This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

A Comprehensive Unsupervised Framework for Chronic Kidney Disease Prediction

Linta Antony[1], Sami Azam[2], Eva Ignatious[3], Ryana Quadir[4], Abhijith Reddy Beeravolu[5], Mirjam Jonkman[6], Friso De Boer[7]

[1]Centre for Materials for Electronics Technology, Kerala, India
[2,3,5,6,7]College of Engineering, IT and Environment, Charles Darwin University, NT, Australia
[4]Daffodil International University, Bangladesh
Corresponding author: Sami Azam (e-mail: sami.azam@cdu.edu.au).

ABSTRACT

The incidence, prevalence, and progression of chronic kidney disease (CKD) conditions have evolved over time, especially in countries that have varied social determinants of health. In most countries, diabetics and hypertension are the main causes of CKDs. The global guidelines classify CKD as a condition that results in decreased kidney function over time, as indicated by glomerular filtration rate (GFR) and markers of kidney damage. People with CKDs are likely to die at an early age. It is crucial for doctors to diagnose various conditions associated with CKD in an early stage because early detection may prevent or even reverse kidney damage. Early detection can provide better treatment and proper care to the patients. In many regional hospital/clinics, there is a shortage of nephrologists or general medical persons who diagnose the symptoms. This has resulted in patients waiting longer to get a diagnosis. Therefore, this research believes developing an intelligent system to classify a patient into classes of 'CKD' or 'Non-CKD' can help the doctors to deal with multiple patients and provide diagnosis faster. In time, organizations can implement the proposed machine learning framework in regional clinics that have lower medical expert retention, this can provide early diagnosis to patients in regional areas. Although, several researchers have tried to address the situation by developing intelligent systems using supervised machine learning methods, till date limited studies have used unsupervised machine learning algorithms. The primary aim of this research is to implement and compare the performance of various unsupervised algorithms and identify best possible combinations that can provide better accuracy and detection rate. This research has implemented five unsupervised algorithms, K-Means Clustering, DB-Scan, I-Forest, and Autoencoder. And integrating them with various feature selection methods. The experiments showed that SHAP (SHapley Additive exPlanations) feature selection method has extracted better features than the other methods. Integrating feature reduction methods with K-Means Clustering algorithm has achieved an overall accuracy of 99% in classifying the clinical data of CKD and Non-CKD.

INDEX TERMS

Chronic kidney disease, Unsupervised learning techniques, Autoencoder, Isolation forest, DB-scan, K means clustering, feature selection, Glomerular Filtration Rate

I. BACKGROUND

Chronic Kidney Disease (CKD) indicates a condition where human kidneys that are damaged [1] and unable to filter the blood stream and get rid of the metabolic waste the way they are supposed to. CKD usually develops gradually over a significant amount of time. More than 800 million people all over the world [2] are found to be affected by kidney disease including the CKD. Identifying someone as having CKD requires two sets of samples, taken at least 90 days apart [8]. Historical values can be used. The estimated Glomerular Filtration Rate (eGFR) depends on creatinine measurement, sex, race, and age. CKD can get worse over time and both kidneys might stop functioning altogether. CKD is often associated with other conditions resulting in poor clinical outcomes, such as obesity, and cardiovascular complications, and can lead to reduced quality of life, obesity, increased healthcare resource utilization, and death [3]. In some cases, CKD may progress to end-stage renal disease (ESRD), resulting in even higher morbidity and mortality [4]. The frequency of ESRD has been increasing rapidly worldwide [5]. The guidelines in diagnosing and staging of define CKD as a state where one is either suffering from severe kidney damage and/or has a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

glomerular filtration rate (GFR) of less than 60 ml/min/1.73 m^2 for more than 3 months. They also advocate the use of GFR as the best indicator of renal function to identify different stages of CKD with each successive stage defining a more severe decrease in GFR and the last stage defining kidney failure with a GFR <15 ml/min/1.73 m^2 [12]. Often kidney disease does not cause any major symptoms in the early stages of the disease, making it difficult to detect. Early detection is considered to be a crucial factor in the management and control of chronic kidney disease.

Our research aims to ascertain whether Chronic Kidney Disease is present at an early stage by deploying various unsupervised algorithms on patients' data and validating the classifications to ensure their accuracy. Intending to support medical personnel and Nephrologists, we are proposing a novel and efficient model for predicting Chronic Kidney Disease at an early stage, even before the clinical diagnosis. We also need to consider that the time and monetary costs of CKD diagnosis have to be minimized by using a limited number of tests to cover the population. This is where the feature selection plays its part as any reduced model which uses fewer features, while still maintaining high performance is preferable. As there is an overlap in the symptoms of CKD with other diseases and there is also a need to select the most important features so that patients do not need to be subjected to a larger number of tests than necessary for diagnosis of CKD [6]. A selection technique is desired to ensure the selection of the most significant features.

There have been a number of research initiatives in the field of Machine Learning for forecasting of kidney disease, but very few use unsupervised feature learning. Unsupervised methods have received attention recently [7] as they do not depend on labeled data and are suitable for training models when the data are imbalanced. We would like to explore and further investigate the prospects of the unsupervised approach for CKD. There have been some notable works based on semi-supervised learning in predicting CKD.

A. RESEARCH APPROACH

This research aims to build an intelligent machine learning model that can be used reliably to *establish* CKD diagnosis. This model will *classify* the clinical data of 'CKD' and 'Non-CKD'. This model can also be used to *confirm* an initial diagnosis. To do so, various feature selection methods and unsupervised machine learning algorithms are implemented, so that a combination of feature selection and machine learning algorithms can be identified which optimizes accuracy. Unsupervised learning can extract patterns from unlabeled CKD-related clinical data. These extracted patterns can be used to classify the patients as 'CKD' and 'Non-CKD'. Various feature selection mechanisms related to filter methods, wrapper methods, embedded methods, and unsupervised methods are implemented to identify the most important features and reduce the number of input variables into the machine learning model. Algorithms such as, K-Means clustering, Isolation Forest, DB-Scan, and Autoencoder are implemented on various sets of selected features. Evaluation metrics are generated and are compared with the performance of existing machine learning models.

II. PREVIOUS WORK

Khamparia et. al. [8] proposed a novel deep learning framework for CKD classification in which a stacked autoencoder model utilizing multimedia data for feature selection with a SoftMax regression was used as a classifier. Autoencoders have been used primarily in supervised learning, but they can also automatically learn the hidden feature representation of data in an unsupervised manner. The learned feature representation can then be used as input to supervised classifiers, which makes the entire model a semi-supervised learning model. They claimed that their multimodal model outperformed conventional classifiers used for chronic kidney disease. In late 2020, Sarah et. al [9] introduced a feature learning and classification approach which integrated unsupervised enhanced sparse autoencoder (SAE) and supervised Softmax regression. The challenge of an imbalanced dataset in applying machine learning algorithms was addressed in their work and a robust semi-supervised learning model was proposed [9]. They applied this to three different diseases, obtaining a 98% accuracy for Chronic Kidney Disease (CKD).

A number of studies have used supervised algorithms, like Random Forest [10, 11], Naive Bayes [12], Gradient Boosting [13], Logistic Regression [14], Fuzzy C Means [15], Support Vector Machine [16, 17] classifiers in detecting Chronic Kidney disease.

Gopika and Vanitha [15] proposed a model based on a clustering algorithm of the test results for detecting Chronic Kidney disease and identifying its different stages, in 2017. Clusters for the different stages in chronic kidney were established. The k-means, k-medoids and Fuzzy C Means were the most commonly used classifiers. Fuzzy C-Means achieved an accuracy of 89%. Polat et.al [18] succeeded in early diagnosis of Chronic Kidney disease using an SVM classifier in 2017. The significance of their work was the use of feature selection algorithms to reduce the dimension of the dataset. The two feature selection methods employed were the wrapper and filter approaches. The filtered subset evaluator with the Best First search engine feature selection method with the SVM classifier resulted in an accuracy of 98.5%. This demonstrated that feature selection methods can play a significant role in terms of the performance of the model. In 2020, Ogunleye et. al. [6] proposed an approach to diagnosing chronic kidney disease using the Extreme Gradient Boosting (XGBoost) model. They used the University of California Irvine (UCI) CKD dataset with all the 25 features and attained an accuracy of 98.7%. Wang et.al [19] also employed the CKD dataset from the UCI machine learning

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

data warehouse in late 2018. An Associative Classification Technique implementing several algorithms ZeroR, OneR, Naive Bayes, J48, IBk (k-nearest-neighbor) based on Apriori associative algorithm was proposed, of which IBk achieved the best result: 99.0% accuracy. No feature reduction technique was used. El-Houssainy et.al [20] compared several data mining techniques for predicting kidney disease stages in 2019. In their work, hidden information was extracted from clinical and laboratory patient data, which assisted physicians in maximizing the accuracy of the disease severity stages identification. However, they only used the 361 CKD Indian patients' data which was only a part of the UCI Machine Learning repository dataset. Different data mining classifiers, Probabilistic Neural Networks (PNN), Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Radial Basis Function (RBF) algorithms were deployed. They concluded that PNN achieved the best classification and prediction performance in terms of accuracy, sensitivity and specificity. Implementing PNN achieved a maximum accuracy of 96.7% for the five stages of CKD. Rustam et.al [21] analysed gene expression data using Random Forest and Support Vector Machine (SVM) for detecting chronic kidney disease in 2019. A hybrid model that combined RF and SVM, called RF-SVM, was proposed to effectively predict CKD using highly dimensional gene expression data. The data were collected from the Gene Expression Omnibus (GEO) database. They used 48 samples where 36 were used for training and 12 for testing. The accuracy of RF-SVM algorithm was 83.4% which outperformed some other hybrid models, but the research was limited by the small dataset.

III. PROPOSED METHOD

Fig. 1 shows the framework of the proposed method and the steps involved. Initially, data preparation and standardization methods were implemented on the dataset to clean and prepare the data for further processing, as can be seen in Fig. 1.

A. DATASET

The dataset is part of the online data repository of the University of California Irvine (UCI) and contains data of 400 patients [22]. It consists of 24 clinical attributes and 1 class attribute. The datasets consist of 250 CKD cases and 150 Non-CKD cases. Missing data is a significant problem in real-world datasets, especially in the medical field. On average, every patient record and attribute have a few missing values. Fig. 2 shows the missing values present in the UCI dataset. Data preparation methods were implemented to handle the missing values. The proportion of missing values for each variable range from 0.3% (1 missing value) to 38% (152 missing values) as shown in Figure 2.

B. CHARACTER ENCODING

Before addressing the missing values in the dataset, character encoding is performed to convert the categorical attribute values into binary numbers. Since most machine learning models only accept numerical variables as input, it is important to convert textual information into binary values. Categorical features such as 'poor' or 'good', 'no' or 'yes', 'not present or 'present' are converted to'0' or '1' binary values.

C. HANDLING MISSING VALUES

After performing the character encoding, missing values in the dataset are handled using the ‘mean imputation’ method, see Fig. 1. Only one feature has attribute values for all cases, whereas the rest of the attributes had some missing values. This is to be expected with real-life patient-data. It is important to handle missing data because any result based on a dataset with non-random missing values could be biased. To tackle the issue, we relied on the following method:

C.1. Mean Imputation:

During the data preparation process, the dataset is analyzed to check for missing attribute values. A statistical method knows as 'mean imputation' is then implemented on the dataset. Mean imputation is a process of replacing missing values of a certain attribute with the mean of non-missing values of that attribute, see equation 1. The imputed values are calculated as the weighted average value of the items for the current or previous instances. Using this method, the missing values in the dataset are filled in.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

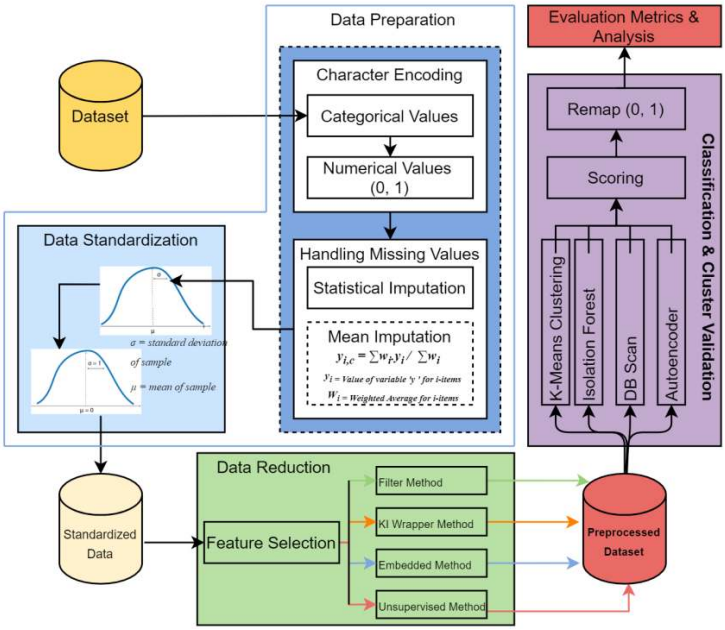

Figure 1. Workflow of the proposed method

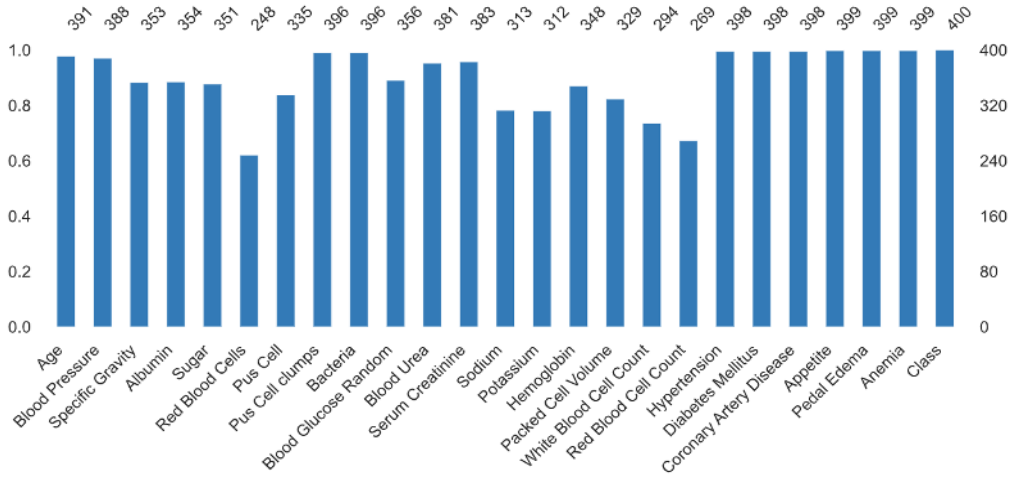

Figure 2. Visualization of missing values in the dataset

D. DATA TRANSFORMATION

Data transformation changes the values of the dataset so that they can be used for further processing. This research uses the data standardization method. Data standardization can increase the accuracy of the machine learning models.

This can be expressed in the following way:

$$y_{i,c} = \sum w_i\, y_i / \sum w_i \quad \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ (1)$$

where, y_i is the value for variable y for i-items

w_i is the weighted average value for i-items

D.1. Standardization of Data:

Standardization converts the data to a mean of 0 and a standard deviation of 1. The conversion formula is given below:

$$Z = (x - \mu)/\sigma \quad \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ (2)$$

where, Z = Standardized score
X = Observed value
μ = Mean of sample
σ = Standard deviation of sample

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

The value ranges of the features before and after standardization of the data, are displayed in Table 1.

E. DATA REDUCTION

Dimensionality reduction, or data reduction is used to reduce the input variables to the machine learning model by identifying the most useful features/attributes in the dataset. It is crucial to implement data reduction because using large number of input variables can result in poor performance of the machine learning algorithms.

E.1. Reason for Feature Reduction:

In order to limit the time and monetary costs of CKD diagnosis the smallest number of tests that is sufficient for the widest range of people need to be selected. This is where the feature selection plays a role as a it is desirable to reduce the number of features while still maintaining high performance. Also, correlated features are redundant and might degrade the performance of machine learning algorithms. Reducing the dimension of the dataset and removing irrelevant features can produce a comprehensive model for classification. The main challenge of the feature reduction procedure is to recognize the best subset of features in order to achieve the best classification result [23].

The correlation between the features is depicted in Figure 3. It can be seen that packed cell volume and hemoglobin, as well as packed cell volume and red blood cell count, have positive correlation coefficients of about 0.85 and 0.7 respectively. Another positive relationship with a correlation coefficient of 0.68 was detected between red blood cell count and hemoglobin. On the other hand, the lowest correlation can be seen for hypertension with hemoglobin and red cell volume with an approximated correlation value of-0.6.

Table 1. Features and their value range before and after standardization

Features	Before Standardization		After Standardization	
	Maximum value	Minimum value	Maximum value	Minimum Value
Age	83	6	2.27187	-2.91873
Blood Pressure	110	50	7.69207	-1.96658
Specific Gravity	1.025	1.005	1.41573	-2.31376
Albumin	4	0	3.13447	-0.80029
Sugar	5	0	4.42507	-0.437797
Red Blood Cells	1	0	4.57849	-1.68748
Pus Cell	1	0	6.77663	-1.13614
Pus Cell clumps	1	0	12.9985	-0.476334
Bacteria	1	0	2.77079	-14.471
Blood Glucose Random	490	70	15.0458	-0.755345
Blood Urea	309	10	1.94397	-3.47483
Serum Creatinine	15.2	0.4	1.85674	-3.67091
Sodium	150	111	7.14025	-2.46268
Potassium	47	2.5	3.92316	-3.10681
Hemoglobin	17.8	3.1	0.995012	-1.00501
Packed Cell Volume	54	9	0.737836	-1.35532
White Blood Cell Count	26400	3800	2.91956	-0.342518
Red Blood Cell Count	8	2.1	4.1451	-0.241249
Hypertension	1	0	1.3119	-0.762252
Diabetes Mellitus	1	0	1.38554	-0.721743
Coronary Artery Disease	1	0	3.28096	-0.304789
Appetite	1	0	1.96928	-0.507801
Pedal Edema	1	0	2.06474	-0.484322
Anemia	1	0	2.38048	-0.420084

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

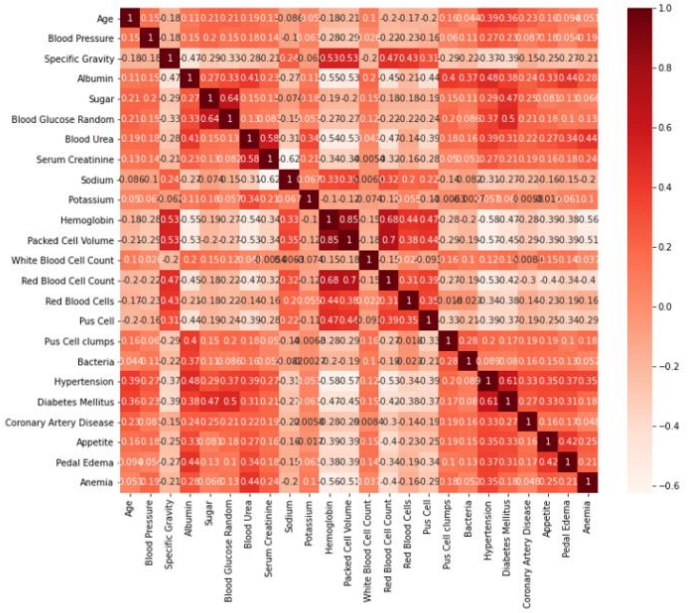

Figure 3. Correlation matrix of the features

E.2. Feature Selection Methods:

Feature selection techniques are important for unsupervised machine learning algorithms as they are essential to extract the best attributes for classification. The main purpose of feature selection is to remove a subset of input features which are not important for classification [18]. This can decrease the cost of the training and obtain higher accuracy [24]. Feature selection allows the machine learning model to remove non-informative and redundant predictors from the model and establish a CKD diagnosis more quickly with less clinical data. Classifying the patients into 'CKD' and 'Non-CKD' classes as quickly as possible can help the clinics/hospitals to allocate hospital resources to the patients that require them. Various feature selection methods are implemented in this research and are integrated with various unsupervised machine learning algorithms.

Feature selection methods are generally divided into three categories: Filter, Wrapper, and Embedded methods. An appropriate feature selection improves the performance of the classifier and reduces the computing time by using optimized data in the dataset [18, 23-26] Although traditional feature selection algorithms are used frequently, they suffer from explainability issues, e.g., when working with clinical data, it is often difficult to explain why some of the features are removed from the provided dataset. Each of the categories of feature selection algorithms has its explainability limitation making it difficult to clarify why certain features are selected without diving deep into the mathematical formulation. The Filter methods do not leverage the model's characteristic to filter the features. Although Wrapper methods do leverage a model's prediction, it chooses a subset of features solely based on accuracy or another similar scoring. For the Embedded method, even though it is calculated as a part of the training process, it has to incorporate each model's individuality and it is often difficult and tedious to provide explanations for every single model. Considering these drawbacks, an unsupervised feature selection technique, based on model agnostic explanations is required for this work and SHAP (SHapley Additive exPlanations) was adopted. This approach assigns the SHAP values, which are contribution values for a model's output for each feature of each data point. These SHAP values determine the feature importance so that the contribution information of each feature can be used to sort the features based on their importance. Selecting a subset of features based on SHAP values means selecting the first features after ordering them based on the feature contributions to the model's prediction. Feature selection methods based on SHAP values has proven their superiority for solving various classification problems in recent years [27]. The motivation to use such an approach is based on the growing need for model interpretation.

In this research, all 24 features were ranked using the 6 feature selection techniques which belong to four different types of feature selection methods. The set-theory-based rule is presented, combining several feature selection methods. The four kinds of feature selection techniques that we utilized are illustrated in figure 4.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

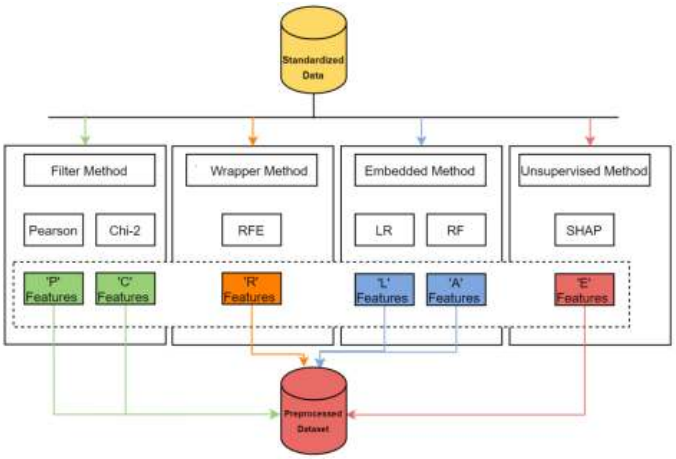

Figure 4. Feature reduction through feature elimination

E.2.1. Filter Methods:

Filter feature selection methods make use of statistical techniques to predict the relationship between each independent input variable and the output (target) variable. The filter methods evaluate the significance of the feature variables based on their inherent characteristics without the incorporation of any learning algorithm. These methods are computationally inexpensive and not subjected to overfitting [24].

E.2.1.1. Pearson:

The correlation coefficient formula quantifies the linear dependence between two continuous variables. It returns values between -1 and +1. The below Pearson correlation coefficient formula is used to measure the correlation of two variables:

$$r = \frac{N\Sigma xy - (\Sigma x)(\Sigma y)}{\sqrt{[N\Sigma x^2 - (\Sigma x)^2][N\Sigma y^2 - (\Sigma y)^2]}} \quad \ldots\ldots\ldots\ldots \quad (3)$$

Where, N = the number of pairs of scores
Σxy = the sum of the products of paired scores
Σx = the sum of x scores
Σy = the sum of y scores
Σx^2 = the sum of squared x scores
Σy^2 = the sum of squared y scores

The Pearson product-moment correlation coefficient, or simply the Pearson correlation coefficient or the Pearson coefficient correlation r determines the strength of the linear relationship between two variables. The stronger the association between the two variables, the closer the answer will be to +1 or -1. Attaining values of 1 or -1 signify that all the data points can be plotted on the straight line of 'best fit.' The closer the answer lies near 0, the larger the independent variation in the variables [43].
After applying the Pearson correlation between each feature and target variable (Class), the features can be ranked in this way illustrated in figure 5: it can be seen that, based on Pearson correlation, hemoglobin is highly correlated to the target variable and potassium is the least correlated one. This makes hemoglobin as a highly important and potassium as a least important feature.

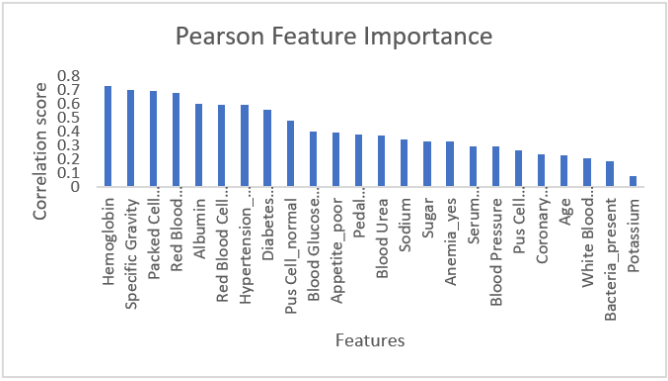

Figure 5. Feature ranking after applying Pearson correlation

E.2.1.2. Chi-2:

A chi-square test is used in statistics to test the independence of two events. Given the data of two features, we can get observed count and expected count. Chi-Square measures how the expected count and observed count deviate from each other[44]. Contigency table and expected values has to be calculated before chi square calculation. Contigency table is a table that represents the distribution of one feature and another in columns. It is used to study the relationship between two features. The expected count for each cell would be the product of the corresponding row and column totals divided by the sample size. Observed values are the actual values calculated from the sample. Then the expected counts will be contrast with the observed counts, cell by cell. The more the difference, the higher the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

resultant statistics, which is the chi square. The formula for chi square is,

$$\chi^2 = \frac{\Sigma((Observed - Expected)^2}{Expected} \quad \ldots \ldots \ldots \quad (4)$$

When two features are independent, the observed count is close to the expected count, thus we will have a smaller Chi-Square value. Inorder to find the feature importance, chi square between each feature and target variable (Class) is calculated. Higher the Chi-Square value between a feature and target column means it more dependent on thetarget column and it can be selected for model training. After applying Chi-2 technique, the features can be ranked in this way illustrated in figure 6.

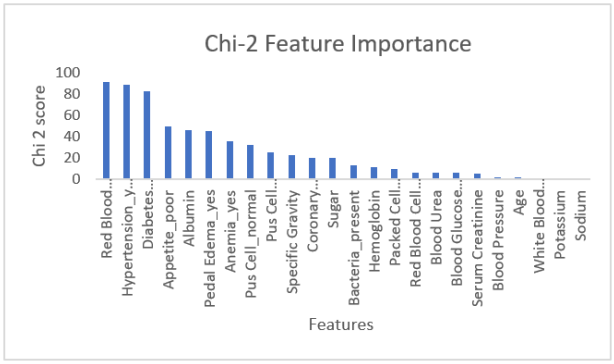

Figure 6. Feature ranking after applying Chi-2 method

E.2.2. Wrapper Methods:

Wrapper methods create several models which have different subsets of input feature variables. Later the features that result in the best performing model according to the performance metric are selected [26]. The main idea behind a wrapper method is to search for the set of features which work best for a specific classifier as shown in figure 7:

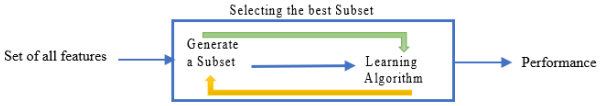

Figure 7. Wrapper feature selection method principle

E.2.2.1. Recursive Feature Elimination:

The Recursive Feature Elimination (RFE) works by recursively removing attributes and building a model on those attributes that remain. It performs a greedy search to find the best performing feature subset [28]. It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute. It iteratively creates models and determines the best or the worst performing feature at each iteration. The subsequent models use the remaining features until all the features are explored. The features are then ranked based on the order of their elimination. In the worst case, if a dataset contains N features RFE will do a greedy search for 2N combinations of features. Here RFE is used with the Logistic Regression classifier to select the top features as depicted in Figure 8.

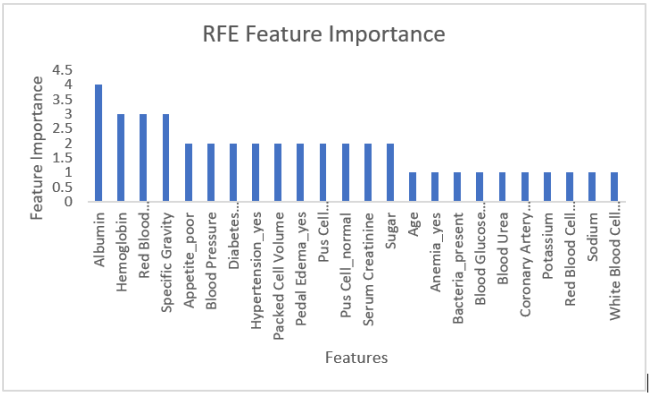

Figure 8. Feature ranking after applying RFE method

E.2.3. Embedded Methods:

Machine learning models that have feature selection naturally incorporated as part of learning are called Embedded feature selection methods [47]. Built-in feature selection is incorporated in some of the models, which means that the model includes predictors that help in maximizing accuracy, as illustrated in figure 9.

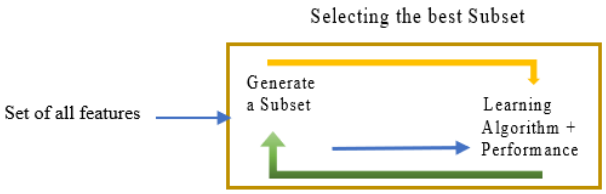

Figure 9. Embedded feature selection method principle

In this method, the machine learning model chooses the best representation of the data. The examples of the algorithms making use of embedded methods are penalized regression models. We utilize Logistic Regression and Random Forest.

E.2.3.1. Logistic Regression:

Rule-based models like Logistic Regression (LR) with L1 penalty (Lasso regression) intrinsically conduct feature selection [45]. It is a linear model that uses this cost function:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

$$\frac{1}{2N_{training}} \sum_{i=1}^{N_{training}} \left(y_{\text{real}}^{(i)} - y_{\text{pred}}^{(i)}\right)^2 + \alpha \sum_{j=1}^{n} |a_j| \ldots. \quad (5)$$

where, a_j is the coefficient of the j-th feature. The final term is called L1 penalty and α is a hyperparameter that tunes the intensity of this penalty term. The higher the coefficient of a feature, the higher the value of the cost function. So, the idea of Lasso regression is to optimize the cost function, reducing the absolute values of the coefficients. Obviously, this works if the features have been previously scaled. For example, using standardization or other scaling techniques. α hyperparameter value must be found using a cross-validation approach. Trying to minimize the cost function, Lasso regression will automatically select those features that are useful, discarding the useless or redundant features. In Lasso regression, discarding a feature will make its coefficient equal to 0. A feature Importance plot created with LR is shown in figure 10.

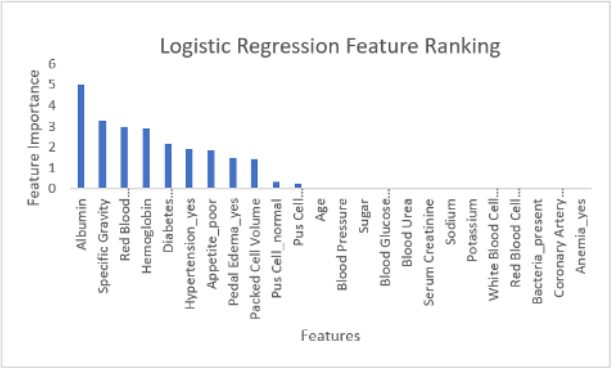

Figure 10. Feature ranking after applying Logistic Regression method

E.2.3.2. Random Forest:

Random forest (RF) is another common feature selection technique. It consists of extracting the feature importance rank from tree-base models [46]. The feature's importance is essentially the Mean of the individual trees' improvement in the splitting criterion produced by each variable. In other words, it is the magnitude of the score or impurity which was improved when splitting the tree using that specific variable. This can be used to rank the features and then select a subset. RF feature importance is biased towards features with more categories. Besides, if two features are highly correlated, both of their scores decrease regardless of the quality of the features. As mentioned, Random Forest uses the mean decrease impurity (Gini index) to estimate a feature's importance. The lower the value, the more important the feature is. Gini index is defined as:

$$Gini = 1 - \sum_{i=1}^{n} (P_i)^2 \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \quad (6)$$

where, the second term is the sum of the squared probabilities of each class for sample i. The Gini index of feature j is measured for each node of a tree where feature j was used and averaged over all trees in the ensemble. If all the samples that reached the node are linked with a single class, then that node can be called pure. This can give a good estimate on the threshold value to set when selecting features based on their importance as shown in figure 11.

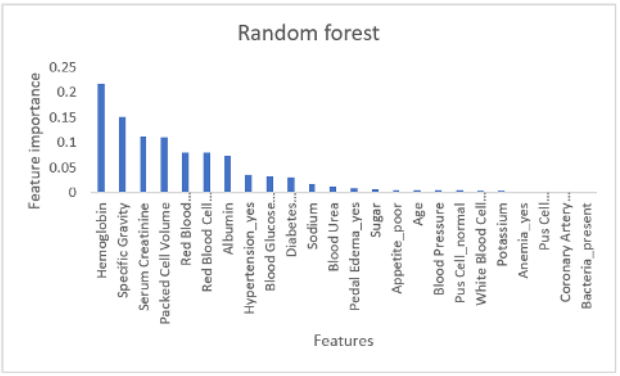

Figure 11. Feature ranking after applying Random Forest method

E.2.4. Unsupervised Feature Selection Method:

E.2.4.1. SHAP (Shapley Additive exPlanations):

The SHAP approach assigns the SHAP values, which are contribution values for a model's output for each feature of each data point [48]. These SHAP values encode the importance of a feature for the model. The mean of the columns of each matrix is calculated and the vectors of mean SHAP values for each class are summed and ordered in a decreasing way. The first position of the resulting vector contains the most important feature, the second position contains the second most important, and so on. Since SHAP can provide a means to interpret the model's decisions by indicating the importance of the dataset features. A feature selection algorithm based on the most important features according to the absolute SHAP values would provide good results [27]. Here, the Tree SHAP explainer approach is used with the Isolation forest model for feature selection and the feature importance plot is shown in figure 12.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

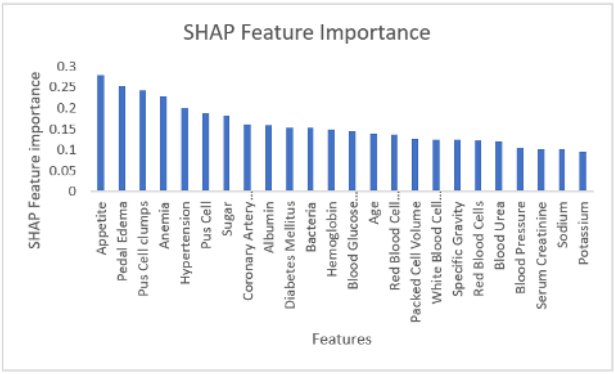

Figure 12. Feature ranking after applying SHAP method

E.3 Outcomes of Feature Selection Process:

The 24 features were ranked using the Pearson, Chi-2, RFE, Random Forest, Logistic Regression, and SHAP. The rankings were shown in Figures 5 – 12. It is clear that the features are ranked differently by different algorithms. The 11 top-ranked features from each feature selection method were selected for the best trade-off between model performance and simplicity. After this selection, the sets of the retained features for Pearson, Chi-2, RFE, Random Forest, Logistic Regression, and SHAP are represented by P, C, R, L, A, G, and E respectively.

P=[Hemoglobin, Specific Gravity, Packed Cell Volume, Red Blood Cells_normal, Albumin, Red Blood Cell Count, Hypertension_yes, Diabetes Mellitus_yes, Pus Cell_normal, Blood Glucose Random, Appetite_poor]

C=[Red Blood Cells_normal,Hypertension_yes,Diabetes Mellitus_yes,Appetite_poor,Albumin,Pedal Edema_yes,Anemia_yes,Pus Cell_normal,Pus Cell clumps_present,Specific Gravity,Coronary Artery Disease_yes]

R=[Albumin, Hemoglobin,Red Blood Cells_normal, Specific Gravity, Appetite, Blood Pressure, Diabetes Mellitus, Hypertension,Packed Cell Volume, Pedal Edema, Pus Cell clumps]

L=[Albumin, Specific Gravity, Red Blood Cells, Hemoglobin, Diabetes Mellitus, Hypertension, Appetite, Pedal Edema, Packed Cell Volume, Pus Cell, Pus Cell clumps]

A=[Albumin, Specific Gravity, Red Blood Cells, Hemoglobin, Diabetes Mellitus, Hypertension, Appetite, Pedal Edema, Packed Cell Volume, Pus Cell, Pus Cell clumps]

G =[Hemoglobin, Specific Gravity, Serum Creatinine, Packed Cell Volume, Red Blood Cells, Red Blood Cell Count, Albumin, Hypertension, Blood Glucose Random, Diabetes Mellitus, Sodium]

E=[Appetite, Pedal Edema, White Blood Cell Count,Anemia, Hypertension, Packed Cell Volume, Sugar, Coronary Artery Disease, Albumin, Diabetes Mellitus, Red Blood Cell Count, Blood Urea]

The features that are included in these sets are depicted in figure 13. Here importance indicates the number of occurrences of a feature in [P U C U R U L U A U G U E]

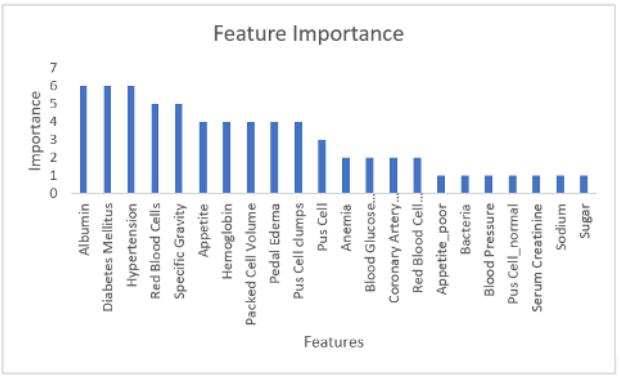

Figure 13. Final outcome of the feature selection process

A feature is only taken in the final reduced feature set if its number of occurrences (importance) is more than 3. This yields the final selected features (SF) set as:

Sf = [Albumin, Diabetes Mellitus, Hypertension, Red Blood Cells, Specific Gravity, Appetite, Hemoglobin, Packed Cell Volume, Pedal Edema, Pus Cell clumps]

F. CLASSIFICATION

This section discusses the unsupervised machine learning algorithms implemented in this research. After applying the feature selection methods described above, preprocessed datasets are created. These datasets are used for training and testing the machine learning models. Since all the classification models are unsupervised separate training and testing data is not required, moreover, dataset is limited in size thus, the whole data comprises of 400 data points were passed to kmeans model as a preliminary training data. The training allows the models to generate a distinct set of data points (Z_0 to Z_3). As illustrated in Fig. 14 each classifier, K-means clustering, DB-scan, Autoencoder, and I-forest have a distinct point that is used to separate the data into clusters of CKD and Non-CKD cases. These clusters are used to classify the data into classes.

F.1: K-Means Clustering:

Unsupervised algorithms can make predictions or inferences from unlabeled data. Clustering unlabeled data based on inferences is very useful when working with clinical data. K-means clustering is a centroid-based unsupervised clustering algorithm that can be used for classification. The preprocessed dataset created with the feature selection methods is used to train the algorithm and extract a data point (Z_0). This data point is used to classify the data in 'CKD' and 'Non-CKD' cases. Similar data points are clustered together to find an underlying pattern for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

assessment. K-means delivers the final output through a process called iterative refinement. It tries to minimize the

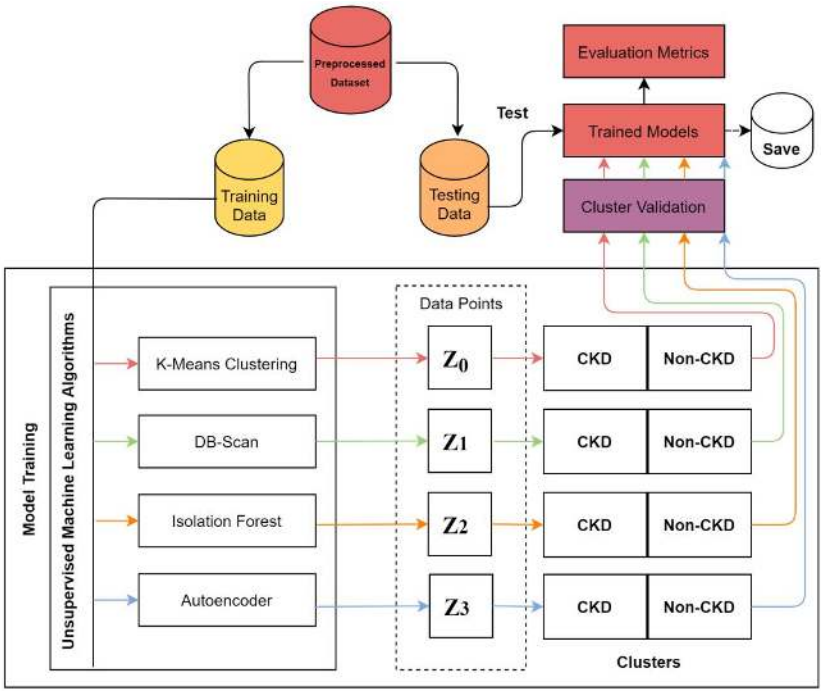

Figure 14. The Application of clustering Algorithms

sum of the squared distance between the data points and the cluster's centroid. The centroid is defined as the arithmetic mean of all the data points that belong to that cluster. The number of groups is denoted by K, and each data point is iteratively assigned to one of these groups of clusters based on the identified similarities among the features. The initial number of clusters 'K' has to be provided as an input. This can sometimes be a delicate issue and users sometimes end up running the system multiple times with different values of K. Afterwards, a comparison is then made to select the best value of 'K'. However, various methods are available for getting a reasonably stable approximation of K. K-means most commonly uses 'Euclidean Distance' to determine the distance between two data points (Z^n and Z^m). One of the key advantages of K-means is that, in case the number of features is really high, it can still complete the computation in a reasonable time if the value of 'K' is kept relatively small [29].

$$Dist(Z^n, Z^m) = \sum_{i=1}^{D} (Z_i^n - Z_i^m)^2 \quad \dots\dots\dots\dots \quad (7)$$

$$S^{(\arg\min)} \sum_{i=1}^{k} \sum_{y \in S_1} ||y - \mu_i||^2$$

$$= S^{(\arg\min)} \sum_{i=1}^{k} |S_i| V(S_i) \dots\dots\dots \quad (8)$$

Given a set of d-dimensional real vector observations (y_1, y_2, . . . , y_n), K-means clustering partitions the n observations into k (≤n) sets S = [S_1, S_2, . . . , S_k] so as to minimize the Variance. μ_i denotes the 'Mean' of S_i and V is the Variance.

The number of clusters was set to six by parameter tuning, and the actual class labels on each cluster were checked. Except for cluster 1, the other clusters reflect CKD patients, as seen in the tables below, where cluster 1 only contains non-CKD cases, while the majority of cases in the remaining clusters are CKD cases. To categorise a new data point in the future, it can be given as test data, and the euclidian distance to each cluster centroid can be calculated to discover which one is closest, and then it can be labeled under that cluster.

Table 2. Clusters of K-Means

	CKD cases	Non CKD cases
Cluster 1	0	147
Cluster 2	51	0
Cluster 3	90	0
Cluster 4	1	0
Cluster 5	30	0
Cluster 6	78	3

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

F.2 DB-Scan:

DB-Scan is Density-Based Spatial Clustering of Applications with Noise. The goal of DB-=Scan is to find core samples of high density and expand them to clusters. It is most suitable for data which contain clusters of similar density [30].

DB-Scan detects density connected clusters by discovering one of its core objects p and computing all objects which are density-reachable from p. The collection of density-reachable objects is found by iteratively computing density reachable objects. DB-Scan checks the neighborhood N of each object p in the database. If N (p) of an object p consists of at least μ objects, i.e., if p is a core object, a new cluster X containing all objects of N (p) is created. Then, the neighborhood of all objects q X, which have not yet been processed, is checked. If object q is also a core object, the neighbors of q, which are not already assigned to cluster X, are added to X and their neighborhood is checked in the next step. This procedure is repeated until no new object can be added to the current cluster X.

DBSCAN aims at discovering clusters which are high-density regions of the dataset. It applies two hyperparameters: Eps (the neighborhood radius) and minPts (minimum number of neighbors) to consider a point a core point. It defines a point as a core-point if there are at least minPts sample points in its Eps neighborhood. The points within the Eps neighborhood of a core-point are said to be directly reachable from that core-point. A point q is reachable from a core-point p if there exists a path from q to p where each point is directly reachable from the next point. The parameter values of MinPts and Eps corresponding to the highest clustering accuracy were selected.

The whole dataset comprising of 400 data points was passed to DB scans model for training. Parameter values for Eps and minPts were selected as 3.6 and 150 respectively by hyper parameter tuning Based on these parameter values, DBscan treats some data points as a cluster and other datapoints as outliers, labeling them as -1. There is only one cluster. Table 3 depicts the number of elements in the cluster and the number of outliers. The cluster consists of 174 datapoints of which 150 cases are non-CKD cases and all the outliers are CKD cases.

Table 3. Clusterpoint and Outliers of DB-Scan

Cluster	Outliers
174	226

To classify a new data point in future, it can be given as test data into this DB scan model which checks whether a given sample is within eps distance of one of the core samples. If it is, it takes the label of the core sample (classify it as non CKD case), if it is not, it us an outlier (CKD case).

F.3 Autoencoder:

An autoencoder neural network is an unsupervised deep learning technique that consists of two components: an encoder and a decoder. The main concept is that both encoder and decoder are trained together, minimizing the discrepancy between the original data and its reconstruction [31].

The encoder e(x) represents a mapping of an input x with higher dimensions to a hidden compressed representation, and the decoder d(x) maps this compressed representation back to a reconstructed version of x,

such that $d(e(x)) \approx x$.

The reconstruction error of autoencoder networks can be used to classify CKD and non-CKD cases. Here the encoder has two layers, one input layer and one hidden layer, whereas the decoder has one hidden layer and one output layer. Encoder/decoder networks are fully or densely connected neural networks with rectified linear unit (ReLu) activation between layers. An encoder network, defined as $e(x) : X \mapsto Z$, maps from the input space $X \in \mathbb{R}M$ to latent embedding $Z \in \mathbb{R}D$, and a decoder network, $d(e(x)) : Z \mapsto X$, maps the embedding Z back to the input space-optimize over encoder and decoder networks as follows:

$$min_{\varphi,\Phi} E\|x - d(e(x))\| \ldots \ldots \ldots (9)$$

where, ϕ and ψ are the parameters of our encoder and decoder neural networks, respectively. The expectation is taken over the training data, and the loss is the squared 2-norm distance between the input x and the reconstructed input. The training parameters for auto encoder are the number of times the algorithm trains on the training data and the number of samples processed before the model is updated.

Loss MSE between inputs and outputs, see equation 10, gives the anomaly score for the Auto-Encoder, for each datapoint that passes through it.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \ldots \ldots \ldots (10)$$

where, MSE is the Mean Squared Error, n is the number of data points xi is the observed values and (xˆ i) is the predicted values. Tuning parameters for autoencoder is given in table 4.

Table 4. Tuning parameters of Autoencoder

	Parameters
Training	● Epochs: The number of epochs is the number of complete passes through the training dataset ● Batch size: A batch size is the number of samples processed before the model is updated.
Testing	● Threshold: for mapping Loss MSE scores to 0 and 1

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

Here the encoder of the model consists of two layers that encode the data into lower dimensions. The decoder of the model consists of two layers that reconstruct the input data. The reconstruction errors are considered to be anomaly scores. The model is compiled with Mean Squared Logarithmic loss and Adam optimizer.
The model is then trained with 40 epochs and a batch size of 50, and in the testing phase, scores are sorted in ascending order and a threshold is set such that scores of more than the threshold result in a cluster of CKD instances, while those below that threshold result in a cluster of non-CKD cases.
Fine-tuning of this threshold is done by comparing the anomaly scores with actual class labels. (Note that class labels are not given as input to the model). As a result, based on this threshold, there are two clusters: cluster 1 contains all cases with a loss MSE of more than the threshold value, which will be mapped as 1 and cluster 2 contain all cases with a loss MSE of less than the threshold value, which will be mapped as 0.
The clusters obtained using the autoencoder with all features considered are shown in the table below. Cluster 1 has a total of 260 datapoints, with 250 of them belonging to CKD. Cluster 2 has 140 datapoints, all of which are non-CKD cases.

Table 5. Clusters of autoencoder

Cluster1	Cluster2
260	140

The model and threshold value can be used to cluster new data in the future.

F.4 Isolation Forest:

Isolation Forest (Iforest) 'isolates' observations by randomly selecting a feature and then randomly selecting a "split value" between the maximum and minimum values of the selected feature. Since recursive partitioning can be represented by a tree structure, the number of splittings required to isolate a sample is equivalent to the path length from the root node to the terminating node. This path length, averaged over a forest of such random trees, is a measure of normality and is used as our decision function. Random partitioning produces noticeably shorter paths for anomalies. A forest of random trees collectively produces shorter path lengths for particular samples [32]. Tuning parameters for isolation forest are given in Table 6.

Table 6. Parameters of Isolation Forest

	Parameters
Training	● n_estimators : number of isolation trees ● max_samples : number of samples ● max_features : number of features to draw to train each base estimator
Testing	● Threshold : for mapping Loss MSE scores to 0 and 1

Training parameters for Isolation Forest are the number of trees to create a forest, the maximum number of features, and the sub-sampling size. During the test phase: Isolation Forest finds the path length of the data point from all the Isolation Trees and finds the average path length. The higher the path length, the more normal the point, and vice-versa. Based on the average path length, it calculates the anomaly score. Decision_function of Iforest can be used to get this. For Iforest, the lower the score, the more anomalous the sample. Scores are sorted and a threshold is set such that scores less than that threshold result in a cluster of CKD instances, while those below that threshold result in a cluster of non-CKD cases. Fine-tuning of this threshold is done by comparing the anomaly scores with actual class labels. (Note that class labels are not given as input to the model). As a result, based on this threshold, there will be two clusters: cluster 1 contains all cases with an anomaly score less than the threshold value and will be mapped as 1; cluster 2 contains all cases with anomaly scores more than the threshold value and will be mapped as 0.
The clusters obtained using Isolation Forest with all features considered are shown in Table 7. Cluster 1 has a total of 250 data points, with 232 of them belonging to CKD. Cluster 2 has 150 data points with 132 of them belonging to nonckd cases.

Table 7. Clusters of Isolation Forest

Cluster 1	Cluster 2
250	150

This model and threshold values can be used to cluster new data in the future.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

G. CLUSTER VALIDATION

The clusters generated from each algorithm are evaluated using cluster validation methods. These methods are used to compare the performance of each cluster.

Validation can be done in two ways:

1. Internally
2. Externally

Table 8. Internal and External Methods and their criteria

Type	Method	Criteria
Internal	• Davies-Bouldin Index - smaller value	Smaller values indicate better defined clusters
	• Calinski-Harabasz Index. • Silhouette score	Higher values indicate better defined clusters
External	• Recall • precision • F1_score • Accuracy • Adjusted rand score • Mutual Information score • V-measure score	Closer to 1 is optimum Less than or equal to zero is poor

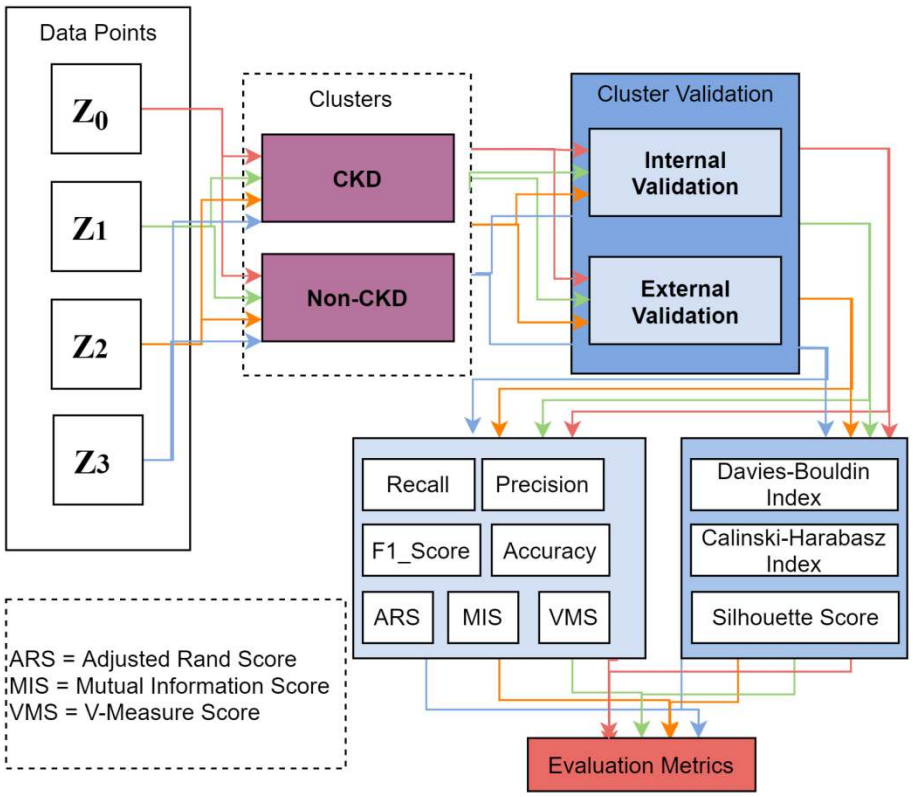

Figure 15. Cluster Validation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

G.1 Internal validation:

Internal validation processes evaluate the connectedness, i.e., how well a pair of data points within the same cluster is connected to each other. Tand the compactness, i.e. how close are the data points, placed inside the same cluster are to each other. Internal measures do not require any prior cluster labelling or ground-truths. Acceptable clusters have minimal 'Connectedness' and 'Compactness' [33, 34].
In this section, we take a look at how the clusters have been validated using various internal metrics. We also discuss about those indexes that we used here.

G.1.1 Davies--Bouldin Index (DBI):

The metric works on the basis of the ratio of within cluster distances to between-cluster distances. The smaller the values are, the better the clustering would be. A factor to note is that, to make it consistent with other indices used in this research, we have used the reverse of Davies-Bouldin Index (1- DaviesBouldin Index) [35]. The Davies Bouldin Index can be calculated for any value of a cluster (n) using the following expression [36]:

$$\mathbf{DBI} = \frac{1}{n_c}\sum_{j=1}^{n_c}\max_{k=1..n_c, k\neq j} R_{jk} \quad \ldots\ldots\ldots (11)$$

$$R_{jk} = \frac{\frac{1}{\|c_j\|}\sum_{y\epsilon c_j} d(y,x_j) + \frac{1}{\|c_k\|}\sum_{y\epsilon c_k} d(y,x_k)}{d(x_j,x_k)} \quad \ldots\ldots\ldots (12)$$

where, d is the Euclidian Distance between the points, c_j is the cluster j having x_j as the centroid.
Figure 16 illustrates the Daviesbouldin score for all the classifier without and with feature reduction. In both cases, it can be seen that kmeans perorming well with good scores.

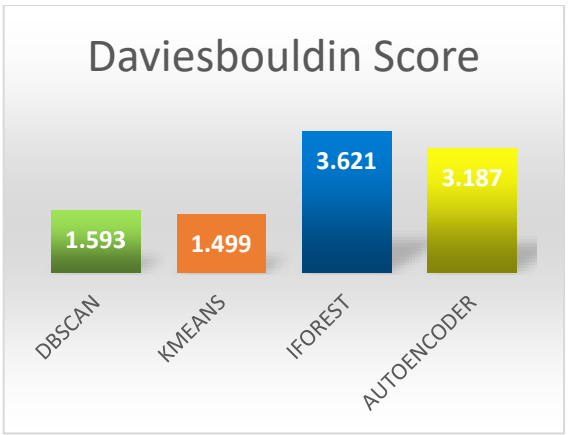

Figure 16(a). Validating by Davies-Bouldin Index for all feature set

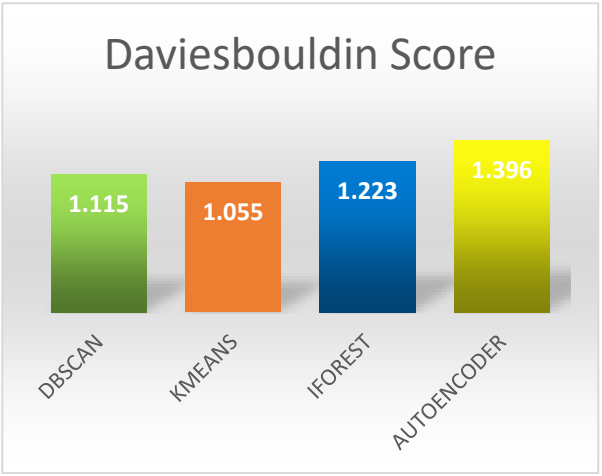

Figure 16(b). Validating by Davies-Bouldin Index for the reduced feature set

G.1.2 Calinski-Harabasz Index:

Calinski-Harabasz is a ratio-type index that evaluates the cluster validity by comparing the average between and within-cluster sum of squares. A higher value indicates better clustering [37].

1.a. The index, CH, is defined as:

$$CH(k) = \frac{Vb/k-1)}{Vw\ /\ N-k} \quad \ldots\ldots (13)$$

where Vb is the overall between-cluster variance, Vw is the overall within-cluster variance, N is the number of observations and k denotes the total number of clusters.
Figure 17 depicts the Calinski Harabasz Index for all the classifier without and with feature reduction.

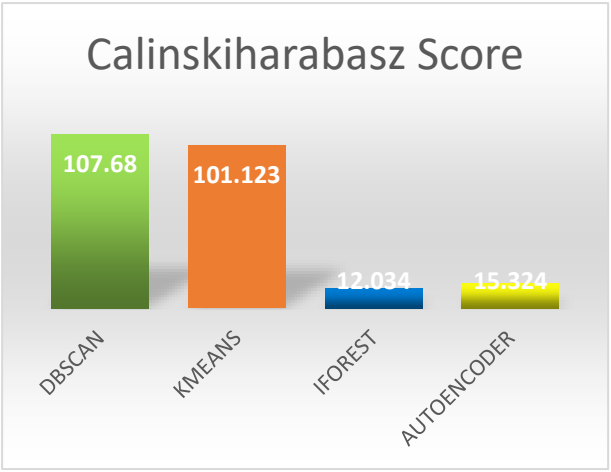

Figure 17(a). Validating by Calinski-Harabasz Index for all feature set

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

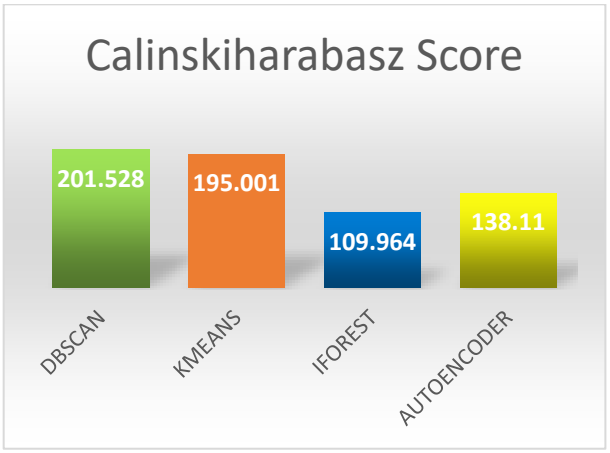

Figure 17(b). Validating by Calinski-Harabasz Index for reduced feature set

G.1.3 Silhouette coefficient score:

Silhouette coefficient score is one of the most widely used internal cluster validation techniques. The Silhouette Coefficient score is derived for each of the samples using the mean within-cluster (intra-cluster) distance and the mean nearest-cluster distance, generally using the following equation [38].

$$c = (q - p)/Max(p, q) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (14)$$

c is Silhouette Coefficient score
where, p is mean within-cluster (intra-cluster) distance
q is the distance between a sample and the nearest cluster that the sample is not a part of.

The metric is primarily an intuitive graphical tool that aids the user in visually assessing cluster quality. Figure 18 depicts the silhoutte score for all the classifier without and with feature reduction.

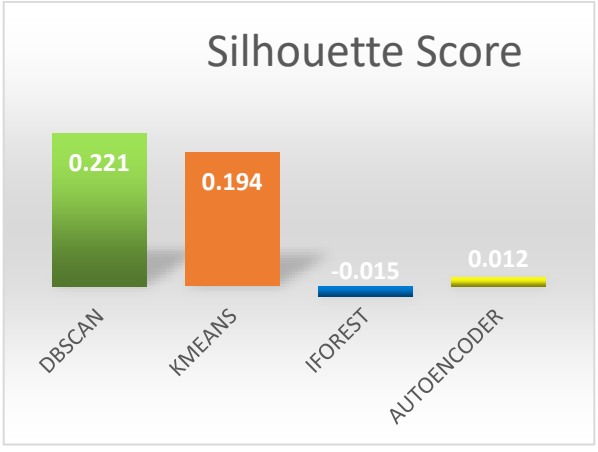

Figure 18(a). Validating by Silhouette coefficient for all feature set

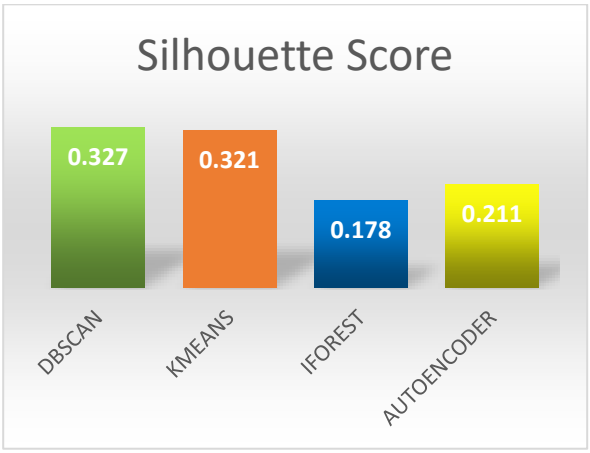

Figure 18(b). Validating by Silhouette Coefficient scores for the reduced feature set

G.2 External validation:

External validation techniques gauge the degree to which cluster labels match class labels supplied externally. These class labels have not been used in any of the processes discussed in previous sections. We will also look at the 'True Rate of Detection' (the 'Recall' measure) for each of the clusters. Several validation methods have been applied. This section will provide a detailed inspection of the quality of the clustering using various External metrics.

G.2.1 Adjusted Rand Index (ARI):

The Rand Index (RI) is a similarity measure between two sets of clusters by considering all pairs of provided samples that are assigned in the same or in different clusters in the predicted and the true clusters. Scores closer to 1 signify better clustering [39, 40]. The ARI results are shown in Figure 19.
The raw RI score is adjusted for chance as follows:

$$\text{ARI} = \frac{\text{RI} - \text{Expected\_RI}}{\max(\text{RI}) - \text{Expected\_RI}} \quad \ldots\ldots\ldots (15)$$

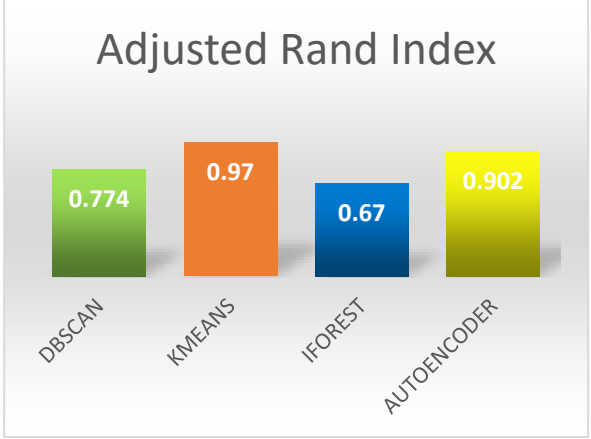

Figure 19(a). ARI scores for all features

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

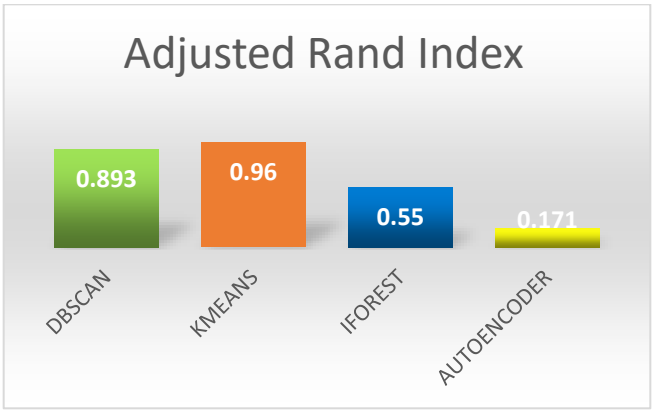

Figure 19(b). ARI scores for the reduced feature set

G.2.2 Mutual information (MI):

The Mutual Information (MI) quantifies the degree of information the two clusters in question have in common. In information theory it is often referred to as 'Correlation Measure'. The classifiers in our model had the following Mutual Information Score:

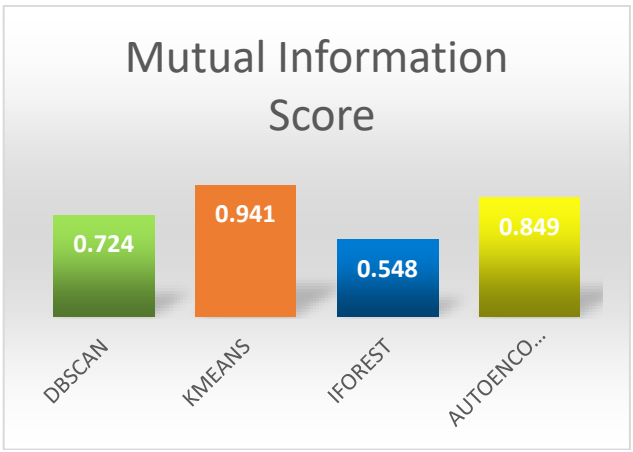

Figure 20(a). Validating by Mutual Information Score for all feature set

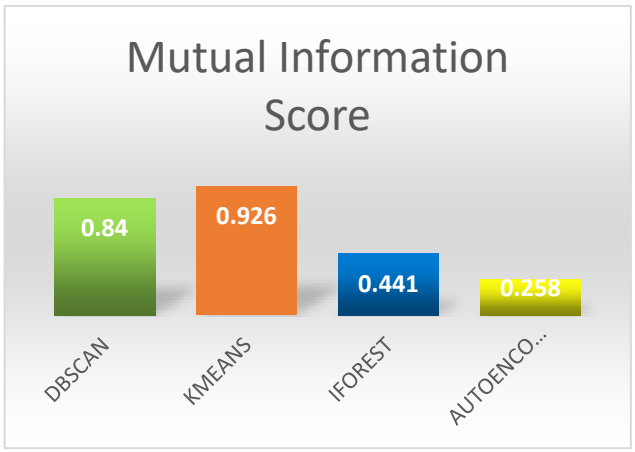

Figure 20(b). Validating by Mutual Information Score for the reduced feature set

G.2.3 V-Measure:

V-measure or Validity measure of a cluster is a metric developed using conditional entropy analysis. Entropy measures the degree of disorder within a cluster. V-measure takes the Harmonic mean of two important characteristics of a cluster, homogeneity– the measure of a cluster holding only members of a single specific cluster, and completeness– whether all members of a given class are allocated to the same cluster [41].

V-measure, v can be expressed as:

$$v = \frac{(1+\beta)*\text{homogeneity}*\text{completeness}}{(\beta*\text{homogeneity}+\text{completeness})} \quad \ldots\ \ldots\ \ldots \ (16)$$

where, v is V-measure v
Default value of β is 1, signifying equal weightage of homogeneity and completeness
Figure 21 shows the Vmeasure score for all the classifier without and with feature reduction. In both cases, it can be seen that kmeans performing well with good scores.

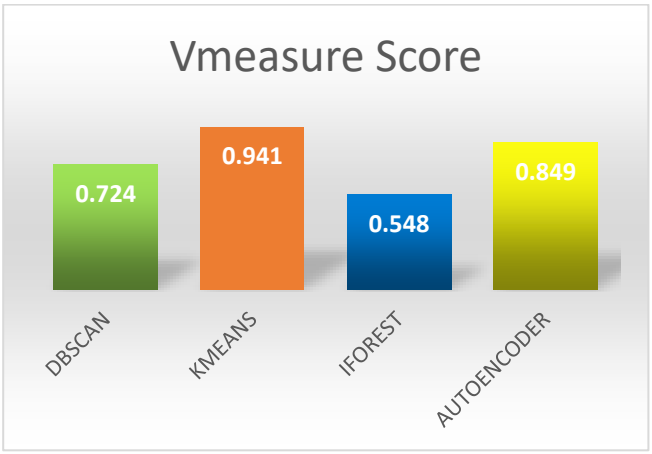

Figure 21(a). Validating by V-measure for all feature set

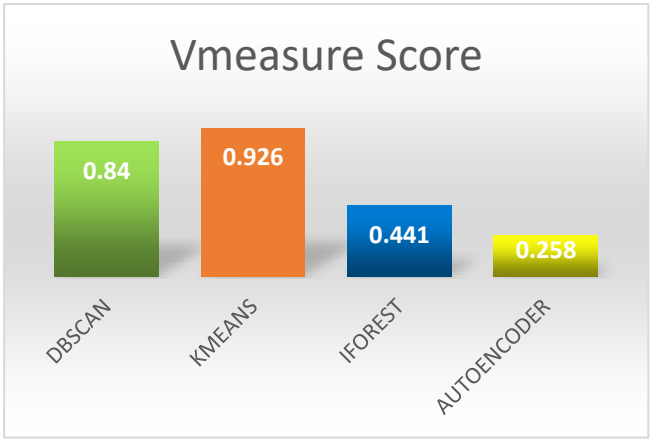

Figure 21(b). Validating by V-measure for the reduced feature set

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

The effectiveness and accuracy of the four unsupervised machine learning methods can be evaluated using performance indicators. Positive classification occurs when a person is classified as having CKD. When a person is not classified as having CKD, he has a negative classification. Similarly, True Positive (TP) indicates instances correctly categorized as CKD, True Negative (TN) instances correctly categorized as non-CKD. False Positive (FP) indicate non=CKD cases, incorrectly classified as CKD and False Negative (FN) indicate CKD cases incorrectly classified as non-CKD. The table 9 gives more explanation.

Table 9. Explanation on different evaluation

TP	True Positive	The Model correctly identified a case as having CKD
TN	True Negative	the model correctly identified a case as having no CKD
FP	False Positive	the model incorrectly identified a case as CKD i.e., identifying non CKD patients as CKD patients
FN	False Negative	the model incorrectly identified a CKD patient as a non-CKD case

G.2.4 *Accuracy*:

Accuracy is the most intuitive performance measure. It is simply a ratio of the correctly predicted observation to the total observations. Accuracy can be expressed as

Accuracy = (TP + TN)/ (TP + TN + FP + FN)

G.2.5 *Precision*:

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Precision can be expressed as

Precision = (TP) / (TP + FP)

G.2.6 *Recall*:

The recall is the ratio of correctly predicted positive observations to all observations in the actual class. Recall can be calculated as

Recall = (TP) / (TP + FN)

G.2.7. *F1-score*:

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. The F1-score can be expressed as

F1-score =2(Precision X Recall) / (Precision + Recall)

III RESULTS AND DISCUSSIONS

Validation scores obtained by considering all the 24 features for DB scan, K-means, I-forest, and Autoencoder are given in the table 10. Both K-means and autoencoder have a 100% recall, indicating that they correctly predicted all CKD cases. K-means clustered 253 anomalies as CKD, although only 250 of these are true CKD cases, giving it a precision of 98 percent. Smaller values for davies bouldin score and higher values for mutual information_scores, adjustedrand scores, Vmeasurescore, silhouette scores and calinskiharabasz scores indicate s how good the clustering is. All the internal validation scores such a s silhouette score, calinskiharabasz score and daviesbouldin score are slightly better for DB scan than for K-means. However, with an accuracy of 99.3 percent and an F1-score of 99.4 percent, K-means clustering outperforms the other three approaches.

The 24 features were ranked using Pearson, Chi-2, RFE, Random Forest, Logistic Regression, and SHAP. The results obtained for the final reduced feature set are shown Table 11. For these highly reduced feature sets, Autoencoder yielded an unsatisfactory result, while DBscan and Isolation Forest produced acceptable results. However, k-means had a low Daviesbouldin score and high other cluster validation scores which indicates that K-means performs well with a reduced feature set. It has ah 99% accuracy and a 99.2 % f1-score.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

Table 10. Validation score for all features

	KMEANS	DBSCAN	AUTOENCODER	IFOREST
Recall	0.904	1	0.928	1
Precision	1	0.988	0.928	0.962
f1 score	0.95	0.994	0.928	0.98
Accuracy score	0.94	0.993	0.91	0.975
Mutualinfo score	0.724	0.941	0.548	0.849
Adjustedrand score	0.774	0.97	0.67	0.902
Vmeasure score	0.724	0.941	0.548	0.849
Silhouette score	0.221	0.194	-0.015	0.012
Calinskiharabasz score	107.68	101.123	12.034	15.324
Daviesbouldin score	1.593	1.499	3.621	3.187
TP	226	250	232	250
TN	150	147	132	140
FP	0	3	18	10
FN	24	0	18	0

Table 11. Validation scores for reduced features

	KMEANS	DBSCAN	AUTOENCODER	IFOREST
Recall	1	0.956	0.556	0.952
Precision	0.984	1	0.965	0.859
f1 score	0.992	0.978	0.706	0.903
Accuracy score	0.99	0.973	0.71	0.873
Mutualinfo score	0.926	0.84	0.258	0.441
Adjustedrand score	0.96	0.893	0.171	0.55
Vmeasure score	0.926	0.84	0.258	0.441
Silhouette score	0.321	0.327	0.211	0.178
Calinskiharabasz score	195.001	201.528	138.11	109.964
Daviesbouldin score	1.055	1.115	1.396	1.223
TP	250	239	139	238
TN	146	150	145	111
FP	4	0	5	39
FN	0	11	111	12

H. COMPARISON OF THE PROPOSED MODEL WITH PREVIOUS WORK

There are only a limited number of studies, using unsupervised systems and algorithms to solve the issue of early detection of CKD. However, in detecting CKD, there were some studies based on semi-supervised and supervised learning which were worth mentioning. Relevant studies have been included for performance comparison in table 12.

From the comparison table it can be seen that no existing work in detecting CKD achieved an accuracy of more than 99.0% whereas our proposed method showed a maximum accuracy of 99.3% using the K-means Clustering algorithm. Most studies did not employ feature selection techniques, and those that did not clearly state why some features were left out. Our research sorting out the most

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

important features for disease prediction leaving out less important ones. Using an unsupervised method, combined with appropriate feature selection techniques led to an improvement in \accuracy for detecting CKD.

Table 12. Comparison of existing and proposed work

Author	Approach	Additional description	Result
Sarah A. Ebiaredoh-Mienye et.al. [9]	Integrated unsupervised learnings-enhanced sparse autoencoder (SAE) and supervised Softmax regression ~ Semi-supervised	Worked with three different diseases: • Chronic Kidney Disease (CKD) • Cervical cancer • Heart disease	CKD accuracy = 98%
Aditya Khamparia et. al [8]	Deep learning framework for chronic kidney disease classification Unsupervised Stacked Autoencoder model utilizing multimedia data with supervised Softmax classifier ~ Semi-supervised	- UCI dataset with 400 CKD patients with 25 attributes - 10 most significant attributes selected - Used Stacked Autoencoder as feature selector	Max accuracy = 100%
El-Houssainy A. Rady, Ayman S. Anwar [42]	Prediction of kidney disease stages using: 1. Probabilistic Neural Networks (PNN) 2. Multilayer Perceptron (MLP) 3. Support Vector Machine (SVM) 4. Radial Basis Function (RBF) algorithms	- UCI dataset of 400 patient-data; used 361 data - Identified 5 disease severity stages - Best performance achieved by: PNN - No feature selection algorithm used; all 25 features (11 numerical, 14 categorical)	Max accuracy = 96.7%
S.Gopika and Dr. M.Vanitha et.al. [15]	• Fuzzy C Means • K-Means clustering • K-medoids	Clusters of different stages in chronic kidney disease according to its severity - Best performance achieved: Fuzzy C-Means	Maximum accuracy Fuzzy C-Means = 89%
Adeola Ogunleye, Qing-Guo Wang et.al.[6]	Extreme Gradient Boosting (XGBoost)	- UCI dataset of 400 patient-data with 250 CKD and 150 CKD-free cases - No feature selection algorithm used; all 25 features taken	Accuracy = 98.7%
Huseyin Polat et. al. [18]	SVM classifier	Feature Selection algorithm were applied- 1. Wrapper approach 2. Filter approach Best method: Filter approach	Accuracy = 98.5%

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3109168, IEEE Access

Zixian Wang, Jae Won Chung et. al. [19]	Associative Classification Technique implementing algorithms: 1. IBk (k-nearest-neighbor) 2. ZeroR 3. OneR 4. Naive Bayes 5. J48	UCI dataset of 400 patient-data with 250 CKD and 150 CKD-free cases - Best performance achieved by: IBk - No feature selection algorithm used; all 25 features taken	Accuracy = 99%
Zuherman Rustam et. al. [21]	Analysis of gene expression data using: 1. Random Forest 2. Support Vector Machine (SVM)	- Used Gene Expression Omnibus (GEO) database - Simulated 48 samples where 36 training and 12 testing - Very small sized dataset	Accuracy = 83.4%
Proposed Method	Algorithm used: 1. K-means clustering 2. DB-scan 3. Autoencoder 4. I-forest	- Different feature selection methods including SHAP used Various unsupervised algorithms on the patient medical record - Best performance achieved by: K-means clustering	**Accuracy = 99.3%**

IV. CONCLUSION AND FUTURE WORK

We developed an approach for improved prediction and detection of Chronic Kidney Disease based on various unsupervised machine learning approaches. The best features were selected using ensemble learning measurement SHAP. The data were then classified and validated. This resulted in a 91% accuracy for I-forest, 94% for DB-Scan, 97.5% for Autoencoder and, the best, 99.3% for K-means clustering. This outperformed other unsupervised algorithms. As an extension of our current work, we would like to detect the five different stages of Chronic Kidney Disease in a similar manner. Thus, would support the medical community in just to detecting the existence of the disease, but also in identifying the stages of the disease.

REFERENCES

[1] B. Khan, R. Naseem, F. Muhammad, G. Abbas, and S. J. I. A. Kim, "An empirical evaluation of machine learning techniques for chronic kidney disease prophecy," *IEEE Access*, vol. 8, pp. 55012-55022, 2020.

[2] C. Thongprayoon *et al.*, "Promises of big data and artificial intelligence in nephrology and transplantation," ed: *Multidisciplinary Digital Publishing Institute*, 2020.

[3] S. Vallabhajosyula *et al.*, "Contemporary National Outcomes of Acute Myocardial Infarction-Cardiogenic Shock in Patients with Prior Chronic Kidney Disease and End-Stage Renal Disease," *Journal of clinical medicine*, vol. 9, no. 11, p. 3702, 2020.

[4] N. Tangri *et al.*, "Multinational assessment of accuracy of equations for predicting risk of kidney failure: a meta-analysis," *Jama*, vol. 315, no. 2, pp. 164-174, 2016.

[5] O. Viktorsdottir, R. Palsson, M. B. Andresdottir, T. Aspelund, V. Gudnason, and O. S. J. N. D. T. Indridason, "Prevalence of chronic kidney disease based on estimated glomerular filtration rate and proteinuria in Icelandic adults," *Nephrology Dialysis Transplantation*, vol. 20, no. 9, pp. 1799-1807, 2005.

[6] A. Ogunleye, Q.-G. J. I. A. t. o. c. b. Wang, and bioinformatics, "XGBoost model for chronic kidney disease diagnosis," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 17, no. 6, pp. 2131-2140, 2019.

[7] N. G. Raju, K. P. Lakshmi, K. G. Praharshitha, and C. Likhitha, "Prediction of chronic kidney disease (CKD) using Data Science," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 642-647: IEEE.

[8] A. Khamparia *et al.*, "KDSAE: Chronic kidney disease classification with multimedia data learning using deep stacked autoencoder network," *Multimedia Tools and Applications,* vol. 79, no. 47, pp. 35425-35440, 2020.

[9] S. A. Ebiaredoh-Mienye, E. Esenogho, and T. G. J. E. Swart, "Integrating Enhanced Sparse Autoencoder-Based Artificial Neural Network Technique and Softmax Regression for Medical Diagnosis," *Electronics,* vol. 9, no. 11, p. 1963, 2020.

[10] S. D. Arasu and R. Thirumalaiselvi, "A novel imputation method for effective prediction of coronary Kidney disease," in *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, 2017, pp. 127-136: IEEE.

[11] S. Y. Yashfi, M. A. Islam, N. Sakib, T. Islam, M. Shahbaaz, and S. S. Pantho, "Risk Prediction Of Chronic Kidney Disease Using Machine Learning Algorithms," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1-5: IEEE.

[12] S. Drall, G. S. Drall, S. Singh, B. B. J. I. J. o. M. Naib, Technology, and Engineering, "Chronic kidney disease prediction using machine learning: A new approach," *International Journal of Management, Technology And Engineering*, vol. 8, no. 5, pp. 278-287, 2018.

[13] M. Almasoud, T. E. J. I. J. o. S. C. Ward, and I. Applications, "Detection of chronic kidney disease using machine learning algorithms with least number of predictors," *International Journal of Soft Computing and Its Applications,* vol. 10, no. 8, 2019.

[14] M. A. Fisher and G. W. J. J. o. p. Taylor, "A prediction model for chronic kidney disease includes periodontal disease," Journal of periodontology, vol. 80, no. 1, pp. 16-23, 2009.

[15] S. Gopika and M. J. M. l. Vanitha, "Machine learning Approach of Chronic Kidney Disease Prediction using Clustering," *Machine learning*, vol. 6, no. 7, 2017.

[16] S. Aljahdali and S. N. J. I. j. o. c. a. Hussain, "Comparative prediction performance with support vector machine and random forest classification techniques," *International journal of computer applications,* vol. 69, no. 11, 2013.

[17] B. Ravindra, N. Sriraam, and M. J. I. J. E. T. Geetha, "Classification of non-chronic and chronic kidney disease using SVM neural networks," *Int. J. Eng. Technol,* vol. 7, no. 1, pp. 191-194, 2018.

[18] H. Polat, H. D. Mehr, and A. J. J. o. m. s. Cetin, "Diagnosis of chronic kidney disease based on support vector machine by feature selection methods," *Journal of medical systems,* vol. 41, no. 4, p. 55, 2017.

[19] Z. Wang *et al.*, "Machine Learning-Based Prediction System For Chronic Kidney Disease Using Associative Classification Technique," *International Journal of engineering & Technology*, vol. 7, pp. 1161-1167, 2018.

[20] E.-H. A. Rady and A. S. J. I. i. M. U. Anwar, "Prediction of kidney disease stages using data mining algorithms," *Informatics in Medicine Unlocked*, vol. 15, p. 100178, 2019.

[21] Z. Rustam, E. Sudarsono, and D. Sarwinda, "Random-Forest (RF) and Support Vector Machine (SVM) Implementation for Analysis of Gene Expression Data in Chronic Kidney Disease (CKD)," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 546, no. 5, p. 052066: IOP Publishing.

[22] D. M. Dr.P.Soundarapandian.M.D., L.Jerlin Rubini, Dr.P.Eswaran Assistant Professor, "Chronic_Kidney_Disease Dataset,".

[23] R. K. Singh and M. J. P. C. S. Sivabalakrishnan, "Feature selection of gene expression data for cancer classification: a review," *Procedia Computer Science,* vol. 50, pp. 52-57, 2015.

[24] C.-T. Su and C.-H. J. E. S. w. A. Yang, "Feature selection for the SVM: An application to hypertension diagnosis," *Expert Systems with Applications*, vol. 34, no. 1, pp. 754-763, 2008.

[25] A. G. Karegowda, M. Jayaram, and A. J. I. j. o. C. a. Manjunath, "Feature subset selection problem using wrapper approach in supervised learning," *International journal of Computer applications,* vol. 1, no. 7, pp. 13-17, 2010.

[26] R. Parthiban *et al.*, "Prognosis of chronic kidney disease (CKD) using hybrid filter wrapper embedded feature selection method," *European Journal of Molecular & Clinical Medicine,* vol. 7, no. 9, pp. 2511-2530, 2021.

[27] X. Xiaomao, Z. Xudong, and W. Yuanfang, "A Comparison of Feature Selection Methodology for Solving Classification Problems in Finance," in *Journal of Physics: Conference Series*, 2019, vol. 1284, no. 1, p. 012026: IOP Publishing.

[28] Ebrahime Mohammed Senan, Mosleh Hmoud Al-Adhaileh, Fawaz Waselallah Alsaade, Theyazn H. H. Aldhyani, Ahmed Abdullah Alqarni, Nizar Alsharif, M. Irfan Uddin, Ahmed H. Alahmadi, Mukti E Jadhav, Mohammed Y. Alzahrani, "Diagnosis of Chronic Kidney Disease Using Effective Classification Algorithms and Recursive Feature Elimination Techniques", Journal of Healthcare Engineering, vol. 2021, Article ID 1004767, 10 pages, 2021. https://doi.org/10.1155/2021/1004767

[29] R. Kiani, S. Mahdavi, and A. J. I. J. o. A. R. i. A. I. Keshavarzi, "Analysis and prediction of crimes by clustering and classification," *International Journal of Advanced Research in Artificial Intelligence,* vol. 4, no. 8, pp. 11-17, 2015.

[30] H. S. Emadi and S. M. J. W. P. C. Mazinani, "A novel anomaly detection algorithm using DBSCAN and SVM in wireless sensor networks," *Wireless Personal Communications*, vol. 98, no. 2, pp. 2025-2035, 2018.

[31] M. Schultz and M. Tropmann-Frick, "Autoencoder neural networks versus external auditors: Detecting unusual journal entries in financial statement audits," 2020.

[32] G. A. Susto, A. Beghi, and S. McLoone, "Anomaly detection through on-line isolation forest: An application to plasma etching," in *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 2017, pp. 89-94: IEEE.

[33] L. J. Deborah, R. Baskaran, A. J. I. J. o. C. S. Kannan, and E. Survey, "A survey on internal validity measure for cluster validation," *International Journal of Computer Science & Engineering Survey,* vol. 1, no. 2, pp. 85-102, 2010.

[34] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE international conference on data mining*, 2010, pp. 911-916: IEEE.

[35] M. Mughnyanti, S. Efendi, and M. Zarlis, "Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation," in *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 725, no. 1, p. 012128: IOP Publishing.

[36] D. L. Davies, D. W. J. I. t. o. p. a. Bouldin, and m. intelligence, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence,* no. 2, pp. 224-227, 1979.

[37] X. Wang and Y. Xu, "An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 569, no. 5, p. 052024: IOP Publishing.

[38] H. B. Zhou and J. T. Gao, "Automatic method for determining cluster number based on silhouette coefficient," in *Advanced Materials Research*, 2014, vol. 951, pp. 227-230: Trans Tech Publ.

[39] R. R. de de Vargas and B. R. C. Bedregal, "A way to obtain the quality of a partition by adjusted rand index," in *2013 2nd Workshop-School on Theoretical Computer Science*, 2013, pp. 67-71: IEEE.

[40] N. X. Vinh, J. Epps, and J. J. T. J. o. M. L. R. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *The Journal of Machine Learning Research,* vol. 11, pp. 2837-2854, 2010.

[41] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, pp. 410-420.

[42] M. Elhoseny, K. Shankar, and J. J. S. r. Uthayakumar, "Intelligent diagnostic prediction and classification system for chronic kidney disease," *Scientific reports,* vol. 9, no. 1, pp. 1-14, 2019.

[43] Nasir, Inzamam M., Muhammad A. Khan, Mussarat Yasmin, Jamal H. Shah, Marcin Gabryel, Rafał Scherer, and Robertas Damaševičius. 2020. "Pearson Correlation-Based Feature Selection for Document Classification Using Balanced Training" *Sensors* 20, no. 23: 6793. https://doi.org/10.3390/s20236793

[44] Mayyadah R. Mahmood 2021m Two Feature Selection Methods Comparison Chi-square and Relief-F for Facial Expression Recognition „*J. Phys.: Conf. Ser.* **1804** 012056

[45] J.Padmavathil,Logistic regression in feature selection in data mining, IJER Volume 3, Issue 8, August-2012 1 ISSN 2229-5518

[46] Chen, RC., Dewi, C., Huang, SW. *et al.* Selecting critical features for data classification based on machine learning methods. *J Big Data* **7,** 52 (2020). https://doi.org/10.1186/s40537-020-00327-4

[47] H. Liu, M. Zhou and Q. Liu, "An embedded feature selection method for imbalanced data classification," in IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 3, pp. 703-715, May 2019, doi: 10.1109/JAS.2019.1911447.

[48] W. E. Marcílio and D. M. Eler, "From explanations to feature selection: assessing SHAP values as feature selection mechanism," 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2020, pp. 340-347, doi: 10.1109/SIBGRAPI51738.2020.00053.