Comparative Analysis of Machine Learning Methods to Detect Chronic Kidney Disease

Madhusree Sankar Roy¹, Ritama Ghosh¹, Diyali Goswami¹, Karthik R²*

¹School of Electronics Engineering, Vellore Institute of Technology, Chennai, India
²Centre for Cyber Physical Systems, Vellore Institute of Technology, Chennai, India
r.karthik@vit.ac.in

Abstract. Chronic Kidney Disease (CKD) is a lifelong health hazard that can cause the failure of kidneys. Symptoms of this develop slowly and are not obvious. Early detection of Chronic Kidney Disease can lead to significant progress in finding the cure for this disease. Through this study, we aim to employ ML techniques for the prediction and diagnosis of Chronic Kidney Disease. The findings obtained from our predictive analysis combined with the expertise of healthcare professionals can help in making an accurate prognosis. For this, we have used a dataset containing data from 400 individuals acquired from the University of California Irvine (UCI) repository. Various feature selection techniques have been used to optimize the number of features affecting Chronic Kidney Disease. Subsequently, these desirable features are chosen and used in different ML models and their accuracy, sensitivity is compared. Multiple Machine learning algorithms have been explored such as Logistic Regression, Naïve Bayes, KNN, SVM, Decision Trees, Random Forest Classifier, and Extra Trees Classifier. It was concluded that Decision Trees using information gain gave six optimal features and the Extra Trees Classifier model gives the best accuracy of 99.36% with Extra Trees Classifier having one of the least execution times.

1. Introduction
Chronic Kidney Disease can lead to complete kidney failure which may result in the need for a kidney transplant. It has become burdensome for the healthcare system due to the growing number of patients, elevated risk of progression to end-stage renal disease, and substandard diagnosis of morbidity and mortality. The highest prevalence of Chronic Kidney Disease has been seen in Asia [1]. From studies, it is seen that the prevalence of CKD is 15% with stage 1 as 6.6%, stage 2 as 5.4%, and stage 3 as 3.02% [2]. The prevalence of CKD in India is no less than in developed countries like the USA. People suffering from diabetes and hypertension are often unaware of the risks with which they are living. As per the findings published in the AJKD, out of the 6.5 million national pediatric hospital discharges, nearly 4% of those children are affected with CKD. It was observed that 50% of children with CKD were likely to die while hospitalized [3]. Researchers have even found that people suffering from Chronic Kidney Disease face a higher risk of mortality due to COVID-19 [4].

In general, CKD is underdiagnosed and kidney biopsy is an invasive procedure with little spatial coverage. Various medical procedures are carried out like testing of creatinine in blood, albumin, and levels of protein in urine for detection. These are then followed by a CT scan, MRI scan, and ultrasound scan as well. These are expensive and laborious procedures.
The healthcare industry has always generated and documented a multitude of medical data, by maintaining records, various compliance & regulatory rules, and general patient care. Governments and companies alike are employing data analytics to fight cancer, Parkinson’s disease, Ebola, and many more diseases with the help of this data. Data science can enable us to diagnose, track, and treat diseases worldwide. Using predictive analysis in machine learning can help us to identify the risk patterns in the occurrence of CKD. It can even assist in providing customized and personalized healthcare solutions for every patient. In this work, a comparative analysis of various Machine learning Algorithms with an optimum number of features was tried out for better diagnosis of CKD.

2. Related Works
Multiple IEEE research papers published in the last few years in this domain were collected and analyzed. Hodneland et al. applied image analysis methods to a dynamic T1-weighted MR sequence to detect renal morphologic changes [11]. Tun-Wen Pai et al. displayed a computer-aided instrument based on analyzing ultrasonography pictures [12]. They created a framework that might distinguish and classify distinctive stages of Chronic disease utilizing image processing and K-means clustering strategies. Jiongming Qin et al. generated models using LOGR, RF, SVM, KNN, NB, and FNN [10]. They finally proposed an integrated model that combined logistic regression and random forest classifiers by using perceptron. The model promised to detect CKD for patients with some missing clinical data and give better accuracy of almost 99.83%. Charleonnan et al. build predictive models using KNN, SVM, logistic regression, and decision tree classifiers [9]. They conclude their study by presenting a model using an SVM classifier. The model gave a 98.3% accuracy and 0.99 sensitivity.

Salekin and Stankovic evaluated the performances of three models-KNN, Random Forest and Neural Networks [8]. They reduced the features from 24 to 12 using the wrapper method and LASSO regularization. Building a random forest classifier using these 12 features, the greatest accuracy of .998 was attained using the F1-measure and 0.107 rmse. They further identified a cost-effective model that attained high accuracy using only 5 attributes using cost analysis by utilizing all 24 attributes. Amirgaliyev et al. used a linear kernel SVM with all the features [5]. The accuracy, specificity, and sensitivity values were tabulated which then informed that the overall performance of SVM is 94.602%. The authors Tazin, Sabab, and Chowdhury used four classification algorithms namely KNN, Decision Tree, Naive Bayes, and SVM [6]. They used 15 attributes in the Decision tree obtaining an accuracy of 99%. Kunwar, Chandel, Sabitha, and Bansal used data mining. The performances of ANN and Naive Bayes were compared using the Rapidminer tool [7]. It was seen that Naive Bayes gave the highest accuracy. Rady and Anwar compared the accuracy and execution times of PNN, SVM, RBF, and MLP to classify the stages of patients who suffered from CKD [13]. The conclusion drawn was that PNN had the highest overall accuracy percentage (96.7%). But, MLP had the least execution time (3s). Aljaaf et al. compared the prediction of CKD with the help of the ML methods RPART, SVM, LOGR, and MLP [14]. The performance metrics and execution time were compared. The MLP model had the maximal True Positive Rate (0.995) and area Under Curve. The RPART model however achieved the maximum True Negative Rate of 1.00. The lowest type 1 error was shown by the MLP model making it the best model in comparison to the others.

3. Proposed System
3.1. Data Acquisition
The CKD data set has been obtained from the UCI gadget learning repository which incorporates a total of 400 cases, out of which 250 of the instances are patients with CKD and the rest 150 are not. The target variable shows whether or not an affected person has a CKD or not. There are 25 attributes within the dataset. The 11 numerical attributes encompass age, bp, bgr, bu, sc, sod, pot, hemo, pcv, wc, rc. The 14 nominal attributes include specific gravity, albumin, sugar, white blood cell count, pus cell, pus cell clumps, bacteria, hypertension, diabetes mellitus, coronary artery disease, appetite, pedal edema, anemia, and class. Parameters that were taken from various blood tests are set at 41.7%. The
parameters extracted from urine tests comprise 29.15%. The ultimate set of parameters that consists of information about other scientific elements that can cause CKD represents 29.15%. Approximately 45% of all statistics units inside the UCI online have some types of missing data. Some parameters have almost 40% incomplete values.

3.2. Data Preprocessing

3.2.1. Missing Values. For missing data, the rows with more than 25% missing values were removed namely 'rbc', 'wc', and 'rc'. Next, the missing data from numerical columns - blood pressure, specific gravity, albumin, sugar, blood glucose random, blood urea, serum creatinine, sodium, potassium, hemoglobin, and packed cell volume were replaced with the numerical mean of the respective columns. The missing categorical values were replaced by taking the mode of each column.

3.2.2. Outliers. Outliers are those data points that are extremely distant from all the other data points in the dataset. Most machine learning algorithms consider their datasets to be gaussian in nature. To avoid this misconception, we checked the skewness. Any value which was not between -1 to +1 indicated the presence of outliers. We used boxplots to visualize the outliers present in each attribute of our dataset. Finally, we treated these outliers by Quantile-based Flooring and Capping. We did 10th percentile flooring for the lower values and 90th percentile capping for the higher values.

3.2.3. Categorical Columns. The categorical parameters that needed to be encoded were - pus cell, pus cell clumps, bacteria, hypertension, diabetes mellitus, coronary artery disease, appetite, pedal edema, anemia, and classification. All values of these columns had to be assigned binary measurements. This was done using label encoding or integer encoding wherein each unique category value is mapped to an integer. This converted the values to a machine-readable format as the values now have a naturally ordered relationship with each other. The original dataset had three more categorical columns 'sg', 'al', and 'su'. But, values of these were numerical, hence these variables were left unchanged.

3.3. Feature Selection

There are 24 attributes in this dataset but not all have a major influence on the prediction. We will employ different feature selection algorithms from each approach to get the best subset of the important features to be used in our model generation.

3.3.1. Wrapper Approach - Recursive Feature Elimination. It is a backward selection of predictors. The algorithm starts by fitting the model with all the attributes. At each iteration, each attribute is ranked according to its importance in the model’s performance. The least contributing features are dropped. The model is refit and its performance is evaluated till we extract the most optimal subset of predictors. We used RFE with cross-validation for feature ranking. We used stratified 10-fold for cross-validation wherein it shuffles the data one time before splitting to ensure that each split is a representation of all strata of data. We can see that the subset of 15 features gives the highest accuracy. From the classification report, we obtained the mean accuracy score to be 98.7%.

3.3.2. Embedded Approach - Decision Trees. Embedded feature selection methods like decision trees integrate feature selection with model generation. The decision tree chooses an attribute at every recursive step of the tree building process and then divides the sample set into smaller subsets. When a tree is being built, the feature importance of each attribute is calculated as the node impurity weighted in a tree decreases. This is calculated based on mean decrease impurity and mean decrease accuracy. These methods not only consider the interaction of features like the wrapper methods but have been seen to give better prediction accuracy as well. The only drawback is that they cost a little more computation time in comparison to other methods. From the classification report, we got a mean accuracy of 99.56%. We obtained a subset of six features-['sg', 'al', 'sc', 'hemo', 'pcv', 'bgr'].

3
3.3.3. Filter Approach - ANOVA and Chi-Square. Univariate ANOVA is used after removing constant, quasi constant, and duplicate features. The f-test for training and testing of the model results in 97.4% accuracy. The univariate selection depicts the relationship between the independent features and the output feature. The chi-square test is used and the selection of 10 best features results in 95.64% accuracy. As our aim is classification, we use logistic regression and print the accuracy for each method used.

3.4. Model Generation

Different classification algorithms have been used for model generation in order to select the model with the best performance and execution time. The models that have been compared are Naive Bayes, Logistic Regression, K nearest neighbors, SVM, Decision Trees, Random Forest Classifier, and Extra Trees Classifier.

3.4.1. Naive Bayes. It works on the principle of the Bayes theorem. The types of Naive Bayes models given in the scikit-learn library are: Gaussian, Multinomial, and Bernoulli Naive Bayes. The Bernoulli Naive Bayes gives us an accuracy of 80.254%. Multinomial gives 90.446% and Gaussian gives 96.815%.

3.4.2. Logistic Regression. It is a supervised classification algorithm that predicts the probability of an output. We observe that logistic regression gives an accuracy of 97.187%.

3.4.3. K nearest neighbors. It is a lazy learning algorithm that considers all the available cases and classifies new cases based totally on a similarity or distance degree. Choosing the best value for K is done by examining the dataset. A large K value is more beneficial because it reduces the overall noise however there is no guarantee. Cross-validation is another way to retrospectively determine the K-
value. K-value for maximum datasets has been between 3-10. We take 6 selected features and find that taking K-value as 2 gives us the most optimum result with an accuracy of 98.29%.

3.4.4. Support Vector Machines. Support Vector Machine (SVM) is a machine learning algorithm (supervised) for finding an optimal boundary between different classes in a dataset [15]. SVM chooses extreme points called the support vectors to create a hyperplane. SVM uses a kernel technique to transform this data into high-dimensional separable classes. We have used SVM with linear, polynomial, RBF, and sigmoid kernels. The linear kernel is the most common and is highly effective when there is a large dataset. A linear kernel is used when data is linearly separable, that is using a single line. A polynomial kernel is a modified form of the linear kernel which gives a more flexible decision boundary in the case of non-linear data. Both linear and polynomial kernels tend to take less computation time than the other kernels. We generated kernels of degrees 2 to 8. But, there was not much improvement by increasing the degree. We used RBF and Sigmoid kernels as well. The best performance was given by the one with a polynomial kernel.

3.4.5. Decision Trees. Decision Trees use a top-down approach to form a tree-like structure. This algorithm recursively chooses the best split according to attribute selection measures (ASM). Each branch here is a decision rule with the leaf node as its outcome. This process continues until there are no more instances or attributes left. The ASM can be information gain, gain ratio, or Gini index. Decision Trees tend to overfit. Pruning is performed to avoid that and reduce the size of the decision trees by removing the classifiers which do not have much significance in classification. It was observed that on pruning the tree to a maximum depth of 5, the accuracy of the model improved by 2%. 10-fold cross-validation was used for evaluating the performance of our model.

3.4.6. Random Forest Classifier (RFC). Random forest works on the concept of bagging. Out of all observations in the training set, samples are taken at random and with replacement. These samples act as a training set for different models (decision trees). Every decision tree is then grown to the maximum extent possible. Like bagging, voting for classification and averaging for regression is used to make a final prediction. In RFC, greedy algorithms determine the split points. For RFC, n_estimators, max_features, max_depth, and min_samples_leaf hyperparameters were optimized. The maximum accuracy obtained was around 99.36%.

![Learning Curve for Random Forest Classifier](image)

**Figure 2.** Learning curve for Random Forest Classifier.
3.4.7. Extra Trees Classifier (ETC). Extra trees classifier is an ensemble method algorithm. In ETC, the training dataset is used to create numerous unpruned decision trees. Just like in bagging and RFC, voting for classification and averaging for regression is used to make a final prediction. In ETC, every decision tree is fitted on the complete dataset. This is different from RFC and bagging where each tree is fitted on a sample dataset. In ETC, splits are taken randomly. Like RFC, an optimized version of ETC has been used.

Bagging is an ensemble method that reduces the variance of an estimate by averaging together multiple estimates. Several models of the same algorithm with different subsets of data are used in a random manner. Bagging has been performed on all of the models mentioned but it did not result in any significant and/or consistent improvement in the performance of the models.

Figure 3. Learning curve for Extra Tress Classifier.
Figure 4. Extra Tree Classifier trees formation.

4. Result and Discussion

4.1. Environment Setup

Google colab environment was utilized. CPU runtime was used. It is an Intel Xeon Processor with two cores @ 2.20 GHz and a RAM capability of 13 GB.

4.2. Performance Analysis

The results of all the feature selection and model generation algorithms were compared. Cross-validation (10 fold) has been used to train all our models and obtain the average of all the performance metrics. We observed that the accuracy of the features selected using Decision Trees is the highest.
Having a high accuracy is of utmost importance as inaccurate predictions can be detrimental to the patient’s health. This subset of features - ['sg', 'al', 'sc', 'hemo', 'pcv', 'bgr'] seems to also have the highest precision of 98.5%. It seems to outperform all the other subsets in recall and f1-score as well. This subset has helped significantly in dimensionality reduction as well.

Out of all the models compared, Extra Trees Classifier(ETC) and Random Tree Classifier(ETC) gave the highest accuracy with the wall time of ETC being lesser than that of RFC. RFC and ETC outperformed most other models in Sensitivity and Precision as visualized below.

**Figure 5.** Comparison of Accuracy obtained.

**Figure 6.** Comparison of Specificity obtained.
The results of all these machine learning models are compared through bar graphs. Cross-validation (of 10 fold) has been used to test and train the data. The average accuracy comparison can be seen in figure 5. The accuracy of RFC and ETC came out to be the highest (both 99.36%) with KNN and Gaussian Naive Bayes closely following with accuracy of 99.14% and 98.30% respectively. RFC and ETC have consistently better performance compared to most other models when it comes to sensitivity, fallout, and precision too.
Table 1. Comparison of Wall Time for different classifiers.

| Models                        | Wall Time (ms) |
|-------------------------------|----------------|
| K Neighbors Classifier        | 16             |
| SVM Classifier                | 8              |
| Decision Trees                | 7.96           |
| Random Forest Classifier      | 286            |
| Random Forest Classifier (Optimized) | 159          |
| Extra Trees Classifier (Optimized) | 119          |

Table 1 shows the wall time comparison for different models. Even though RFC and ETC have similar accuracy, ETC has a significantly lower wall time thus making it more suitable for real-time applications.

5. Conclusion

All the machine learning algorithms give a satisfactory performance in terms of accuracy, sensitivity, fallout, and precision by using the subset of features obtained with the help of decision trees. On further comparing all the models, Random Forest Classifier and Extra Trees Classifier give the highest accuracy of 99.36%. With regards to time taken to train and predict, Extra Trees outperforms Random Forest. One limitation of these results can be the generalization of the performance because the dataset has only 400 samples. In the future, this proposed model can further be applied to other diseases as well. It will make the process of disease detection less invasive and more cost-efficient.

References

[1] Hasan, M., Sutradhar, I., Gupta, R.D.: Prevalence of chronic kidney disease in South Asia: a systematic review. BMC Nephrol 19, 291 (2018), https://doi.org/10.1186/s12882-018-1072-5
[2] Varma P. P.: Prevalence of chronic kidney disease in India - where are we heading?. Indian journal of nephrology, 25(3), 133–5 (2015)
[3] Modi,J.Z., Waldo,A., Selewski,D.T., Troost,J.P., Gipson,D.S.: Inpatient pediatric health care utilization and mortality in the United States. American Journal of Kidney Diseases, 12(10), (2020)
[4] Ajainy,M., Melamed, M. L.: COVID-19 in patients with kidney disease. CJASN, 15 (8) 1087-1089; doi: 10.2215/CJN.09730620
[5] Amirgaliyev,Y., Shamiltuluu,S., Serek,A.; "Analysis of chronic kidney disease dataset by applying machine learning methods," .In 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), Almaty, Kazakhstan, 2018, pp. 1-4, doi: 10.1109/ICAICT.2018.8747140.
[6] Tazin,N., Sabah,S.A ., Chowdhury,M.T.: Diagnosis of chronic kidney disease using effective classification andfeatureselection technique. IEEE International Conference on Medical Engineering, Health Informatics and Technology (MediTec).pp. 1-6. IEEE,Dhaka (2016); doi: 10.1109/MEDITEC.2016.7835365.
[7] Kunwar,V., Chandel,K., Sabitha, A. S., Bansal,A.: Chronic kidney disease analysis using data mining classification techniques. 6th International Conference - Cloud System and Big Data Engineering (Confluence), pp. 300-5. IEEE,Noida ( 2016); doi: 10.1109/CONFLUENCE.2016.7508132.
[8] Salekin, A., Stankovic, J.: Detection of chronic kidney disease and selecting important predictive attributes. In: 2016 IEEE International Conference on Healthcare Informatics (ICHI), pp. 262-270. IEEE, Chicago, IL (2016).

[9] Charleonnan, A., Fufaung, T., Niyomwong, T., Choa Chu Pattanakit, W., Suwannawach, S., Ninchawee, N.: Predictive analytics for chronic kidney disease using machine learning techniques. In: 2016 Management and Innovation Technology International Conference (MITicon), pp. MIT-80-MIT-83. IEEE, Bang-San (2016).

[10] Qin, J., Chen, L., Liu, Y., Liu, C., Feng, C., Chen, B.: A machine learning methodology for diagnosing chronic kidney disease. vol. 8, pp. 20991-21002. IEEE (2020).

[11] Hodneland, E., Keilegavlen, E., Hanson, E.A., Andersen, E., Monsen, J.A., Rorvik, J., Leh, S., Marti, H.P., Lundervold, A., Svarstad, E., Nordbotten, J.M.: In vivo detection of chronic kidney disease using tissue deformation fields from dynamic MR imaging. In: IEEE Trans Biomed Eng. 2019 Jun, vol. 66, no. 6, pp. 1779-90. IEEE (2019).

[12] Ho, C., et al.: Ultrasoundography image analysis for detection and classification of chronic kidney disease. In: 2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, pp. 624-9. IEEE, Palermo (2012), DOI: 10.1109/ECACE.2019.8679388

[13] Rady, E.A., Anwar, A.S.: Prediction of kidney disease stages using data mining algorithms. Informatics in Medicine Unlocked. vol. 15, pp. 100178 (2018); https://doi.org/10.1016/j.imu.2019.100178

[14] Aljaaf, A. J., et al.: Early prediction of chronic kidney disease using machine learning supported by predictive analytics. IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, 1-9 (2018); doi: 10.1109/CEC.2018.8477876.

[15] Awad, M., Khanna, R.: Support Vector Machines for Classification. In: Efficient Learning Machines. Apress, Berkeley, CA (2015), https://doi.org/10.1007/978-1-4302-5990-9_3