applied sciences

MDPI

Article

Efficient Data Preprocessing with Ensemble Machine Learning Technique for the Early Detection of Chronic Kidney Disease

Vinoth Kumar Venkatesan [1], Mahesh Thyluru Ramakrishna [2], Ivan Izonin [3,*], Roman Tkachenko [4] and Myroslav Havryliuk [3]

[1] School of Information Technology and Engineering, Vellore Institute of Technology University, Vellore 632014, India
[2] Department of Computer Science and Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-Be University), Bangalore 562112, India
[3] Department of Artificial Intelligence, Lviv Polytechnic National University, 79013 Lviv, Ukraine
[4] Department of Publishing Information Technologies, Lviv Polytechnic National University, 79013 Lviv, Ukraine
* Correspondence: ivan.v.izonin@lpnu.ua

Abstract: It is a serious global health concern that chronic kidney disease (CKD) kills millions of people each year as a result of poor lifestyle choices and inherited factors. Effective prediction tools for prior detection are essential due to the growing number of patients with this disease. By utilizing machine learning (ML) approaches, this study aids specialists in studying precautionary measures for CKD through prior detection. The main objective of this paper is to predict and classify chronic kidney disease using ML approaches on a publicly available dataset. The dataset of CKD has been taken from the publicly available and accessible dataset Irvine ML Repository, which included 400 instances. ML methods (Support Vector Machine (SVM), K-Nearest Neighbors (KNN), random forest (RF), Logistic Regression (LR), and Decision Tree (DT) Classifier) are used as base learners and their performance has been compared with eXtreme Gradient Boosting (XGBoost). All ML algorithms are evaluated against different performance parameters: accuracy, recall, precision, and F1-measure. The results indicated that XGBoost outperformed with 98.00% accuracy as compared to other ML algorithms. For policymakers to forecast patterns of CKD in the population, the model put forth in this paper may be helpful. The model may enable careful monitoring of individuals who are at risk, early CKD detection, better resource allocation, and management that is patient-centered.

Keywords: chronic kidney disease (CKD); machine learning (ML); data preprocessing; ensemble methods; entropy

check for updates

Citation: Venkatesan, V.K.; Ramakrishna, M.T.; Izonin, I.; Tkachenko, R.; Havryliuk, M. Efficient Data Preprocessing with Ensemble Machine Learning Technique for the Early Detection of Chronic Kidney Disease. *Appl. Sci.* **2023**, *13*, 2885. https://doi.org/10.3390/app13052885

Academic Editors: José Salvador Sánchez Garreta, Vicente García and Yu-Dong Zhang

Received: 21 January 2023
Revised: 15 February 2023
Accepted: 21 February 2023
Published: 23 February 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The kidney is a very important organ in the human body. Excretion and osmoregulation are two of the most important functions of this organ. In layman's terms, the kidney and excretion systems gather and dispose of all poisonous and useless material from the body. In India, 1 million instances of CKD are diagnosed each year. The patient can live for a long period with poor kidney functions if the condition is detected early on and the appropriate treatments are started [1,2]. However, it is common that the patient's condition deteriorates over time and the number of nephrons falls rapidly. As a result, increased hazardous materials in the blood endanger the patient's life. The final stage is referred to as "terminal kidney failure." To keep the patient alive at this stage, one of hemodialysis, continuous peritoneal dialysis, or kidney transplantation procedures should be used. CKD, which eventually leads to renal failure, can cause a variety of issues in society, including economic, social, and medical issues. The high cost of hemodialysis might be used as an illustration of economic issues. Patients experience emotional and social problems as a result of their continued use of medicines and the consequences thereof. Furthermore, the medications

utilized can interact with various organs and systems, resulting in a considerable reduction in life expectancy [3].

Chronic kidney disease is referred to as "chronic" because it develops gradually and often lasts for a long period, impairing the urinary system's function. High blood pressure, diabetes, as well as cardiovascular disease are all risk factors for CKD patients [4]. Patients with CKD experience adverse effects, particularly in the late stages that harm the neurological and immunological systems. Patients in countries that are developing may come to the end of their lives and require dialysis. GFR is calculated using information such as the patient's age, blood test results, gender, and other criteria [5]. The various parameters affecting CKD are shown in Figure 1.

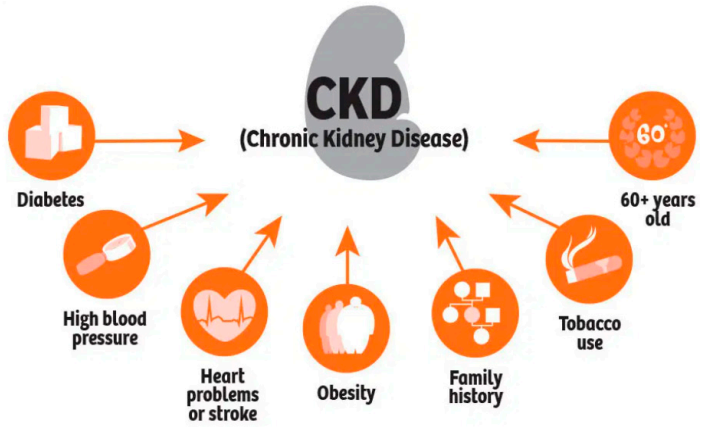


Figure 1. Factors affecting CKD.

The patient's age, race, blood creatinine, gender, and other criteria can be used to assess chronic kidney disease. The severity of impairment caused by CKD is classified into five phases, as shown in Table 1, based on GFR [6]:

Table 1. Different stages present in CKD.

Stage	Description	GFR (mL/min)
I	Kidney function is normal	≥90
II	Mild increase in GFR	60–89
III	Moderate increase in GFR	30–59
IV	Severe increase in GFR	15–29
V	Kidney failure	15 or dialysis

As there is a growing number of CKD patients, there is a scarcity of CKD-specialized physicians [7]. Computer-assisted diagnostics are required to assist physicians as well as radiologists in making diagnostic judgments. Machine learning (ML) as well as deep learning (DL) methods have been used in the early phases of CKD prediction as well as diagnosis. Artificial intelligence (AI) techniques have played a major role in medical and health industries [8]. CKD is one of the key issues in medical science because it involves significant criteria and complexity to effectively forecast this disease. Algorithms, such as Naive Bayes (NB), Decision Trees (DTs), K-Nearest Neighbors (KNN), and Neural Networks, are used to predict CKD risk. Each algorithm has a specialty, such as Naive Bayes' use of probability to predict CKD, Decision Trees' use of classification to produce a classified report for CKD, and Neural Networks' use of opportunities to minimize CKD prediction error [9].

The contribution of this study are summarized as follows:

- Detection of chronic kidney disease in the early stages to avoid critical health issues.
- Early detection of chronic kidney disease will be helpful for low-income people so that they can take prior precautions as they cannot undergo or afford major surgery.

To provide better clarification and highlight the core theme of this research, this article is structured as follows. Section 1 briefs the essential basic concepts and significance of the proposed model with key attributes. Section 2 discusses the contributions of related research. It also summarizes different related studies, their proposed techniques, and the performance achieved. Section 3 enumerates the flow of the research process from elaborating the dataset, the preprocessing technique, the feature selection process, and classification. Section 4 discusses and exhibits the empirical observations and analysis in comparison with all other existing key methodologies. Section 5 includes the vital end notes of this study with planned future works.

2. Related Works

Many academics are consistently working on the prediction of CKD using various classification algorithms. Numerous studies have been undertaken in the last few decades to better understand and investigate CKD. Many researchers have employed various classification strategies for the early detection of CKD in the literature. Data mining (DM) is the methodology of extracting specific data from a large dataset. Medical diagnosis, facial recognition, and data filtering are just a few of the uses of these techniques.

The authors of one study [9] described an ultrasonography (USG)-based approach for diagnosing the phases of CKD. The algorithm works to detect fibrotic instances at various times. Authors of another study [10] performed a comparison of the outcomes of various models. They found and claimed that the Multiclass DT approach outperforms previous algorithms, with an accuracy of roughly 98% for a smaller dataset of 14 attributes. To decide whether the urinary system is harmless or harmful, the authors [11] suggested an expert system based on fuzzy logic. The authors [12] employed Softmax to categorize the final class after using a stacked autoencoder prototype to extract the features of CKD. In one study, the authors [13] introduced a genetic algorithm (GA) in which the GA optimized the weight vectors to train the NN. For CKD diagnosis, the system outperforms using standard neural networks.

The authors of another study [14] used multiple machine learning classification algorithms to reduce diagnosis time and enhance the accuracy of diagnosis. The planned study looks at how different phases of CKD are classified based on their severity. The RBF algorithm performs better than other classifiers, with an accuracy of 85.3%. In one study, the authors [15] examined the missing values in the CKD dataset as they greatly decreased accuracy as well as prediction.

Authors of one study [16] incorporated a unique methodology to figure out CKD using ML algorithms. The dataset consists of 400 instances with 24 dependent variables and one independent variable or class variable whose values are “ckd” or “nockd.” They used KNN, Neural Networks, and RF in order to obtain the results. Pinar Yildirim [17] performed an investigation on the impact of class imbalance. Using a sampling algorithm, a comparison study was done. In another study, the authors [18] evaluated 12 alternative categorization algorithms. They calculated the accuracy of their prediction findings by comparing their calculated results to real results. Accuracy, sensitivity, precision, and specificity were utilized as assessment criteria.

Clustering, classification, and other data mining techniques play an important role in extracting undisclosed knowledge from large databases. The technique of classification is one of the supervised learning techniques for defining subgroups ahead of time. The data attribute value must be used to identify the classes in the classification process. It establishes the classes by taking into account the quality of the data. These preconfigured specimens are used by the training algorithm to identify the set of parameters required for proper segregation [19–21]. In one article, the authors [22] utilized Apriori as well as

k-means algorithms to construct a prediction system, and these were also employed to forecast individuals with kidney failure. The authors used machine learning algorithms to assess 42 data characteristics and used Receiver Operating Characteristic (ROC) plots to evaluate the data.

The authors of one study [23] developed a diagnosis and prediction system and also used the tool Weka to evaluate the data. The authors analyzed ROC plots using the J48 classifier, AD Trees, the K-Star method, the NB classifier, and the random forest (RF) classifier. In their research, the K-Star algorithm and the RF classifier were found to be the best approaches for the dataset. Sinha [24] suggested a method for comparing the performance of two data mining approaches. To assess accuracy and precision, the author employed KNN and also support vector machine (SVM). In terms of precision as well as accuracy performance metrics, the author found that the KNN outperformed the SVM classifier.

To extract the most critical vital characteristics, the authors of one study [25] presented an approach called LVW-FS. In another study, the author [26] tested diabetes in patients with chronic kidney illness using a model. The statistical method of chi-square was used to minimize the dimensionality of large amounts of data. The model uses parameters including glucose, age, albumin rate, and others to predict kidney health. The authors of one study [27] used two strategies to minimize the dimensionality of the dataset so as to strongly choose the attributes linked with CKD. Table 2 summarizes some of the recent related work being carried out in the detection CKD with techniques and performance.

Table 2. Summary of recent related research on the detection of CKD.

Study and Year	ML Techniques	Performance
Ilyas, H. et al. (2019) [28]	Random forest and J48 algorithms	Random forest accuracy—78.25% and J48 accuracy—85%
Sabanayagam et al. (2020) [29]	Deep Learning Algorithm (DLA)	DLA accuracy—95%
Poonia, R.C. et al. (2022) [30]	LR model + chi-square feature selection (K > 14), where K is number of features	Accuracy—97.5%
Elhoseny, M. et al. (2019) [31]	Ant Colony-based Optimization (D-ACO) algorithm	D-ACO accuracy—95%

In this study, the different ML models (KNN, SVM, RF, DT Classifier, Logistic Regression, and XGBoost) are used to diagnose CKD. In order to determine the optimal attributes, the Recursive Feature Elimination (RFE) approach is also used. For promising accuracy, all six machine learning techniques were utilized and compared among themselves, and it was observed that XGBoost exhibited the best performance.

3. Proposed Methodology

To analyze the CKD dataset, a sequence of experiments has been conducted utilizing ML methods such as SVM, KNN, DT, RF, LR, and XGBoost classifier. Figure 2 depicts the general pattern of CKD classification.

The mean approach was used to calculate missing values that are numeric, while the mode has been utilized to calculate missing values that are nominal during preprocessing. RFE, which is a Recursive Feature Elimination approach, has been utilized to select the aspects of relevance related to the attributes of high importance for the diagnosis of CKD. For disease diagnosis, these chosen features were given as inputs to the classifiers.

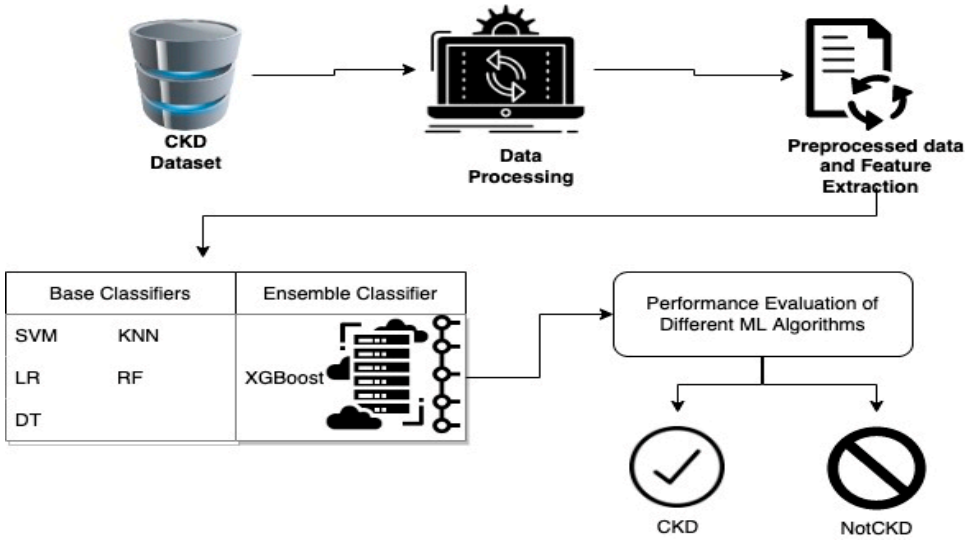


Figure 2. Proposed pattern of CKD classification.

3.1. Dataset

The dataset of CKD has been taken from the Irvine ML Repository [6], which included 400 patients. Besides the class characteristics, namely "ckd" and "notckd," the dataset has twenty-four features separated into eleven numeric features as well as thirteen category features. Age, specific gravity, blood pressure, albumin, red blood cells, sugar, pus cells, red blood cells, pus cell clumps, bacteria, blood glucose random, blood urea, serum creatinine, sodium, potassium, hemoglobin, packed cell volume, white blood cell count, red blood cell count, hypertension, diabetes mellitus, coronary artery disease, appetite, pedal edoema, and anaemia are some of the characteristics. Ckd and notckd are the two diagnostic classes. Figure 3 depicts count of diagnosis of kidney disease in the dataset.

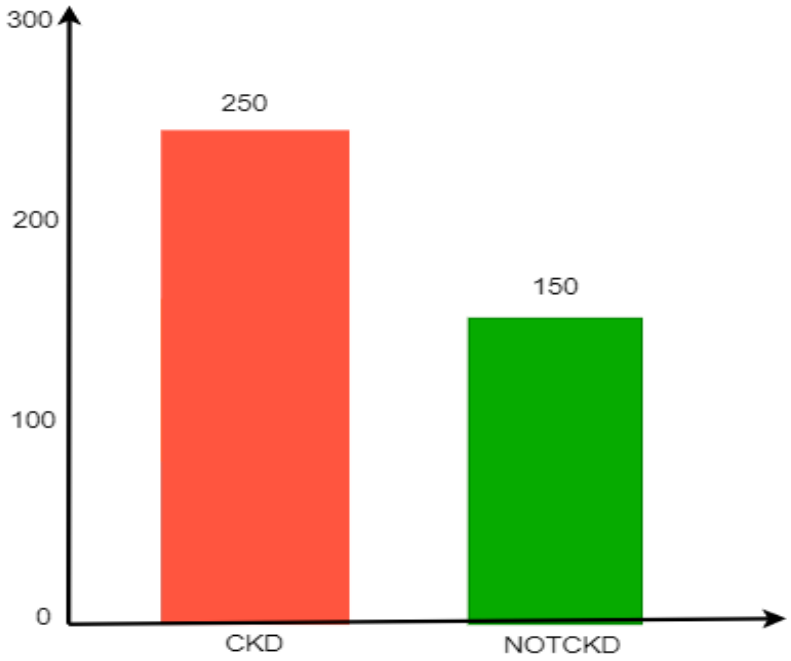


Figure 3. Count of diagnosis of kidney disease.

3.2. Preprocessing

To ensure satisfactory accuracy, missing values must be handled in the preprocessing step depending on their distributions. Little's MCAR (Missing Completely At Random) test was used in this study to confirm the randomization of missing values. The possibility of bias as a result of missing data is dependent on the mechanism that caused the data to be missing. The chi-square test of MCAR for multi-variate quantitative data is used to examine the analytical approaches used to correct the missingness and all the inference/computation work. In this study, the KNN Imputer method was employed to fill in the missing values.

Many classification algorithms strive to collect only pure examples for learning and to make the boundary between each class as clear as possible to improve prediction. The majority of classifiers find it significantly more difficult to learn how to classify synthetic cases that are close to the boundary than those that are far from it. Based on these results, authors of one study presented an enhanced preprocessing method (A-SMOTE) for imbalanced training sets [27]. The SMOTE method is carried out in 3 stages, as explained below.

Step A: Using Equation (1) synthetic instance is created.

$$N = 2 * (r - z) + z \tag{1}$$

where $r \rightarrow$ majority class samples, z $\rightarrow$ minority class samples and $N \rightarrow$ newly created synthetic instance.

Step B: The below-mentioned steps are carried out to remove the outlier; that is, noise.

If, $\hat{S}$ ={$\hat{S}_1$, $\hat{S}_2$,$\hat{S}_3$, ...$\hat{S}_n$} is a new instance received by Step A, then we will calculate the distance among $\hat{S}_i$ with original minority S_m, $Min_{Rap}(\hat{S}_i, \hat{S}_m)$ defined using Equation (2).

$$Min_{Rap}(\hat{S}_i, \hat{S}_m) = \sum_{k=1}^{z} \sum_{j=1}^{M} \sqrt{(\hat{S}_i{}^{(j)} - S_{mk}{}^{(j)})^2} \tag{2}$$

where

$Min_{Rap}(\hat{S}_i, \hat{S}_m) \rightarrow$ samples rapprochement and as per Equation (2), L is calculated using Equation (3).

$$L = \sum_{i=1}^{n} (Min_{Rap}\ (\hat{S}_i, S_m)) \tag{3}$$

Step C: Calculate distance between $\hat{S}_i$, and every original majority S_a, $Maj_{Rap}(\hat{S}_i, S_a)$, described using Equation (4).

$$Maj_{Rap}(\hat{S}_i, S_a) = \sum_{i=1}^{r} \sum_{j=1}^{M} \sqrt{(\hat{S}_i{}^{(j)} - S_{al}{}^{(j)})^2} \tag{4}$$

$Maj_{Rap}(\hat{S}_i, S_a) \rightarrow$ samples rapprochement and as per Equation (4), H is computed using Equation (5).

$$H = \sum_{i=1}^{n} (Maj_{Rap}\ (\hat{S}_i, S_a)) \tag{5}$$

Entropy is nothing but the measure of disorder. The following Equations (6) and (7) were used by Claude E. Shannon to put this link between probability and heterogeneity.

$$H(X) = -\sum (p_i * log_2 p_i) \tag{6}$$

$$\text{Entropy (p)} = -\sum_{i=1}^{N} p_i log_2 p_i \tag{7}$$

There are missing values in all the features except diagnostic class. In this study, the KNN Imputer method was employed to fill in the missing values. Since the dataset comprises 250 cases of the "ckd" class (62.5%) and 150 cases of "notckd" (37.5%), the dataset is uneven or unbalanced. The analysis of the features before and after application of the SMOTE technique is shown in Table 3.

Table 3. SMOTE analysis.

Components		Before SMOTE	After Smote	(Mean ± SD)
		Overall (N = 400)	Overall (N = 750)	
Age (Mean ± SD)		38.52 ± 16.49	38.40 ± 16.53	-
Gender, [N (%)]	Male	280 (70)	375 (50)	38.13 ± 16.51
	Female	120 (30)	375 (50)	
CKD		250 (62.5)	300 (40)	38.25 ± 16.55
Non-CKD		150 (37.5)	300 (40)	
Testing Samples		-	150 (20)	38.29 ± 16.49

3.3. Feature Selection

A heat map represents the Pearson correlation coefficient matrix. The linear correlation between two properties can be described by the Pearson correlation coefficient. Figure 4 depicts the heat map.

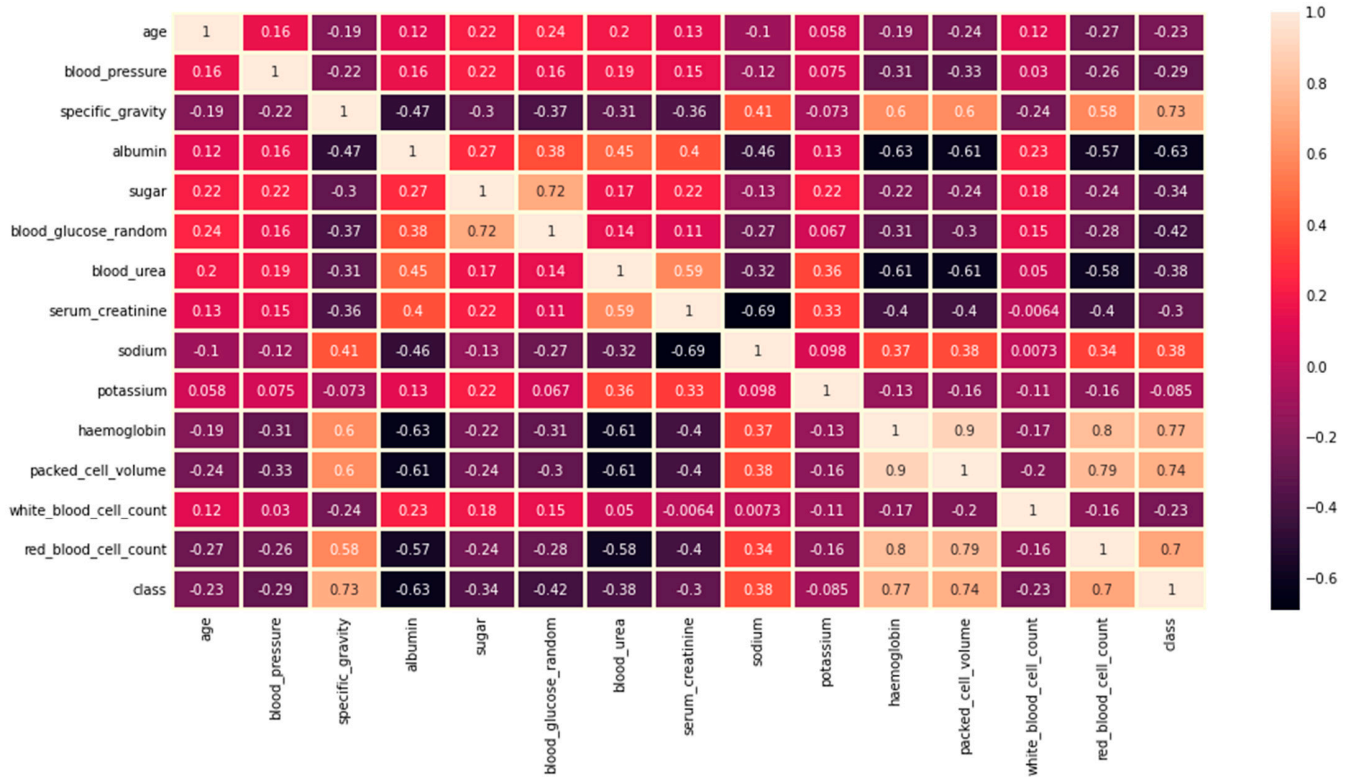


Figure 4. Correlation heat map.

We observe that several of the features we developed have a stronger linear link with one another. The Pearson correlation coefficient between these two features is close to one, which supports the heat map's message. Two features that are completely correlated with one another can be expressed by one another, which means that the information these two features carry is identical. The accuracy of the model can be increased while causing less data information loss for the two features with a high linear association by deleting one of them. Figure 4 shows that the absolute values of the heat map of attribute correlations to the class label demonstrate that age, specific gravity, and blood pressures have the highest correlations (more than 0.5). The secondary characteristics of sugar and albumin, for example, have correlations greater than 0.2.

The RFE approach is selected to choose the most prominent features by identifying those that have a high correlation with the target. DT is used as a base model for the RFE approach for feature selection. From the 24 dependent attributes included in the dataset, we chose 14 relevant features apart from class label to develop the prediction model. Table 4 provides the attributes that were chosen. During preprocessing, the missing values are replaced with the mean value for a few attributes, such as age, and the highest threshold was taken for a few attributes, including BP.

Table 4. Attributes/features chosen.

Attributes/Features	Values Used
Age (A)	Discrete integer values
Blood pressure (BP)	Discrete integer values
Specific gravity (sg)	Nominal values
Albumin (Al)	Nominal values
Sugar(su)	Nominal values
Blood glucose random (bgr)	Numerical value
Blood urea (bu)	Numerical value
Serum creatinine (sc)	Numerical value
Sodium (sod)	Numerical value
Potassium (pot)	Numerical value
Haemoglobin (haem)	Numerical value
Packed cell volume	Numerical value
White blood cell count	Numerical value
Red blood cell count	Numerical value

The nominal error rate, as determined by Little's MCAR test on a sample of 400 records, is shown in Figure 5. Incorporating incomplete data flow factors into the inference approach is useless if they do not correlate with the result. It is a fallacious perception to integrate any attributes in the inference process only because there is a possibility that they are associated with multicollinearity in the result.

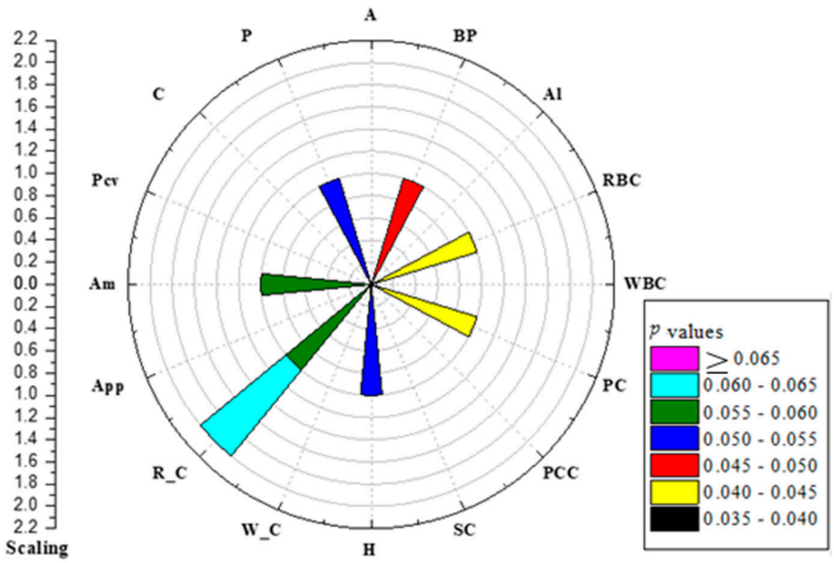


Figure 5. Little's MCAR test on 16 attributes.

3.4. Classification

The dataset is divided into 2 subsets, each having 14 features. The training dataset consists of 75% of the total CKD dataset, i.e., 300 out of 400 records, and the testing dataset consists of the remaining 100 records out of 400 in the overall CKD dataset.

3.4.1. Regression Analysis (RA)

With a linear regression line, the dependent variable is continuous, and the independent variable is more typically continuous or discrete. For logistic regression, we use the natural logarithm, as shown in Equation (8).

$$log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x \tag{8}$$

where β_0 is the intercept and β_1 is the slope. LR can be extended to situations involving outcome variables with three or more categories.

3.4.2. K-Nearest-Neighbors (k-NN)

Despite the fact that the k-NN approach has higher processing costs than other methods, it is still the best choice for applications where predictions are not demanded frequently but accuracy is critical. For distance metrics, the Euclidean metric can be used, and last, the input x gets allocated to that class that has the largest probability, as shown in Equation (9).

$$\begin{aligned} d\left(x, x^{|}\right) &= \sqrt{\left(x_1 - x_1^{|}\right)^2 + \ldots . + \left(x_n - x_n^{|}\right)^2} \\ \mathrm{P(y = j|X = x)} &= \tfrac{1}{K} \textstyle\sum_{i \in A}^{n} I\left(y^{(i)} = j\right) \end{aligned} \tag{9}$$

3.4.3. Decision Tree (DT) Classifier

A DT is a flowchart-like structure in which each internal node represents an attribute "test," each branch reflects the test's conclusion, and each leaf node represents a class label [28]. It is one of the most well-known modeling strategies because it was one of the first elite regression analysis methods individuals learned when learning predictive modeling. With a linear regression line, the dependent variable is usually continuous, while the independent variable is more commonly discrete.

3.4.4. Random Forest (RF) Classifier

A machine learning (ML) method called a random forest is employed to resolve regression and classification issues. It makes use of ensemble learning, a method for solving complicated issues by combining a number of classifiers. In a random forest algorithm, there are many different decision trees. The random forest algorithm creates a "forest" that is trained via bagging or bootstrap aggregation. Based on the decision trees' predictions, the RF algorithm determines the result. It makes predictions by averaging or averaging out the results from different trees. The accuracy of the result grows as the number of trees increases. The decision tree algorithm's shortcomings are eliminated with a random forest. It improves precision and lowers dataset overfitting.

3.4.5. Support Vector Classifiers (SVM)

Groups of supervised learning techniques called SVMs are employed in the identification of outliers, regression, and classification. Because they select the decision boundary that optimizes the distance from the nearest data points of all the classes, SVMs vary from other classification techniques. The maximum margin classifier or maximum margin hyper plane is the name of the decision boundary produced by SVMs.

3.4.6. XGBoost

XGBoost, an ensemble method, is used with k-fold cross-validation. The working of XGBoost and its evaluation with respect to the performance metrics is discussed in the next section.

Boosting is an ensemble modeling technique for creating a classifier that is strong from a huge number of classifiers, which are weak. It is performed by putting together a model from a series of weak models. The training data are used to develop a model to start the process. The second model is then developed, with the goal of correcting the original model's flaws. This method is repeated until all of the training data are correctly predicted. The overall functioning of XGBoost is shown in Algorithm 1.

Algorithm 1: Overall functioning of XGBoost

A: booster:

- At each iteration, choose the kind of model to run. There are two choices:
  - ○ gradboostree: tree-based models.
  - ○ gradboostree: linear models.

B. silent:

- When this is set to 1, no running messages will be printed.
- It is best to leave it at 0, as the messages may aid in comprehending the model.

C. nthread

- Total cores in the system must be put here for parallel processing.

If one wishes to run on all cores, leave the value blank and the algorithm will figure it out.

Despite the fact that there are two sorts of boosters, we will only consider the tree booster because it consistently performs better than the linear booster, which is why the second one is rarely utilized.

The function $y = f(x)$ is recovered by approximately estimating $\hat{f}(x)$ while measuring how good the mapping is using a loss function $l(y,x)$ and then taking the average over all the dataset points to obtain the final cost, given the dataset $\{(xi, yi)\}i = 1 \ldots ., n$, where x are the features and y is the target, to solve any supervised machine learning problem.

XGBoost uses K additive trees to create the ensemble model as shown in Equation (10)

$$\hat{y} = \hat{f}(x) = \sum_{i=0}^{K} \hat{f}_l\ (x),\ \hat{f}_l(x) \in F \tag{10}$$

where $F = f(x) = \omega_{q(x)}\left(q:\ R^m \to T,\ \omega \in R^T\right)$.

The tree structure that transfers an input to the relevant leaf index at which it finishes up is represented by q. The number of leaves on the tree is denoted by T. Each leaf of a regression tree consists of a continuous score. The score on the i^{th} leaf is represented by ωi.

Then, minimize the following regularized objective to learn the set of functions employed in the model as shown in Equations (11) and (12).

$$l(\varnothing) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{11}$$

where

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda||\omega|k|^2 \tag{12}$$

To minimize overfitting, the additional regularization term smooths out the final learned weights. Trees with deeper depth have more leaf nodes, which can lead to overfitting on the training data, as only a few samples end up in each leaf node. As a result, we impose a penalty for the amount of leaf nodes to limit the depth and overfitting. The objective reverts to classic gradient tree boosting when the regularization parameter is set to zero.

To minimize the following objective function, as shown in Equation (13), it is necessary to add f_t for the t^{th} iteration.

$$l^{(t)} = \sum_i l(\hat{y_i}^{t-1} + f_t(x_i), y_i) + \sum_t \Omega(f_k) \tag{13}$$

The extension of a real function $f(x)$ around a point $x = a$ is called a one-dimensional Taylor series, and it is given by the Equation (14).

$$f(x) = f(a) + f^{|}(a)(x-a) + \frac{f^{||}}{2!}(x-a)^2 + \ldots + \frac{f^n}{n!}(x-a)^n + \ldots, \tag{14}$$

Then, the function can be approximated as shown in Equation (15) by using second order approximation while disregarding higher-order terms.

$$l^{(t)} = \sum_i^n [\, l(\hat{\mathrm{y}}_\mathrm{i}^{\,t-1},\, y_i) + g_i f_t(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega\,(f_k) \tag{15}$$

To make the objective function simpler, delete the constant terms as depicted in Equation (16).

$$l^{(t)} = \sum_i^n [g_i f_t(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega\,(f_k) \tag{16}$$

Let $I_j = \{i \mid q(x_i = j)\}$ represent the instance collection of leaf j; that is, the collection of all the input data points that ended up in the j-th leaf node. So, for a particular tree, if one of the input data points ends up at the j-th leaf node after all of the decisions, then include it in the set I_j as shown below in Equation (17).

$$\begin{array}{rl} l^{(t)} = & \sum_i^n [g_i f_t(x_i) + \frac{1}{2} h_i f_i^2(\, x_i)] + \gamma T + \frac{1}{2}\lambda ||\omega||^2 \\ & = \sum_{i=1}^n [(\, \sum_{i \in I_j} g_i)\omega_\mathrm{j} + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda)\omega_\mathrm{j}^2] + \gamma T \end{array} \tag{17}$$

The optimal weight ω_j^* of leaf j for a fixed tree structure $q(x)$ can be determined by differentiating the previous equation with regard to w and equating to 0 as shown in Equation (18).

$$\omega_\mathrm{j}^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{18}$$

For the time being, let us assume that we have a tree structure q for which we have determined the best weights at each leaf node. If you look at the above ω_j^* equation, you will note that the leaf nodes are missing, i.e., I_j has not been calculated yet. The following step is used to locate the best tree structure that minimizes the loss, and then we would be finished with our tree search.

Now, replace ω_j^* in the preceding equation as shown below in Equation (19) with the appropriate optimal value for the provided tree structure q.

$$l^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{19}$$

The quality "q" of a tree structure can be measured using the above equation. The score is very much similar to that of impurity score in terms of evaluating trees, but it is calculated for a broader variety of objective functions. Generally, it is difficult to list all of the feasible tree architectures q. Instead, a greedy algorithm is used, which starts with a single leaf and later iteratively adds branches to the tree. For instance, assume that IL and

IR are the left and right node instance sets after the split. If we assume I = IL∪IR, the loss decrease after the split is shown in Equation (20).

$$\begin{aligned} l_{split} &= \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i+\lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i+\lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i+\lambda}\right] - \gamma(t+1-t) \\ &= \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i+\lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i+\lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i+\lambda}\right] - \gamma \end{aligned} \tag{20}$$

i.e., Total loss after splitting minus total loss prior to splitting.

Similar to the Gini index or entropy, this score can be used to evaluate split candidates.

XGBoost uses well-known metrics, namely Mean Squared Error (MSE), as shown in (21), and Mean Absolute Error (MAE), as shown in (22), to assess a regression model's performance.

$$MSE = \frac{\sum_{i=1}^{n}\left(y_i - y_i^p\right)^2}{n} \tag{21}$$

$$MAE = \frac{\sum_{i=1}^{n}\left|y_i - y_i^p\right|}{n} \tag{22}$$

As discussed earlier, XGBoost computes the Gain of the root node by making use of Similarity Score as shown in Equations (23)–(25), respectively.

$$SimilarityScore = \frac{(\sum \text{Residual}_\text{i})^2}{\sum[\text{Previous Probability}_\text{i} * (1 - \text{Previous Probability}_\text{i})] + \lambda} \tag{23}$$

$$Gain = \text{Left}_{\text{Similarity}} + \text{Right}_{\text{Similarity}} - \text{Root}_{\text{Similarity}} \tag{24}$$

$$\text{Output Value} = \frac{(\sum \text{Residual}_\text{i})}{\sum[\text{Previous Probability}_\text{i} * (1 - \text{Previous Probability}_\text{i})] + \lambda} \tag{25}$$

4. Results and Discussion

Figure 6 displays Area under the (ROC) Curve. The LR, SVM, DT, RF, KNN, and XGBoost models are represented by lines of a different color in the figure. The black curve's AUC value of XGBoost is somewhat higher than that of the other four curves, indicating that it has the best performance out of the five models.

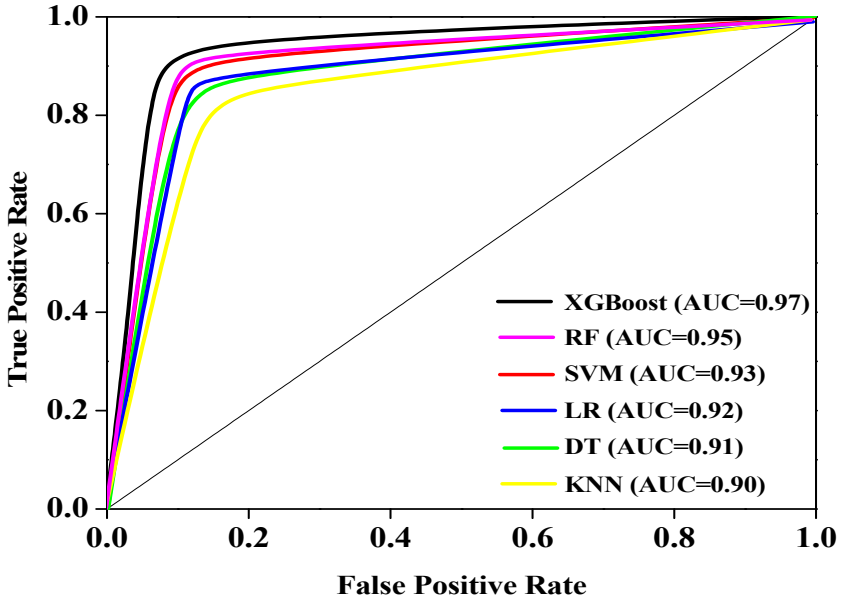


Figure 6. AUC for different ML algorithms.

Initially, the few vital steps taken in the preprocessing stages are evaluated. The essential process is to assess the estimation accuracy of mean and mode methods in identifying the missing values over 400 considered instances. Thus, Figure 7a,b depicts the resultants of the process in comparison with the randomized manual verifications.

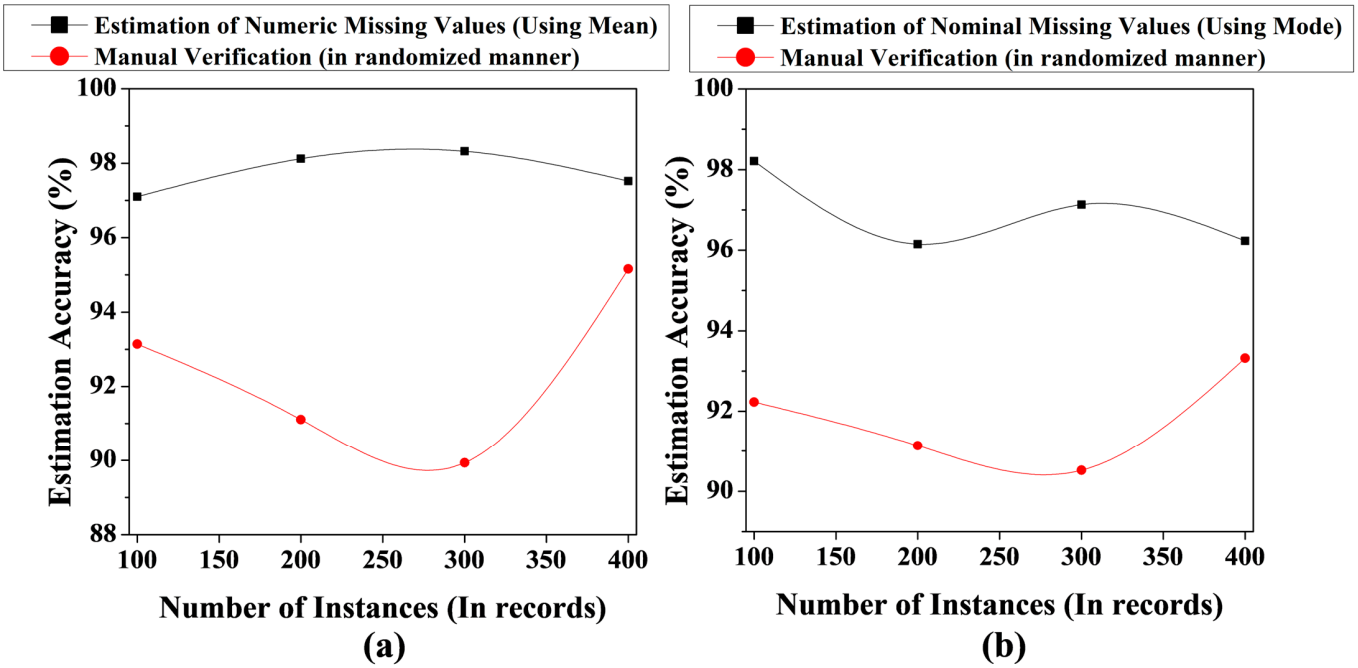


Figure 7. Missing value estimation accuracy using mean and mode method.

The results precisely state that the average estimation accuracy of both mean and mode methods in identifying numerical and nominal missing values are 97.77% and 96.93%, respectively, which is 5.71% more accurate than the randomized manual verifications. The values in the confusion matrix for all ML algorithms being implemented is shown in Figure 8.

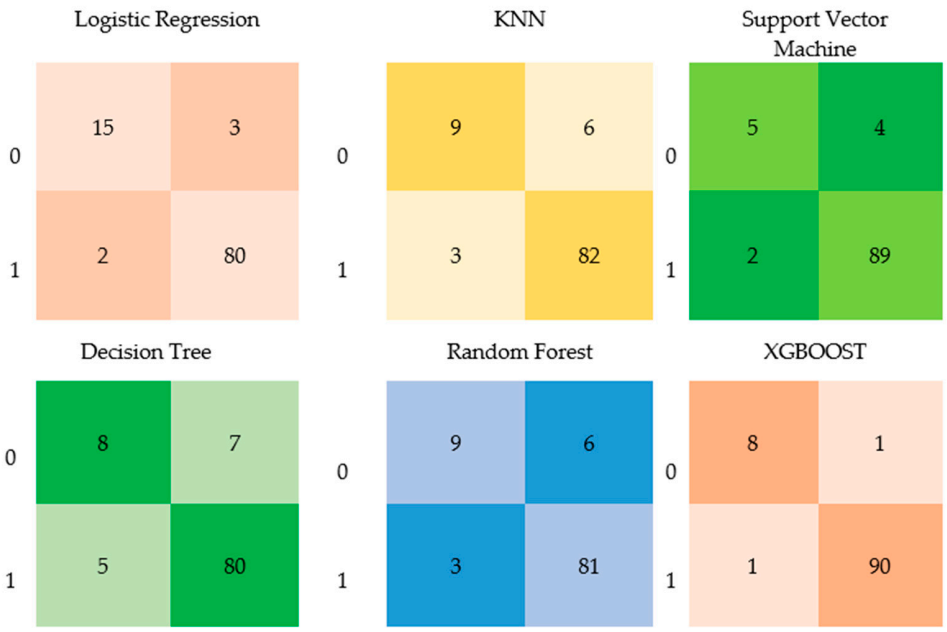


Figure 8. ML algorithms—Confusion matrix.

Precision indicates what amount of the total positive anticipated is genuinely positive. The ratio of true positives (TPs) to total number of TPs and true negatives are called precision (TNs). Precision is a good statistic to utilize when the cost of false positives (FNs) is high. Using Equation (26), the precision is calculated.

$$\text{Precision} = \frac{\text{TPs}}{\text{TPs} + \text{FPs}} \tag{26}$$

The precision values of LR, SVM, KNN, DT, RF, and XGBoost classifiers are 96.3%, 95.69%, 91.18, 91.95, 93.10%, and 98.9%, respectively.

Recall quantifies how many right positive predictions were made. The equation used to calculate the recall is (27) shown below.

$$\text{Recall} = \frac{\text{TPs}}{\text{TPs} + \text{FNs}} \tag{27}$$

Recall of LR, SVM, KNN, DT, RF, and XGBoost classifiers are 97.56%, 97.8%, 96.47%, 94.11%, 96.42%, and 98.9% respectively.

The F1-score considers both recall and precision. F1-score is computed using the below Equation (28).

$$\text{F1 Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{28}$$

The F1-score of the SVM, LR, DT, KNN, RF, and XGBoost classifiers is 96.73%, 96.93%, 93.01%, 94.75%, 94.73%, and 98.9%, respectively. The performance evaluation of these ML algorithms is shown in Figure 9.

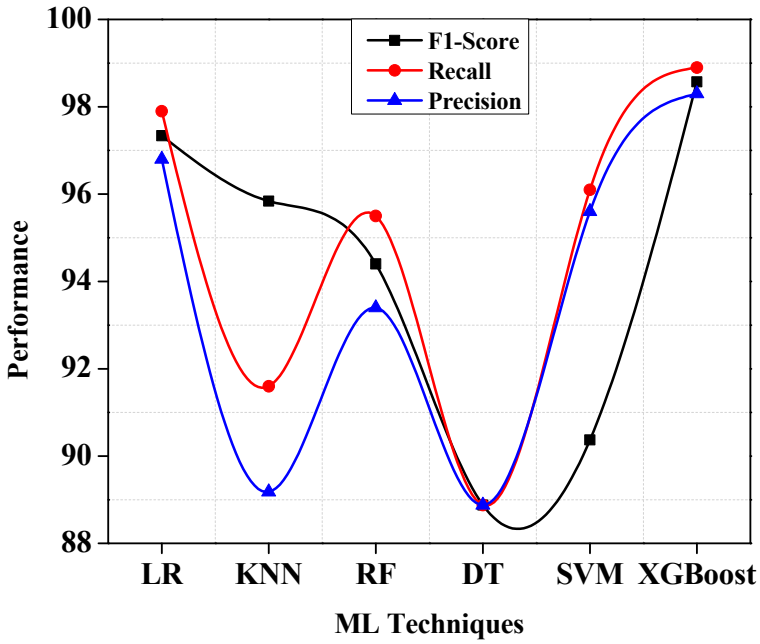


Figure 9. Different ML techniques—Performance evaluation.

One technique to determine how often an ML classification system properly classifies a record or instance is looking at accuracy. The Equation (29) is used to compute accuracy.

$$\text{Accuracy} = \frac{\text{TNs} + \text{TPs}}{\text{TNs} + \text{TPs} + \text{FPs} + \text{FNs}} \tag{29}$$

The performance of different ML Techniques for CKD classification is shown in Table 5 in terms of percentage.

Table 5. Performance in classification of CKD.

ML Techniques	Accuracy	Precision	Recall	F1-Score
LR	95.00	96.30	97.56	96.93
KNN	91.00	91.18	96.47	94.75
RF	90.00	93.10	96.42	94.73
DT	88.00	91.95	94.11	93.01
SVM	94.00	95.69	97.80	96.73
XGBoost (EL Method)	98.00	98.90	98.90	98.90

Table 6 shows the comparison of XGBoost performance with other ensemble techniques and XGBoost again outperforms with an accuracy of 98.33%.

Table 6. Performance comparison with other ensemble methods.

ML Techniques	Accuracy	Precision	Recall	F1-Score
Boosting	95.44	94.22	96.66	98.01
Bagging	93.62	91.22	92.55	94.44
Voting	94.66	94.70	96.77	95.34
Rotation Forest	91.20	91.88	92.24	90.40
XGBoost (EL Method)	98.00	98.90	98.90	98.90

Figure 10 exposes the superior performance of the proposed EL method during the testing phase in the classification of CKD. Such test validation is required to monitor the development of the trained algorithm and fine-tune or maximize its performance. Therefore, the learnt logic of the EL technique is regularly evaluated on a sample size of about 100 instances, and the findings (average accuracy is 97.78%) acquired provide concise evidence that the system maintains ideal consistency and precision.

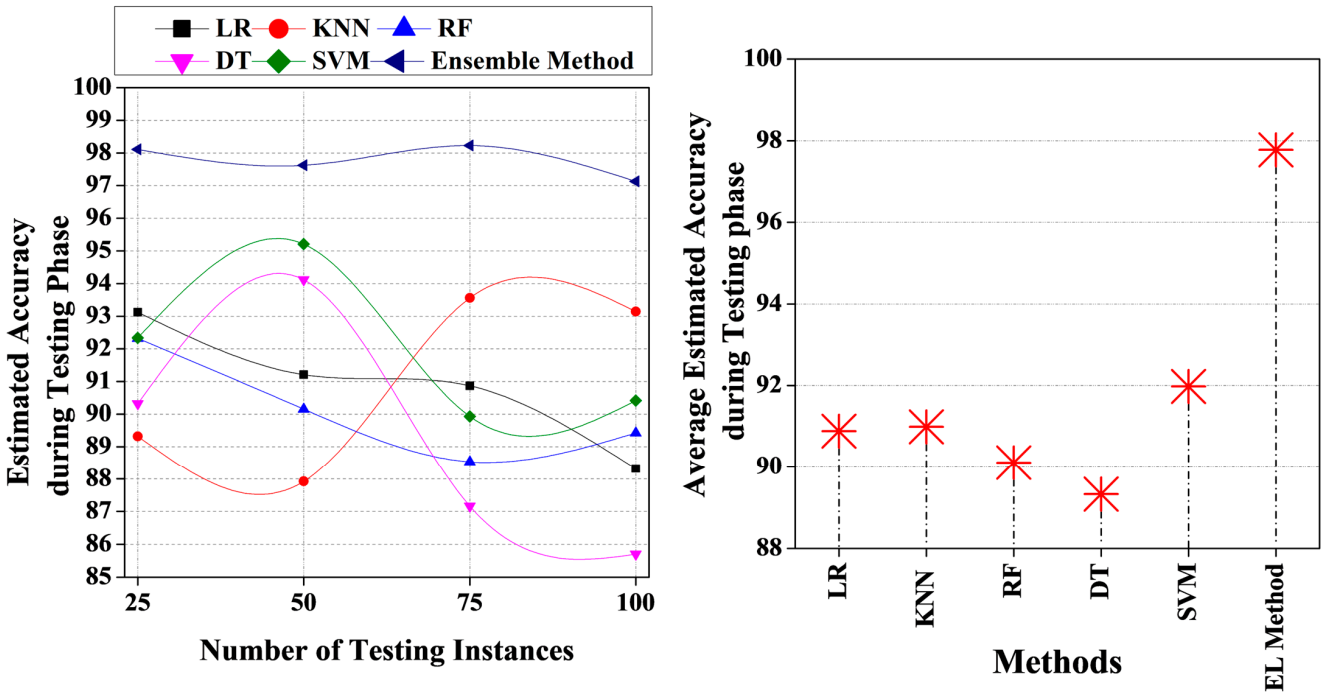


Figure 10. Evaluation of various ML methods during testing phase.

This accuracy provides the correctly classified instances of CKD. The accuracy of different ML techniques in the classification of CKD is shown in Figure 11. It is noted that XGBoost exhibits the best prediction with an accuracy of 98.93% and outperforms

compared to other algorithms in classifying the instances of the records of the dataset into "ckd" and "notckd."

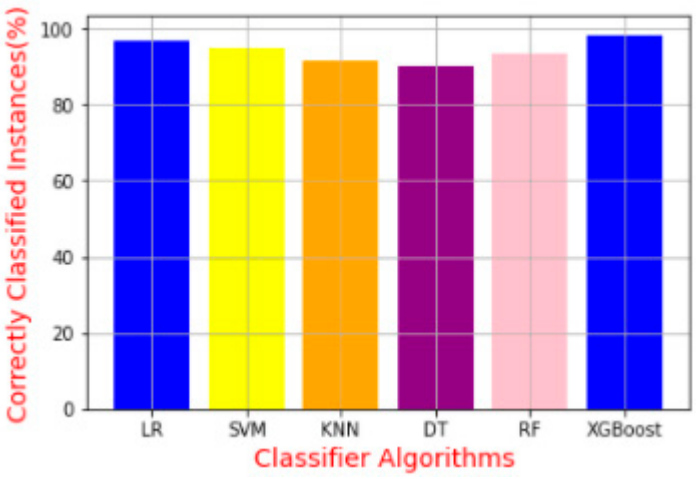


Figure 11. Accuracy of different ML techniques in classification of CKD.

All the performance measures reported in this study are statistically different compared to the other studies, as shown in Table 7.

Table 7. Comparison of the outcomes of this study with other studies found in the literature.

Study and Year	Sampling Strategy	Accuracy
Ifraz et al. (2021) [32]	70–30% training-testing	97.00%
S. Y. Yashfi et al. (2020) [33]	10-fold cross-validation	97.12%
Cheng-Sheng Yu et al. (2020) [34]	10-fold cross-validation	90.40%
Kovesdy et al. (2018) [35]	70–30% training-testing	95.0%
Proposed model	75–25% training-testing	98.00%

5. Conclusions

This study sheds light on how CKD patients might be diagnosed and treated early on in the disease's progression. A total of 400 patients contributed to the dataset, which included 24 distinct characteristics. After the estimation of missing values using the mean and mode methods, KNN imputation is employed to fill in the missing values. Later, the RFE process is applied to select the desirable attribute of CKD, thus enhancing the data quality in the preprocessing stages. To classify features, KNN, SVM, RF, DT, LR, and XGBoost were used. All of the classifiers' parameters were tweaked to achieve the best classification results, and all of the algorithms produced promising results. The proposed model also demonstrated decreased entropy by showing very less impurity measure. All other algorithms were outperformed by the XGBoost technique, which achieved the best precision, recall, F1-score, and accuracy of 98.90%, 98.90%, 98.90%, and 98.00%, respectively. Future work is planned to include the deep learning method, especially to deal with raw CKD informative image datasets.

Author Contributions: Conceptualization, M.T.R., V.K.V. and R.T.; methodology, M.T.R. and M.H.; software, M.H.; validation, V.K.V. and I.I.; formal analysis, V.K.V.; investigation, M.T.R.; resources, I.I.; data curation, M.H.; writing—original draft preparation, M.T.R.; writing—review and editing, M.T.R., V.K.V. and I.I.; visualization, V.K.V.; supervision, R.T.; project administration, I.I.; funding acquisition, I.I. All authors have read and agreed to the published version of the manuscript.

Funding: The National Research Foundation of Ukraine funded this research under project number 2021.01/0103.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting this study's findings are openly available in [6].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Im, C.-G.; Son, D.-M.; Kwon, H.-J.; Lee, S.-H. Tone Image Classification and Weighted Learning for Visible and NIR Image Fusion. *Entropy* **2022**, *24*, 1435. [CrossRef]
2. Jha, K.K.; Jha, R.; Jha, A.K.; Hassan, A.M.; Yadav, S.K.; Mahesh, T. A Brief Comparison On Machine Learning Algorithms Based On Various Applications: A Comprehensive Survey. In Proceedings of the 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 16–18 December 2021; pp. 1–5. [CrossRef]
3. Roopashree, S.; Anitha, J.; Mahesh, T.; Kumar, V.V.; Viriyasitavat, W.; Kaur, A. An IoT based authentication system for therapeutic herbs measured by local descriptors using machine learning approach. *Measurement* **2022**, *200*, 111484, ISSN 0263-2241. [CrossRef]
4. Dan, Z.; Zhao, Y.; Bi, X.; Wu, L.; Ji, Q. Multi-Task Transformer with Adaptive Cross-Entropy Loss for Multi-Dialect Speech Recognition. *Entropy* **2022**, *24*, 1429. [CrossRef]
5. Yang, Y.; Tian, Z.; Song, M.; Ma, C.; Ge, Z.; Li, P. Detecting the Critical States of Type 2 Diabetes Mellitus Based on Degree Matrix Network Entropy by Cross-Tissue Analysis. *Entropy* **2022**, *24*, 1249. [CrossRef] [PubMed]
6. "UCI Machine Learning Repository" [Online]. Available online: https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease (accessed on 28 August 2022).
7. Chronic Kidney Disease Overview. Available online: https://www.webmd.com/a-to-z-guides/tc/chronickidney-disease-topic-overview (accessed on 24 February 2018).
8. Mahesh, T.R.; Vinoth Kumar, V.; Muthukumaran, V.; Shashikala, H.K.; Swapna, B.; Guluwadi, S. Performance Analysis of XGBoost Ensemble Methods for Survivability with the Classification of Breast Cancer. *J. Sens.* **2022**, *2022*, 1–8. [CrossRef]
9. Gunarathne, W.H.S.D.; Perera, K.D.M.; Kahandawaarachchi, K.A.D.C.P. Performance Evaluation on Machine Learning Classification Techniques for Disease Classification and Forecasting through Data Analytics for Chronic Kidney Disease (CKD). In Proceedings of the 2017 IEEE 17th International Conference on Bioinformatics and Bioengineering, Washington, DC, USA, 23–25 October 2017.
10. Gowramma, G.S.; Mahesh, T.R.; Gowda, G. An automatic system for IVF data classification by utilizing multilayer perceptron algorithm. *ICCTEST-2017* **2017**, *2*, 667–672.
11. Kim, D.-H.; Ye, S.-Y. Classification of chronic kidney disease in sonography using the GLCM and artificial neural network. *Diagnostics* **2021**, *11*, 864. [CrossRef]
12. Ramya, S.; Radha, N. Diagnosis of Chronic Kidney Disease Using Machine Learning Algorithms. *Proc. Int. J. Innov. Res. Comput. Commun. Eng.* **2016**, *4*, 812–820.
13. Mahesh, T.R.; Kaladevi, A.C.; Balajee, J.M.; Vivek, V.; Prabu, M.; Muthukumaran, V. An Efficient Ensemble Method Using K-Fold Cross Validation for the Early Detection of Benign and Malignant Breast Cancer. *Int. J. Integr. Eng.* **2022**, *14*, 204–216. [CrossRef]
14. Ahmed, S.; Kabir, M.T.; Mahmood, N.T.; Rahman, R.M. Diagnosis of kidney disease using fuzzy expert system. In Proceedings of the 8th International Conference on Software, Knowledge, Information Management and Applications Journal of Healthcare Engineering (SKIMA 2014), IEEE, Dhaka, Bangladesh, 18–20 December 2014; pp. 1–8.
15. Khamparia, A.; Saini, G.; Pandey, B.; Tiwari, S.; Gupta, D.; Khanna, A. KDSAE: Chronic kidney disease classification with multimedia data learning using deep stacked auto encoder network. *Multimed. Tools Appl.* **2019**, *79*, 35425–35440. [CrossRef]
16. Mahesh, T.R.; Vivek, V.; Kumar, V.V.; Natarajan, R.; Sathya, S.; Kanimozhi, S. A Comparative Performance Analysis of Machine Learning Approaches for the Early Prediction of Diabetes Disease. In Proceedings of the 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 28–29 January 2022; pp. 1–6. [CrossRef]
17. Yildirim, P. Chronic Kidney Disease Prediction on Imbalanced Data by Multilayer Perceptron: Chronic Kidney Disease Prediction. In Proceedings of the 41st IEEE International Conference on Computer Software and Applications (COMPSAC), IEEE, Turin, Italy, 4–8 July 2017. [CrossRef]
18. Sarveshvar, M.R.; Gogoi, A.; Chaubey, A.K.; Rohit, S.; Mahesh, T.R. Performance of different Machine Learning Techniques for the Prediction of Heart Diseases. In Proceedings of the 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS), Bengaluru, India, 21–22 December 2021; pp. 1–4. [CrossRef]
19. Ramakrishna, M.T.; Venkatesan, V.K.; Izonin, I.; Havryliuk, M.; Bhat, C.R. Homogeneous Adaboost Ensemble Machine Learning Algorithms with Reduced Entropy on Balanced Data. *Entropy* **2023**, *25*, 245. [CrossRef]
20. Chang, Y.; Chen, X. Estimation of Chronic Illness Severity Based on Machine Learning Methods. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1–13. [CrossRef]
21. Soni, V.D. Chronic disease detection model using machine learning techniques. *Int. J. Sci. Technol. Res.* **2020**, *9*, 262–266.
22. Chaudhary, A.; Garg, P. Detecting and Diagnosing a Disease by Patient Monitoring System. *Int. J. Mech. Eng. Inf. Technol.* **2014**, *2*, 493–499.

23. Mahesh, T.R.; Sivakami, R.; Manimozhi, I.; Krishnamoorthy, N.; Swapna, B. Early Predictive Model for Detection of Plant Leaf Diseases Using MobileNetV2 Architecture. *Int. J. Intell. Syst. Appl. Eng.* **2023**, *11*, 46–54.
24. Mahesh, T.R.; Kumar, V.V.; Vivek, V.; Raghunath, K.M.K.; Madhuri, G.S. Early predictive model for breast cancer classification using blended ensemble learning. *Int. J. Syst. Assur. Eng. Manag.* **2022**. [CrossRef]
25. Venkatesan, V.K.; Ramakrishna, M.T.; Batyuk, A.; Barna, A.; Havrysh, B. High-Performance Artificial Intelligence Recommendation of Quality Research Papers Using Effective Collaborative Approach. *Systems* **2023**, *11*, 81. [CrossRef]
26. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'16, New York, NY, USA, 13–17 August 2016; pp. 785–794.
27. Mahesh, T.R.; Kumar, D.; Vinoth Kumar, V.; Asghar, J.; Mekcha Bazezew, B.; Natarajan, R.; Vivek, V. Blended Ensemble Learning Prediction Model for Strengthening Diagnosis and Treatment of Chronic Diabetes Disease. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–9. [CrossRef]
28. Ilyas, H.; Ali, S.; Ponum, M.; Hasan, O.; Mahmood, M.T.; Iftikhar, M.; Malik, M.H. Chronic kidney disease diagnosis using decision tree algorithms. *BMC Nephrol.* **2021**, *22*, 273. [CrossRef]
29. Sabanayagam, C.; Xu, D.; Ting, D.S.W.; Nusinovici, S.; Banu, R.; Hamzah, H.; Lim, C.; Tham, Y.-C.; Cheung, C.Y.; Tai, E.S.; et al. A deep learning algorithm to detect chronic kidney disease from retinal photographs in community-based populations. *Lancet Digit. Health* **2020**, *2*, e295–e302. [CrossRef]
30. Poonia, R.C.; Gupta, M.K.; Abunadi, I.; Albraikan, A.A.; Al-Wesabi, F.N.; Hamza, M.A. Intelligent Diagnostic Prediction and Classification Models for Detection of Kidney Disease. *Healthcare* **2022**, *10*, 371. [CrossRef] [PubMed]
31. Elhoseny, M.; Shankar, K.; Uthayakumar, J. Intelligent Diagnostic Prediction and Classification System for Chronic Kidney Disease. *Sci. Rep.* **2019**, *9*, 9583. [CrossRef] [PubMed]
32. Ifraz, G.M.; Rashid, M.H.; Tazin, T.; Bourouis, S.; Khan, M.M. Comparative Analysis for Prediction of Kidney Disease Using Intelligent Machine Learning Methods. *Comput. Math. Methods Med.* **2021**, *2021*, 6141470. [CrossRef] [PubMed]
33. Yashfi, S.Y.; Islam, A.; Pritilata; Sakib, N.; Islam, T.; Shahbaaz, M.; Pantho, S.S. Risk Prediction of Chronic Kidney Disease Using Machine Learning Algorithms. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; pp. 1–5. [CrossRef]
34. Yu, C.S.; Lin, Y.J.; Lin, C.H.; Wang, S.T.; Lin, S.Y.; Lin, S.H.; Wu, J.L.; Chang, S.S. Predicting metabolic syndrome with machine learning models using a decision tree algorithm: Retrospective cohort study. *JMIR Public Heal. Surveill.* **2020**, *8*, e17110. [CrossRef]
35. Kovesdy, C.P.; Matsushita, K.; Sang, Y.; Brunskill, N.J.; Carrero, J.J.; Chodick, G.; Hasegawa, T.; Heerspink, H.L.; Hirayama, A.; Landman, G.W.; et al. Serum potassium and adverse outcomes across the range of kidney function: A CKD Prognosis Consortium meta-analysis. *Eur. Heart J.* **2018**, *39*, 1535–1542. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.