Journal of Applied Research and Technology

www.jart.icat.unam.mx

Instituto de Ciencias Aplicadas y Tecnología

Journal of Applied Research and Technology 24 (2026) 211-224

Original

Optimized Machine Learning Models for Early Prediction of Kidney Failure

Araek Tashkandi[1]*

[1]*College of Computer Science and Engineering, Department of Information Systems and Technology, University of Jeddah, Jeddah, Saudi Arabia.*

Received: 31 12 2024; Accepted: 03 06 2025
Available: 30 04 2026

Abstract: Kidney failure's poor prognosis, prevalence, and progression to end-stage renal disease pose serious health risks and healthcare burdens, yet early detection and treatment save lives and decrease the demand on the healthcare system. On average, a person may survive only 18 days without functioning kidneys. Kidney failure can be predicted using machine learning approaches, and the optimal algorithms for accurate diagnosis can be identified by using clinical data to identify relevant variables. Our research uses clinical data to help medical professionals anticipate kidney failure using machine learning techniques. Using variables such as creatinine, diabetes, and hypertension, we employ machine learning to predict kidney disease and select the most informative parameters for precise prediction. To predict kidney failure from multiple parameters, various algorithms were used to develop predictive models. The most effective model was identified among Random Forest, decision trees, support vector machines, K Nearest Neighbor, and logistic regression. The modified Random Forest model outperformed the default model with an F1-measure of 99% and an exceptional prediction accuracy of 99.75%, according to the results. In the end, this will help doctors save lives by enabling early diagnosis of kidney disease, allowing early treatment, and proper monitoring of patients' health before dialysis is necessary.

Keywords: **Chronic Kidney Disease; Machine Learning; Kidney Failure; CKD prediction.**

*Corresponding author.
E-mail address: astashkandi@uj.edu.sa (A. Tashkandi).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

Kidney failure occurs when the kidneys are unable to remove excess water and waste from the blood, resulting in an imbalance of chemicals in the body. Early detection and treatment can lead to a cure, while chronic kidney failure can progress over time due to conditions like diabetes or high blood pressure. This can eventually lead to end-stage Chronic Kidney Disease (CKD), requiring dialysis or a kidney transplant.

The Glomerular Filtration Rate (GFR) test is used to determine kidney function by measuring how much blood passes through the tiny filters in the kidneys per minute. Decreases in GFR can indicate the severity of renal disease, even before symptoms appear. GFR is crucial for diagnosing and treating both chronic and acute kidney failure, as it helps evaluate the kidney's ability to filter blood and remove waste products (Levey et al., 2002).

Countries prioritize patient data given the importance of health and the need to increase health awareness. Kidney failure is a common complication among hospitalized patients, necessitating predictive methods. Utilizing IoT data from healthcare devices on a cloud platform allows for more efficient medical services. Kidney failure, a severe condition, can be caused by diseases or impaired kidney function.

Late detection of kidney failure is a major challenge in the medical field, leading to limited options like dialysis or transplants. This can result in 11% of hospital deaths due to failure to promptly recognize deteriorating patients. Early prediction of kidney failure is crucial for providing timely treatment and potentially saving lives through interventions like dialysis or transplantation. Ultimately, early predictions of kidney failure can significantly impact patient outcomes and treatment strategies. Research is being conducted to predict kidney failure using Machine Learning (ML) algorithms, with the aim of improving accuracy and identifying key influencing factors. The process involves stages such as data collection, pretreatment, and classification of medical data to facilitate prediction.

To improve outcomes in this challenging condition, healthcare providers may be able to better identify and treat patients at risk of acute renal injury early by using data mining and machine learning algorithms. By analyzing medical data, machine learning (ML) helps predict CKD and identify early symptoms of the condition. ML models are trained on massive datasets that include clinical data about patients, such as blood pressure, blood sugar, serum creatinine, glomerular filtration rate (GFR), and medical history. These characteristics help the model identify trends related to the onset or progression of CKD. Traditional statistical techniques might overlook subtle, nonlinear interactions in data, but machine learning can identify them. This aids in the early detection of CKD, the prediction of CKD risk in high-risk individuals, and the prompt intervention to stop additional kidney damage. Support Vector Machines (SVMs), Random Forests, Decision Trees, Neural Networks, and Logistic Regression are common machine learning methods for predicting chronic kidney disease. (Delrue et al., 2024; Sanmarchi et al., 2023; Islam et al., 2023).

The process for predicting chronic kidney disease using different machine learning models is shown in Figure 1. Gathering relevant medical data from patient records or publicly available datasets is the first step in predicting CKD. In order to prepare the data for analysis, preprocessing is done after it has been gathered. This includes addressing missing values, encoding category variables, and normalizing numerical features. Following preprocessing, feature selection is performed to identify the critical factors influencing CKD prediction. The next step is model selection, in which machine learning algorithms are selected based on their likely efficacy. Examples of these algorithms are Decision Tree, Random Forest, and Support Vector Machine. After that, model training is performed on the selected models using the prepared dataset. In the model evaluation phase, metrics such as accuracy, precision, recall, and F1 Score are used to assess the models' performance after training. This evaluation identifies the best-performing model by analyzing the outputs of various methods. To help with new-patient CKD prediction, the best model is then transferred to the deployment stage and integrated into a clinical decision support system or application.

The research paper aims to use machine learning to predict kidney disease by analyzing parameters such as creatinine, diabetes, and hypertension, and to select the most informative parameters for accurate prediction. Various algorithms will be applied to create predictive models and determine the most efficient one based on multiple criteria. This will help doctors predict kidney failure early, enabling effective treatment and monitoring of patients' health.

2. Literature Review

Research shows that machine learning models effectively predict and diagnose serious illnesses, including chronic kidney disease CKD. Recent studies highlight the Potential for these models to facilitate early detection and improve treatment options for CKD is garnering increasing attention from researchers, analysts, and doctors.

Ahmad et al. (2017) used SVM to detect chronic kidney failure. The classification model included 5 stages: data collection, preparation, grouping, classification, and rules extraction. Data was gathered from the UCI Machine Learning dataset by Dr. P. Soundarapandian, with 400 patients and 25 parameters. Only 5 parameters were selected for classification. The system, based on the R language, achieved 98.34% accuracy in predicting chronic kidney disease. The methodology involved classification modeling and system development based on extracted rules.

Al-Hyari et al. (2013) used a decision tree to diagnose Chronic Kidney Failure, based on data collected from 102 patients at Prince Hamza Hospital. Fifteen parameters, such as age, weight, and blood pressure, are used for training and classification. The DT calculates if the patient has CKD and determines the disease stage using GFR with the Cockcroft-Gault formula, known for lower accuracy than MDRD and CKD-EPI formulas.

Salekin and Stankovic (2016) evaluated three classifiers (k-nearest neighbors, random forest, neural networks) using 24 predictive parameters to develop a machine-learning classifier for CKD detection. The approach was tested on a dataset of 400 individuals, with 250 having CKD, and a random forest achieved a detection accuracy of 0.993, F1-score, and 0.1084 root mean square error. New predictive attributes were identified but not used in the GFR estimator equations.

Revathy et al. (2019) conducted research comparing algorithms for detecting kidney failure using traditional data mining methods. Large CKD datasets were collected, prepared, and preprocessed using traditional data mining methods. Three machine learning algorithms - Decision tree, Random Forest, and Support Vector machines were used to predict early CKD occurrence. The models were trained and validated with input parameters from CKD patients. Random Forest Classifier showed the highest accuracy of 99.16%, outperforming Decision trees and Support Vector machines. The results indicate that Random Forest is the best classifier for predicting CKD, achieving the highest accuracy among the models.

The study by Guldogan and Kucukacali (2021) aimed to classify chronic kidney failure using tree-based methods. The study used decision trees (J48), Random Forest, and Gradient Boosted Trees to create classification models for the ”Chronic Kidney Disease” data set, comprising 400 patients, 62.5% of whom had chronic kidney failure. Columns such as Blood Pressure, Albumin, and Serum Creatinine were selected as factors associated with kidney failure. The Random Forest model achieved the best classification metrics among the four methods. Decision trees had an accuracy of 96.25%, J48 had 97.75%, Random Forest had 99.25%, and Gradient Boosted Trees had 98.00% accuracy. The focus was on the accuracy of Decision Trees and Random Forest, with Random Forest achieving the highest accuracy of 99.25%. The study demonstrated the effectiveness of tree-based methods for classifying chronic kidney failure, with Random Forest the most accurate method in this context.

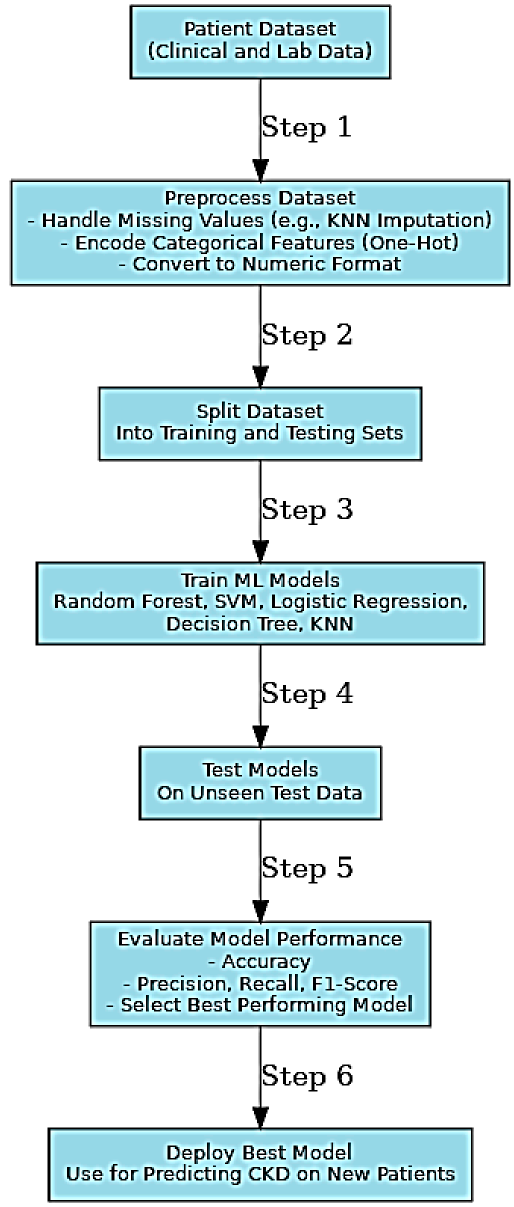

Figure 1. Chronic Kidney Disease (CKD) Prediction Process.

The study by De Almeida et al. (2020) focused on using machine learning techniques like Decision Trees, Random Forest, and Support Vector Machine for early kidney failure detection to aid health professionals. Cross-validation with K-Fold, confusion matrix, and classification report were used to evaluate the algorithms due to their precision and wide applicability. Decision trees showed the highest accuracy of 87%.

Hamida Ilyas et al. (2021) conducted a study using Random Forest and J48 algorithms to develop a model for accurately detecting different stages of CKD. The stages ranged from normal kidney function to end-stage kidney failure. Results showed that J48 achieved higher predictive accuracy across all stages than Random Forest, with 85.5% accuracy in detecting CKD. J48 also outperformed Random Forest in terms of speed, with an accuracy of 85.5% in 0.03 seconds compared to 78.25% in 0.28 seconds.

In their study, Polat et al. (2017) used a Support Vector Machine classification algorithm to diagnose CKD. Two types of feature selection methods, wrapper and filter approaches, were employed to reduce the dataset dimension. The wrapper approach included a classifier subset evaluator with a greedy stepwise search engine and a wrapper subset evaluator with the Best First search engine. The filter approach used a correlation feature selection subset evaluator with a greedy stepwise search engine and a filtered subset evaluator with the Best First search engine. The results indicated that the Support Vector Machine classifier using the filtered subset evaluator with the Best First search engine feature selection method achieved a higher accuracy of 98.5% for diagnosing CKD compared to other methods.

Saif et al. (2024) used deep learning and ensemble learning techniques to create a CKD prediction framework. Despite the need for early CKD prediction. The study addresses data imbalance, feature selection, optimizer optimization, and creates deep learning models for 6–12-month CKD prediction. The ensemble model combines the best individual models, achieving 98% and 97% accuracy for 6 and 12 months, respectively.

Su et al. (2022) evaluated five machine learning classifiers, Logistic Regression, Random Forests, XGBoost, Support Vector Machines, and Gaussian Naive Bayes, to predict the course of pre-dialysis CKD in 858 patients at a Taiwanese veteran's hospital. The random forest classifier with the synthetic minority over-sampling method (SMOTE) had the best predictive performance for early-stage CKD patients who advanced over three to five years and for advanced-stage CKD patients who advanced within one to three years. Urine and serum creatinine levels were found to be significant indicators of CKD progression.

Alturki et al. (2024) investigated the use of machine learning algorithms to detect CKD early. Using a CKD dataset from the University of California, UC Irvine, the study presents TrioNet, an ensemble model that combines random forest, tree classifier, and extreme gradient boosting. The model performs exceptionally well at predicting CKD, handling class imbalance with synthetic minority oversampling (SMOTE) and missing values with the KNN imputer. The model outperforms other models with 98.97% accuracy in CKD detection, demonstrating its potential for early diagnosis.

Praveen et al. (2024) used a machine learning technique called the Neuro-Fuzzy model to predict CKD patients, based on image processing techniques. The model can identify fibrosis proportions in renal tissues and predict the risk of CKD patients. The ML-based Neuro-Fuzzy logic method achieved 97% accuracy in CKD prediction, compared with traditional methods such as Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). This approach can be assessed using metrics such as precision, accuracy, recall, and F1 Score, enabling the identification of patients at high risk of developing chronic diseases.

A literature review was conducted on the worldwide health crisis of kidney disease, also known as kidney failure. Research focused on machine learning algorithms for predicting kidney failure, including decision trees, Support Vector Machines, Random Forests, J48, Logistic regression, Naive Bayes, k-Nearest Neighbors, and Gradient Boosting. Specifically, this literature review and a recent review of ML models used for CKD prediction (Delrue et al., 2024; Sanmarchi et al., 2023) found that Support Vector Machines, Random Forest, and Decision Trees were the most successful in accurately forecasting kidney failure. Thus, in this study, we employ those renal failure prediction models and evaluate them to determine which is the most reliable.Our study's achieved accuracy in predicting kidney failure was 99.75%. Although Revathy et al. (2019) used a smaller dataset than we do (their dataset only included 120 patients, whereas our dataset included 400 patients), they were nevertheless able to reach 99.16% accuracy. We employ the same dataset as Ahmad et al. (2017) and Guldogan and Kucukacali (2021); however, our accuracy is 0.41% higher than that of Ahmad et al. (2017) and 0.50% higher than that of Guldogan and Kucukacali (2021).

Table 1. Comparative Accuracy analysis of the diagnosis/prediction of Kidney Failure from the literature.

Paper	Used Algorithms	Best Performance Model	Accuracy
(Ahmad et al., 2017)	Support vector machine	Support Vector Machine	98.34%
(Al-Hyari et al., 2013)	Artificial Neural Networks (ANNs), Naïve Bayes and Decision Tree	Decision Tree	92.2%
(Salekin & Stankovic, 2016)	K-Nearest Neighbors, Random Forest, and Neural Network	Random Forest	.993 F1-measure and 0.107 RMSE
(Revathy et al., 2019)	Decision Tree, Support vector machine, and Random Forest	Random Forest	99.16%
(Guldogan & Kucukacali, 2021)	Decision trees, J48, Random Forest, and Gradient Boosted Trees	Random Forest	99.25%
(De Almeida et al., 2020)	Decision trees, Random Forest and Support Vector Machine	Decision Tree	87%
(Ilyas et al., 2021)	J48 (C4.5) Decision Tree , and Random Forest	J48 (C4.5) Decision Tree	85.5%
(Polat et al., 2017)	Support Vector Machine	Support Vector Machine	98.5%
(Su et al., 2022)	Logistic Regression, Random Forests, XGBoost, Support Vector Machines , and Gaussian Naive Bayes	Random Forest	93%
(Saif et al., 2024)	Convolutional neural network, long short-term memory (LSTM), and LSTM-BLSTM	LSTM-BLSTM	98%
(Alturki et al., 2024)	TrioNet: ensemble model combining extreme gradient boosting, random forest, and extra tree classifier	TrioNet	98.97%
(Praveen et al., 2024)	Neuro-Fuzzy , Support Vector Machine (SVM) and K-Nearest Neighbor (KNN).	Neuro-Fuzzy	97%
This research paper	Decision tree, Support Vector Machine, Random Forest, Logistic regression, k-nearest neighbors.	Random Forest	99.75%

3. Dataset and Attributes

The dataset titled "Chronic Kidney Disease" (Rubini et al., 2015) was collected from an Indian hospital in 2015 and is valuable for kidney failure research. It is the only kidney failure dataset available online, thus, it is used in various studies to predict chronic kidney disease using different machine learning algorithms as mentioned in literature review (Section 2).

The dataset from India includes 24 real attributes such as blood cell counts, blood pressure, and hypertension, collected over 2 months. It provides detailed laboratory test results. It contains 24 attributes and the class which in total 25 (11 numeric ,14 nominal); which are described in Table 2. 62.5% of the dataset: 250 Chronic Kidney Disease patients; 150 which is 37.5% non-CKD patients out of 400.

Across all numerical and categorical attributes, the dataset contains approximately 421 outliers and 1009 missing values. There are missing values in the data, especially in some features. One option is to drop them, but it is preferred to use an imputation technique to preserve distributions. KNNImputer from sklearn is chosen for imputation. Before imputation, categorical features are encoded using One-Hot Encoding to maintain the same dimensionality as they are binary. This approach is suitable for the dataset's characteristics. We evaluate the model prediction value on the dataset with and without KNN Imputation in order to determine the impact of KNN imputation for missing values on model prediction accuracy, as presented in Figure 2. Without KNN Imputation (removing rows that contain missing values): Because rows with missing data were removed, certain models (such as Random Forest and Logistic Regression) achieved 100% accuracy, but only on a smaller and somewhat biased sample. If the test set doesn't accurately represent missing in the real world, this could result in overfitting or misleading performance. Nevertheless, KNN Imputation Models retain more information because they are trained on a larger, more comprehensive dataset. Even with k = 3, accuracy remained extremely high and was more consistent among models. To determine the optimal k, we evaluate the KNN imputation using a range of k values (3, 5, 7, and 10) as presented in Figure2. Since it continuously produced high accuracy across a variety of models—particularly for Random Forest, Decision Tree, and KNN classifiers—we determine that the best k value for KNN imputation is k = 3. It seems that k=3 finds a balance between avoiding overfitting that could result from smaller k values and capturing local data patterns. In conclusion, by replacing missing values without eliminating data, KNN Imputation with k = 3 provides a strong, dependable enhancement in predictive modeling. Despite its great accuracy, "No Imputation" might not generalize well because of the bias in smaller, cleaner samples. Consequently, we use the KNN imputation.

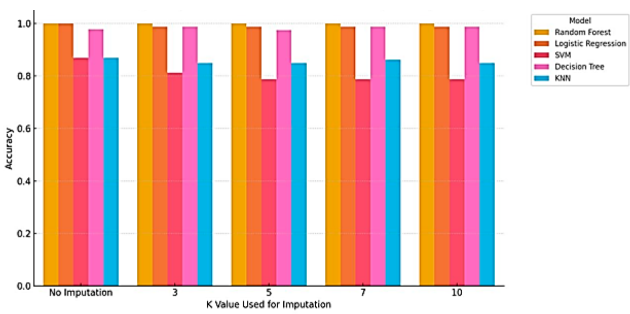

Figure 2. Model Accuracy comparison without KNN Imputation and with KNN Imputation Across Different K Values.

The dataset analysis revealed the following: the age range is from 2 to 90 years, with an average of 51.59 years. The minimum blood pressure is 50 mm Hg, which is too low, and the maximum is 180 mm Hg, indicating hypertension. The normal specific gravity values range from 1.010 to 1.030, with an average of 0.017449. Abnormal levels may indicate kidney disease. Normal albumin levels range from 3.5 to 5.5 g/dL, with values in the dataset ranging from 0.0 to 5.0.

High blood sugar can lead to kidney problems. Blood urea levels in the dataset range from 1.5 to 391, with higher levels observed in older individuals. Serum creatinine levels are essential for monitoring kidney disease but are gender-dependent. Normal sodium levels range from 135 to 145, with one outlier in the dataset at 4.5, possibly erroneous data. The Blood Glucose Random attribute's values vary in the dataset, making it challenging to determine significance without additional context.

To determine feature importance, we examined the relationship between the medical dataset attributes and the prediction of chronic kidney disease. The correlation coefficients between different medical attributes (or biomarkers) associated with chronic kidney disease are shown in Figure 3, whereas White/Neutral color (around 0) indicates little to no linear correlation between features, red color (closer to +1) indicates high positive correlation between features, while Blue color (closer to -1) indicates strong negative correlation. Results showed a high positive correlation between hypertension and kidney failure (0.59), diabetes mellitus and kidney failure (0.56), and Albumin and kidney failure (0.55). The microalbumin test can detect early signs of kidney damage. Additionally, there was a weak positive correlation between Pus Cell Clumps and kidney failure (r = 0.27) and between age and kidney failure (r = 0.23). Moreover, high negative correlation between blood urea and CKD (-0.49). Major findings are the features associated with CKD: Blood urea (0.58) and serum creatinine (0.77) showed positive correlations with CKD, as predicted. Red blood cells (-0.55), packed cell volume (-0.65), and hemoglobin (-0.64): they are frequently negatively associated in CKD. Diabetes Mellitus (0.40) and hypertension (0.58) are highly correlated with CKD. Figure4 illustrates how each feature correlates to chronic kidney disease. There is a strong positive correlation between the top features (e.g., serum creatinine, hypertension: yes); higher levels are associated with CKD. Whereas there is a strong negative correlation between features at the bottom (e.g., Red Blood Cells: Normal, Hemoglobin) and CKD; lower or abnormal values are associated with CKD. In summary, as Figure 4 illustrates, the most significant features are those pertaining to blood health (hemoglobin, red blood cells) and kidney function (serum creatinine, blood urea). Chronic kidney disease has a negative correlation with indicators of a healthy blood composition and a positive correlation with indicators of impaired kidney function. Given their moderate to strong positive associations, diabetes and hypertension are also major causes of CKD.

Table 2. The list of 24 Attributes that contain quality data involved in the present work.

Attribute	Description	Type	Information
Age	The patient's age	numerical	age in years
Blood Pressure	A measure of Heart force used to pump blood	numerical	mm/Hg
specific gravity	ratio of the substance density to the standard	nominal	(1.005,1.010,1.015, 1.020,1.025)
albumin	Numerous heat-coagulable water-soluble proteins in blood or muscle	nominal	(0,1,2,3,4,5)
sugar	The main sugar in blood glucose	nominal	(0,1,2,3,4,5)
red blood cells	Hemoglobin protein transports oxygen from the lungs to the body	binary	(normal =1, abnormal= 0)
pus cell	Neutrophil white blood cell in pus	binary	(normal=1, abnormal = 0)
pus cell clumps	Long-standing infection	binary	(present = 1, not present =0)
bacteria	Microorganisms can be free-living or dependent parasites	binary	(present =1, not present =0)
Blood glucose random	Measure glucose or sugar levels in blood	numerical	mgs/dl
blood urea	Nitrogen from urea, breakdown of protein in the liver, in the blood	numerical	mgs/dl
serum creatinine	Muscle waste product from regular daily activities	numerical	mgs/dl
sodium	Main positive ion in the body's cell fluid	numerical	mEq/L
potassium	An important metallic element in the body for blood pressure regulation.	numerical	mEq/L
hemoglobin	Protein in red blood cells transports oxygen from lungs to body organs	numerical	gms
packed cell volume	Percentage of red blood cells in total blood volume	numerical	value
white blood cell count	Counts white blood cells	numerical	cells/cumm
red blood cell count	blood test to count red blood cells	numerical	millions/cmm
hypertension	Blood vessels have raised pressure	binary	(yes=1, no =0)
diabetes mellitus	Group of diseases affecting body's blood glucose usage	binary	(yes=1, no =0)
coronary artery disease	caused by plaque buildup in the wall of the arteries that supply blood to the heart	binary	(yes=1, no=0)
appetite	Craving for food triggered by sensory cues	binary	(poor =1, good =0)
pedal edema	Fluid accumulation in feet, legs	binary	(yes=1, no=0)
anemia	Not enough red blood cells lead to insufficient oxygen	binary	(yes=1, no=0)
Chronic Kidney Disease (CKD)	kidney failure	binary	(CKD =1, notCKD=0)

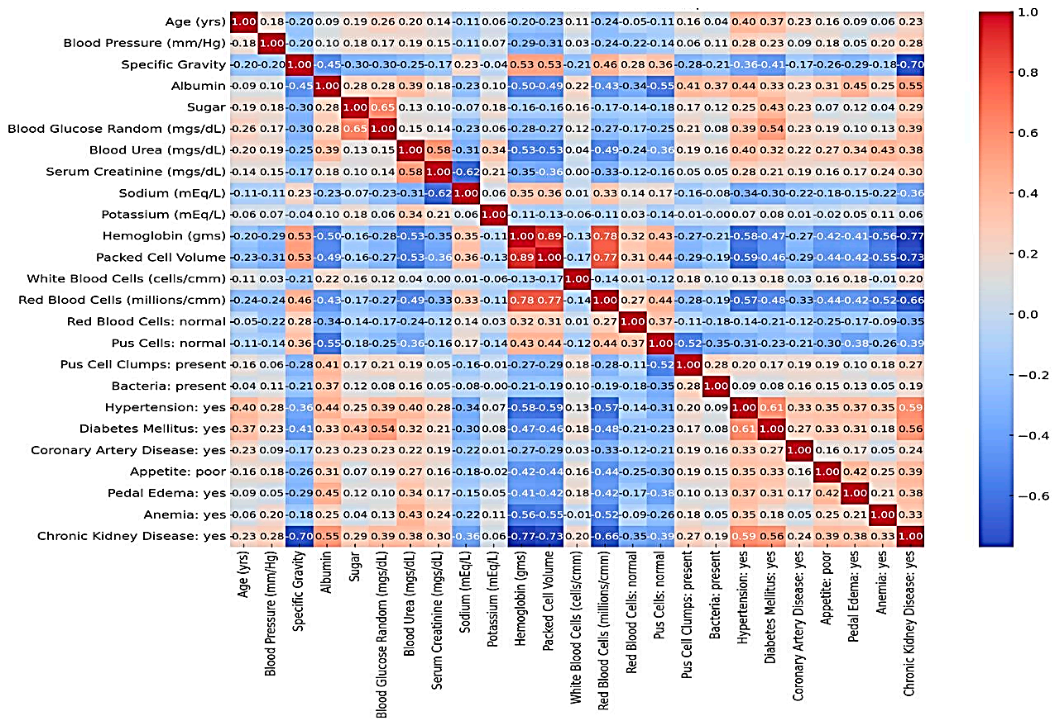

Figure 3. Feature Correlation Heatmap.

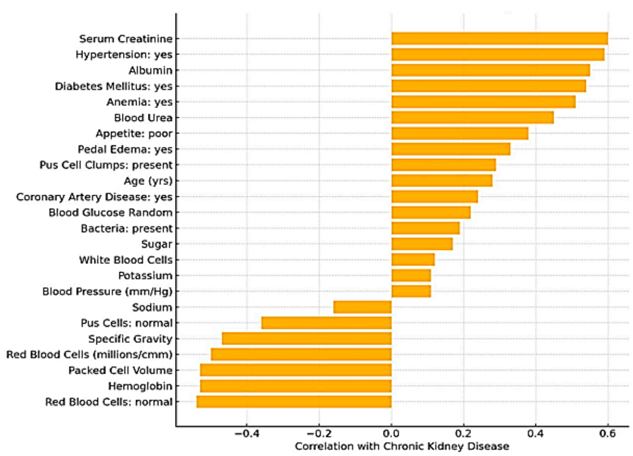

Figure 4. Feature Correlation with Chronic Kidney Disease.

4. Proposed Methodology

As mentioned in Section 2, research on machine learning algorithms for predicting kidney failure used decision trees, Support Vector Machines, Random Forests, Logistic regression, and k-nearest neighbors. These were found to be highly effective in prediction. To predict kidney failure, machine learning algorithms were implemented, and parameters were optimized using grid search to maximize accuracy.

The Python libraries used in this implementation are Pandas, NumPy, Matplotlib, Seaborn, Lux, and Sklearn. Pandas is used for reading and writing various file formats, while NumPy is a module for numerical computation that provides fast mathematical functions. Matplotlib is used for visualization, and Seaborn is a data visualization library based on matplotlib. Lux automates data exploration, while Sklearn is a machine learning library for Python that includes various methods, such as accuracy score, Decision Tree Classifier, and train-test split. These libraries help simplify data manipulation and analysis.

The process of applying models to predict kidney failure involves importing the sklearn ensemble library, setting identifiers, fitting the model, and using the predict function to predict data labels. For testing and training models the used dataset was split into two parts: 20% for testing and 80% for training. The Confusion Matrix is used to evaluate the model's accuracy and F-score, comparing actual and predicted information. The accuracy value is determined by comparing the percentage of the correct prediction value and the f-measure value by weighting the combination of precision and recall. The accuracy of the model is then determined using the score function. A classification report is then printed, showing the precision, recall, F1 Score, and support of the trained classification model. A Grid search algorithm in Python is used to optimize model parameters, selecting the best parameter based on the results.

4.1 Random Forest

Its technique using decision trees involves random decision tree combinations via bagging, followed by voting among the trees in the forest to determine the final result.

Initially, important libraries were rendered for a new random forest classifier. Parameters were set, and grid search was used with a cross-validation value of 10 to test n-estimators values. The code in (Figure 5) uses GridSearchCV from scikit-learn to do hyperparameter tuning for a Random Forest Classifier. It selects the model with the highest accuracy by running a grid search over a set of Random Forest hyperparameters using 10-fold cross-validation. The parameters for GridSearch. 'n_estimators' is the number of trees in the forest (tried 12 to 17),'max_depth' is the maximum depth of each decision tree (2 to 5 and no limit None), and 'class_weight' is How classes and 'random_state' are balanced by the algorithm (None = no weight, a custom weight dict {2: 0.33, 1: 0.67}, or 'balanced' = automated weighting): establishes a predetermined seed to guarantee reproducibility (42). GridSearchCV is In order to determine the optimal model utilizing the given parameter combinations, setup and model fitting employ the RandomForestClassifier with the set of tuned parameters and 10-fold cross-validation, using accuracy as the metric.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score

tuned_parameters = {'n_estimators': [12, 13, 14, 15, 16, 17], 'max_depth': [2, 3, 4, 5, 6, None],
                    'class_weight': [None, {0: 0.33, 1: 0.67}, 'balanced'], 'random_state': [42]}

clf = GridSearchCV(RandomForestClassifier(), tuned_parameters, cv=10, scoring='accuracy')
clf.fit(X_train, y_train)
```

Figure 5. Random Forest with grid search.

4.2 Support Vector Machine

SVM is a supervised machine learning method based on Statistical Learning Theory that classifies dataset points by mapping them to a multidimensional feature space using a kernel function. It is popular in bioinformatics for its high precision, scalability to large data, and flexibility in modeling different data types. SVMs generalize well and are resistant to overfitting, offering various kernels such as linear, polynomial, RBF, and sigmoid.

GridSearchCV is a metaestimator that creates a new estimator, such as SVC, and behaves like a classifier. Adding refit=True and setting the verbose level to a higher value allows for a detailed process description. The fit function performs cross-validation to identify the best parameter combination, then builds a new model with those parameters on all data. The code in (Figure 6) uses GridSearchCV from scikit-learn to do hyperparameter tuning for a Support Vector Machine (SVM) to determine the ideal C and gamma value combination for an SVM with a linear kernel. The model will be automatically retrained (refit=True) with the optimal parameters once they have been identified.

```
# Support vector machine
from sklearn.model_selection import GridSearchCV

# defining parameter range
param_grid = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['linear']}

grid = GridSearchCV(SVC(), param_grid, refit = True, verbose = 3)
```

Figure 6. Support Vector Machine with grid search.

4.3 K-Nearest Neighbors

KNN is a simple, easy-to-implement algorithm for classification and regression problems. We are creating a k-NN classifier and testing 24 values of the n-neighbors hyperparameter by conducting a grid search with 10-fold cross-validation. This involves splitting the dataset into 10 groups, where one is used as the test set and the rest for training. The model is trained and scored for each group to determine the best n-neighbor value. This process is repeated 10 times to ensure each group serves as the test set. Finally, we will use the 'fit' function to run the grid search and identify the optimal hyperparameter value for our k-NN classifier.

4.4 Decision Tree

The technique uses supervised statistical models to classify and predict data. It involves a tree-like structure with nodes, branches, and leaf nodes, where each node checks an attribute, each branch represents a possible outcome, and each leaf node represents a classification. The model can be constructed using entropy, which measures the information gain, or gini, which measures data heterogeneity. A lower result indicates a purer dataset.

The decision tree model from sklearn is imported and parameters are set, including criteria like ”gini”, ”entropy”, and ”log loss” for assessing split quality. The default criterion is ”gini”. The ”splitter” parameter decides on the method for selecting splits at each node, with options ”best” or ”random”. The ”min samples leaf” parameter determines the minimum number of samples required at a leaf node. Only split points with enough training samples on both sides will be considered, potentially smoothing the model, particularly in regression. This setup allows for testing different hyperparameters to optimize the decision tree model, as shown in Figure 7. The code in Figure 7 uses GridSearchCV from scikit-learn to adjust the hyperparameters for a Decision Tree Classifier. Using 10-fold cross-validation, this script conducts a thorough search to identify the optimal decision tree settings for a classification problem, choosing the setup that yields the highest accuracy.’Entropy’ (information gain) or ‘gini’ impurity are the two criteria used to assess the quality of a split.• max_depth: The tree’s maximum depth.• max_features: The maximum number of features to take into account when determining the optimal split. The minimum number of samples required to be at a leaf node is known as min_samples_leaf.The minimum number of samples required to split an internal node is known as min_samples_split.• Split: ‘best’ vs. ‘random’ is the strategy used to select the split.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV


# hyper parameter tuning of decision tree
param_grid={'criterion': ['gini', 'entropy'],
                        'max_depth': [3, 5, 7, 10],
                        'max_features': ['auto', 'sqrt', 'log2'],
                        'min_samples_leaf': [1, 2, 3, 5, 7],
                        'min_samples_split': [1, 2, 3, 5, 7],
                        'splitter': ['best', 'random']}

dtc = GridSearchCV(DecisionTreeClassifier(),param_grid, cv=10,scoring='accuracy')
dtc.fit(X_train, y_train)
```

Figure 7. Decision tree with grid search.

4.5 Logistic Regression

Logistic regression calculates the odds ratio with multiple explanatory variables, similarly to linear regression, but with a binomial response variable. Logistic regression default parameter values include a penalty for the punishment level, smaller values for stronger regularization, C for inverse regularization strength, solver for optimization algorithm, and maxiter for maximum solver convergence iterations.

5. Results and Discussions

Performance evaluation metrics are essential when assessing a machine learning model’s capabilities on unfamiliar datasets. While models may excel on the training dataset, real-world applicability requires handling new and dynamic data. Employing a robust model is essential for effective machine-learning problem-solving. In this context, performance evaluation metrics play a crucial role in determining a model’s capability to address specific challenges.

Various criteria, such as those used in regression and classification tasks, can quantify a model’s effectiveness. Using Python, we can evaluate and explain these criteria for assessing machine learning models. Reviewing related work on kidney failure prediction, many studies utilized similar datasets, highlighting the importance of selecting appropriate performance evaluation metrics.

Evaluation of models using Confusion Matrix involves determining accuracy and f- measure values of an algorithm. The matrix provides insight into correct and incorrect predictions, aiding in assessing actual versus predicted data. Accuracy is gauged by the proportion of correct predictions, while f-measure considers precision and recall in determining predictive success.

The confusion matrix is a table used in categorization problems to show where model errors occurred. The rows represent the actual outcomes, while the columns represent the model’s predictions (Figure 8). TP: True positives are when the model correctly predicts a positive outcome (i.e., correctly predicts kidney failure), while FP: false positives are incorrect positive predictions (i.e., incorrectly predict no kidney failure). TN: True negatives are correct negative predictions (i.e., correctly predict no kidney failure), and FN: false negatives are incorrect negative predictions (i.e., incorrectly predict kidney failure). This helps identify the model’s prediction accuracy and errors. When the kidney failure prediction model returns a false negative, it indicates that the patient actually has kidney failure, but the model predicts otherwise, which could have dangerous health effects. Therefore, the FN is the most critical metric for medical applications in evaluating disease prediction model accuracy, and we should work to reduce it.

Figure 9 compares the performance of various machine learning (ML) models with and without optimization using confusion matrices. Figure 9a shows moderate performance with some misclassifications. LR with optimization (Figure 9f) shows improved accuracy and fewer misclassifications, resulting in fewer FPs and FNs, better balance, and higher precision. Figure 9b shows good accuracy with a few misclassifications. RF with optimization (Figure 9g) shows near-perfect classification, with a slight boost in TP and a reduction in FP.

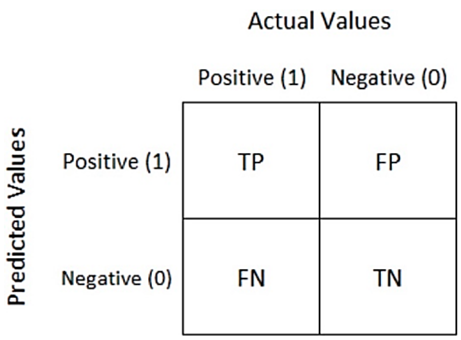

Figure 8. Confusion Matrix.

Figure 9c misclassified some instances from both classes, but the SVM with optimization did not. Figure 9h gives a significant improvement. Optimization improved TN and reduced FP, although TP decreased by 1. Figure 9i shows better class separation than DT without optimization (Figure 9d). Moreover, TP increased, FN dropped to 0, and sensitivity improved significantly.

Figure 9e has many misclassifications, while KNN with optimization. Figure 9j achieved noticeable improvement, but still the lowest among all models, and higher FN, leading to lower sensitivity (recall).

The confusion matrix for the model's evaluation with the default parameters is shown in Figure 9. The logistic regression model had some false positives and false negatives, whereas the random forest model had none. The kernel linear SVM model had more false positives and false negatives than the other models. The decision tree model had the fewest false positives, and the k-Nearest Neighbor model had the fewest false negatives. Overall, the models showed varying performance in correctly classifying data points as positive or negative.

The confusion matrix of the models evaluated with optimization is represented in Figure 9. Overall, the Logistic regression model had the highest true positive rate, while the Random Forest model had the highest true negative rate. The Kernel linear SVM model had the lowest false positive rate, and the Decision tree model had the lowest false negative rate. The k-nearest neighbor model had the highest false-positive and false-negative rates. The evaluation of these models shows their performance in correctly classifying positive and negative data points, as well as the classification errors.

The comparison was based on the prediction accuracy of multiple algorithms, as shown in Figure 10, which shows improved performance post-optimization for the compared models. Overall, logistic regression, random forest, and decision tree models performed well, while the kernel linear SVM model performed well but exhibited some misclassifications. The k-nearest neighbor model produced less accurate predictions than other models. These evaluations provide insights into the strengths and weaknesses of each model in accurately classifying data points.

In conclusion, all models are much enhanced by optimization. After optimization, Random Forest achieves the highest accuracy (99.75%), indicating it is the best model for predicting kidney failure, with high TP and low FP/FN. Significant gains are also observed in Kernel SVM and Logistic Regression, surpassing 96%. Before and after optimization, KNN performs the worst, with a high FN, suggesting it might not be a good fit for this dataset or that it requires more fine-tuning. The accuracy result is supported by the confusion matrices, which show that models with fewer misclassifications are more accurate.

6. Conclusion

Detecting kidney failure late is a major challenge in healthcare, leading to high mortality rates. Data mining techniques can help healthcare professionals identify patients at risk of acute kidney injury early, enabling prompt treatment. This research paper focuses on using machine learning algorithms to predict kidney disease and analyze data efficiently. By determining the most effective algorithm based on multiple criteria, we aim to provide healthcare professionals with better alternatives for managing kidney failure. This approach has the potential to improve patient outcomes and reduce the burden on the medical field in the treatment of late-stage kidney failure. Prompt recognition and treatment of acute kidney injury remain difficult, but our approach offers potential for early intervention and improved outcomes.

The aim was to compare and identify the most efficient model among Random Forest, decision tree, support vector machine, K Nearest Neighbor, and logistic regression for predicting kidney failure. Results showed that optimized Random Forest had outstanding prediction accuracy of 99.75%, surpassing default model performances and achieving a 99% F1-measure. While other models had lower accuracy rates, such as KNN at 78%, the top model stood out for its exceptional predictive power.

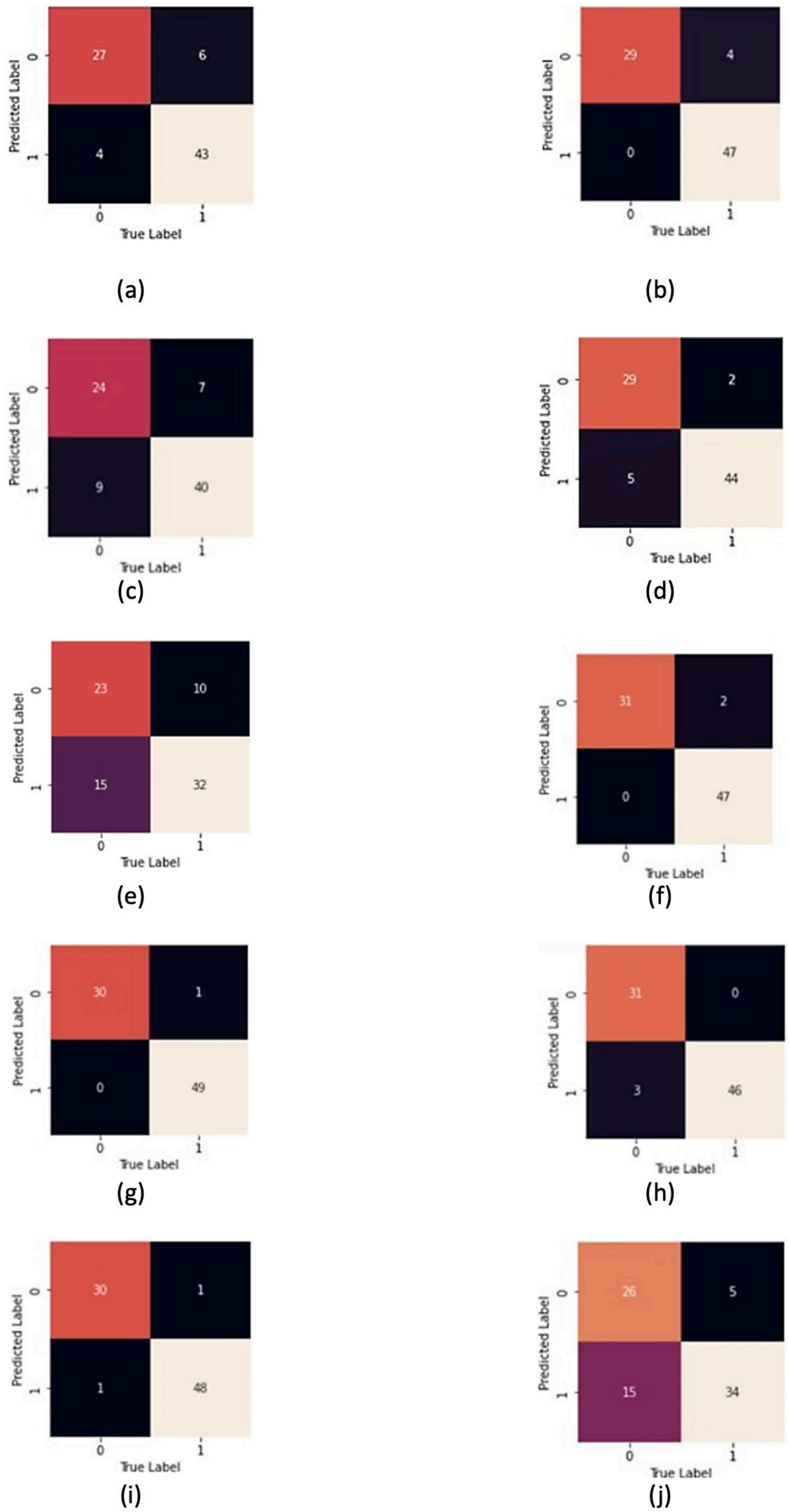

Figure 9. Confusion Matrix of the used ML models with and without optimization: a) Logistic Regression; b)Random Forest; c)Kernel linear SVM; d)Decision Tree; e) K-Nearest Neighbors; f) Logistic Regression with optimization; g) Random Forest with optimization; h) Kernel linear SVM with optimization; i) Decision Tree with optimization; j) K-Nearest Neighbors with optimization.

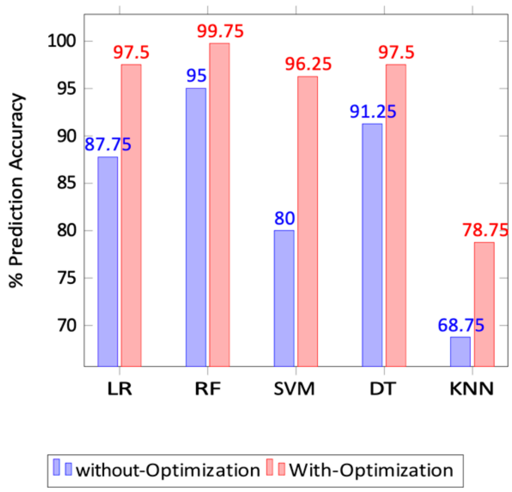

Figure 10. Models' Prediction Accuracy before and after Optimization.

Overall, our research demonstrated significant advancements in kidney failure prediction through the use of machine learning algorithms.

Through Electronic Health Records (EHR) systems, where ML models may be embedded into hospital EHR platforms and provide a model for Clinical Decision Support Systems (CDSS), the resulting ML model from this study for CKD prediction can be included into clinical practice. The ML model automatically evaluates the data and highlights possible CKD concerns whenever a patient's vitals or test results are updated. The ML model helps doctors make decisions by producing an alert or risk score. This may indicate the need for additional testing, a referral to a specialist, or changes to medication.

However, incorporating a ML model for CKD prediction into clinical practice is more than just creating a model. It requires extensive preparation to ensure its relevance, data security, usability, and trust from healthcare professionals. To successfully integrate a ML model into real-world clinical practice, it is crucial to train clinicians, ensure easy integration into current workflows, and make the model explicable. Regulatory and ethical considerations are also crucial, ensuring the system adheres to regulations like HIPAA, GDPR, and local health data laws. To prevent biased predictions, the model should be evaluated across various populations while considering bias and fairness. Prospective clinical trials should be used to assess the model's efficacy in the real world. Challenges include model drift, interoperability, and data quality and consistency. These include changing patient populations and medical practices over time, interoperability between different healthcare systems using different EHR formats, and data quality and consistency, where missing or inconsistent medical records can affect model accuracy. These challenges are the main real-world deployment challenges in the field of ML in clinical practice and are limitations for this study.

The implementation of deep learning and a comparison with the best machine learning model found in this study will be the focus of future research. We will also be concerned about developing the model understandable and offering a list of the clinical parameters that most significantly influenced the predicted risk outcome in order to gain the clinicians' trust. Additionally, attempt to implement and evaluate the developed machine learning model for CKD prediction in real-world healthcare system.

Funding

The author received no specific funding for this work.

Conflict of interest

The author has no conflict of interest to declare.

Acknowledgements

The author would like to thank Futon Alharbi, Samar Halawani, Renad Filmban and Arwa Kalantan, who are students at the College of Computer Science and Engineering, University of Jeddah, for supporting this study by developing the machine learning prototype.

References

Ahmad, M., Tundjungsari, V., Widianti, D., Amalia, P., & Rachmawati, U. A. (2017). Diagnostic decision support system of chronic kidney disease using support vector machine. In *2017 second international conference on informatics and computing (ICIC)* (pp. 1-4). IEEE.
https://doi.org/10.1109/IAC.2017.8280576

Al-Hyari, A. Y., Al-Taee, A. M., & Al-Taee, M. A. (2013). Clinical decision support system for diagnosis and management of chronic renal failure. In *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)* (pp. 1-6). IEEE.
https://doi.org/10.1109/AEECT.2013.6716440

Alturki, N., Altamimi, A., Umer, M., Saidani, O., Alshardan, A., Alsubai, S., ... & Ashraf, I. (2024). Improving Prediction of Chronic Kidney Disease Using KNN Imputed SMOTE Features and TrioNet Model. *Computer Modeling in Engineering & Sciences (CMES)*, *139*(3).
https://doi.org/10.32604/cmes.2023.045868

Chittora, P., Chaurasia, S., Chakrabarti, P., Kumawat, G., Chakrabarti, T., Leonowicz, Z., ... & Bolshev, V. (2021). Prediction of chronic kidney disease-a machine learning perspective. IEEE access, 9, 17312-17334.
DOI:10.1109/ACCESS.2021.3053763

De Almeida, K. L., Lessa, L., Peixoto, A. B., Gomes, R. L., & Celestino, J. (2020). Kidney failure detection using machine learning techniques. In *8th international workshop on advances in ICT infrastructures and services (ADVANCE 2020)* (pp. 1-8).
https://hal.science/hal-02495264/

Delrue, C., De Bruyne, S., & Speeckaert, M. M. (2024). Application of machine learning in chronic kidney disease: current status and future prospects. *Biomedicines*, *12*(3), 568.
https://doi.org/10.3390/biomedicines12030568

Guldogan, E., & Kucukacali, Z. (2021). Classification of chronic kidney failure by applying different tree-based methods on a medical data set. *Medicine Science*, *10*(2).
https://doi.org/10.5455/medscience.2021.05.153

Ilyas, H., Ali, S., Ponum, M., Hasan, O., Mahmood, M. T., Iftikhar, M., & Malik, M. H. (2021). Chronic kidney disease diagnosis using decision tree algorithms. *BMC nephrology*, *22*(1), 273.
https://doi.org/10.1186/s12882-021-02474-z

Islam, M. A., Majumder, M. Z. H., & Hussein, M. A. (2023). Chronic kidney disease prediction based on machine learning algorithms. *Journal of pathology informatics*, *14*, 100189.
https://doi.org/10.1016/j.jpi.2023.100189

Levey, A. S., Coresh, J., Bolton, K., Culleton, B., Harvey, K. S., Ikizler, T. A., ... & Briggs, J. (2002). K/DOQI clinical practice guidelines for chronic kidney disease: evaluation, classification, and stratification. *American Journal of Kidney Diseases*, *39*(2 SUPPL. 1), i-ii+.
https://pure.johnshopkins.edu/en/publications/kdoqi-clinical-practice-guidelines-for-chronic-kidney-disease-eva-3/

Polat, H., Danaei Mehr, H., & Cetin, A. (2017). Diagnosis of chronic kidney disease based on support vector machine by feature selection methods. *Journal of medical systems*, *41*(4), 55.
https://doi.org/10.1007/s10916-017-0703-x

Praveen, S. P., Jyothi, V. E., Anuradha, C., VenuGopal, K., Shariff, V., & Sindhura, S. (2024). Chronic kidney disease prediction using ML-based neuro-fuzzy model. *International Journal of Image and Graphics*, *24*(06), 2340013.
https://doi.org/10.1142/S0219467823400132

Revathy, S., Bharathi, B., Jeyanthi, P., & Ramesh, M. (2019). Chronic kidney disease prediction using machine learning models. *International Journal of Engineering and Advanced Technology*, *9*(1), 6364-6367.
https://doi.org/10.35940/ijeat.A2213.109119

Rubini, L., Soundarapandian, P., & Eswaran, P. (2015). Chronic kidney disease. *UCI Machine Learning Repository*, *10*, C5G020.
https://doi.org/10.24432/C5G020

Saif, D., Sarhan, A. M., & Elshennawy, N. M. (2024). Early prediction of chronic kidney disease based on ensemble of deep learning models and optimizers. *Journal of electrical systems and information technology*, *11*(1), 17.
https://doi.org/10.1186/s43067-024-00142-4

Salekin, A., & Stankovic, J. (2016). Detection of chronic kidney disease and selecting important predictive attributes. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)* (pp. 262-270). IEEE.
https://doi.org/10.1109/ICHI.2016.36

Sanmarchi, F., Fanconi, C., Golinelli, D., Gori, D., Hernandez-Boussard, T., & Capodici, A. (2023). Predict, diagnose, and treat chronic kidney disease with machine learning: a systematic literature review. *Journal of nephrology*, *36*(4), 1101-1117.
https://doi.org/10.1007/s40620-023-01573-4

Su, C. T., Chang, Y. P., Ku, Y. T., & Lin, C. M. (2022). Machine learning models for the prediction of renal failure in chronic kidney disease: A retrospective cohort study. *Diagnostics*, *12*(10), 2454.
https://doi.org/10.3390/diagnostics12102454