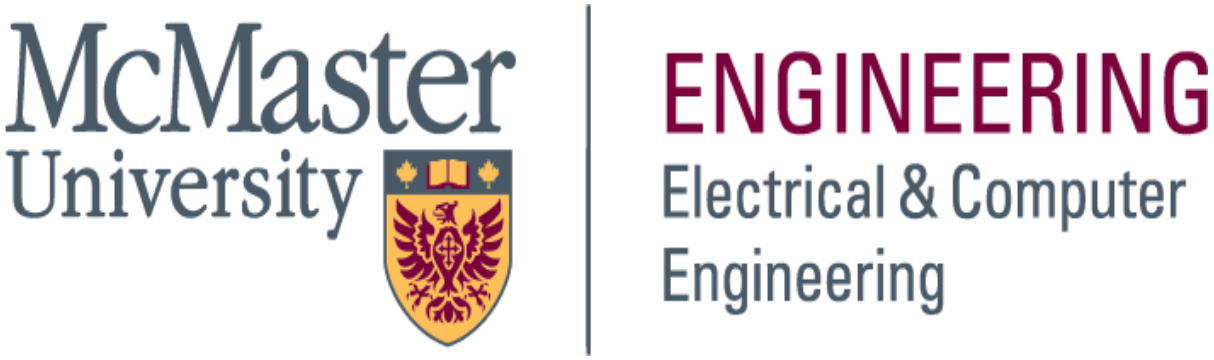

Project: Chronic Kidney Disease Diagnosis

ECE 718: Special Topics in Computation. Machine Learning: An Introduction

Faculty Advisor:
Dr. Sorina Dumitrescu
Department of Electrical and Computer Engineering
McMaster University

Prepared and submitted by:
Neelanjan Goswami
400414867
M.Eng. Electrical and Computer
McMaster University

INTRODUCTION

Kidney illnesses manifest themselves in a variety of ways all around the world. Chronic Kidney Disease (CKD), has no symptoms in its early stages, thus testing may be the only method to determine whether the patient has renal disease. Early identification of CKD in its early stages can assist patients to receive successful therapy and avoid the development of the disease into ESRD. The sooner people are aware of their illness, the sooner they may seek therapy. A cheap and fast Clinical Decision Support System (CDSS) for automatic diagnosis is particularly advantageous to battle illnesses due to the scarcity and expensive cost of professionals for manual disease diagnosis. Artificial Neural Networks (ANN), Support Vector Machine (SVM), Naive Bayes, decision tree, extreme gradient boosting one (XGBoost), logistic regression, and fuzzy set theory, when combined with medical expertise, go a long way. Lung cancer, TB, cardiovascular disease, and malaria have all been diagnosed using them. The rule-based and non-rule-based systems meant to do such a diagnosis task are divided into two categories. The primary objectives are:

1) The 80:20 distinction has been used to test the evaluation process of several models.
2) To provide more credible results, all missing value concerns were rectified using the K-Nearest Neighbors imputation approach.
3) All features are prepossessed with the ability to keep their values inside the range of [0, 1] using a conventional scaler approach.
4) This study also looks at accuracy, error rate, execution time, AUC, and ROC statistics to demonstrate the efficacy of many classifiers.

A patient with CKD who is in the acute phases responds to therapy and can be resurrected. Non-Dialysis-Dependent Chronic Kidney Disease (NDD-CKD) is the term for these acute phases, which range from stage one to stage four. Stage five, often known as End-Stage Kidney Disease, is the last stage (ESKD). Dialysis and careful care are required at this point for the patients. The white and Indian populations of South Africa and the rest of the globe are less affected by CKD than the African black population. A very accurate Clinical Decision Support System (CDSS) is urgently needed to correctly diagnose CKD. The number of persons who require kidney replacement due to CKD is

increasing every day. With a mortality rate of 6 percent. Furthermore, as the prevalence of CKD increases, specialists and the healthcare system face new challenges. As a result, the need of a preventive diagnostic predictive solution that enables early and precise identification of CKD cannot be overstated.

The suggested paradigm for this study is depicted in the following Fig. 1.

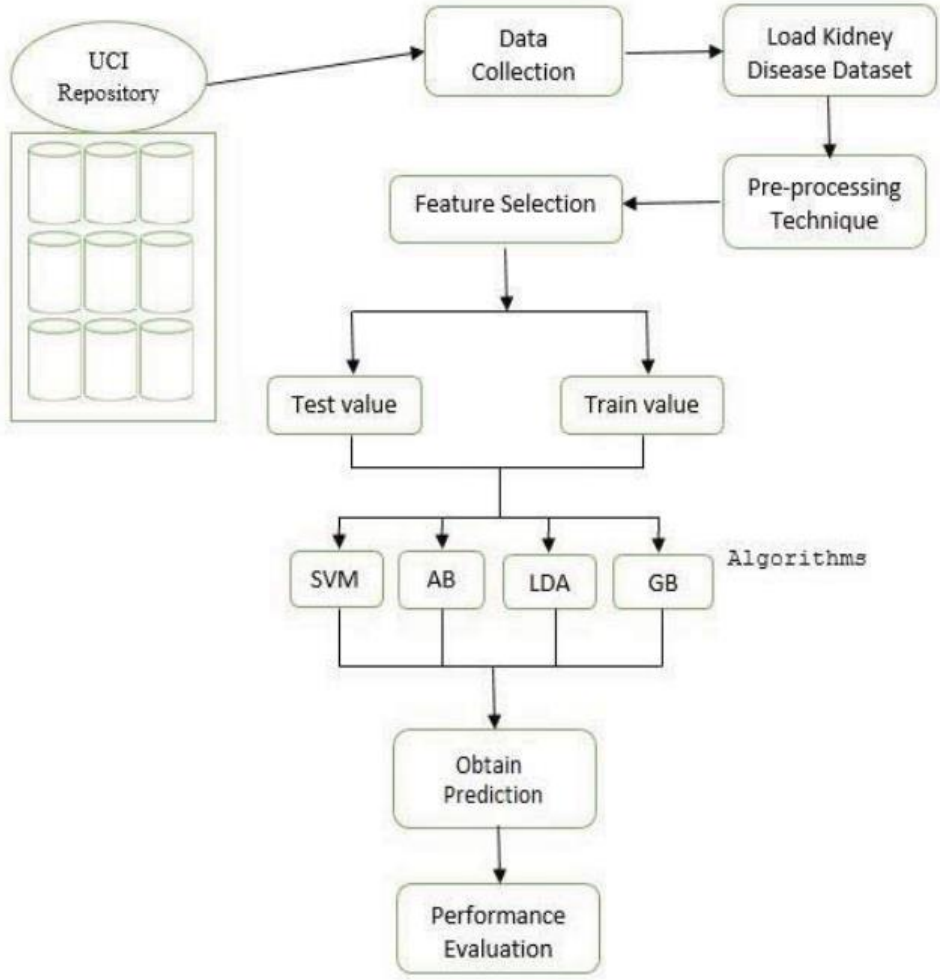

Figure 1: The suggested kidney disease detection mechanism [3].

All missing value problems were addressed utilizing the K-Nearest Neighbors imputation technique to offer more believable findings. Several models' assessment processes have been tested using the 80:20 distinction. All features are built with the ability to retain their values inside the range of a traditional scaler method. This study investigates accuracy, error rate, execution time, AUC, and ROC statistics to demonstrate the efficacy of numerous classifiers.

MACHINE LEARNING ALGORITHMS

The planned study's main goal is to develop a renal disease system that is totally based on machine learning. The goal of the study is to solve several algorithms, such as XGBoost, SVM, AB, LDA, and GB, in order to categorize persons with renal illness. This study uses various performance assessment metrics such as Accuracy (ACC), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), F1 Score (F1), Precision (PRE), Mean Squared Error (MSE), Standard Deviation (SD), Specificity (SPE), False Negative Rate (FNR), Sensitivity (SEN), Negative predictive value (NPV), False Discovery Rate (FDR) and False Positive Rate (FPR).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$f-measure = \frac{2\times precision \times recall}{precision \times recall}$$

1. XGBoost is a distributed gradient boosting toolkit that is optimised for efficiency, flexibility, and portability. It uses the Gradient Boosting framework to create ML algorithms. XGBoost is a parallel tree boosting (also known as GBDT, GBM) algorithm that addresses a variety of data science issues quickly and accurately. The same algorithm may tackle problems with billions of instances in a distributed environment (Hadoop, SGE, MPI).
2. Multilayer Perceptron (MLP) is one of the most extensively used neural networks. As illustrated in figure 2, MLP is made up of three types of layers, each of which is built up of artificial neurons and connected by weighted linkages. Some neurons will be triggered to a certain value depending on the weights and a specified value termed the activation value, while others will not. The activation pattern of one layer influences the activation pattern of the following layer.

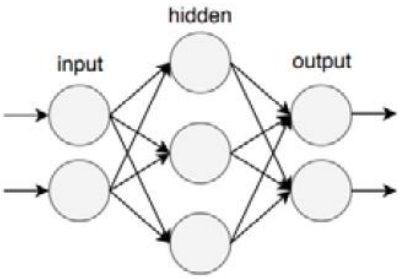

Figure 2: Basic Structure of MLP [2]

3. SVM also has been a popular choice among many academics when it comes to classification challenges since it consistently beats other classification methods. Furthermore, SVM works effectively even with a small number of samples. SVM has been utilized in a variety of applications, including optical character recognition, spam detection in email, and medical diagnostics.

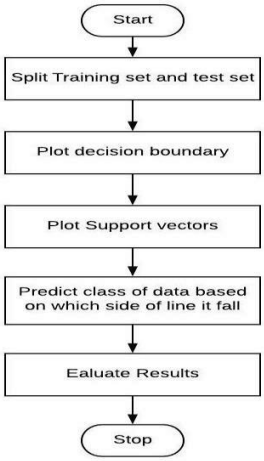

Figure 3: The working process of the SVM algorithm [3]

4. The Bayesian probability theorem is the foundation of the Naive Bayes classification method. The classifier is based on the Naive Bayes assumptions, which indicate that each feature's contribution to the final result is independent and equal, suggesting that the presence/absence of one feature is unrelated to the presence/absence of another. Naive Bayes is used to compute the likelihood of an instance belonging to each target class, and

the instance is deemed to belong to the target class with the highest probability using the following rule:

$$P(A|B) = \frac{P(B \mid A) \,.\, P(A)}{P(B)}$$

P(A|B) is the probability of instance B belonging to the target class, A, in the rule above. A is the target class, and B is the features vector characterizing an instance. P(A) is the target class's prior probability in the training set. Given the target class in the training set, P(B) is the probability of the features vector. KNN is a straightforward classifier that has grown in popularity in recent years. It is referred to as a lazy classifier because it does not need the construction of a training model; instead, for each query, the k-closest neighbors are identified from the training database regardless of their class label, and the majority vote predicts the instance's class label.

5. To improve classification accuracy, boosting algorithms combine various weak classifiers to generate strong classifiers. Adaptive Boosting is another useful method. LogitBoost was demonstrated to be able to overcome this problem by using stronger generalizations. Boosting algorithms are used to tackle a range of medical problems, such as detecting protein structure classes, cancer detection, and breast cancer detection. Figure 4 depicts the Adaptive Boost process:

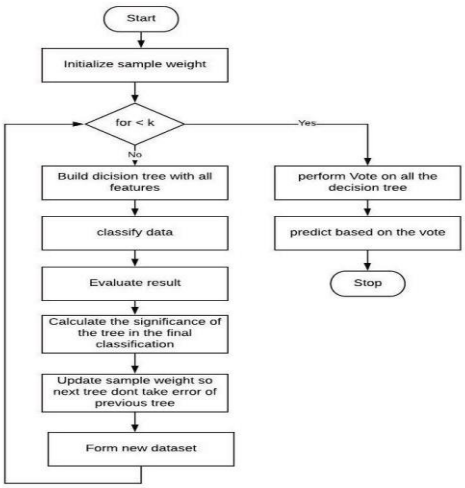

Figure 4: The illustration process of the AdaBoost algorithm [3]

6. Gradient boosting solves the classification and regression problems. This results in a diagnostic model in the form of a collection of weak predictive analytics, commonly known as decision trees. The model is phase-wise constructed, which is consistent with existing boosting approaches, and it allows for the optimization of any differentiable loss function. The insertion of a new estimator $h_m(x)$ in equation 3 should be no problem for our approach: $F_{m+1}(x) = F_m(x) + h_m(x) = y$

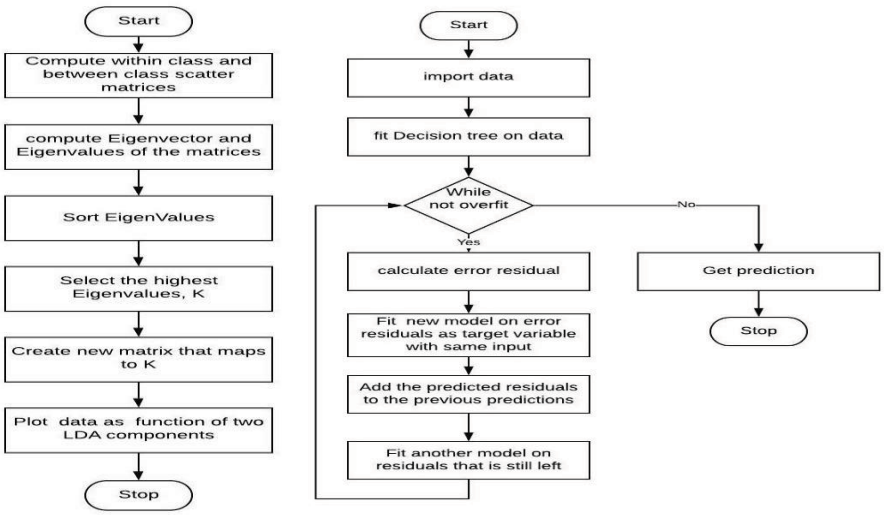

Figure 5: The process of LDA algorithm and the flow of the Gradient classifier [3]

DEVELOPMENT OF THE MODEL

The dataset D = { (x_i; y_i), i = 1, 2, ..., N}, where xi = [x_{i1} , x_{i2} , x_{i3} , ... , x_{ip}] , is a row vector whose entries are real-valued input variables (or features), and y = {0,1} (2), is a scalar whose element is the output of integer-value. The job at hand is to create a model, y = f(x), based on the training data, which is a binary classification issue. Then we may use the model to forecast $y_k = f(x_k)$ on the test data in the hopes that the anticipated output y_k matches the real output y_k for as many test points as feasible. Figure 6 depicts the suggested model. The first problem in using Machine Learning (ML) to develop models is feature selection and reduction. Collecting more features is expensive, and using them in model training is time-consuming. The characteristics must then be reduced by picking just the relevant and crucial aspects that contribute to accurate output prediction.

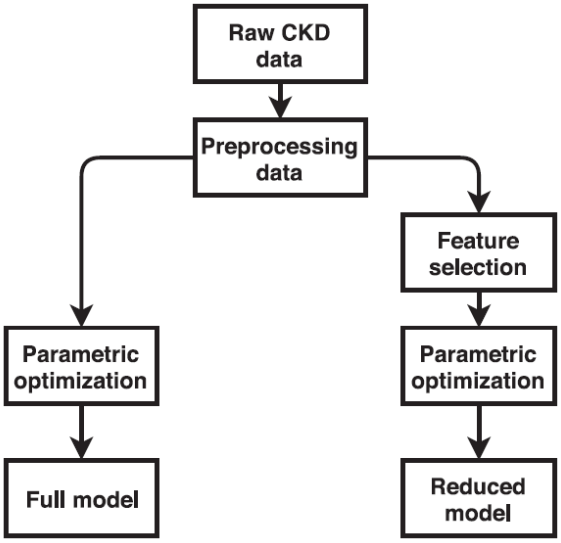

Figure 6: Methodology of model [1]

1. Recursive Feature Elimination (RFE): With repeated modeling, RFE recursively decreases the features in a dataset. To fit a model, the RFE algorithm first uses all of the information. After that, the features will be rated in order of their relevance. Let L denote the number of characteristics to be maintained (L1 > L2 > L3...). After each round of recursive model development, estimators are trained and each feature is assigned a weight; the features with the lowest weights are removed. Features that worked exceptionally well are retained.
2. Extra Classifier for Trees (ETC): It was proposed by Geurts and is also known as the Extreme Randomised Classifier (ERC). ETC creates a top-down ensemble of the unpruned decision or regression trees. The technique is carried out using the feature representation and the partitioning of the nodes into right and left. A tree will continue to grow until it reaches a certain depth. A random subset of features is employed during the bagging process and with each attribute split. This method does its work using greedy optimization and the Gini index as its split criteria. Selection with only one variable (US). Univariate Selection selects variables that performed well in univariate statistical tests like Pearson's Correlation, LDA, ANOVA, and Chi-Square. The Chi-Square test is a statistical test that uses the frequency distribution of characteristics to assess the likelihood of correlation or connection between

them. For the US method, we used " SelectKBest " from the SciKit learn Python module, with the scoring function and K set to Chi2 and Chi4 respectively.

Table 1: Confusion Matrix [3]

		Predicted Class	
		III	**Healthy**
III		TP	FN
Actual Class	**Healthy**	FP	FN

- Step 1: Create a validation set of 10% of the dataset and a test set of 10% of the dataset. Use the remaining 80% to create our XGBoost CKD model, which has been tuned. To check overfitting, the n-estimators are set to 100 and early-stopping round as 10.
- Step 2: Find the best learning rate and gamma at the same time because they have a direct impact on the model's performance. The learning rate grid values are 0.01, 0.02, 0.03, 0.06, 0.1, 0.2, and 0.3, while the gamma grid values are 0.1, 0.2, 0.5, 1, 1.5, 2, and 10. For model tuning, all potential combinations of these two parameter values are tested, and the ones that perform best are kept as the ideal values.
- Step 3: Perform a grid-search across the max depth and min child weight in specified ranges of 1 to 10 using the optimal learning rate and gamma values.
- Step 4: Perform a grid-search over the L2 regularisation parameter reg lambda and subsample in specified ranges 0.1 to 1 at the same time.
- Step 5: Use a grid-search value of 1 to 5 to account for imbalance in the dataset over max delta step.
- Step 6: Check for discrepancies between the optimal values using a simultaneous grid search over gamma, reg lambda, and subsample.

RESEARCH METHODOLOGY

The model is represented using four alternative learning strategies, as well as performance measures such as accuracy, recall, F1-score, and specificity. TP, TN, FP, and FN influence the results of performance measure indices.

1) True Positive (TP) = A list of instances that have been correctly classified as having CKD.
2) False Positive (FP) = A list of confirmed occurrences that were wrongfully labelled as having CKD.
3) True Negative (TN) = A list of recorded cases that have been accurately identified as having CKD.
4) False Negative (FN) = A list of occurrences that have been verified as having CKD.

The data used in this study was obtained over a two-month period in 2015 from CKD patients at Apollo Hospital in India. The data may be found at the data repository of the University of California, Irvine (UCI). There are missing and noisy values in this data set of 400 observations. There are 250 records of individuals with CKD and 150 records of those who do not have CKD in the data. As a result, 63 percent of people in each class have CKD, while 38 percent do not. These observations range in age from two to ninety years old. The CKD dataset comprises 24 features, 11 numeric characteristics and 13 nominal features, and the 25th feature represents the categorization or condition of CKD, as shown in Figure 1.

Table 2: CKD Dataset [3]

Name	Description	Type: unit/ values
Age (age)	Patient's age	Numeric: years
Blood pressure (bp)	Blood pressure of the patient	Numeric: mm/Hg
Specific gravity (sg)	The ratio of the density of urine	Nominal: 1.005, 1.010, 1.015, 1.020,1.025
Albumin (al)	Albumin level in the blood	Nominal: 0,1,2,3,4,5
Sugar (su)	Sugar level of the patient	Nominal: 0,1,2,3,4,5
Red blood cells (rbc)	Patients' red blood cells count	Nominal: normal, abnormal
Pus cell (pc)	pus cell count of patient	Nominal: normal, abnormal
Pus cell clumps (pcc)	Presence of pus cell clumps in the blood	Nominal: present, not present
Bacteria (ba)	Presence of bacteria in the blood	Nominal: present, not present
Blood glucose (bgr)	blood glucose random count	Numeric: mgs/dl

Blood urea (bu)	blood urea level of the patient	Numeric: mgs/dl
Serum creatinine (sc)	serum creatinine level in the blood	Numeric: mgs/dl
Sodium (sod)	sodium level in the blood	Numeric: mEq/L
Potassium (pot)	potassium level in the blood	Numeric: mEq/L
Hemoglobin (hemo)	hemoglobin level in the blood	Numeric: gms
Packed cell volume (pcv)	packed cell volume in the blood	Numeric
White blood cell count (wc)	white blood cell count of the patient	Numeric: cells/cumm
Red blood cell count (rc)	red blood cell count of the patient	Numeric millions/cmm
Hypertension (htn)	Does the patient has hypertension on not	Nominal: yes, no
Diabetes mellitus (dm)	Does the patient has diabetes or not	Nominal: yes, no
Coronary artery disease (cad)	Does the patient has coronary artery disease or not	Nominal: yes, no
Appetite (appet)	Patient's appetite	Nominal: good, poor
Pedal Edema (pe)	Does patient has pedal edema or not	Nominal: yes, no
Anemia (ane)	Does patient has anemia or not	Nominal: yes, no
Class	Does the patient has kidney disease or not	Nominal: CKD, not CKD

SIMULATION OF THE MODEL

A CKD dataset may be found at the University of California Irvine's (UCI) online repository. There are 400 individuals in this collection, including 250 instances of CKD and 150 cases of no CKD. The dataset is reasonably balanced in terms of the ratio of CKD-free and CKD patients. Table 2 shows the characteristics. The patient's age (Age) ranges from 2 to 90 years. Blood pressure (BP) ranges from 50 to 120 mmHg, whereas blood glucose levels vary from 70 to 490 mg/dl (BGR). The blood urea concentration (BU) ranges from 15 to 424 mg/dl. Outliers and incorrect data are found in four rows, which are deleted before modeling. Then there are 396 data points/rows. The categorical characteristics in the dataset were changed to binary variables, i.e., bad or good, no or yes, not present or present were converted to zero and one.

Table 3: CKD Attributes with Abbreviations [1]

Number	Attribute	Abbreviation
1	Age	AGE
2	Appetite	APPET
3	Anemia	ANE
4	Albumin	AL
5	Bacteria	BA
6	Blood Pressure	BP
7	Blood Glucose Random	BGR
8	Blood Urea	BU
9	Coronary Artery Disease	CAD
10	Diabetes Mellitus	DM
11	Pus Cell Clump	PCC
12	Serum Creatinine	SC
13	Sodium	SOD
14	Potassium	POT
15	Hemoglobin	HEMO
16	Pack Cell Volume	PCV
17	White Blood Cell Count	WC
18	Red Blood Cell Count	RBCC
19	Hypertension	HTN
20	Pus Cell	PC
21	Specific Gravity	SG
22	Red Blood Cell	RBC
23	Petal Edema	PE
24	Sugar	SU
25	Class	CLASS

Other non-binary columns were normalized in order for an ML algorithm to perform consistently on them; however, some ML algorithms, such as trees, are resistant to normalization. We attempted removing the rows with missing data, which reduced the number of instances from 396 to 157, a significant reduction in the dataset that would have a negative impact on the model due to the short sample size. Fill-forward and fill-backward, mean, median, and mode imputation were all choices for substituting missing data. After viewing the dataset after imputation using the Gaussian curve, we decided on the median imputation since it generated a regular bell curve. The occurrences of the same class are clustered together in the original dataset. To disrupt the clear trend, we decided to shuffle the dataset. The whole dataset is the output of this process.

$$z_{ij} = \frac{x_{ij} - min(x_{.j})}{max(x_{.j}) - min(x_{.j})},$$

There are several ML algorithms. Because of the magnitude of the dataset, advanced approaches like ANN are inappropriate due to over-fitting. Deep learning is the most advanced use of ANN, and it works best with enormous datasets. We picked the approaches of K-Nearest Neighbour (KNN), logistic regression (LG), Linear Discriminant Analysis (LDA), Classification and Regression Tree (CART), Support Vector Machine (SVM), and XGBoost because tree-based algorithms, SVM, and regression perform well with tiny datasets (XGB). They were applied to the entire dataset with the default/normal settings without any adjustment. Accuracy, Root Mean Square Error (RMSE), f1 score, precision, sensitivity, and Area Under Curve are among the performance characteristics. (AUC). XGBoost offers the best overall performance. The XGBoost model was then improved on the entire features using the parameter optimization approach. On the training dataset, the model was trained and fine-tuned. To avoid over-fitting the model, an early stop value of 10 was used. The model was evaluated using the test set. Table 5 compares the performance of our model to those of other CKD models available in the literature. Correlated characteristics are redundant and may cause ML algorithms to perform poorly. As a result, we had to reduce the number of features. The 24 characteristics were ranked using Recursive Feature Elimination, Extra Tree Classifier, and Uni-variate Selection. The rankings are presented, where it can be seen that different algorithms rate the attributes differently. To choose the top-ranked characteristics for the optimal trade-off between model performance and simplicity, a cut-off point is set. RFE, ETC, and US have cut-off thresholds of 10, 2.5, and 2.5, respectively. Following this, the sets of retained features for RFE, ETC, and US are created.

Table 4: AI Methods' Performance [1]

Abbreviation	Accuracy	F1_score	Sensitivity	AUC	RMSE
KNN	0.968 ± 0.032	0.93 ± 0.10	0.87 ± 0.16	0.98 ± 0.04	0.18 ± 0.18
LG	0.981 ± 0.026	0.95 ± 0.10	0.91 ± 0.16	1.00 ± 0.00	0.14 ± 0.17
SVM	0.981 ± 0.026	0.95 ± 0.10	0.91 ± 0.16	1.00 ± 0.00	0.14 ± 0.17
LDA	0.981 ± 0.025	0.95 ± 0.10	0.93 ± 0.16	1.00 ± 0.00	0.14 ± 0.20
CART	0.981 ± 0.016	0.96 ± 0.07	0.95 ± 0.10	0.99 ± 0.04	0.14 ± 0.17
XGBoost	0.987 ± 0.016	0.97 ± 0.06	0.98 ± 0.08	1.00 ± 0.00	0.11 ± 0.16

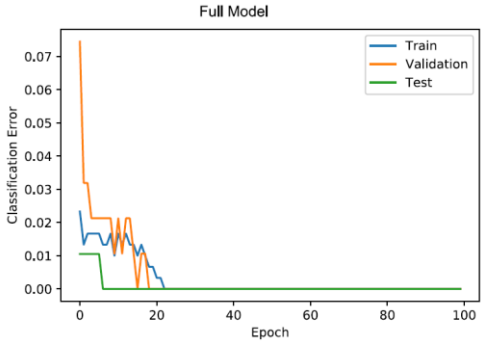

Figure 7: Classification error of the full model [1]

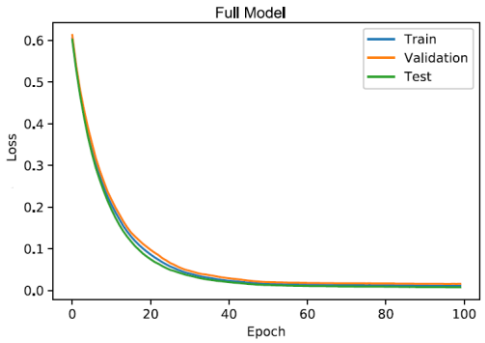

Figure 8: Loss function of the full model [1]

The XGBoost model was then improved on the entire features using the parameter optimisation approach described in Section 2. On the training dataset, the model was trained and fine-tuned. An early-stop value of 10 was selected to avoid over-fitting the model. Figures 2 and 3 demonstrate the classification error and loss function of the optimization procedure. The validation dataset was used to test the trained model's performance. We will test it if it is satisfactory. The model was evaluated using the test set.

Table 4: Optimal Parameter Values for Full Model [1]

Hyperparameter	Value
learning_rate	0.1
n_estimator	100
gamma	0.1
reg_lambda	0.2
subsample	0.8
min_child_weight	1
max_depth	4

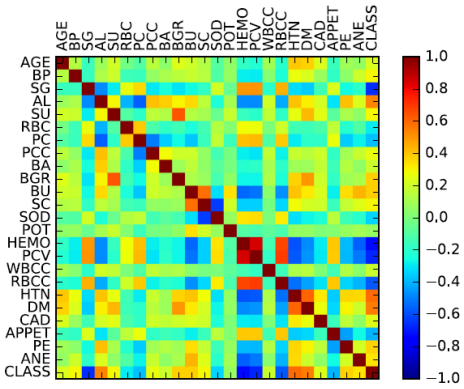

Figure 9: Correlation plot for CKD attributes [1]

Correlated characteristics are repetitive and may cause ML algorithms to perform poorly. Figure 4 depicts the relationships between the characteristics. As a result, we had to reduce the number of features. RFE, ETC, and US have cut-off thresholds of 10, 2.5, and 2.5, respectively. Following this selection, S1, S2, and S3 reflect the sets of maintained characteristics for RFE, ETC, and US, respectively. S1 has 15 features, S2 has 13, and S3 has 13 features apiece.

- S1 = {DM; AL; SG; HTN; PCC; HEMO; AGE; BP; PE; PCV; SU; ANE; WBCC; RBCC; BGR}
- S2 = {HEMO; DM; SG; AL; SC; PCV; APPET; WBCC; PC}
- S3 = {DM; HTN; AL; ANE; PE; PCC; SG; SU; APPET; CAD; PC}

OPTIMIZATION STRATEGY

The settings of the learning algorithms must be tweaked to generate models that can answer a classification task optimally. An extensive grid search with cross-validation was employed for this. Grid search is a technique for narrowing down the search space by specifying parameters and their range of potential values. Then it creates all conceivable combinations of the parameters' values in order to simultaneously explore the parameters. Validation was done using 10-fold cross-validation for each combination, and the average accuracy of the models was obtained.

Table 5: Optimal Parameters for Each Classifier [2]

Classifier	Parameter	Value
ANN	Number of Layers	1
	Number of Neurons	100
	Activation Function	Linear
	Optimization Algorithm	L-BFGS
SVM	Kernel Type	Linear
	C	0.001
	Epsilon	0.001
KNN	K	5
	Weight	Distance

FEATURE SELECTION

The correlation coefficient and recursive feature removal are used in this study's feature selection approach. As indicated in the table, the correlation coefficient is used to rank the qualities from highest to lowest correlation with the target variable. The recursive feature elimination process then creates subsets by iteratively removing the bottom half of the ranking characteristics until only one is left. To determine the subset with the best performance, each subgroup runs all four classifiers with 10-fold cross-validation.

Table 6: Result of Different Feature Subsets [2]

Number of Attributes	Accuracy of Classifier (%)				Average Accuracy
	ANN	SVM	NB	k-NN	
57 (All)	0.926	0.938	0.737	0.754	0.83875
29	0.934	0.931	0.762	0.837	0.866
15	0.656	0.939	0.774	0.881	0.8125
8	0.946	0.942	0.623	0.902	0.85325
4	0.943	0.946	0.959	0.910	0.9395
2	0.951	0.943	0.956	0.939	0.94725
1	0.943	0.931	0.935	0.939	0.937

Subsequently, we used the feature selection rule from sec. 2 to acquire our reduced set of the feature Sr: a feature is only chosen if it is a member of at least two of the three sets above. This gives

- Sr = {DM; AL; SG; HTN; HEMO; WBCC; AGE; PCV; ANE; PE; SU; APPET; PC}

Table 7: Optimal Parameter Values for the Selected Features [1]

Hyperparameter	RFE	ETC	US	Reduced
Learning_rate	0.1	0.3	0.1	0.2
N_estimator	100	100	100	100
gamma	0.5	0.2	0.1	0.3
reg_lambda	0.2	0.6	0.3	0.2
subsample	0.5	0.5	0.8	0.7
min_child_weight	2	1	2	2
max_depth	3	3	2	3
max_delta_step	2	1	2	2

Table 8: Model Comparison [1]

Model	Accuracy	Precision	Sensitivity	Specificity	MAE

RFE	0.989	0.985	1.000	0.983	0.011
ETC	0.979	0.981	0.981	0.979	0.021
US	0.979	1.000	0.969	0.974	0.021
Reduced model	1.000	1.000	1.000	1.000	0.000
Full model	1.000	1.000	1.000	1.000	0.000

Sr. was given the best XGboosting modelling. Figures depict the tuning process, while Table 6 lists the best parameter values. The entire model and the models derived above using individual feature selection methods are then compared in Table 7 with the reduced model. It received flawless ratings on all of the tests. It's worth noting that if a smaller collection of characteristics is used well to construct a model to detect CKD, the patient will spend less money and time on medical testing.

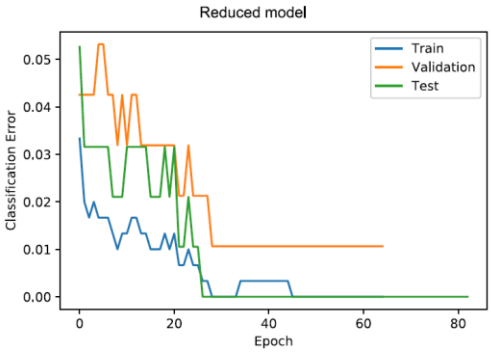

Figure 10: Classification error of reduced model [1]

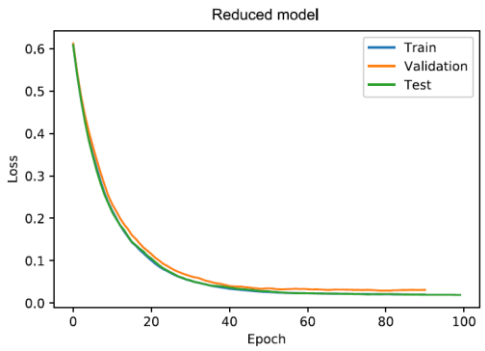

Figure 11: Loss function of reduced model [1]

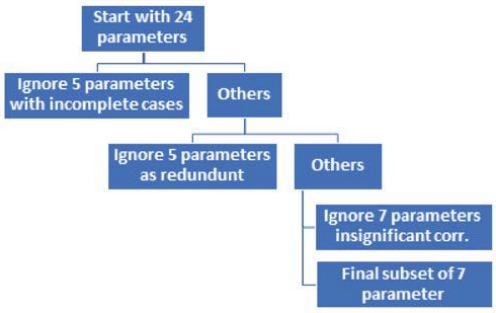

Figure 12: Parameters selection procedure [4]

Table 9 shows the results of SVM, K-NN, ANN, and NB on the best parameters and features with direct partitioning. In the instance of SVM, we discovered that Table 9 shows that SVM, ANN, and Naive Bayes have the best performance across all metrics (precision, recall, accuracy, and f-measure). The FN rate is the most important measure to look at in the confusion matrices below.

Because failing to diagnose a CKD patient might result in subsequent difficulties. ANN, SVM, and Naive Bayes were used to find the lowest FN rate, followed by K-NN. Adjusting C = 100 produced a better result than our first guess.

Table 9: Classification Performance on Optimal Features [2]

Classifier	Accuracy	Precision	Recall	f-measure
ANN	0.980	0.964	1	0.98167
SVM	0.980	0.964	1	0.98167
NB	0.980	0.964	1	0.98167
k-NN	0.939	0.929	0.963	0.9457

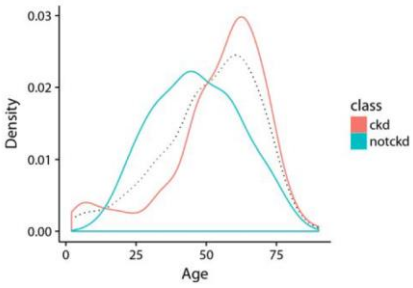

Figure 13: Age distribution [4]

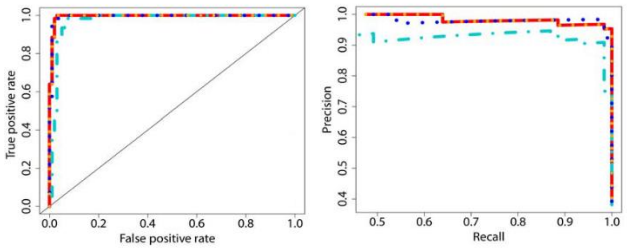

Figure 14: Sensitivity, Specificity, & Accuracy [4] Figure 15: Precision, F1, & Area under ROC curve [4]

Table 10: Confusion Matrix for the Reduced Model [1]

		Predicted Class	
		CKD	**NoCKD**
Actual Class	**CKD**	11	0
	NoCKD	0	29

CONCLUSION

For CKD diagnosis, different machine learning algorithms were used in this work. The GB classifier, in contrast to all other techniques, produced the most beneficial results. While AB and LDA (97.91

percent) produce a low score, these models effectively yield a 99.80 percent accuracy rate. For feature selection, the authors employed the correlation coefficient and recursive feature removal. Then, SVM, Naive Bayes, ANN, and kNN, were investigated. The classification accuracy, precision, recall, and f-measure attained by each of these classifiers were used to evaluate their performance. ANN, SVM, and NB all had a 98 percent accuracy rate, whereas k-NN had a 93.9 percent accuracy rate., the XGBoost technique has been examined and optimized. The resultant CKD models are compared to the domain's existing CKD models. The accuracy, sensitivity, and specificity of the suggested complete model were all 1.000, 1.000, and 1.000, respectively. By combining the strengths of each approach, three feature selection techniques are created. The accuracy, sensitivity, and specificity of a reduced model with around half of the complete features are 1.000, 1.000, and 1.000, respectively.

SIMULATION OF THE PROGRAM (CKDD.ipynb)

In the simulation, the same 24 features are used having 400 patient records. After training the model on SVM, k-NN, Decision tree and Random Forest, the

Classification Algorithm	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
Support Vector Machine	95.8	94.7	96.4	95.6
K Nearest Neighbour	92.5	93.0	91.4	92.1
Decision Tree	93.3	89.5	96.2	92.7
Random Forest	97.5	94.7	100	97.3

Based on the above table, it can be noted that Random Forest has the highest accuracy (97.5%), highest recall (94.7%), highest precision (100%) and highest F1-score (97.3%) of all the other algorithms. From the metrics recorded which are unrealistically very good, there are some improvements that can be done to improve the fitting performance:

1. The dataset given is very small (only 500 records). Either, a larger dataset must be used or we can also apply data augmentation (usually used for deep learning) to generate more training samples.

2. Use better data pre-processing such as min-max scaling.
3. Use PCA for Dimensionality reduction.
4. From the correlation heatmap produced above, it can be seen that there are several features with very weak correlation/no correlation. In this case, it is better to perform efficient feature selection and/or feature engineering.

References

[1] A. Ogunleye and Q.-G. Wang, “XGBoost Model for Chronic Kidney Disease Diagnosis”, IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 6, Nov./Dec. 2020, pp. 2131-2140.

[2] Reem A. Alassaf, Khawla A. Alsulaim, Noura Y. Alroomi, Nouf S. Alsharif; ”Preemptive Diagnosis of Chronic Kidney Disease Using Machine Learning Techniques”, 2018, pp. 18374828, doi: 10.1109/INNOVATIONS.2018.8606040.

[3] Pronab Ghosh, F. M. Javed Mehedi Shamrat, Shahana Shultana, Saima Afrin, Atqiya Abida Anjum, Aliza Ahmed Khan, "Optimization of Prediction Method of Chronic Kidney Disease Using Machine Learning Algorithm", in IEEE Access, pp. 20492838, 2021, doi: 10.1109/iSAI-NLP51646.2020.9376787.

[4] Ahmed J. Aljaaf, Dhiya Al-Jumeily, Hussein M. Haglan, Mohamed Alloghani, Thar Baker, Abir J. Hussain, Jamila Mustafina, "Early Prediction of Chronic Kidney Disease Using Machine Learning Supported by Predictive Analytics", in IEEE Access, pp. 18133484, 2018, doi: 10.1109/CEC.2018.8477876.