nodes. The highest accuracy (99.25%) was observed when the value of the learning rate was 0.001.

The following Table 5 represents the optimal parameters of ANN for predicting CKD according to the results of the accuracies achieved by changing the three parameters.

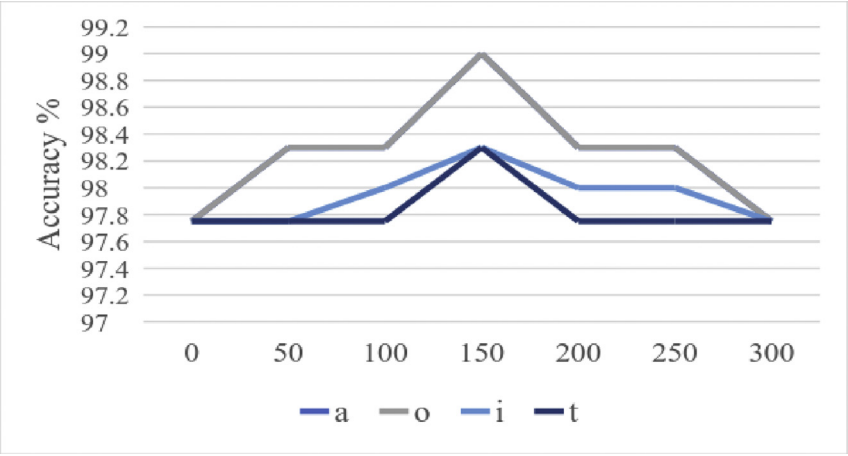

Fig. 9. Seed values and their accuracies with different no. of hidden layer node values.

Table 5
Optimum parameters of the proposed ANN model.

Parameters	Optimal Value Chosen
Hidden Layers	O
Seed	150
Learning Rate	0.0011

4.3.2. SVM optimized parameters

The SVM parameters were manipulated to identify the best combination of parameters. Two parameters were changed, and their accuracies were analyzed. The two parameters used were the kernel type and cost. The default kernel type used for SVM in WEKA was the radial basis function, and the default value of cost was one. The kernel type can include the following:

1. Radial basis function.
2. Linear.
3. Polynomial.
4. Sigmoid.

The cost can range from one to infinity. After changing the kernel type while holding the cost value at the default value, the plot shown in Fig. 11 was obtained. The highest accuracy was achieved with the linear kernel type with an accuracy of 96.75%.

The cost parameter of each kernel type was changed and varied within a range from 1 to 100, and the accuracies were analyzed. Fig. 12 shows the results obtained. The plots of the sigmoid and radial basis functions overlap as they have the same accuracy values. It was found that the highest accuracy (96.75%) was obtained when the cost was set to one with the linear kernel type.

Table 6 presents the optimal parameters of SVM and ANN to predict CKD according to the results of the accuracies achieved by changing the two parameters.

5. Results and discussion

In this section, we define and analyze the results obtained by investigating the effect of feature selection using the dataset, and the results were based on training/testing data partitioning using 10-fold cross-validation.

5.1. Results of investigating the effect of feature selection using the dataset

Using the correlation coefficients shown in Table 7, we can select the features as follows and compare the effects on the results. We used 10-fold cross validation at this stage to run the classifier while performing the feature selection because we do not know which partition

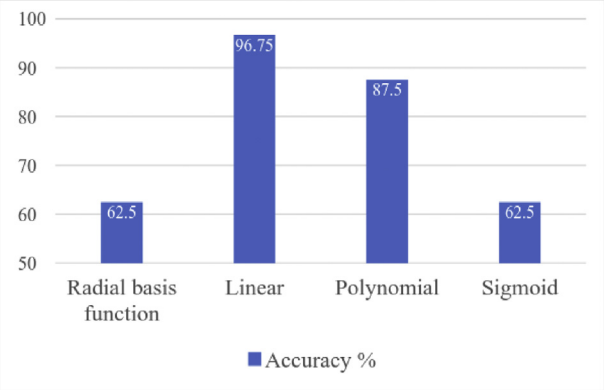

Fig. 11. SVM kernel types and their accuracies.

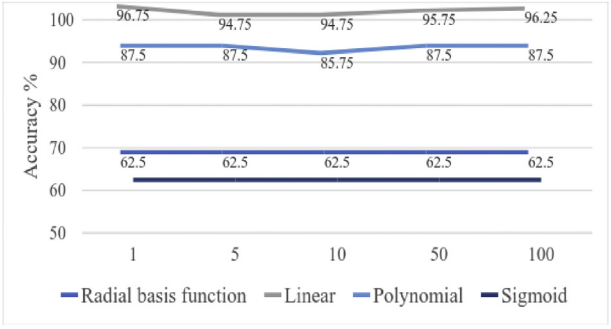

Fig. 12. Cost values varied for each kernel type and their accuracies.

Table 6
Optimum parameters of the proposed SVM and ANN model.

SVM	ANN
C = 1.0 Kernel = Ploy Kernel Epsilon = 1.0E-12 Calibrator = Logistic	Learning Rate - 0.3 Hidden Layers Momentum = 0.2

ratio is the best, and using 10-fold cross validation is the best option. To eliminate the feature, the following algorithm is used:

1. Build X named classifiers using V features (i.e., all features).
2. Compute the correlations between all features and the target (and then arrange in descending order) and choose the best V/2 features.
3. Repeat until one feature is left.
4. Choose the feature subset that exhibits the best performance (using cross-validation) and use this subset to build the final models.

As we decreased the number of features used in each technique, the time continues to decrease in both, and the time required to build the ANN model decreased more than the time required using SVM. Regarding accuracy, the accuracies of both techniques decreased as the number of features decreased. Initially, SVM seemed to have a more accurate result, but as we reached 2 features, stage ANN had much better accuracy. The experimental results above were used to develop the final model by considering the optimum parameters. The necessary graphs and table, such as a confusion matrix and ROC curve, were plotted. Table 8 shows the confusion matrix of the ANN model using the best 12 features from the previous steps. The matrix shows the number of correct predictions made by the model. As shown in Table 8, only one prediction was wrong.

5.2. ANN confusion matrix

Since our experiment is based on binary classification, the following equations can be used to evaluate the performance of the model:

1. Accuracy = TP + TN / total = (249 + 150) / 400 = 0.9975
2. Recall (Sensitivity) = TP / TP + FN = 249 + (249 + 1) = 0.996
3. Precision (Specificity) = TP / TP + FP = 249 / (249 + 0) = 1
4. F1 Score = 2 * (Recall * Precision) / (Recall + Precision) = 2 * (0.996 * 1) / (0.996 + 1) = 0.997

5.3. ANN ROC curve

Fig. 13 shows the ROC curve of the ANN model. The curve shows that the results are almost equal to one on the Y axis because the ROC value is 0.9999 for both classes. This value indicates that the classifier performs well.

Table 7
Results of subsets with different features.

	SVM				ANN			
	Time required to build a model (seconds)	Correctly classified instances	Incorrectly classified instances	Relative absolute error	Time required to build a model (seconds)	Correctly classified instances	Incorrectly classified instances	Relative absolute error
Using features: All 24 features	0.06	391 (97.75%)	9 (2.25%)	4.80%	3.16	391 (97.75%)	9 (2.25%)	4.41%
Using features: Best 12 features	0.05	394 (98.5%)	6 (1.5%)	3.20%	1.02	392 (98%)	8 (2%)	4.06%
Using features: Best 6 features	0.02	374 (93.5%)	26 (6.5%)	13.86%	0.4	388 (97%)	12 (3%)	9.01%
Using features: Best 3 features	0.01	373 (93.25%)	27 (6.75%)	14.39%	0.13	373 (93.25%)	27 (6.75%)	21.69%
Using features: Best 2 features	0.01	372 (93%)	28 (7%)	14.93%	0.13	376 (94%)	24 (6%)	21.57%
Using features: Best feature	0.01	354 (88.5%)	46 (11.5%)	24.52%	0.09	371 (92.75%)	29 (7.25%)	26.75%

Table 8
ANN confusion matrix.

	Predicted CKD	Predicted Not CKD
Actual CKD	249	1
Actual Not CKD	0	150

5.4. SVM confusion matrix

Table 9 shows the confusion matrix of the SVM model using the best 12 features from the previous steps. The matrix shows the number of correct predictions made by the model. Table 9 shows that 9 predictions were wrong.

Since our experiment is based on binary classification, the following equations can be used to evaluate the performance of the model:

1. Accuracy = TP + TN / total = (241 + 150) / 400 = 0.9775
2. Recall (Sensitivity) = TP / TP + FN = 241 / (241 + 9) = 0.964
3. Precision (Specificity) = TP / TP + FP = 241 / (241 + 0) = 1
4. F1 Score = 2 * (Recall * Precision) / (Recall + Precision) = 2 * (0.964 * 1) / (0.964 + 1) = 0.982

5.5. SVM ROC curve

Fig. 14 shows the ROC curve of the SVM model. We can observed that the line approaches its goal, which is to obtain the curve nearest to one on the Y-axis because the ROC value is 0.982 for both classes. This value indicates that the classifier almost performs well.

5.6. Further discussion

Based on the previous figures, tables, and results, clearly, ANN outperformed SVM in almost all models of this dataset. Although the ANN classifier had the highest accuracy, the runtime of the classifier is considerably longer compared with that of SVM. Table 10 summarizes the outcome of the experiment by presenting the accuracy and optimum parameters of each model. Table 10 shows that ANN's performance is better than SVM's performance by 2%.

To additionally compare the output of the experiment, the accuracies of the techniques used in this experiment are compared to the accuracies of the techniques used in the review of the related literature.

In the study performed by Ref. [17], an accuracy of 99.8% was achieved using the random forest algorithm. Their accuracy is very similar and close to the accuracy achieved in this work, which is 99.75%.

Regarding the SVM technique, the accuracy achieved was 99% in the study performed by Ref. [2], which is significantly better than the accuracy achieved in this work by using SVM (97.75%). In Ref. [14], the authors also used one of the methods mentioned in this paper, which is SVM. Their study was conducted based on a UCI repository with the goal of detecting CKD. However, their feature selection method was performed by implementing a ranking algorithm yielding an output of 10 features that were subsequently used for the prediction. However, the feature selection method in this paper used a recursive feature elimination algorithm that uses the correlation between the attributes and target variable to rank the features. Ultimately, the result of using SVM in Ref. [14] yielded the same results of 97.75% accuracy. In another paper [16], the authors also applied SVM to a UCI repository for the prediction of CKD. The features were selected using wrapper and filter techniques, and each technique was applied using two different evaluators. The study concluded that a filtered subset evaluator with the best first search engine technique to select features from the dataset was the most accurate technique with an accuracy of 98.5%. By comparing our paper to Ref. [16], we can note that the latter yielded a higher accuracy with a difference of 0.75%, which could be due to the differences in the feature selection approaches.

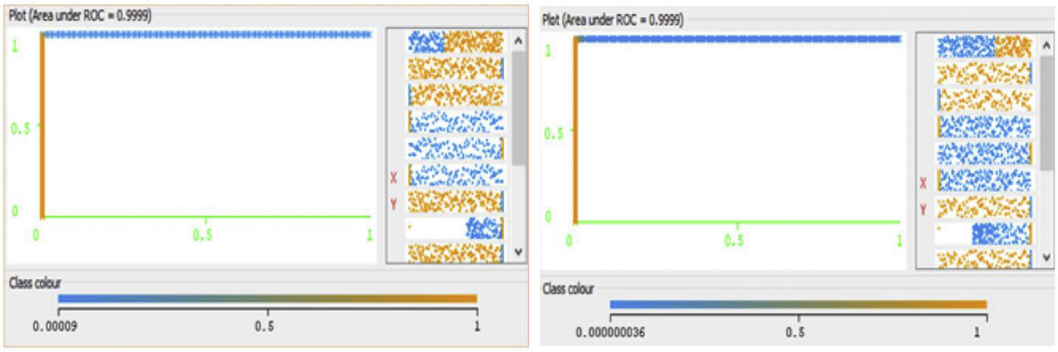

Fig. 13. ANN ROC curve illustrating the true positive rate against the false positive rate.

Table 9
SVM confusion matrix.

	Predicted CKD	Predicted Not CKD
Actual CKD	241	9
Actual Not CKD	0	150

Table 10
Comparison of the ANN and SVM models.

Classifier	No. of features	Best optimum parameters	Accuracy
ANN	12	Hidden layer: a Learning Rate: 0.3 momentum: 0.2	99.75%
SVM	12	Kernel type: linear Cost: 1.0	97.75%

In Ref. [2], the authors used three different models, and the highest accuracy in predicting CKD in this study was achieved by the SVM model, which reached 99% accuracy by addressing the noise disturbance in the composite dataset. This model not only reached the highest accuracy but also obtained the highest score in other evaluation factors, such as specificity (97.6%) and the F1 score (99.7%). Compared to our paper, in Ref. [2], the authors achieved higher accuracy and sensitivity used SVM than achieved in this paper. In this paper, we reached an accuracy of 97.75%, sensitivity of 96%, and F1 score of 98%. In contrast, we achieved better specificity (100%) in our work.

In a research study [11], the C4.5 decision tree algorithm was applied to the UCI repository dataset to diagnose CRF in patients. After applying the algorithm, eight if-then rules were developed to predict CRF. The accuracy of the algorithm was found to be 98.25%, which is higher than the accuracy achieved in a study performed by Al-Taee, Al-Hyari, & Al-Taee [10], who use the same dataset. However, the results obtained applying ANN in our study yielded a higher accuracy (99.75%). In contrast, applying SVM to our dataset resulted in a lower accuracy of 97.75%.

6. Conclusion

Support Vector Machine (SVM) and Artificial Neural Network (ANN) are two of the most known machine learning techniques. Both techniques have advantages and have been proven to perform excellently in several fields. ANN has been proposed as a new model to better predict CKD and has been compared against SVM, which has exhibited the highest accuracy in previous studies. The dataset was first preprocessed, and the missing values were replaced. A 10-fold cross-validation was used as the algorithm for the partitioning of the training and test datasets with the ratio (90: 10). The optimized parameters for Support Vector Machine and Artificial Neural Network were identified. Several experiments were performed using different values of the parameters for both techniques. It was found that ANN performs better with an accuracy of 99.75% using the optimized features, while SVM had a performance accuracy of 97.75%.

7. Recommendations

Despite the promising contribution presented in this paper, the utilization of deep learning techniques for CKD prediction remains open for further studies and investigations. Deep learning certainly remains significant and represents a very important technique currently explored for a host of machine learning based solutions; among these techniques, the convolutional neural network (CNN) remains at the forefront [27,28,30,33–35]. However, notably, deep learning concepts are commonly used because they are mostly superior and cost-effective for image-oriented feature problems or other areas in which manual or preprocessing based standard feature extraction is mostly a concern and could lead to poor accuracy of the classifiers [29–35]. However, in the

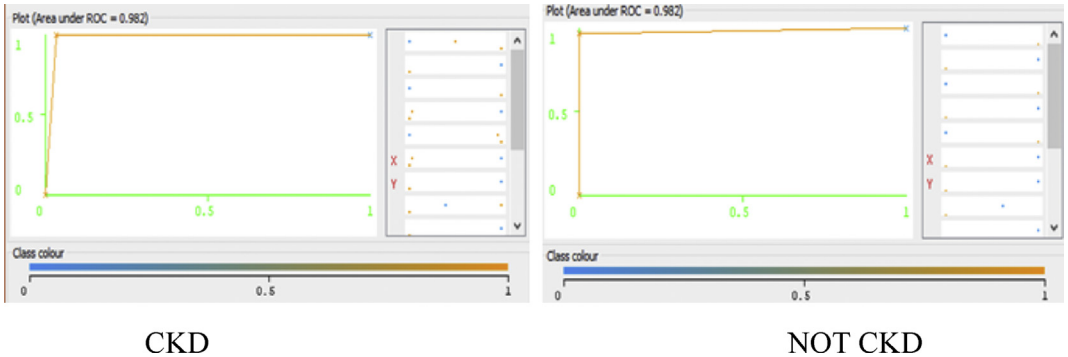

Fig. 14. SVM ROC curve illustrating the true positive rate against the false positive rate.

present application, we used ready-made features, thereby rendering the need to use deep learning as the starting point a slightly less necessary at the beginning of this study because one of the principles of machine learning is to start with simple classifiers and then gradually proceed with complex classifiers depending on the need [36,37]. Notwithstanding, future works should explore how deep learning compares to this particular work using the same dataset.

Furthermore, notably, we aim to use deep learning systems in some of our future studies, including those in this field, especially to explore image-based features related to schizophrenia diagnosis using brain images, brain cancer imaging classification for early diagnosis, and a host of other interesting and important medical applications. Further recommended future works include the following: investigating the use of deep learning in oncology radiation treatment planning, the effective and early diagnosis of breast cancers, using other techniques or an intelligent mixture of classifiers (ensemble) along with several types of attributes to increase the accuracy rate for medical prediction applications, and a host of other medical conditions, especially those in which early and accurate diagnosis are critical to facilitate survival and an early cure. Certainly, this type of recommended future work has great potential for drastically improving patients' quality of life.

Conflict of interest statement

The authors declared that there is no conflict of interest regarding this paper.

Acknowledgments

Imam Abdulrahman Bin Faisal University is acknowledged for some of the facilities utilized in this research.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2019.04.017.

References

[1] A. a Al-Sayyari, F. a. Shaheen, End-stage chronic kidney disease in Saudi Arabia. A rapidly changing scene, Saudi Med. J. 32 (2011) 339–346.
[2] Z. Chen, X. Zhang, Z. Zhang, Clinical risk assessment of patients with chronic kidney disease by using clinical data and multivariate models, Int. Urol. Nephrol. 48 (12) (2016) 2069–2075 http://doi.org/10.1007/s11255-016-1346-4.
[3] R.G. Brereton, G.R. Lloyd, Support vector machines for classification and regression, Analyst 135 (2) (2010) 230–267 http://doi.org/10.1039/B918972F.
[4] A. Survey, Importance of artificial neural networks for location of faults in transmission systems, Surveyor (2017) 357–362.
[5] T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler, Support vector machine classification and validation of cancer tissue samples using microarray expression data, 16 (10) (2000) 906–914.
[6] A.A.A.A. Adewumi, T.O. Owolabi, I.O.I.O. Alade, S.O.S.O. Olatunji, Estimation of physical, mechanical and hydrological properties of permeable concrete using computational intelligence approach, Appl. Soft Comput. 42 (2016) 342–350 http://doi.org/10.1016/j.asoc.2016.02.009.
[7] K.O. Akande, S.O. Olatunji, T.O. Owolabi, A. AbdulRaheem, Feature Selection-Based ANN for Improved Characterization of Carbonate Reservoir, CPAPER, Society of Petroleum Engineers, 2015, http://doi.org/10.2118/178029-MS.
[8] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 297 (1995) 273–297.
[9] M.M. Luck, A. Yartseva, G. Bertho, E. Thervet, P. Beaune, N. Pallet, Metabolic profiling of H-1 NMR spectra in Chronic Kidney Disease with local predictive modeling, 2015 IEEE 14th International Conference on Machine Learning and Applications (Icmla), 2015, pp. 176–181 http://doi.org/10.1109/icmla.2015.155.
[10] A.M. Al-Taee, A.Y. Al-Hyari, M.A. Al-Taee, Clinical decision support system for diagnosis and management of chronic renal failure, 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2013.
[11] A.A. Serpen, Diagnosis rule extraction from patient data for chronic kidney disease using machine learning, Int. J. Biomed. Clin. Eng. 5 (2) (2016) 64–72 http://doi.org/10.4018/IJBCE.2016070105.
[12] R. Ani, G. Sasi, R.S. U, O.S. Deepa, Decision support system for diagnosis and prediction of chronic renal failure using random subspace classification, 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 1287–1292 http://doi.org/10.1109/ICACCI.2016.7732224.
[13] Z. Chen, Z. Zhang, R. Zhu, Y. Xiang, P.B. Harrington, Diagnosis of patients with chronic kidney disease by using two fuzzy classifiers, Chemometr. Intell. Lab. Syst. 153 (2016) 140–145 http://doi.org/10.1016/j.chemolab.2016.03.004.
[14] N. Tazin, S.A. Sabab, M.T. Chowdhury, Diagnosis of Chronic Kidney Disease Using Effective Classification and Feature Selection Technique, (2013) http://doi.org/10.1109/MEDITEC.2016.7835365.
[15] S.B. Akben, Early stage chronic kidney disease diagnosis by applying data mining methods to urinalysis , blood analysis and disease history, IRBM 39 (5) (2018) 353–358 https://doi.org/10.1016/j.irbm.2018.09.004.
[16] H. Polat, H. Danaei Mehr, A. Cetin, Diagnosis of chronic kidney disease based on support vector machine by feature selection methods, J. Med. Syst. 41 (4) (2017) 55 http://doi.org/10.1007/s10916-017-0703-x.
[17] A. Salekin, J. Stankovic, Detection of Chronic Kidney Disease and Selecting Important Predictive Attributes, (2016) http://doi.org/10.1109/ICHI.2016.36.
[18] M. Bhattacharya, C. Jurkovitz, H. Shatkay, Chronic Kidney Disease Stratification Using Office Visit Records : Handling Data Imbalance via Hierarchical Meta-Classification vol. 18, (2018).
[19] DACS, Artificial Neural Networks Technology, (2014).
[20] A. Krogh, What are artificial neural networks? Nat. Biotechnol. 26 (2) (2008) 195–197 http://doi.org/10.1038/nbt1386.
[21] M. Kamuda, J. Stinnett, C. Sullivan, Automated isotope identification algorithm using artificial neural networks, IEEE Trans. Nucl. Sci. 9499 (c) (2017) 1–1 http://doi.org/10.1109/TNS.2017.2693152.
[22] I.N. da Silva, D. Hernane Spatti, R. Andrade Flauzino, L.H.B. Liboni, S.F. dos Reis Alves, Artificial Neural Networks vols. 21–29, (2017) http://doi.org/10.1007/978-3-319-43162-8.
[23] W.S. Noble, What is a support vector machine? Nat. Biotechnol. 24 (12) (2006) 1565–1568 http://doi.org/10.1038/nbt1206-1565.
[24] S. Canu, Y. Grandvalet, V. Guigue, A. Rakotomamonjy, SVM and Kernel Methods Matlab Toolbox, (2008).
[25] L.-P. Ni, Z.-W. Ni, Y.-Z. Gao, Stock trend prediction based on fractal feature selection and support vector machine, Expert Syst. Appl. 38 (5) (2011) 5569–5576 Journal Article. Retrieved from http://www.sciencedirect.com/science/article/B6V03-51F7PMS-B/2/f3645bc7144b2047233ac753849dccce.
[26] D. Dua, C. Graff, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2019http://archive.ics.uci.edu/ml.
[27] Yanming Guo, Yu Liu, Oerlemans Ard, Songyang Lao, Song Wu, S. Michael, Lew; Deep learning for visual understanding: a review, Neurocomputing 187 (2016) 27–48.
[28] T.J. Brinker, A. Hekler, A.H. Enk, et al., A convolutional neural network trained with dermoscopic images performed on par with 145 dermatologists in a clinical melanoma image classification task, Eur. J. Cancer 111 (2019) 148–154, https://doi.org/10.1016/J.EJCA.2019.02.005.
[29] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: a review, Neurocomputing 187 (2016) 27–48, https://doi.org/10.1016/j.neucom.2015.09.116.
[30] E. Cetinic, T. Lipic, S. Grgic, Fine-tuning convolutional neural networks for fine art classification, Expert Syst. Appl. 114 (2018) 107–118, https://doi.org/10.1016/J.ESWA.2018.07.026.
[31] H.-I. Suk, An introduction to neural networks and deep learning, Deep Learn Med Image Anal. January (2017) 3–24, https://doi.org/10.1016/B978-0-12-810408-8.00002-X.
[32] R. Raoui-Outach, C. Million-Rousseau, A. Benoit, P. Lambert, Deep Learning for Automatic Sale Receipt Understanding, https://www.abbyy.com/en-eu/ , Accessed date: 27 March 2019.
[33] Y. Zhang, C. Zhou, F. Chang, A.C. Kot, Multi-resolution attention convolutional neural network for crowd counting, Neurocomputing 329 (2019) 144–152, https://doi.org/10.1016/J.NEUCOM.2018.10.058.
[34] P. Ding, Y. Zhang, W.-J. Deng, P. Jia, A. Kuijper, A light and faster regional convolutional neural network for object detection in optical remote sensing images, ISPRS J. Photogrammetry Remote Sens. 141 (2018) 208–218, https://doi.org/10.1016/J.ISPRSJPRS.2018.05.005.
[35] S. Shichijo, S. Nomura, K. Aoyama, et al., Application of convolutional neural networks in the diagnosis of Helicobacter pylori infection based on endoscopic images, EBioMedicine 25 (2017) 106–111, https://doi.org/10.1016/J.EBIOM.2017.10.014.
[36] R.C. Holte, Very simple classification rules perform well on most commonly used datasets, Mach. Learn. 11 (1993) 63–91 https://www.mlpack.org/papers/ds.pdf , Accessed date: 2 April 2019.
[37] Edwin Chen, Choosing a Machine Learning Classifier. (Retrieved from):.