CHRONIC KIDNEY DISEASE PREDICTION USING ENSEMBLE MACHINE LEARNING

A thesis submitted to the Department of Electronics and Communication Engineering in partial fulfillment of the requirements for the degree of Master of Science in Electronics and Communication Engineering

Submitted by

Md. Minarul Islam Raju

Student ID: 2005107

Session: 2020-2021

Supervised by

Professor Md. Mehedi Islam

Department of ECE, HSTU

Department of Electronics and Communication Engineering

Hajee Mohammad Danesh Science and Technology University (HSTU)

Dinajpur-5200, Bangladesh.

2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

FACULTY OF POST GRADUATE STUDIES

HAJEE MOHAMMAD DANESH SCIENCE AND TECHNOLOGY UNIVERSITY, DINAJPUR-5200, BANGLADESH

CERTIFICATE

This is to certify that the work entitled as **"Chronic Kidney Disease Prediction using Ensemble Machine Learning"** by Md. Minarul Islam Raju has been carried out under our supervision. To the best of our knowledge, this work is an original one and was not submitted anywhere for a diploma or a degree.

Supervisor

……

Md. Mehedi Islam

Professor

Department of Electronics and Communication Engineering.

Hajee Mohammad Danesh Science and Technology University Dinajpur-5200, Bangladesh.

Co-Supervisor

……..

Md. Selim Hossain

Lecturer

Department of Electronics and Communication Engineering.

Hajee Mohammad Danesh Science and Technology University Dinajpur-5200, Bangladesh.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
FACULTY OF POST GRADUATE STUDIES

HAJEE MOHAMMAD DANESH SCIENCE AND TECHNOLOGY UNIVERSITY, DINAJPUR-5200, BANGLADESH

DECLARATION

The work entitled **"Chronic Kidney Disease Prediction using Ensemble Machine Learning"** has been carried out in the Department of Electronics and Communication Engineering, at Hajee Mohammad Danesh Science and Technology University is original and conforms to the regulations of this University. We understand the University's policy on plagiarism and declare that no part of this thesis has been copied from other sources or been previously submitted elsewhere for the award of any degree or diploma.

……...

Md. Minarul Islam Raju

Student ID: 2005107

Session: 2020-2021

The thesis titled **"Chronic Kidney Disease Prediction using Ensemble Machine Learning"** submitted by Md. Minarul Islam Raju, Student ID: 2005107 and Session 'January-June' 2023, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Electronics and Communication Engineering.

BOARD OF THESIS EXAMINERS

Professor Dr. Md. Mahabub Hossain — Examiner and Chairman
Department of ECE, HSTU, Dinajpur.

Professor Md. Mehedi Islam — Examiner (Internal)
Department of ECE, HSTU, Dinajpur.

Md. Abubakar Siddik — Examiner (Internal)
Assistant Professor
Department of ECE, HSTU, Dinajpur.

Professor Dr. Sajjad Waheed — Examiner (External)
Department of ICT,
MBSTU, Santosh, Tangail-1902.

ACKNOWLEDGMENT

I am grateful to Allah, the Almighty for providing me the strength to carry out this research work because I could not have done without it. I would like to express my sincere and heartiest to my honorable thesis supervisor, Professor Md. Mehedi Islam, and Co-Supervisor, Lecturer, Md. Selim Hossain, Department of Electronics and Communication Engineering (ECE), Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur-5200, Bangladesh, for their ongoing inspiration, direction, and keen support that helped me throughout the time of my research work. Nothing is comparable to their enthusiastic advice and the freedom they gave me in my research. I am thankful to them for their collaboration throughout my thesis work. I would like to thank each member in this department. Also, I would like to thank the Institute of Research and Training (IRT), of this university which supported and enabled me to achieve my goal. I want to express my gratitude to all my friends for their cooperation. I want to express my gratitude to my parents for their unwavering support and collaboration.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The system of clinical support is implicated by the problem of the large variance in chronic illness prognosis. One of the main factors contributing to the deaths of vast populations worldwide who have severe conditions like chronic kidney disease (CKD) is that healthcare systems are very concerned when this condition is diagnosed. CKD is currently regarded as one of the primary disorders. Most people are impacted by their erratic lifestyle. A disease of the chronic kidney damages the internal organs of the human body by accumulating junk and hazardous fluids in the blood. It usually denotes kidney damage that worsens over time until end-stage renal disease occurs. A healthy lifestyle can be recommended, and it can be reduced by early-stage prediction. A huge quantity of historical information is used in Machine Learning (ML), which is the technique of educating computers to make wise classification judgments. In our study, we provide appropriate methods and findings for this disease's precise and early identification to save people from this life-threatening condition. To obtain balanced data from imbalanced data, we build a pipeline that includes outlier elimination, data standardization, and imbalance handling approaches. Therefore, utilizing machine learning methods for analysis, including Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), k-Nearest Neighbors (k-NNs), and Stochastic Gradient Descent (SGD), we forecast kidney illness from balanced data. By evaluating the baseline results, we have seen that SVM, k-NN, and RF perform at 98% efficiency, while LR and SGD work at 97% efficiency. Then we considered RF, SVM, and LR but not k-NN because k-NN isn't fast in actual time. And, it needs to keep updated with all training data and discover neighbor nodes, whereas LR can efficiently extract information. And we have also been told that RF, SVM, and LR algorithms have better performance than others for predicting chronic kidney diseases. That's why we have ensemble RF, SVM, and LR machine learning algorithms. The proposed ensemble RSL also shows a better result, which is 99%, that is superior to other classifiers. The suggested ensemble RSL makes it easier for the domain of medical analysis to predict chronic kidney disease than previous approaches.

CHAPTER 1

INTRODUCTION

1.1 Introduction

A critical concern that requires protection by our population is chronic kidney disease (CKD). This chapter contains the background of this study, the motivation, the objectives of our research work, and the organization of the remaining chapters.

1.2 Background

Chronic nephritic illness (CNS) is another name for chronic kidney disease (CKD). It outlines restrictions that have an impact on our kidneys and lower our chance of remaining healthy. Anemia (low blood count), high blood pressure, brittle bones, and injury to the nerves are a few of the consequences to be concerned about [1]. Prior detection and therapy are usually necessary to prevent chronic uropathy from getting worse. As the number of people suffering from renal diseases rises quickly, there is a great deal of concern worldwide. Chronic Kidney Disease (CKD), a devastating kind of renal illness, damages the body inside by increasing dangerous fluids and waste in the blood. Such physical deterioration causes kidney failure, which ultimately results in mortality, [2]. A variety of end-stage organ failures, including kidney failure, are still brought on by chronic illnesses like disease of cardiovascular and eyesight loss. The sole artificial method for maintaining kidney function is dialysis, which is unpleasant, costly, and challenging. The Global Health Organization indicates that the likelihood of dying from renal disease is increasing yearly and affects millions of individuals worldwide. Due to this, it is essential to obtain a precise prognosis as soon as possible so that any necessary safeguards may be put in place without delay. We must first identify the disease pattern to reduce the serious symptoms of this harmful kidney functional issue. Many academics are focusing on that problem, especially on the chronic kidney disease (CKD) database utilizing both statistical and ML approaches. Machine learning algorithms [3] are effective forecasters and generalizers. Typically, an ML algorithm consists of a classifier or two separate algorithms running in parallel. A clinical presentation, symptoms, and blood test may be used to forecast kidney disease by using a decision support system that performs illness diagnosis. However, Machine Learning (ML) algorithms must practice on the test data first [4]. But ML methods also appear to be reasonably effective in making judgments in the context of disease automated diagnosis, [5]. Several models are used in ML, and they are applied to computer programs in different ways. There are a few examples of popular ML models, including neural networks, decision trees, Nave Bayes, and logistic regression (LR). Which one is utilized depends on a variety of factors, including the domain, the model's established

classification accuracy, and its processing power. To enhance classification accuracy, "model ensembles" [6] of combined models are frequently utilized in machine learning competitions. Using several machine learning methods, we project kidney illness in this study. This study includes a few classification methods that operate well on CKD, including Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), k-Nearest Neighbors (k-NN), and Stochastic Gradient Descent (SGD) [7]-[8]. Additionally, we have put forth an ensemble approach that also predicts chronic kidney disease (CKD) by combining RF, SVM, and LR. Finally, we compare the ensemble RSL model to other baseline models and show the result in the ROC curve. ROC analysis is also being utilized more and more in machine learning, and related applications have also appeared in economics [9]. It emphasizes the value of transferring information from one field to another and has mostly been used in the medical sector. The correlation between the rate of false positives and true positives is depicted graphically by the ROC [10]curve.

1.3 Motivation

Most people in Bangladesh are unaware of kidney-related issues. They are genuinely unsure about whether they have renal disease or not. Globally, an estimated 1.2 million individuals die every year from kidney related problems. Out of around 18 million individuals, 35,000-40,000 CKD patients in Bangladesh experience renal failure each year. Kidney illness is more likely to affect persons over the age of 40, per a study [11]. The Kidney Prediction that works well has not been the subject of many studies. In addition to this, we observe that the emphasis in the modern world is mostly on systems of recommendation. A system must allow users to make decisions independently of other people to be recommended. It should be necessary to have categorized data to take decisions alone. In this type of research, we categorize data and forecast kidney disease, which has been done for all the reasons. Numerous studies have been done on predicting kidney disease. The methodologies used to categorize data on renal disease are few and few between, nevertheless. To find the greatest accuracy in our results, we are currently using five different classification approaches. Additionally, the best three classification algorithms are identified, as they can be combined to produce an effective ensemble ML technique for forecasting chronic kidney disease (CKD).

1.4 Objectives

In order to reach the following objectives, this study creates a model to identify the chronic kidney disease (CKD):

- Finding the most effective classifier for chronic kidney disease (CKD) using balanced data is the major objective.
- We also propose an ensemble machine-learning model for how we can predict CKD more accurately.
- To accurately forecast chronic kidney disease (CKD), we also compare the expected results of machine learning algorithms and ensemble RSL.
- By contrasting these outcomes, we may also determine which model is more effective in accurately predicting chronic kidney disease (CKD).

1.5 Thesis Outline

The remaining portions of the thesis are arranged as follows:

- Chapter 1: Introduction. The current situation in chronic kidney disease, the necessity to detect diseases mean the problem statement, motivation, and our contribution added.
- Chapter 2: A literature review. Related recent work on this problem is added to this chapter. The techniques, related results, and analysis are also added.
- Chapter 3: Machine learning techniques and tools. We focus on major machine learning techniques (like supervised, unsupervised, and reinforcement learning) and important tools, including theory.
- Chapter 4: Methods and Materials. The proposed methodology is discussed here, with a proper diagram and related algorithms.
- Chapter 5: Results and Discussion. The experiment performed by our methodology is described and analyzed with the necessary figures, diagrams, and tables.
- Chapter 6: Conclusion and Future work. This is the ending part of our work which is described. It is a summary of our overall work. Future work related to our model is also added at the end of this section. This section represents what we do in this thesis.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

We will talk about various disease prediction methods in this chapter, along with a pertinent literature review.

2.2 Review of Related Works

Researchers are particularly interested in predicting CKD. To increase classification accuracy, most of them use ML algorithms, conventional techniques, and ensemble techniques. Additionally, various hybrid methods or algorithms are used to categorize kidney illness from secondary sources found at UCI and Kaggle.

Estimated glomerular filtration rate [eGFR] and albuminuria were two chronic kidney disease (CKD) indicators that were commonly assessed in hospitals and enhanced the identification of emergent illnesses of the heart. However, there is no consistent strategy for incorporating these markers into CVD risk assessment in most important clinical guidelines. “CKD patch” was a proven technique for calibrating and enhancing the risk predictions made by well-established equations using CKD measurement. Matsushita et al. [12]constructed multiple "CKD Patches" utilizing the Pooled Cohort Equation (PCE) using 4,143,535 adult data points from 35 datasets. These patches include albuminuria and eGFR. Additionally, they enhance the PCE-based method for evaluating cardiovascular risk, the risk of atherosclerotic cardiovascular illness, and projecting CVD mortality. For the cardiovascular atherosclerotic illness and PCE-based cardiovascular risk methods, they noticed an improvement in their forecasting accuracy using the chronic kidney failure validation datasets.

To predict acute kidney damage, Song et al. [13] compared the effectiveness of ML algorithms with traditional techniques. The t-tests of post hoc techniques and One-way analysis of variance techniques were performed to compare the average differences in area under the curve between various ML methods. According to these data, ML models work equally to LR scenarios, although ML model performance varies and has better results.

By measuring their effectiveness by their needs and awareness, machine learning methods for diabetes patients offer choice-based hierarchies to produce trustworthy results with predicted precision. A series of data-extracting pointers with regular outcomes on the generated model serve as the foundation for the learning scheme's actions. Analyzing 600 clinical records for diabetes using machine learning techniques, Balusamy, et al. [14] looked at the accuracy of

the initial forecasting of severe renal illnesses, which are the most devastating kidney disorders for people with diabetes. The categorization of the dataset was verified using choice-based structures and naive Bayes. Analyzing the categorizing schedule in terms of naive byes techniques and the structure of choice-based hierarchy depending on selection, it was possible to obtain a result with a high level of accuracy of 90.2%. Neural networks and information grouping techniques were employed to increase the accuracy of the estimation results, which greatly aided in achieving their objectives and provided opportunities for future investigations.

Shanthakumari, et al. [15] intended to create a model named machine learning (ML) that considers hypertension, pharmacological data, and population prevalence projections. For predicting CKD, they applied ML methods with clinical evidence. A contrast analysis was approximated considering different indicators, such as accuracy of classification, error percentage, the f1-measure, and others. The simulation results showed that their suggested ensemble machine learning classifier, such as Ensemble SVM, identified chronic kidney illness from the datasets more accurately than other hybrid approaches.

Ventrella, et al. [16] wanted to speed up the strategic scheduling of therapy by predicting how frequently a CKD patient might need to undergo dialyzation. They created a computer model using a supervised ML algorithm to precisely forecast the length of time a CKD patient will require dialysis. The test has a 94% accuracy rate, a 96% sensitivity rate, and a 91% specificity rate, showing that total renal disease can occur sooner rather than later. Nephrologists can forecast the patient's clinical course with tremendous assistance from the built in computational model. The model's promising outcomes, when combined with the physicians' expertise and knowledge. It can successfully result in more individualized treatment and effective management of both patient needs and facility assets.

A Correlational Neural Network (CorrNN) learning algorithm that is effective in terms of computation and a computerized examination tool were created by Bhaskar, et al. [17] to identify CKD. By adding a classifier, support vector machine, their CorrNN model's prediction accuracy was improved. They developed a combination of models and taught and verified them using a unique sensing approach. To identify the illness, they kept an eye on the amount of urea in the measured saliva. Real-time samples were used in experiments to test the model. Additionally, its effectiveness was evaluated against that of other well-known data classification methods as well as traditional neural networks using convolution. With a success

rate for forecasting of 98.67%, the outcomes demonstrate that their suggested approach outperforms more traditional methods.

A CKD dataset was classified using several classifiers by Almustafa, et al.[18]. They created the techniques to classify CKD datasets by using some classifiers, including Decision Tree (DT), Random Tree (RT), Statistical Gradient Descent (SGD), the J48, Closest Neighbor (k-NN), and Bayesian Naive. Furthermore, they had created a prediction model founded on a selection of features that could accurately predict the occurrence of CKD symptoms. The resultant findings demonstrated that J48 and the tree of decision techniques beat other techniques with 99% accuracy rates.

The possibility of various algorithms for machine learning for making a swift CKD prognosis has been examined throughout their investigation, according to Islam et al. [19]. On this subject, a significant quantity of research has been done. However, by applying predictive modeling, they were enhancing their method. As a result, they examined how the features of the target class related to the data items predictive in their technique. Due to the enhanced assessments of qualities that may be brought about by probabilistic modeling, they can develop a variety of forecasting methods with the use of machine learning and probabilistic analytics. Beginning with 25 factors in addition to the class attribute, this study eventually reduced the list to 30 of those parameters, which it found to be the most effective subset for identifying CKD. A supervised learning setting has been used to test 12 different machine learning-based algorithms. The XgBoost classification technique had the strongest indicator of usefulness out of all 12 alternative machine learning-based predictors. And it had been examined in the supervised learning environment, with accuracy at 0.983, precision at 0.98, recall at 0.98, and an f1-score of 0.98, correspondingly. Their study found that one of the most intriguing ways to benefit from recent developments in machine learning and predictive modeling was to develop new approaches. However, when predicting the field of renal illness, they can be used to evaluate the accuracy.

Dekka, et al. [20] included healthcare and disease diagnosis for every area of research that has room for Artificial Intelligence (AI). To increase public knowledge of serious health hazards, including chronic kidney disease (CKD), they employed machine learning techniques. Kidney disease prevents them from doing their normal job of filtering blood. Due to the potential for long-term health effects, it was difficult to predict, recognize, and prevent such a

condition. To address this issue early on, machine learning techniques enabled more accurate forecasting. When they were utilized to solve these issues, several approaches, including Logistic Regression, Naive Bayes, and Decision Trees named ML algorithms, were used. Experiments were carried out in a MATLAB environment, leveraging a large data set. By improving prognosis and diagnosing patients more accurately at an early stage, Logistic Regression suggests the possibility of decreasing mortality from chronic renal disease.

By developing a website-based risk identification system, Kanda, et al. [21] evaluated the efficacy of a machine learning system's capacity to anticipate these hazards in CKD patients. They created 16 risk evaluation machine learning methods utilizing Random Forest (RF), Gradient Boosting, Decision Tree, and Extreme Gradient Boosting using 22 chosen attributes to forecast the key outcome. These models were built using data from the electronic health records of CKD patients that had been continuously examined. The effectiveness of the models was assessed using information from a three-year perspective survey of people with CKD. Because they were so good at foreseeing the results, two RF models, one with 22 features and another using 8 features of time-series data were selected to be employed in a risk-prediction method. The C-statistics for the 22 and 8 attribute RF models were excellent in verification, at 0.932 and 0.93 respectively, for the prediction of the outputs. The model of Cox proportional hazards using splines shows a considerable correlation between high likelihood and a high chance of an occurrence. Furthermore, the hazard ratio for the 22-variable model was 104.9, whereas the hazard ratio for the 8-variable model was 90.9, showing that patients with high possibilities were at greater risk than patients with low possibilities. After that, a website-based risk-identification system was created to employ models in hospitals. Table 1 shows how risk categories of outcome incidents are characterized using machine learning algorithms [21].

Table 1. Based on machine learning models, outcome events of risk groups are categorized.

Names of techniques	**All the time-series variables in the RF model**			**Eight variables of time-series data for the RF model**		
Hazard groups	The group of high hazard	The group of low hazard	Value of p	High hazard group	Low hazard group	Value of p
Initial outcomes (%)	43 (2.9)	235 (0.9)	<0.0001	40 (2.7)	238 (0.9)	<0.0001

ESKD (%)	34 (2.3)	153 (0.6)	<0.0001	31 (2.1)	156 (0.6)	<0.0001
Death (%)	9 (0.6)	82 (0.3)	0.099	9 (0.6)	82 (0.3)	0.009
Follow-up period (days)	710 [407, 1038]	901 [527, 1052]	<0.0001	738.5 [428, 1043]	898 [526, 1052]	<0.0001
Median (interquartile range) values are displayed for continuous variables. Variables with categories are displayed as n (%). ESKD stands for end-stage kidney disease.						

The 'MedAi' smart watch-based prediction system, reported by Himi, et al. [22], was characterized by a variety of machine learning (ML) algorithms to identify multiple ailments, including cardiovascular disease, pulmonary disease, and renal failure. A "Sense O'clock" smartwatch prototype that has eleven sensors for tracking bodily data. The three key parts of their solution are an ML model to examine the data and a smartphone application platform to show the forecast outcomes. According to ethical principles, including getting the previous approval of both patients and doctors. A local hospital provided a dataset of patient physical characteristics. They tested several methods, including long and Short-Term Memory (LSTM), Statistical Vector Machine (SVM), Support Vector Regression (SVR), k closest neighbor (k-NN), X- Gradient Boosting (XGBoost), and random forest theory (RF), to see which ML approach fared the best. According to the experimental dataset, the RF algorithm predicts the diseases with a 99.4% accuracy rate, outperforming other machine learning algorithms like SVM, k-NN, XGBoost, etc. And the system reports the user's physical health and suggests necessary treatments, offering full-time help. It was a substantial improvement over current early sickness detection systems and can anticipate several health hazards before they worsen to the point where they are irreversible. Then, using the pertinent, pre-existing methodologies, they built their strategy.

Debal, et al. [23] evaluated both professionals and patients, and making an early forecast is essential to preventing and delaying the progression of renal disease. They used three ML techniques, such as RF, SVM, and DT, as well as the selection feature methods RFECV and UFS, in their study to create the suggested models. The model was assessed using tenfold cross-validation methods. First, with all 19 characteristics the original datasets were applied to the four ML techniques. When they applied the models to the main dataset, they found the highest value of accuracy by using these techniques, including RF, SVM, and

XGBoost. And the accuracy was 99.8%, while for the five-class system, it was 82.56%. DT had the lowest rank when contrasted with RF. But the highest values of the f1-score were produced by RF. SVM with RF and RFECV created binary classes with 99.8% accuracy, and XGBoost produced an accuracy of up to 82.56% for five class datasets. They concluded that multiple-classification work was necessary for comprehending the illness's phases and offering patients the medicines they needed to survive longer. An optimal subset of features was chosen for the models by using feature selection techniques and a supervised ML algorithm. Using unsupervised or deep learning algorithm models was preferable for comparing performance results. Experts can take decisions quickly because of this model. However, it would be better to develop a mobile-based system that would enable professionals to keep an eye on patients' health and help patients use it to learn about their diseases.

For their research, Pal, et al. [24] employed ongoing renal illness attributes from the UCI machine computing library. Additionally, they used trees of decision, logistic regression, and support vector systems as three algorithms to develop a model for forecasting chronic renal failure. A model's performance is influenced by its sensitivity, accuracy, recall, f1-score, support, confusion matrix, and other performance matrices. The forecasting model was taught using both category and non-categorical characteristics from the attributes of CKD. When the base classifiers were applied, it was seen that the choice tree classifier generated better results with respect to accuracy, f1-score, and the precision, and recall, with values of 95.92%, 0.99, 0.98, and 0.98. The decision tree predictor performs superiorly to logistic regression and vector support systems. They found that a decision tree had the best accuracy (97.23%) while using the packing hybrid model to increase the accuracy of fundamental classifiers. They noted that collaboration between patients and physicians could result in the early diagnosis of CKD and possibly save lives.

To make successive models for forecasting the hazard of developing CKD, Dritsas, et al. [25] created a supervised learning technique that mainly combines probabilistic, tree, and collective learning-based models. According to the obtained results, the rotating Forest outperformed the techniques, having a 100% AUC but 99.2% precision, recall, F-measure, and accuracy. The suggested models performed more accurately than previous research that used a comparable dataset. They intend to determine whether CNN and LSTM (Long-Short-Term Memory) models enhance accuracy and will concentrate their study on Deep Learning approaches. To make the most of the possibilities of these models, they achieve two goals. The

previous method will employ a method for enhancing the data, including an SVR-based additional input multiplication approach, to enhance a small dataset before giving it to the ML models. For the latter, they will experiment with a sizable non-synthetic dataset right away.

Singh, et al. [26] built and submitted a deep learning model for the early identification of chronic illness. The investigators of their research applied the Recursive pattern Elimination method to identify the characteristics that were essential for prediction. Serum creatinine, particular gravity, the protein hemoglobin albumin, cell count, and high blood pressure were the most important CKD symptoms. The algorithms used for segmentation received a set of features. Several factors, including categorization reliability, recall, precision, and f-measure, were used to calculate the parallelism analysis. The presented deep neuron framework performed better than the five previous techniques by achieving 100% accuracy, beating out Support Vector Machine, k-Nearest Neighbor, Logistic Regression, Random Forest, and the Naive Bayes technique. The accuracy rates for k-NN, SVM, Naive Bayes, RF, and LR are respectively 92%, 92%, 95%, 97%, and 99%. Several recent academic studies, including Ant Colony-based Optimization Classifier, Neural network, k-NN, Convolutional Neural Networks, SVM, and SAE and Softmax Regression presented, were used to compare the performance to their suggested model. The accuracy of the current works ranged from 85% to 98.5%, but their proposed model achieved 100%. The recommended technique may be effective in the initial stage of CKD identification, according to nephrologists. The anticipated model had the flaw of only having been evaluated on a small number of data points. The experts in pathology will gather the clinical data. The effectiveness of the proposed model will be assessed in the feature using a significant medical database using acid-based factors, excessive parathyroid, hormone inorganic phosphate attention, and night urine production. Additionally, to acquire a more comprehensive viewpoint on the helpful criteria for CKD illness, new features will be employed to evaluate the prediction accuracy. Table 2, [13] depicts the Examine the characteristics of various diseases using a range of machine learning (ML) classifiers.

Table 2. Examine the characteristics of various diseases using a range of machine learning (ML) classifiers.

Researchers	Year	Usable data (N)	Type of model	Qualities of the subject	Findings (best models/predictors)
R.J. Kate et al. [27]	2016	25,521	LR, BN, ENS, RF, SVM	Inpatient care (>60, y. o.)	AKI prediction by using the best ENS.
S. E. Davis et al. [28]	2017	170,675	RF, BN, ANN, LR	Every admittance	Both the ANN and the RF models were better antiemetic.
P. Cheng et al. [29]	2017	48,955	AB, RF, LR	visits with patients	The most effective model was RF.
J. L. Koyner et al. [30]	2018	121,158	GB	Full-age patients that are inpatients	Lower AKI stage prediction causes a decline in extreme GB performance.
H. C. Lee et al. [31]	2018	2,010	ANN, LR, GB, DT, SVM, RF	Surgery of heart	The GB model performed better than expected.
N. Park et al. [32]	2018	21,022	RF and LR	Crab	For predicting AKI, RF has higher accuracy, sensitivity.

L. Adhikari et al. [33]	2019	2,911	RF	An anxious surgical process	The RF techniques containing both interop and preop parameters scored better.
C. Chiofolo et al. [34]	2019	6,530	RF	ICU for health and surgery	The RF technique works better for ongoing AKI population projection in the ICU.
M. Flechet et al. [35]	2019	252	RF	Seriously ill in the ICU	With less overestimation of the risk of AKI, the RF model was assessed similarly to physicians in AKI forecasts.
J. He et al. [36]	2019	76,957	ENS, RF, BN, LR,	Visits with patients	The best performance was displayed by ENS.
J. Parreco et al. [37]	2019	151,098	GB, ANN, LR	ICU admissions	The efficiency of GB was superior.
M. Sun et al. [38]	2019	16,558	LR, ANN, BN, RF, SVM	Critical care unit (ICU)	The best results were achieved by SVM and LR.

Cronin et al. [39]	2015	1,620,898	The RF and LR	Visiting patients in hospitals	Performance on the models was comparable.
Thottakkara et al. [40]	2016	50,318	LR, BN, SVM	Any significant surgery	More effectively than BN were LR and SVM.
N.K. Tran et al. [41]	2019	50	KNN	Burn	When used to forecast AKI in burn victims, KNN is excellent.
Z. Zhang et al. [42]	2019	6,682	LR, GB	Intensive care unit admittance	When predicting whether an AKI will respond to volume changes or not, Extreme GB outperformed LR.

Ghosh, et al. [43] said that one of the biggest issues with mortality rates in the clinical sector today is kidney illness a slow-developing disease that is often discovered too late. It was a serious concern that every year, a large number of people endured it because of the absence of quick detection tools and sufficient ambition. However, prompt disease detection in the earliest stages can save patients' lives. A trustworthy dataset can also help a machine learning algorithms evaluation procedure identify this dangerous diseases stage much more quickly. The entire study was executed using four trustworthy methods: hereafter AB (AdaBoost), hereafter SVM, hereafter LDA (Linear Discriminant Analysis), and hereafter GB (Gradient Boosting). These approaches were chosen because they are well-known and have a track record of providing reliable results. On a dataset available publicly in the UCI machine learning repository, these techniques were put into practice. GB classifiers produced results with a predictably high accuracy of 99.80%. Additionally, a number of indicators of performance assessment were shown to show the successful results. Finally, these standards can be employed to select the most effective and optimal methods for the desired job.

A stunning 63,538 cases of chronic kidney disease (CKD) have been identified, according to an analysis of Indian health records by Gudeti, et al. [44]. Nephropathy in humans often appeared between the ages of 48 and 70. Men had a greater frequency of CKD than women did. Unluckily, since 2015, India has slipped among the top 17 countries for CKD. A state in which the function of the excretory organs gradually deteriorated over time. Early sickness detection and effective treatment may help keep this dreadful disease at bay. Machine learning (ML) is being used in real-world applications throughout many sectors, including fraud detection and the interpretation of medical research findings. The forecasting of chronic diseases uses a variety of ML techniques. The main target was to compare how different machine learning algorithms perform, which were mainly focused on accuracy. Recode had been praised in their study's analysis of performance. Their study's main objective was to use an analysis of the chronic renal disease sample to categorize those with CKD and non-CKD. The benefit of this technique was that it categorized a larger group of victims more quickly, and the prediction procedure took considerably less time, allowing doctors to treat those suffering from CKD as soon as feasible. Because the dataset used in this research was so tiny, with just 400 samples, it would have been preferable to deal with greater amounts of data moving forward or to contrast the outcomes of this database with those of another database with similar features. Additionally, using the appropriate information, it was tried to determine whether a person with this syndrome had a higher likelihood of developing chronic risk syndromes.

Devika, et al. [45] , have previously seen programs for system learning and statistics mining in the medical sector. A novel selection aid technique was employed in their investigation to forecast the occurrence of CKD. Even though their models were also effective at predicting other illnesses. In their study, they used distinct classifiers to predict chronic kidney disease, and it concluded with a comparison of the classifiers' overall performance. The analysis showed that the most effective classifiers were Naive Bayes, Random Forest, and k-NN. Regarding the precision of the price of CKD prediction, the classifier using Random Forests fared better than the alternative. Different workable evolutionary algorithms could be applied to enhance the execution of the expected methods. Here, Random Forest, k-NN, and Naive Bayes techniques were used to find CKD. They also compared the efficiency of the old classification algorithms to other ones now in use. Identification of CKD quickly is important for the timely treatment of those who are ill and for preventing the condition from worsening. The medical zone required early disease diagnosis and prompt treatment.

Numerous renal problems have been reported by people, with chronic kidney illness being one of the main causes, according to Baidya et al. [46] . It was important to diagnose CKD as soon as possible to stop renal disease from worsening while it was still in its initial phases. The disease can currently be detected earlier than in any other way by doctors using ML classifier algorithms. In their research, they suggested a method for promptly identifying CKD illness by considering a dataset of the patient's medical history and eight different ML algorithms. A dataset provided by the medical center that covered roughly two months of that time was used to evaluate the risk of chronic renal condition. They also utilized several classifiers, such as the EXT (Extra Tree), ADB (AdaBoost), k-NNs (k-Nearest Neighbors), GB (Gradient Boosting), XGB (Extreme Gradient Boosting), DT (Decision Tree), GNB (Gaussian Nave Bayes), and RF (Random Forest), to provide the highest identification outcomes. After preparing the data to provide an accurate output, they applied ML algorithms and compared their results. Statistical approaches such as the f1-score, precision, accuracy, recall, and AUC score were utilized to evaluate the results. The results of the investigation demonstrate that k-NNs (k-Nearest Neighbors) and EXT (Extra Tree) approaches surpassed other approaches in receiving an accuracy of 99% prior to GB (Gradient Boost), which reached 98% at the time.

Basak et al. [47] determined approximately 422 million people worldwide have diabetes as of today, which contained 30% having type 1 (adolescent onset) and 10 to 40% having type 2 (adult onset), both of which might eventually result in kidney impairment. It was clear that adult-onset damage from chronic kidney disease (CKD) can be reduced with early identification. The effectiveness of five distinct classifiers involving NB (Naive Bayes), IBK (Instance-Based Learning), RF (Random Forest), DS (Decision Stump), and DT-J48 (Decision Tree-J48) were compared to anticipate CKD in individuals with diabetes simply using a urinary examination. Of all the classifiers, the IBK technique delivered the greatest outcome. People with diabetes will be able to determine if they have CKD or not by comparing several algorithms.

Chen, et al. [48] developed a simple ultrasonography imaging technology that was utilized for the risk of patients with CKD as an additional clinical approach to detect the disease of the chronic renal failure at its initial stage. They developed an identification system using image analysis methods and ML strategies to identify chronic renal failure. Support-vector machine approaches and decisive area-proportional and textural aspects were used for effective analysis. Several clustered groups of CKD victims were evaluated and compared using their

approximated rates of glomerular filtration. The recommended approach may be used in their study as additional proof to help distinguish between distinct clinical diagnoses through the observations of developing abnormalities via ultrasound images.

Biomarker researchers Soreide, et al. [49], enhanced their research and result presentation by using ROC curves properly. The best classification criteria were found using ROC curves. Confounding brought on by shifting criteria and subjective judgments was avoided through ROC curves. Its result should always be viewed in context. A most effective technique does not ensure the final medical results when unique statistical precautions and procedures are needed. Table 3 contrasts the ROC research strategies for past and present studies [50].

Table 3. ROC study designs for future and past investigations.

Features of the Study	Future investigation	Past investigation
References type	Allows the study process to be applied evenly across all disciplines to the corresponding criterion.	Since the reference to the standard was previously in use, it is possible that its performance and perception are commonly non-standardized. Bias might enter a study when some subjects are not exposed to the reference standard.
Subjects for studies	Makes sure there are enough participants with and without the condition, but significant numbers are needed to provide a representative sample that can be generalized.	Gives the researchers the option to select subjects for study who will be present in a large number of different situations.
Analysis readers	Need a clinical reader to examine the data immediately, while many readers commonly do so	Lends itself to being consistently understood by many readers who are not involved in the patient's

	after the fact.	treatment.
Measures of results	For subgroup analysis, there is typically more patient data available. Accuracy of diagnostic tests, impact of medical decisions on patient care, and results for patients.	Accurate tests for diagnosis
Competence	Finding enough topics usually takes a lot of effort and money.	Because chart reviews are used to identify subjects, has a tendency to be significantly shorter as well as less expensive.

CHAPTER 3

MACHINE LEARNING TECHNIQUES AND TOOLS

3.1 Introduction

Methods utilizing machine learning have become widely employed in various applications recently, including speech and picture identification, natural language processing, and predictive analytics for kidney disorders. We cover the fundamentals of machine learning (ML) in this chapter, as well as the various ML methods, such as supervised learning, unsupervised learning, and reinforcement learning, as well as the field of ML tools.

3.2 Machine Learning (ML)

Humanity faces new challenges in the digital age. Every second, massive amounts of data are generated by the internet, smart devices, and ever-advancing technology. In recent times, companies, institutions, scientists, and academics have been looking for new methods and technologies to use this data for a variety of objectives, including detection, recognition, analysis, evidence identification, and recommendations. Almost every industry now uses Artificial Intelligence (AI) to improve its processes and methods, such as medicine, engineering, finance, and manufacturing. One of these trends is machine learning in AI and computer science, which has become a focus for technology specialists. Author Samuel invented the term "Machine Learning" in 1959, and it has been explored and modified by various academics since then. A branch of artificial intelligence known as Machine Learning (ML). It employs statistical techniques to let computers learn from data in real time, without prior guidance or predefined procedures [51]. ML accomplishes this by discovering patterns and relationships between data and events using computer algorithms. These algorithms gain knowledge from the datasets to anticipate the outcomes of similar datasets. Unlike traditional applications that operate within strict, pre-defined guidelines, the capacity of ML algorithms to grow and advance with time. Making it possible to build data-driven applications like computer vision or email filtering, which would be almost impossible with traditional programming methods. ML has three main categories:

• Supervised • Unsupervised • Reinforcement

3.2.1 Supervised Learning

A division of artificial intelligence and machine learning is supervised learning. The phrase "supervised machine learning" is another name for it. And it is determined by its capacity to develop algorithms that correctly classify data and forecast consequences. Additionally, it trains computers how to use the information at hand to uncover hidden insights. A supervised learning technique can be grouped into two different strategies, such as

classification and regression [52]. A concern with these algorithms is that they only work with labeled data, and acquiring data for learning can be expensive. There have been several supervised algorithms investigated, and each has merits and flaws of its own. In ML, selecting the appropriate method is essential because no single algorithm is best for all problems. Among the greatest number of popular supervised algorithms are:

- Support Vector Machines (SVM)
- Random forest (RF)
- Logistic regression (LR)
- Linear regression
- Naive Bayes (NB)
- Decision trees (DT)
- K-Nearest Neighbor (k-NN)
- Neural Network

3.2.2 Unsupervised Learning

In contrast to supervised machine learning, unsupervised learning uses raw data [53]. As an alternative to all of this, we use unlabeled input data that is not segmented based on different attributes and does not also include associated outputs for supervision. Later, the machine learning model receives the unlabeled input data to finish the training phase. It can be categorized into clustering and association problems.

Clustering: This type of task involves grouping similar data points together, such as grouping customers based on their purchasing behavior.

Association: The algorithm looks for underlying patterns and linkages in the data, such as the correlation between buying shirts and pants. Algorithms like K-means clustering, and the priority association algorithm are two examples of unsupervised learning.

3.2.3 Reinforcement Learning

Machine learning models are trained via Reinforcement learning to make a series of judgments [54]. The agent acquires the capacity to carry out a task in a potentially complex and unpredictable environment. Reinforcement learning places artificial intelligence in an environment analogous to competition. The computer employs a method involving experimentation and failure to find a solution. The actions artificial intelligence takes to get the machine to accomplish what the developer wants are honored or penalized.

3.3 Important tools

In this study, we employed various tools and applications in the creation of models and experiments, like most research and studies. Several tools are necessary for these types of studies, such as the programming language Python, which was utilized by the author to develop the model. For chronic kidney illness prediction, a huge amount of data has also been used for training purposes. Additionally, several Python libraries deemed necessary or helpful in creating machine learning models were utilized. All the tools that were used in the thesis have also been described.

Python:

A versatile programming language is Python created by Guido van Rossum with the goal of making programming accessible to everyone. Its simplicity, sensitivity to spacing, and ease of use have made it a popular choice, especially for those without a programming background. Due to its effective and user-friendly libraries that accelerate and simplify development, it has gained tremendous popularity in the disciplines of artificial intelligence and machine learning over the years. In this thesis, the author chose Python as the programming language for model development due to their knowledge and interest in the language.

Pandas:

Pandas is a free software tool that enables effective computation with arrays and matrices in two dimensions. It contains a wide range of functions that make working with this type of data more straightforward. When working with large amounts of data, arrays are a vital tool for data specialists since they can be utilized to speed up operations and increase efficiency. Its memory size is larger than NumPy, and it can also contain dissimilar data types. It is particularly useful for detection and forecasting tasks, where models need to be optimized to perform quickly.

Matplotlib and Seaborn:

Matplotlib and Seaborn are popular plotting tools used to create a variety of graphs and figures. Its ease of use means that it can produce high-quality plots and figures with just a few lines of code, making it a go-to choose for data visualization. Pandas and NumPy are used by Matplotlib to plot a variety of graphs, but Seaborn is an expanded version of Matplotlib that employs Matplotlib in addition to Pandas and NumPy. That’s why Seaborn has been used in this thesis to extract color features and create histograms.

Jupiter Notebook:

The Jupiter Notebook software enables the creation and editing of documents that show a Python program's input and output. These files can be shared with others after being saved. By default, Python and R languages are supported, but Notebook can run a variety of additional kernel environments with some customization.

Computer:

The following features are present in the computer used to train and test the models:

Device: HP ProBook G2-450

Memory: 4 GB RAM

CPU: Core i5 Quad and Core

Graphics: Intel HD 2.20 GB"

System type: 64-bit operating system

CHAPTER 4

PROPOSED METHOD AND MATERIALS

4.1 Introduction

The overall workflow of this study mainly focuses in this chapter. A diagram that summarizes the complete methodology is included, along with information on the dataset, data pretreatment procedures, correlation between the variables, equations, an explanation of the algorithms, and performance measurement tools. This chapter shows a clear scenario of our work.

4.2 Proposed Mechanism

A prediction model is used to forecast chronic kidney disease and identify people with the condition. Many academics have been interested in studying chronic kidney disease datasets to create an effective algorithm to support healthcare professionals in keeping track of probable CKD patients. A variety of trustworthy and effective models to help healthcare professionals have been produced by machine learning (ML) algorithms. Some techniques in ML have been utilized in various disorders relevant to healthcare. Five methods of machine learning (ML), namely Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), k-Nearest Neighbors (k-NN), and Stochastic Gradient Descent (SGD), have been used in this work. In addition, we developed an ensemble (RSL) including RF, SVM, and LR. Firstly, we have created a library function and have also loaded the data by using a method named Pandas to predict chronic kidney diseases. We also applied the SMOTE Tomek technique to the imbalanced data and produced balanced data for improving the accuracy level of prediction.

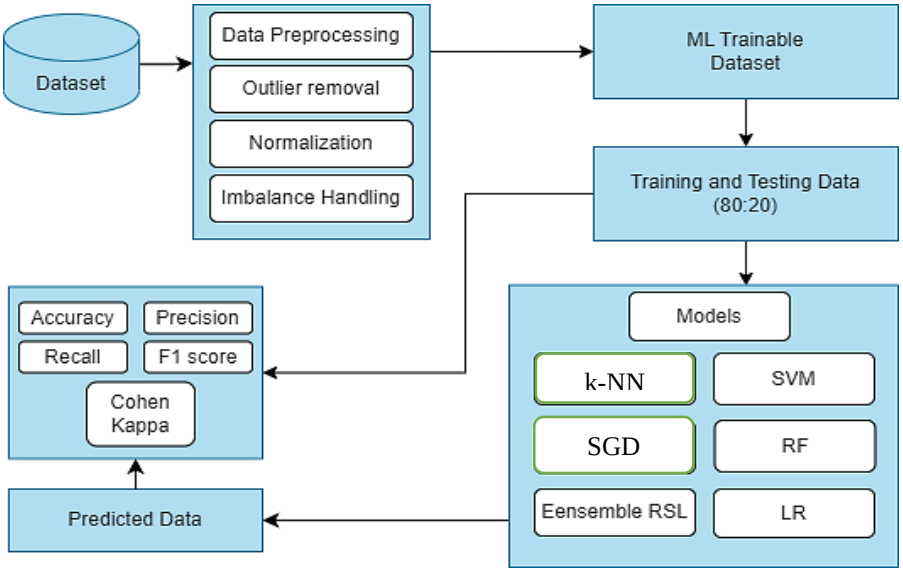

Figure 1. The block diagram for our proposed technique

The dataset is randomly separated into testing and training segments after any outliers have been removed and the normalizing technique has been applied. The testing data is then utilized to provide the expected values after the models have been trained to use the training data. Although the outcomes are good, the level of precision of the algorithms could yet be improved. This improves the model's overall performance by allowing it to make predictions that are more reliable.

The total process is broken into numerous subsections, from data preprocessing to final assessment. All the subsections are explained below, while figure 1 depicts the diagram of our suggested technique. The complete data analysis was divided into two primary stages after the data preparation. On balanced data, CKD was forecast through several machine learning techniques. The training and testing set in each dataset are randomly split (in an 80:20 ratio), and each set is used individually for analysis. To measure the performances of the algorithms, this study utilizes the f1-score, the accuracy, the recall, the precision, the Cohen Kappa, and the ROC score. The bar chart displays the outcomes, while the ROC curve depicts the algorithms' performance.

4.3 Description of Data Sets

The dataset for chronic kidney disease was gathered from UCI [55] and used these in this investigation. A total of 400 instances made up the CKD dataset, which was chosen over two months (samples). Each instance had 24 attributes and one class attribute, of which 11 were numerical and 14 were nominal, as shown in table 4. After collecting the data, the library function is created and imports these data using the Pandas method.

Table 4. Variables with details and data types of the variables

Serial	Variable	Full Name	Type	Values
1.	age	age	numerical	age in years
2.	al	albumin	nominal	al - (0,1,2,3,4,5)
3.	sg	specific gravity	nominal	sg - (1.005,1.010,1.015,1.020,1.025)
4.	bp	blood pressure	numerical	bp in mm/hg
5.	su	sugar	nominal	su - (0,1,2,3,4,5)
6.	rbc	red blood cells	nominal	rbc - (normal, abnormal)
7.	pc	pus cell	nominal	pc - (normal, abnormal)

8.	pcc	pus cell clumps	nominal	pcc - (present, notpresent)
9.	ba	bacteria	nominal	ba - (present, notpresent)
10.	bgr	blood glucose random	numerical	bgr in mgs/dl
11.	bu	blood urea	numerical	bu in mgs/dl
12.	sc	serum creatinine	numerical	sc in mgs/dl
13.	sod	sodium	numerical	sod in meq/l
14.	pot	potassium	numerical	pot in meq/l
15.	hemo	hemoglobin	numerical	hemo in gms
16.	pcv	packed cell volume	numerical	____
17.	wc	white blood cell count	numerical	wc in cells/cumm
18.	rc	red blood cell count	numerical	rc in millions/cmm
19.	htn	hypertension	nominal	htn - (yes, no)
20.	dm	diabetes mellitus	nominal	dm - (yes, no)
21.	cad	coronary artery disease	nominal	cad - (yes, no)
22.	appet	appetite	nominal	appet - (good, poor)
23.	pe	pedal edema	nominal	pe - (yes, no)
24.	ane	anemia	nominal	ane - (yes, no)
25.	class	class	nominal	class - (ckd, notckd)

4.4 Correlation among the variables

The correlation test is a popularly used statistical technique and provides the foundation for numerous applications, including the analysis of data for exploration, structural estimation, data science, etc. [56]. There are many kinds of correlation [57], which are shown in Table 5**.**

Table 5. Different classifications of correlation for entire variables

Different classifications of correlations	
Pearson's correlation	Point-Biserial and biserial correlation
Kendall's rank correlation	Polychoric correlation
Bit_Weight mid correlation	Tetrachoric correlation
Percentage bend correlation	Partial correlation
Distance correlation	Multilevel correlation

In this paper, we use Pearson's Correlation because the variables are normally distributed, their relationship is linear, and both variables are quantitative [58].

4.4.1 Pearson's Correlation

A statistical technique called Pearson's correlation is applied to determine the graphical relationship within the two parameters, x and y. It has a range of -1 to 1+. A correlation can be expressed as +1 for a positive correlation, -1 for a negative correlation, or 0 for none at all. Figure 2 displays correlations that are positive, negative, and zero. We are aware that both variables x and y are continuous. The following equation can be used to get the Pearson's correlation [59] coefficient:

$$P_{x,y} = \frac{\mathrm{Cov}(X,Y)}{d_x d_y} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

From this equation, we know that Cov is the covariance in this instance**,** and dx and dy are the respective x-axis and y-axis standard deviations.

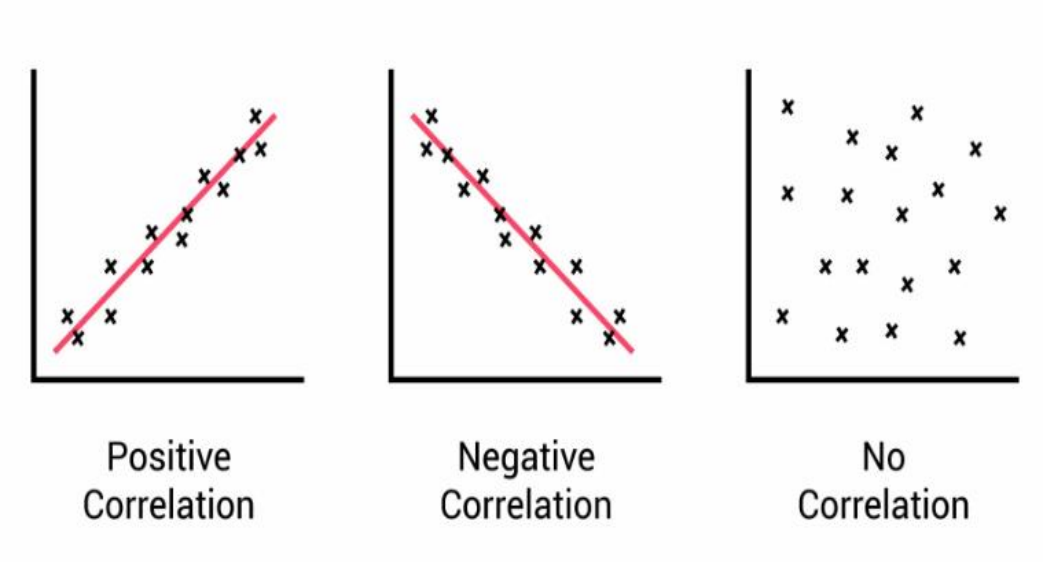

Figure 2. Pearson's correlation with positive, negative, and zero correlation

4.4.2 Correlation heatmap

An illustration of the strength of correlations between numerical data is a correlation heatmap. To determine which variables are correlated with one another and the significance of this association, correlation graphs are utilized. A column usually represents one of the many numerical variables, and the rows show the relationship between each set of variables. Positive values carry a dynamic relationship, while negative values carry a weak relationship. The values in the cells illustrate the quality of the relationship. To find potential connections among variables and measure how potential these connections are, we can use correlation heatmaps. Correlation plots are also used to find outliers and determine both linear and nonlinear correlations [86]. The scheme of colors in the cells makes it simple to see the connections between variables briefly. Finding both linear and nonlinear associations between data is another purpose for correlation heatmaps. To comprehend the linear connection among these variables in the chronic kidney diseases (CKD) dataset, a correlation heatmap is built here. A matrix known as a correlation matrix is created by the set of correlation values between pairs of its characteristics [60]. Finding a correlation between the independent variables is very important. We know that if we feed the model incorrect data, the weight's value will change, and the model won't produce the desired results if the independent variables are correlated. Here, we use Pearson's correlation techniques to plot the correlation between the variables. So, we remember that a strong correlation between an independent variable and any other independent variable or dependent variable is not an issue and has no bearing on the algorithm. But the correlation between the independent variables is a problem that affects the algorithm.

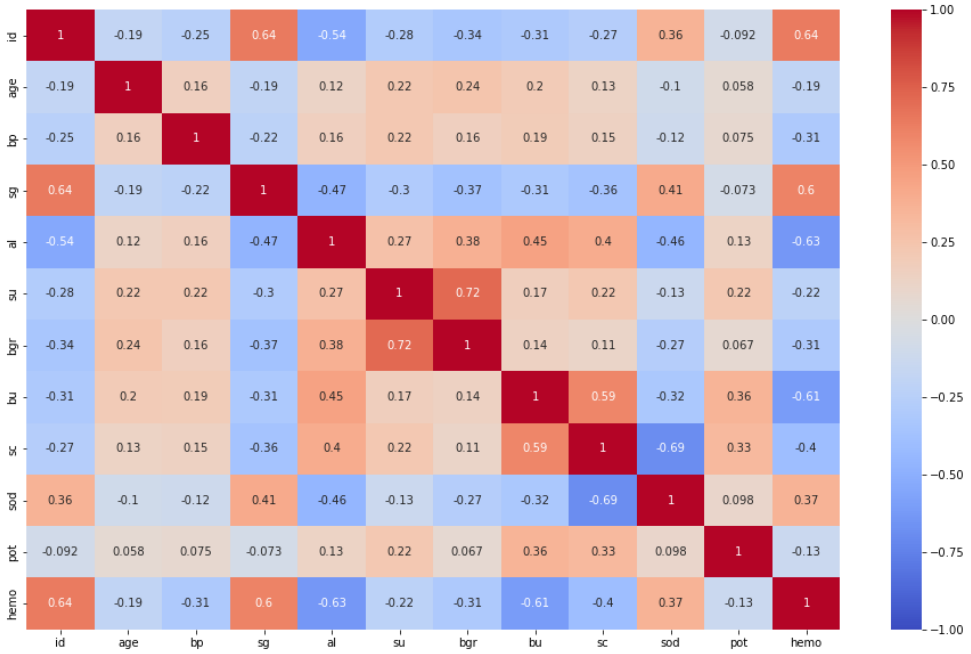

Figure 3. Correlation Heatmap for Chronic Kidney Diseases (CKD)

Figure 3 is shown in the diagram of correlation. There is a scale that is on the right side of the diagram, and the lighter the color, the more correlation there is. The correlation of 1 is happens when we find the same column. In this figure, for example, we see that the bottom age and left age are in the same column, and the correlation is coming in at 1, while rest of the correlation is not high. Instead of being a correlation, the id column is not significant in this model because it is just unique ids. And rest every other place where there is no correlation, then we have to drop this. So, we can now go ahead and finalize the data.

4.5 Data preprocessing techniques

A technique for transforming unprocessed data into a complete form is known as data preparation. The dataset is preprocessed to identify outliers, missing values, noisy values of data, and other irregularities before running it. The following is a description of several common data preparation techniques used in this investigation.

4.5.1 Outlier Detection

Outlier detection is used to correctly remove unnecessary observations from the datasets [61]. By detecting faults, removing their harmful impacts, and using other techniques, it can clean up data before analyzing it. Extreme values that are positioned far from the feature's central tendency are considered outliers. Data input errors, often known as noise in the data, are the cause of illegal outliers, [62]. When dealing with outliers, Clinical data is unable to be altered like other data since these outliers may be real (valid) or significant. To determine if an

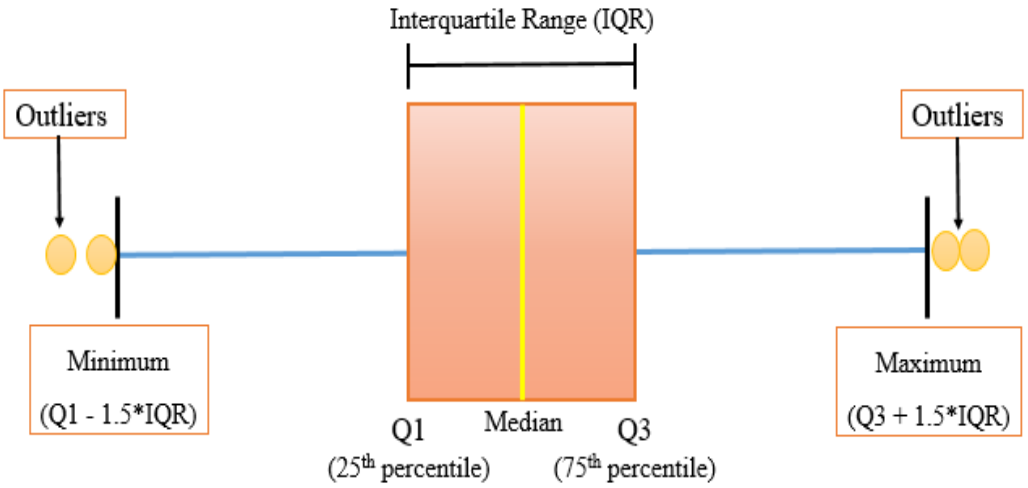

Figure 4. Outlier detection and removing using IQR

outlier is realistic or not, each one that is found in the CKD dataset is examined, [63]. Extreme data points that fall outside of the permissible range in terms of medicine have been handled as missing data and then adjusted. In figure 4, we have discussed the technique, which is outlier detection and removal technique using an Inter Quartile Range (IQR) [64]. The dataset is divided into four equal quartiles, or Q1, Q2, Q3, and Q4, to calculate the IQR, which measures volatility. The first quartile (Q1) separates the 25% of data with the smallest range from the other 75% with greater ranges. The Third Quartile (Q3) separates the lowest 25% of data from the highest 75% of data. IQR stands for the distinction between the first and third quartiles of a population. So,

$$IQR = Q3 - Q1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$\text{The lower limit} = Q1\text{-}1.5*IQR\ldots\ldots\ldots\ldots\ldots (3)$$

$$\text{The upper limit} = Q3+1.5*IQR\ldots\ldots\ldots\ldots\ldots (4)$$

Any values in the dataset that are outside of the upper and lower limits (Q3+1.5*IQR and Q1-1.5*IQR, respectively) are referred to as outliers.

4.5.2 Normalization

The normalization of data is a crucial step in machine learning that helps to ensure a more precise forecast. The greatest datasets are created by combining and analyzing data from many sources. This method's primary benefits are monetary savings, space investments, and accuracy enhancements [65]. Every dataset does not need to be normalized. Min-max normalization is only used when there are outliers. Data normalization has the benefit of making features easier to compare because every feature will have the same scale, according to [66]. So, whenever we apply features with different scales, the normalization step will be

used. A scale with the values [0, 1] is an example of a numerical property that has been normalized. The MinMaxScaler from the sklearn library must be imported and applied to our dataset to normalize our data. This can be done using the following formula [67]:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \dots\dots\dots\dots\dots\dots\dots\dots\dots (5)$$

First, we determine the dataset's minimum and maximum values, which are represented by the X_{min} and X_{max} respectively. After that, measure the dataset's range with the smallest value subtracted from the highest value, which is represented by the formula$range = X_{max} - X_{min}$. The next step is to subtract the variable's minimum value, $(X - X_{min})$, in order to calculate how much more data in the variable needs to be normalized from it. Finally, as indicated in equation 5, the method for determining the normalization of the variable x is derived by dividing $(X - X_{min})$ and $(X_{max} - X_{min})$.

4.5.3 Imbalance Data Handling Technique

In the analysis of medical records, data imbalance is a major problem. Clinical datasets frequently have unbalanced class labels [68]. The classical classifiers may be significantly affected by the data's imbalanced class arrangement. This is so that they can maximize accuracy overall without considering how each class is distributed relatively. When the categorization classes of a set of data are not nearly comparable, the dataset is said to be unbalanced. To address the imbalance issue, we use Tomek and SMOTE. The SMOTE approach is used to oversample the minority group, and once examples of under-sampling from the majority groups are found and eliminated from Tomek, a balanced distribution is ensured.

Oversampling:

Oversampling is utilized to enlarge the minority class through arbitrary duplication. Once the instances of the majority and minority classes are equal, this process is repeated (Figure 5).

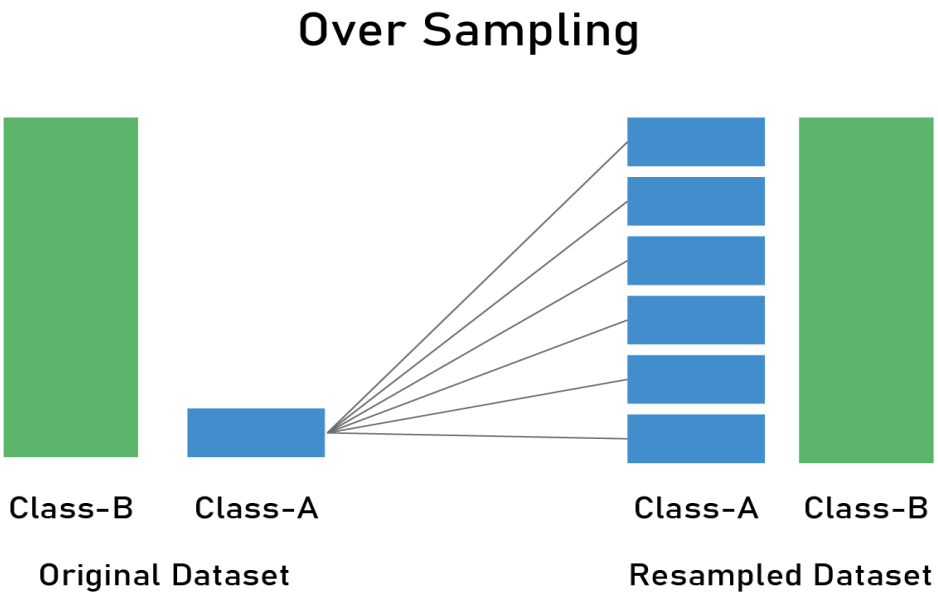

Figure 5. Imbalance Data handling technique (Oversampling)

Under-sampling:

The majority class is reduced by under-sampling, which involves randomly removing members of the majority class. This is carried out until the instances of the majority and minority classes are equal (Figure 6).

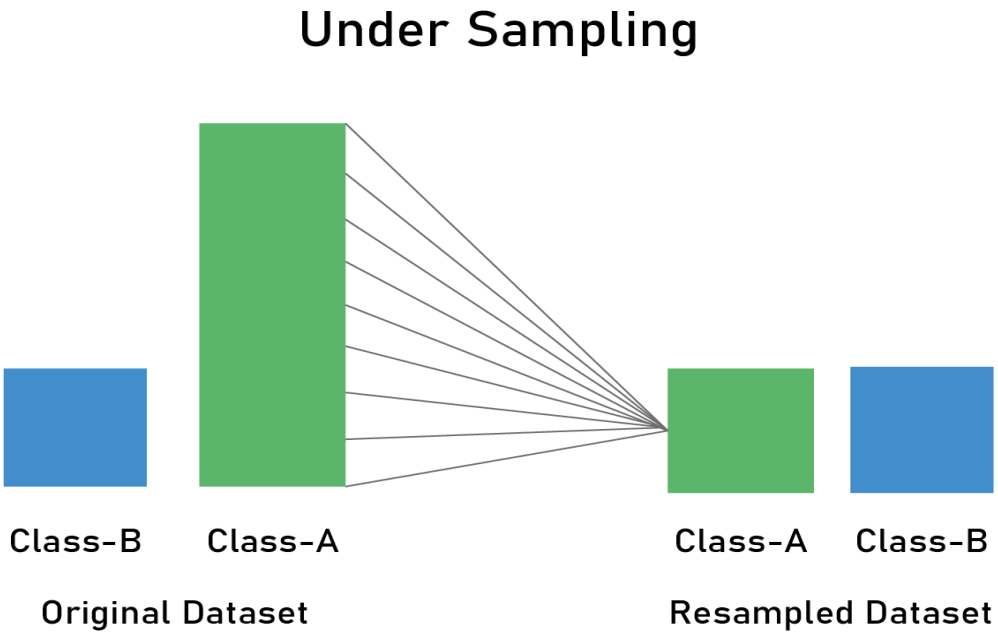

Figure 6. Imbalance Data handling technique (Under-sampling)

4.6 Description of the algorithm

The machine is educated to manage the data more effectively by using Machine Learning (ML). Sometimes, even after seeing the data, we are unable to figure out what information it contains. For a number of uses, such as analytics for prediction, processing of images, and statistical extraction, we use ML methods and their algorithms [69]. The major advantage of ML is that an algorithm may function independently once it learns how to apply data. Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), k-Nearest Neighbor (k-NN), and Stochastic Gradient Descent (SGD) are five ML methods that we use in this investigation. The following is a summary of each ML method:

4.6.1 Support Vector Machine (SVM)

The most popular Supervised Learning technique is Support Vector Machine (SVM), which is classified for the problems of Classification and Regression. Nevertheless, it is utilized primarily to solve categorization problems in the field of machine learning. The goal of the SVM [70] is to construct the optimal path or decision-making threshold. It is capable of classifying space with n dimensions, enabling us to swiftly classify a fresh data item in the appropriate region. In figure 7, the term "hyperplane" refers to the best-option area. It finds the most severe vectors and places to build a hyperplane. The data items or vectors that are nearest to the hyperplane and have the biggest impact on the position in which the hyperplane is situated are known as support vectors. As a result of these vectors supporting the so-called support vector hyperplane, the technique is known as a Support Vector Machine (SVM).

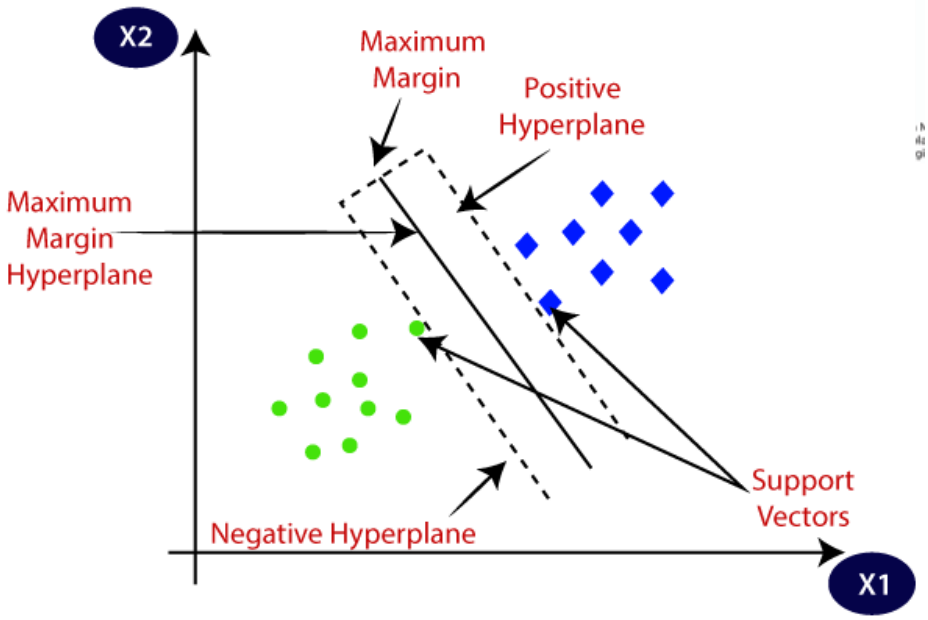

Figure 7. Support Vector Machine (SVM) with margin and hyperplane

The highest margin, or the greatest possible separation among the data items, is continuously employed to form a hyperplane. There are two different forms of SVM:

Linear SVM: When two distinct categories can be distinguished from an array of data employing only one direct path, it is called linearly distinct and is utilized in linear SVM. Afterward, such data are known as linearly separable data, and a classifier named the Linear SVM classifier is used.

Non-linear SVM: Non-Linear SVM is employed to analyze data that cannot be linearly separated. According to this definition, a database is considered non-linear when it is not classifiable using a direct path and a non-linear SVM classification algorithm is used to categorize it.

4.6.2 Random Forest (RF)

RF is a classification algorithm that keeps track of a variety of decision chains or trees on several subgroups of a specific dataset and uses the median to improve the dataset's productivity [71]. Instead of taking into account a single decision tree, RF predicts from every branch of the tree and projects the final outcomes depending on the number of ballots or choices. The factors are ranked in order of importance for the categorization problem. A greater number of trees in a forest improves accuracy, which reduces the harmful effects of excess fitting. An ensemble learning technique is the foundation of a method for merging multiple classifiers to deal with complex issues and increase the efficiency of the models. The most preferable number of trees in the natural environment minimizes overfitting problems and improves accuracy. Figure 8 of the following diagram shows how the RF algorithm [72] works:

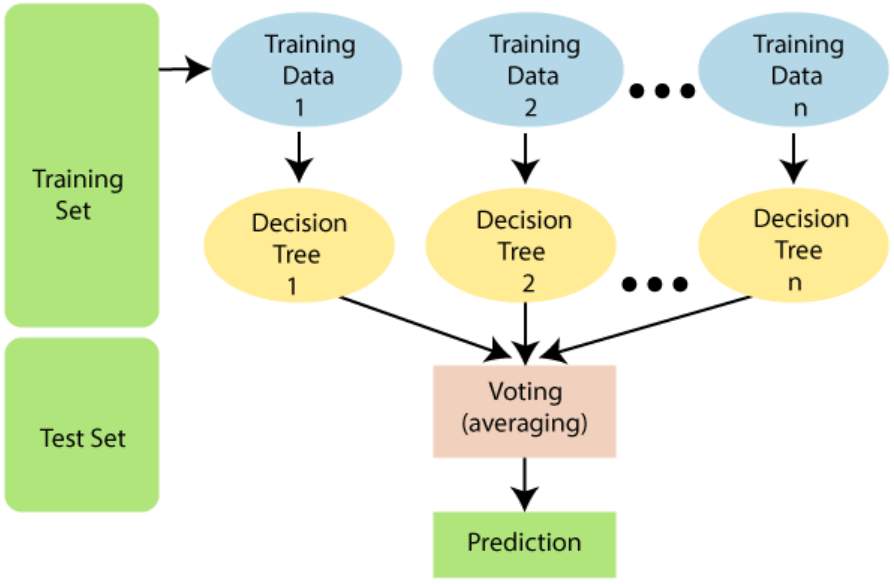

Figure 8. Working flow diagram of Random Forest (RF)

A two-phase process is used by an RF classifier. The Random Forest (RF) is created in the first phase by combining N decision trees. And the second page is given prediction accuracy by using these decision trees.

4.6.3 Logistic Regression (LR)

Observations are stored into discrete classes using the Logistic Regression (LR) method for classification. In general, there are many similarities between linear Regression and Logistic Regression (LR). While LR is used for applications requiring categorization, linear regression techniques are used to forecast values. The fitting of an S-shaped logistic function indicates two distinct values (0 or 1) rather than a regression curve in Logistic Regression (LR). The most effective variables used for classification can be basically completed using LR, which is frequently utilized to separate the observations using different types of data. The logistic regression technique [73] is shown in figure 9. Logistic Regression (LR) is calculated using the following formula [73]:

$$y = \frac{e^{(b_o + b_1 X)}}{1 + e^{(b_o + b_1 X)}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (6)$$

Where,

x = the input value

y = the predicted output

b_o = bias or intercept term

b_1 = coefficient for input (x)

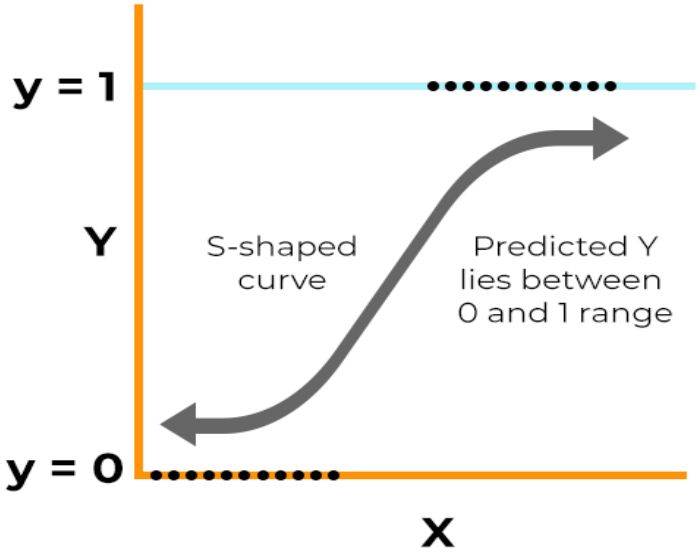

Figure 9. Logistic Regression (LR) technique

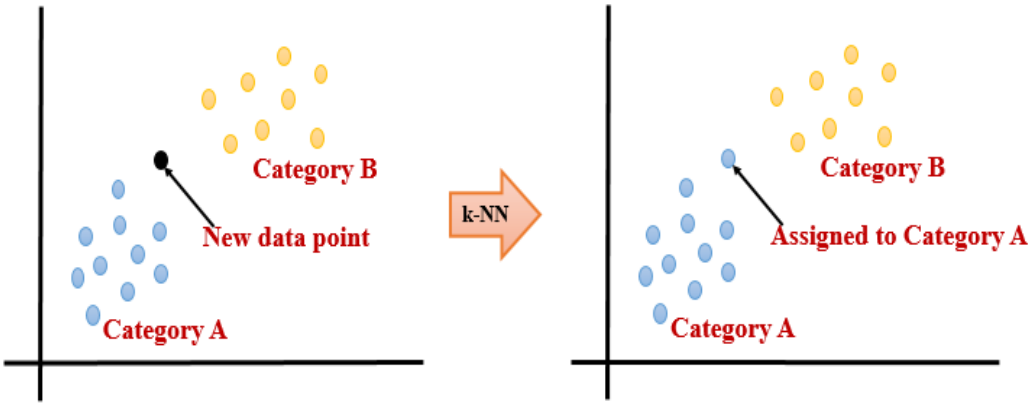

Figure 10. The machine learning technique k-Nearest Neighbor (k-NN)

4.6.4 k-Nearest Neighbors (k-NN)

A classification technique known as the k-Nearest Neighbors (k-NNs) is the most significant categorization strategy in machine learning. It is a part of the supervised learning field and performs several tasks such as identification of patterns, manufacturing, and detection of attacks [74]. All applicants are assigned a location in the vector space of the predictor, which is described by the k-NN method. The proportion of favorable hazards among the k-closest values in the teaching set is another way it evaluates future possibility, [75]. Figure 10 displays the k-NN algorithm. The following algorithmic stages are used by machine learning's most popular classification method, k-NN [75]:

- Select the “K” values or the number of neighbors that will be used to forecast the resulting class.
- Take the k-number of the data item that is nearest to the new one and is the furthest away after measuring the distance between it and the current data points.
- Calculate the category's statistics after collecting the "K" closest ones.
- Assign the maximum number of categories per class to the new data point “N”.

4.6.5 Stochastic Gradient Descent (SGD)

A relatively efficient approach for fitting symmetric cost functions using linear classification and regression analyses is Stochastic Gradient Descent (SGD) [76]. The number of hyper-parameters used in this method. SGD [74] is a successful approach for logistic regression and linear classifiers that uses both initial work and selective learning. Its key benefits is that it is extremely effective and makes it simple to build these algorithms. The drawback is that SGD calculations need a variety of hyper-parameters including regularization

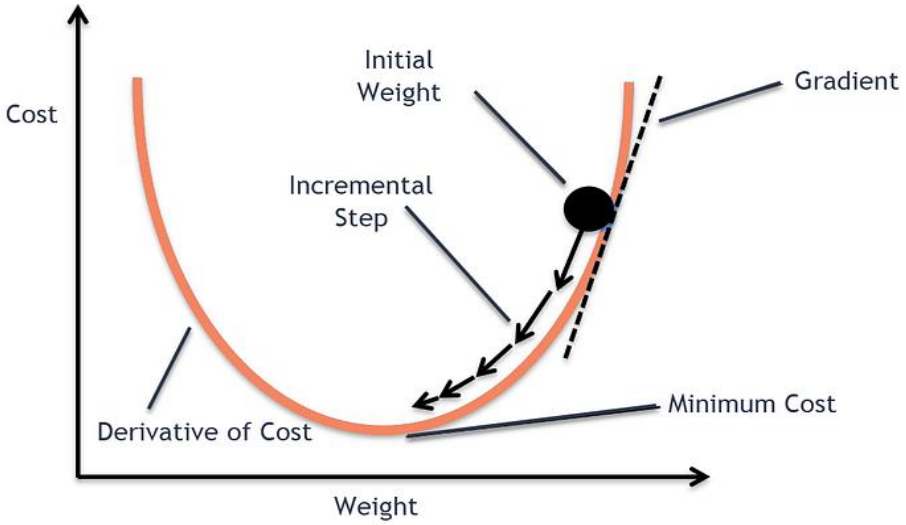

Figure 11. Stochastic Gradient Descent (SGD) algorithm

and different cycles. In SGD [77], if there is a huge dataset, then it will just take a random sample from that and calculate appropriate weights for them, and then those calculated weights are then used for rest of the data as well. It is very helpful to reduce the consumption of machine resources, and leads to quick result. Figure 11 shows the explanation of the Stochastic Gradient Descent (SGD) algorithm [77].

4.6.6 Stacking Random Forest, Support Vector Machine, and Logistic Regression (RSL)

A strategy collectively referred to as an "ensemble approach" [78] combines the best elements of various ML techniques into an effective individual model. In general, the model performs better than the learners on a personal basis. To create the new model known as Ensemble RSL, we employed the stacking ensemble process. As a basic model, the Support Vector Machine (SVM) is combined with the Random Forest (RF) and Logistic Regression (LR) algorithms to create the ensemble RSL. With the aid of the three algorithms, it increases categorization reports' accuracy and precision.

4.7 Performance measure techniques

This study used six evaluation methodologies, [79], including accuracy, recall, and precision, f-1 Score, Cohen Kappa, and ROC, to evaluate the effectiveness of Machine Learning (ML) models. Machine Learning (ML) models use TP, FP, TN, and FN [80] as outcome measurement strategies to obtain the correct results. The following diagram [76], which is illustrated in figure 12, represents the structure of an array of confusion in which the parameters are provided.

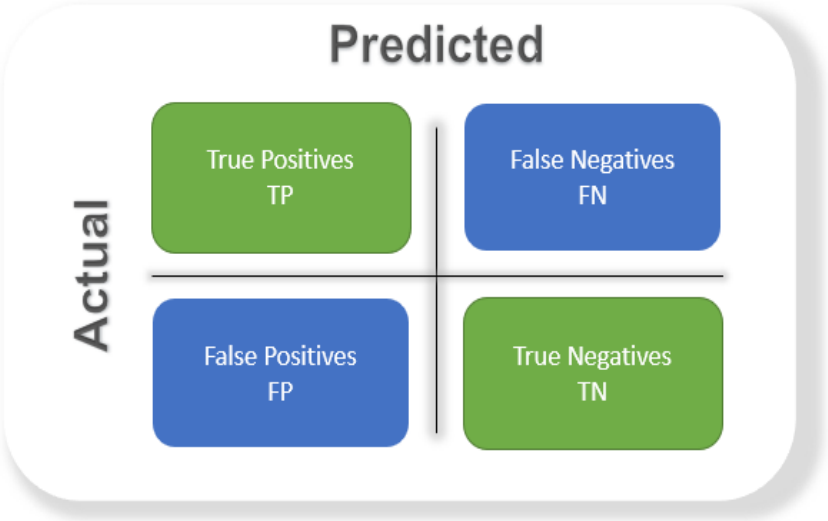

Figure 12. An array of confusion

The four potential outcomes [80] that are outlined in the following are the basis for all the performance measurements that are described:

True Positive (TP): The quantity of occurrences that were positively identified as having chronic kidney disease.

False Positive (FP): How many cases were misdiagnosed as having chronic kidney disease.

True Negative (TN): The exact count of cases where chronic kidney disease was diagnosed.

False Negative (FN): There were just how many cases of chronic kidney disease.

4.7.1 Accuracy

Accuracy is defined as the percentage of exactly classified data items compared to the average number of data items [81]. Even though accuracy is one of the most widely used performance metrics, it occasionally yields inaccurate results, particularly for datasets with unbalances. Mathematically,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (7)$$

4.7.2 Precision

The number of TP is divided by the total of the TP and FP to determine the level of precision [80] for binary categorization. Whether the objective is to reduce FP, precision executes accurately on imbalance data. Even if the FP ratio is large, it is still a valuable statistic. Mathematically, $\text{Precision (P)} = \frac{\text{TP}}{\text{TP+FP}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (8)$

4.7.3 Recall

Sensitivity or True Positive Rate (TPR) are other terms for recall [81]. It is often ascertained by splitting the total amount of TP by the sum of TP and FN. It is also suitable in the situation of removing FN from the unbalanced dataset. Mathematically,

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

4.7.4 f1-score

The f1-score is referred to as the harmonic mean of Precision and Recall [82]. Only accuracy is insufficient to decide whether the model is appropriate. A logical model will only exist when Precision and Recall are high. For this reason, the performance of two classifiers is compared using the f1-score. The more f1-score points there are, the more logical a model emerges. Mathematically,

$$\text{f1} - \text{score} = \frac{2\text{PR}}{\text{P+R}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

4.7.5 Cohen Kappa

A dataset could contain multiple classifications or be out of balance at any given time. Metrics like precision, accuracy, or recall are ineffective under these circumstances. An excellent metric for handling unbalanced and multi-class situations is Cohen's kappa statistic [83]. Mathematically,

$$\text{Cohen Kappa} = \frac{\text{Po} - \text{Pe}}{1 - \text{Pe}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (11)$$

Where Pe stands for the predicted agreement and Po for an observed agreement.

4.7.6 Receiver Operating Characteristics (ROC)

The industry norm for comparing and discussing the correctness of diagnostic procedures is now known as ROC curves. Clinical chemists utilize these curves quite frequently [84]. The ROC curve is used to visualize, organize, and select the classifiers considering the performance. The True Positive Rate (TPR) and False Positive Rate (FPR) are plotted on a probability curve at various threshold levels, with the True Positive Rate (TPR) on the Y-axis and the False Positive Rate (FPR) on the X-axis. However, using the following formulas [85], which are provided below, we may get the True Positive Rate (TPR) and False Positive Rate (FPR).

$$\text{TPR/Recall/Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (12)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (13)$$

$$\text{FPR} = 1 - \text{Specificity} = \frac{\text{FP}}{\text{TN} + \text{FP}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (14)$$

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Introduction

An important component of this study is the results and discussion chapter which focuses on presenting and debating the findings of the experiments conducted for the research. This chapter summarizes the outcomes of the suggested materials and methods for predicting kidney disease, together with how well they performed in terms of evaluation metrics and prediction values. To determine which of the suggested approaches or algorithms works best, the findings are evaluated against current best practices. This chapter aids in reaching findings and offering suggestions, advancing the study of kidney disease prognosis.

5.2 Result analysis on Baseline Model

The result includes an investigation of several classification methods of the suggested model, which is a chronic kidney disease prediction system. The investigation concentrated on how well the model identified kidney illness. Utilizing several classification methods (such as SVM, RF, LR, SGD, and k-NN) and efficacy metrics like accuracy, recall, and precision, f1-score, and Cohen kappa, the models were trained with 400 attributes or data and examined to predict chronic kidney disease. Identifying the chronic kidney diseases with high values for accuracy, f1-score, precision, recall, Cohen Kappa, sensitivity, and ROC score, then the result showed that the suggested model accurately performed. The results of several performance evaluation techniques on balanced data are shown in table 6.

Table 6. The performance results of different models using specific datasets.

Method	Accuracy	Precision	Recall	f1-score	Cohen Kappa
SVM	0.98	0.98	0.98	0.98	0.96
RF	0.98	0.99	0.98	0.98	0.97
LR	0.97	0.97	0.97	0.97	0.94
k-NN	0.98	0.98	0.98	0.98	0.97
SGD	0.97	0.97	0.97	0.97	0.95
Proposed Ensemble RSL	0.99	0.99	0.99	0.99	0.98

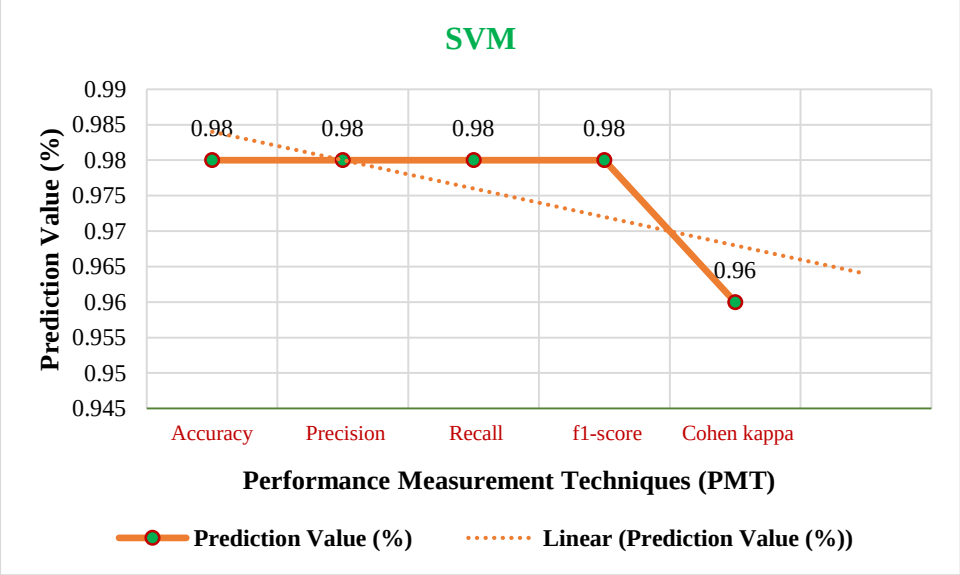

Figure 13. Analysis of the CKD prediction results using the SVM model

5.2.1 SVM

Figure 13 shows the Support Vector Machine (SVM) Classification methods and its result in a line plot of prediction value versus performance measurement techniques (PMT), including accuracy, precision, recall, f1-score, and Cohen Kappa. The x-axis represents the performance measurement techniques (PMT), which help to measure the outcome of the entire method. The y-axis represents the prediction accuracy of this model. From the figure, the SVM model consists of a plane line and dotted line. With respect to the dotted line the plane line represents the values of performance techniques (98% accuracy, 98% precision, 98% recall, 98% f1-score, and 96% Cohen kappa) that emphasize the SVM model's performance to predict accurately chronic kidney diseases. So, the SVM model correctly predicts the disease of CKD by about 98%.

5.2.2 RF

In figure 14, the Random Forest (RF) classifier is depicted with a line plot of prediction value versus performance measurement techniques (PMT). The x-axis represents the performance measurement techniques (PMT), which create a decision for the entire model to give the proper outcomes. The y-axis represents the model prediction value. In the figure, the RF model consists of a plane line and dotted line. The dotted line is the reference line by predicting CKD of this model. The plane line represents the values of performance techniques (98% accuracy, 99% precision, 98% recall, 98% f1-score, and 97% Cohen kappa) at a time when they increase and decrease. So, the RF model predicts chronic kidney disease about 98% accurately, and its highest precision value is 99%.

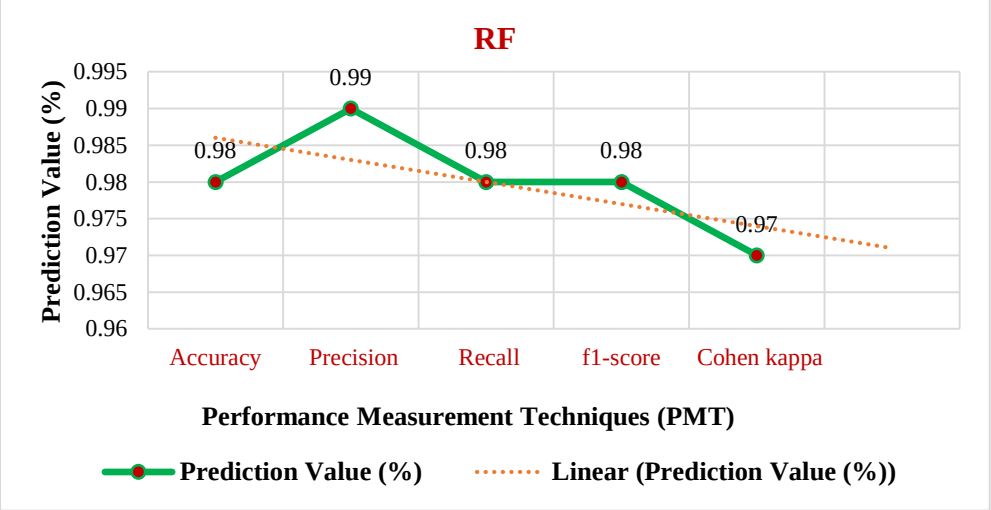

Figure 14. Analysis of the results for the RF model's CKD prediction

Similarly, figures 15, 16, 17, and 18 all illustrate a line plot of prediction value against performance measurement techniques (PMTs), including accuracy, precision, recall, fi-score, and Cohen Kappa.

5.2.3 LR

Figure 15 represents the result for accurate CKD prediction using this Logistic Regression (LR) model. Showing this figure, we said that the LR model predicted chronic kidney disease about 97% accurately, which is represented by a plane line. Also the dotted line represent the reference line by predicting CKD in this model.

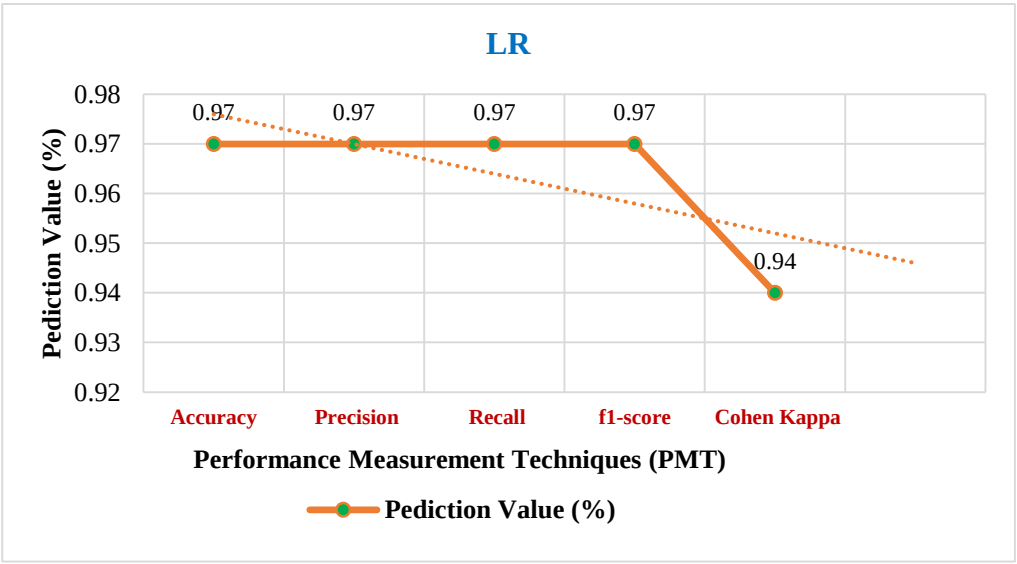

Figure 15. Results analysis for CKD prediction using the LR model.

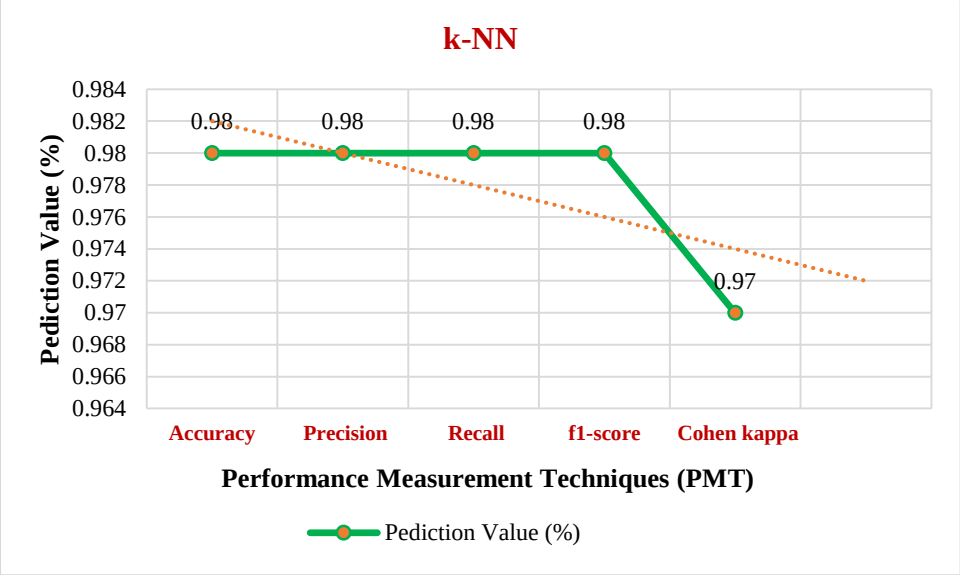

Figure 16. Analysis of the k-NN model's predictions for CKD

5.2.4 k-NN

Here, the results are shown in figure 16 for accurate forecasting CKD by using this k-Nearest Neighbors (K-NN) model. With respect to the dotted line we can see that the k-NN prediction model predicted chronic kidney disease 98% correctly, which is shown by a plane line. But its Cohen Kappa value is 97%.

5.2.5 SGD

The Stochastic Gradient Descent (SGD) model, which also plots some prediction values from the training attributes and gets a plane line and dotted line is shown in figure 17. The dotted line is the reference line for this model. This model correctly identified the disease of kidney about 97% of the time, but with a 95% Cohen Kappa score.

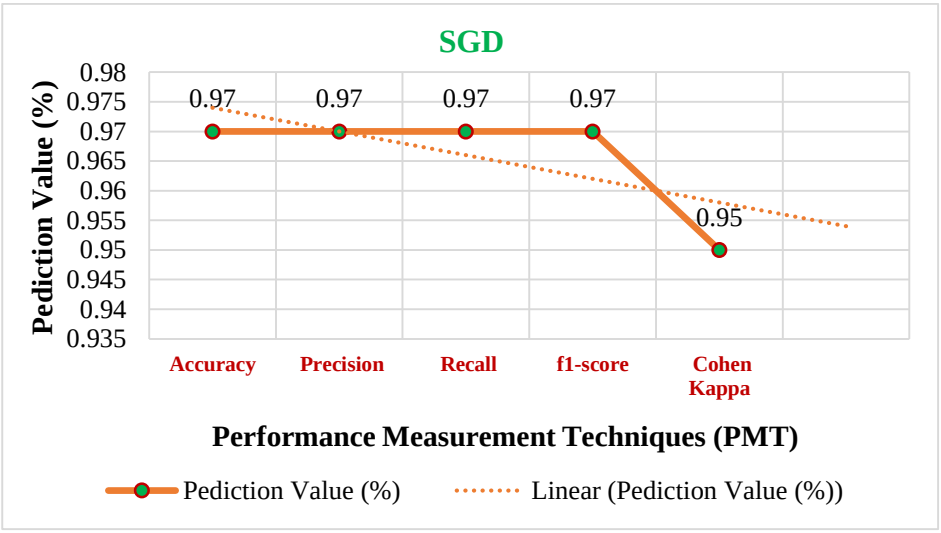

Figure 17. Analysis of the results for SGD-based CKD prediction

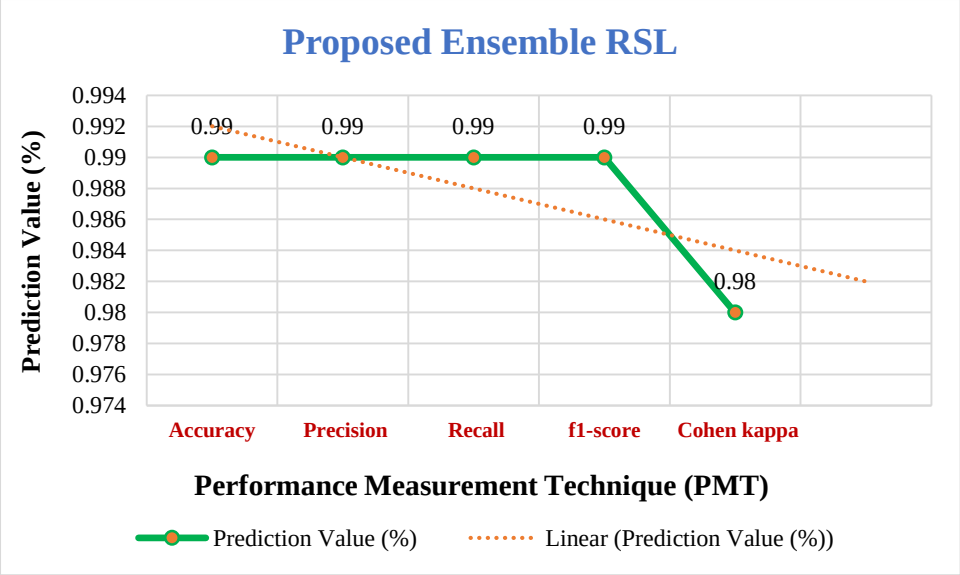

Figure 18. Results analysis for CKD prediction using the Ensemble RSL model.

5.3 Proposed Ensemble RSL

Figure 18 displays the Ensemble RSL model, whose x-axis represents the performance measurement techniques (like accuracy, recall, etc.) and y-axis represents the prediction values for properly detecting the CKD. In this figure, the results depict the plane line with respect to the dotted line, and it shows a value of 99% with respect to accuracy, precision recall, and f1-score, but Cohen kappa is about 98%. That's why the proposed Ensemble RSL is better than the other models because its prediction accuracy is high at 99%. We have also told that the dotted line represents the reference line for predicting CKD in this model.

5.4 Comparison and Results Analysis among Baseline & Proposed Ensemble RSL models

Most classification algorithms perform well to classifying the kidney disease by using balanced data. Based on several performance evaluation metrics, the results of all the chosen methods are depicted in Table 6.

According to the dataset and figure 19, SVM, k-NN, and RF performed at 98%, but LR and SGD at 97%. On the other hand, the proposed ensemble RSL performed at 99% on the balanced data.

In addition to accuracy, our suggested technique outperforms previous benchmark algorithms in terms of f1 score, precision, and recall. SVM and k-NN display precision of 98%, LR and SGD display precision of 97%, and RF and ensemble RSL display precision of 99%.

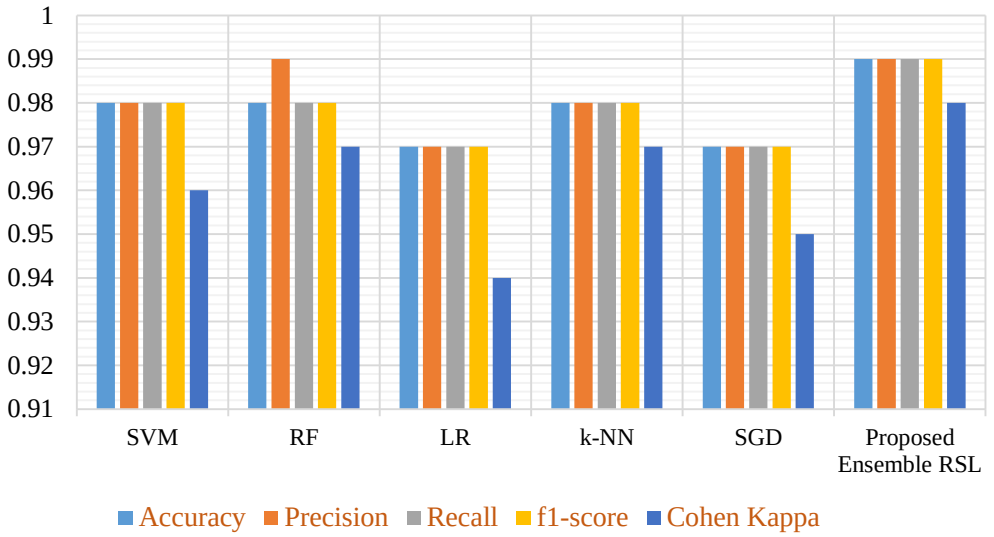

Figure 19. Accuracy and classification matrices report on different models

In figure 20, the five baseline models and the proposed ensemble RSL model depict the plot on the x-axis with Performance Measurement Techniques (PMT) and the y-axis with the prediction attributes. Showing this figure, we can understand that the proposed Ensemble RSL model is superior to baseline models because its prediction accuracy is better, which is 99%. The plane lines in these figures represent the prediction line with accurate results that is used to accurately forecast Chronic Kidney Disease (CKD). And all the dotted line represents the reference line for predicting CKD among these models. Among these lines, we told that the proposed Ensemble line is more powerful to detect kidney disease than baseline models.

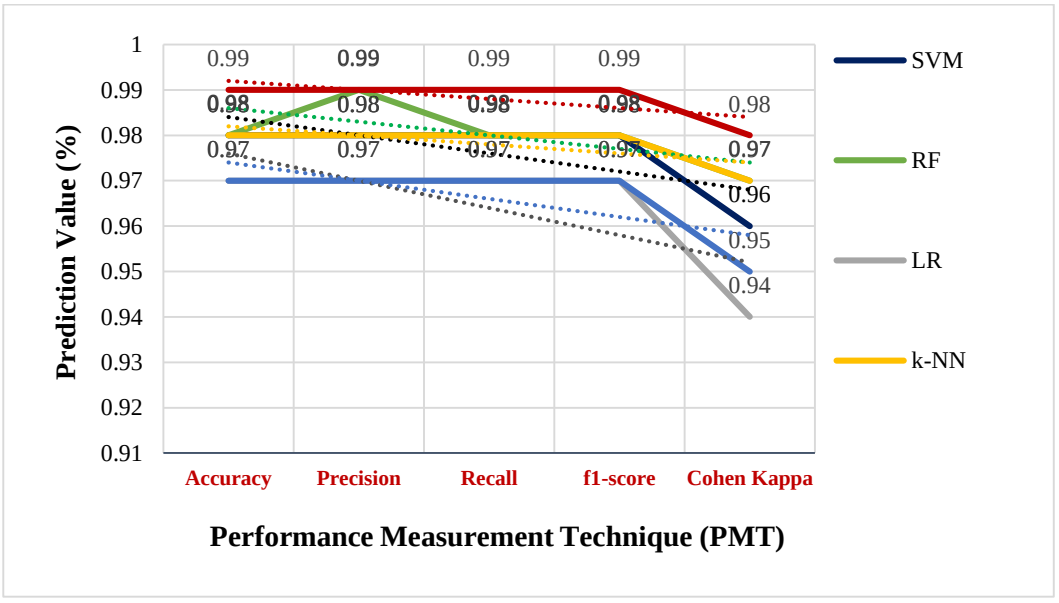

Figure 20. Comparison on different models with accuracy and classification metrics

5.5 ROC Curves among Baseline & Proposed Ensemble RSL models

In figure 21, the total result is displayed utilizing a ROC curve along with five baseline and one ensemble models. Here, the False Positive Rate (FPR) is plotted on the x-axis, and the True Positive Rate (TPR) is also plotted on the y-axis. In the respected figure, describe all these baseline models, and the proposed ensemble RSL model together, and all are represented by the dotted line with various colors. The blue line is the reference and we have plotted difference threshold or cutoff values then we get curve of different models like SVM, RF, LR, k-NN, SGD and proposed ensemble RSL. The area of the proposed ensemble RSL is greater than the area of other baseline models. According to these curves the proposed ensemble RSL model performed well with 99% accurately classified CKD. The curve is not shown properly due to the crossover of the accuracy.

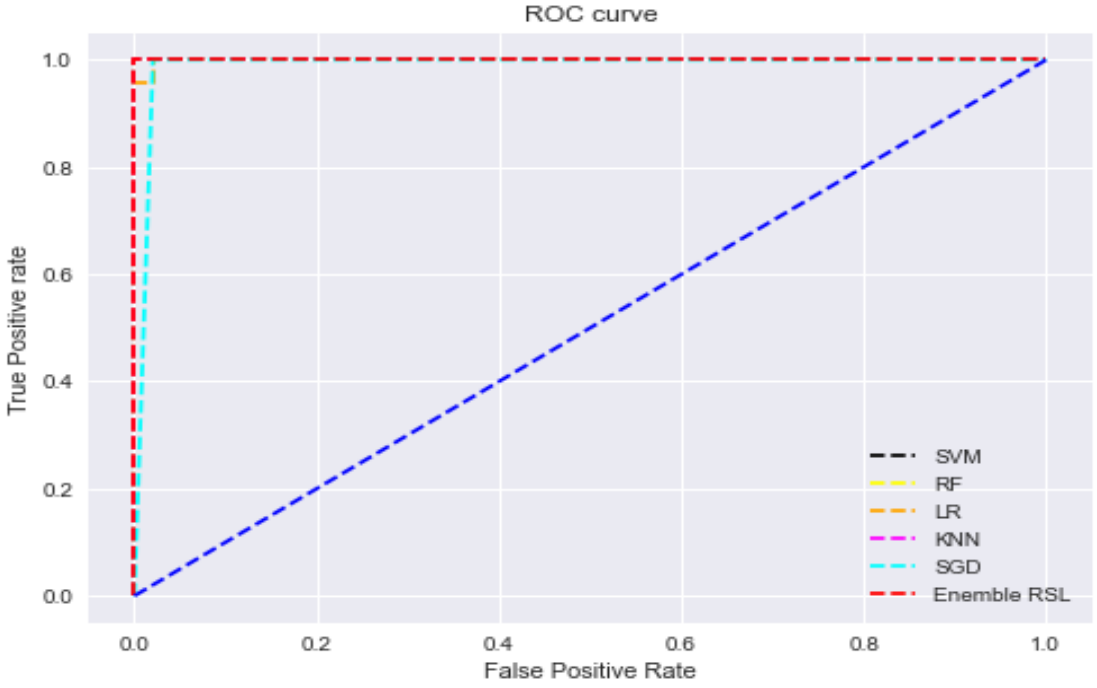

Figure 21. ROC curve of diverse classifiers employing SMOTETomek

5.6 Discussion

All classifiers' parameters are adjusted for the best classification performance, and the results from all methods are positive. By analyzing the baseline results, we have seen that the performance of SVM, k-NN, and RF is 98%, but that of LR and SGD is 97% respectively. On the other hand, RF demonstrated 99% precision, SVM and k-NN demonstrated 98% precision, and LR and SGD demonstrated 97% precision. However, we have chosen the best three baseline algorithms from the evaluation results, which are RF, SVM, and LR, and combined these to create ensemble RSL. Then, the results were compared among the baseline models and the proposed ensemble RSL models. The final result demonstrated that our suggested ensemble RSL model has a maximum value of 99% for forecasting Chronic Kidney Disease (CKD).

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Introduction

Conclusion and future work are the two distinct sections of this chapter. The work that was performed in this study is summarized in this chapter. The outcome and future direction of the work related to our study are also discussed here.

6.2 Conclusion

The kidney is one of the major organs. Any damage or disease to the kidney can be fatal to the human body. Most people are impacted by their irregular lifestyle. Chronic Kidney Disease (CKD) damages the internal parts of our body by causing waste blood, and toxic fluids. In this study, we provided appropriate methods and findings for this disease's precise and early identification to save people from this life-threatening condition. To obtain balanced data from imbalanced data, we have also developed a pipeline that includes outlier removal, data standardization, and imbalance handling approaches. Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), k-Nearest Neighbors (k-NN), and Stochastic Gradient Descent (SGD) were also used in this thesis. The classification algorithm known as an SVM is capable of classification, regression, and outlier detection. A dataset is increased by RF that maintains a variety of decision trees. Multi-class, non-linear, and overfitting issues resolve the LR. The k-NN classification method is widely used to classify several applications. The reduction coefficient value of a cost function is determined through SGD. The balanced data is divided into two subsets (on a ratio of 80:20), which are the training and testing datasets. Using the training data, the models are then trained, and the projected values are produced using the test data. To obtain expected data, these ML techniques are applied to training and testing datasets, and which model correctly predicts kidney disease is also shown. In addition, we have analyzed these results by using performance measurement techniques for predicting Chronic Kidney Disease (CKD). We therefore evaluated the baseline results and found that the performance of SVM, k-NN, and RF was 98%, but 97% for LR and SGD, respectively. On the other hand, RF demonstrated 99% precision, SVM and k-NN demonstrated 98% precision, and LR and SGD demonstrated 97% precision. All classifiers' variables were improved for the greatest classification performance, and the results from all methods are positive. Then we considered RF, SVM, and LR but not k-NN because k-NN runs slowly in real time and needs to track every training statistic in addition to finding nearby nodes, whereas LR outputs quickly. We have also seen that RF, SVM, and LR algorithms give better performance for predicting chronic kidney diseases than other baseline algorithms, and that’s why RF, SVM, and LR machine learning algorithms are combined in this paper. By evaluating these outcomes, we

conclude that the benchmark ML (SVM, RF, LR, k-NN, and SGD) algorithms are less accurate than the proposed ensemble RSL model for predicting chronic kidney illness. We achieved the highest prediction accuracy of 99% with our chosen model. In terms of f1 score, precision, recall, and Cohen Kappa score, we agreed that the suggested RSL was particularly better than the competing classifiers. Ultimately, we can state that the medical domain analyzer using our proposed ensemble RSL can predict chronic kidney disease more accurately than previous approaches.

6.3 Future work

In the future, we will try to work with more than one dataset and compare the performances. A significant use of deep learning techniques is in the field of health care. The result might be enhanced using deep learning techniques. To obtain more precise findings, dimension reduction and feature selection may be helpful on this subject. We will create our own kidney disease dataset in the perspective of Bangladesh and implement the proposed methodology and evaluate how well the proposed methodology work. It can also be classified as a multi-category problem to assess the severity of the disease.

REFERENCES

[1]. Devika, R., Sai Vaishnavi Avilala, and V. Subramaniyaswamy, "Comparative study of classifier for chronic kidney disease prediction using naive Bayes, KNN, and random forest," 3rd International conference on computing methodologies and communication (ICCMC), IEEE, 2019.

[2]. Kumar, Narander, and Sabita Khatri, "Implementing WEKA for medical data classification and early disease prediction," In 2017 3rd international conference on computational intelligence & communication technology (CICT), pp. 1-6, IEEE, 2017.

[3]. Burrell, Jenna, "How the machine 'thinks': Understanding opacity in machine learning algorithms," Big data & society 3, no. 1, 2053951715622512, 2016.

[4]. Burrell, Jenna, "Invisible users: Youth in the Internet cafés of urban Ghana," MIT Press, 2012.

[5]. Khosravi, Khabat, Himan Shahabi, Binh Thai Pham, Jan Adamowski, Ataollah Shirzadi, Biswajeet Pradhan, Jie Dou, "A comparative assessment of flood susceptibility modeling using multi-criteria decision-making analysis and machine learning methods," Journal of Hydrology 573, 311-323, 2019.

[6]. Ayodele, Taiwo Oladipupo, "Types of machine learning algorithms," New advances in machine learning 3, 19-48, 2010.

[7]. Huang, Jiale, Tao Jin, Menglin Liang, and Houlei Chen, "Prediction of heat exchanger performance in cryogenic oscillating flow conditions by support vector machine," Applied Thermal Engineering 182, 116053, 2021.

[8]. Zhou, Ying, Zhixiang Yu, Limin Liu, Lei Wei, Lijuan Zhao, Liuyifei Huang, Liya Wang, and Shiren Sun, "Construction and evaluation of an integrated predictive model for chronic kidney disease based on the random forest and artificial neural network approaches," Biochemical and Biophysical Research Communications 603, 21-28, 2022.

[9]. Lasko, Thomas A., Jui G. Bhagwat, Kelly H. Zou, and Lucila Ohno-Machado, "The use of receiver operating characteristic curves in biomedical informatics," Journal of biomedical informatics 38, no. 5, 404-415, 2005.

[10]. Gonçalves, Luzia, Ana Subtil, M. Rosário Oliveira, and Patricia de Zea Bermudez, "ROC curve estimation: An overview," REVSTAT-Statistical journal 12, no. 1, 1-20, 2014.

[11]. Rahman, Tanjina, “Nutrition and Health Status of Hemodialysis Patients in Dhaka, Bangladesh,” Wayne State University, 2020.

[12]. Matsushita, Kunihiro, Simerjot K. Jassal, Yingying Sang, Shoshana H. Ballew, Morgan E. Grams, Aditya Surapaneni, Johan Arnlov, "Incorporating kidney disease measures into cardiovascular risk prediction: development and validation in 9 million adults from 72 datasets," EClinicalMedicine 27, 2020.

[13]. Song, Xuan, Xinyan Liu, Fei Liu, and Chunting Wang, "Comparison of machine learning and logistic regression models in predicting acute kidney injury: A systematic review and meta-analysis," International journal of medical informatics 151: 104484, 2021.

[14]. Balusamy, Balamurugan, "A novel algorithm for prediction of chronic kidney risks using machine learning schemes," Materials Today: Proceedings, 2021.

[15]. Shanthakumari, A. S., and R. Jayakarthik, "Utilizing support vector machines for predictive analytics in chronic kidney diseases," Materials Today: Proceedings, 2021.

[16]. Ventrella, Piervincenzo, Giovanni Delgrossi, Gianmichele Ferrario, Marco Righetti, and Marco Masseroli, "Supervised machine learning for the assessment of chronic kidney disease advancement," Computer Methods and Programs in Biomedicine 209, 106329, 2021.

[17]. Bhaskar, N., and M. Suchetha, "A computationally efficient correlational neural network for automated prediction of chronic kidney disease," IRBM 42.4, 268-276, 2021.

[18]. Almustafa, Khaled Mohamad, "Prediction of chronic kidney disease using different classification algorithms," Informatics in Medicine Unlocked 24, 100631, 2021.

[19]. Islam, Md Ariful, Md Ziaul Hasan Majumder, and Md Alomgeer Hussein, "Chronic kidney disease prediction based on machine learning algorithms," Journal of Pathology Informatics, 100189, 2023.

[20]. Dekka, Mr. Satish, "Utilizing Machine Learning Algorithms for Kidney Disease Prognosis," European Journal of Molecular & Clinical Medicine, v. 10, no. 1, 2023.

[21]. Kanda, Eiichiro, Bogdan Iuliu Epureanu, Taiji Adachi, and Naoki Kashihara, "Machine-learning-based Web system for the prediction of chronic kidney disease progression and mortality," PLOS Digital Health 2, no. 1, e0000188, 2023.

[22]. Himi, Shinthi Tasnim, Natasha Tanzila Monalisa, M. D. Whaiduzzaman, Alistair Barros, and Mohammad Shorif Uddin, "MedAi: A Smartwatch-Based Application Framework for the Prediction of Common Diseases Using Machine Learning," IEEE Access 11, 12342-12359, 2023.

[23]. Debal, Dibaba Adeba, and Tilahun Melak Sitote, "Chronic kidney disease prediction using machine learning techniques," Journal of Big Data 9.1, 1-19, 2022.

[24]. Pal and Saurabh, "Chronic Kidney Disease Prediction Using Machine Learning Techniques," Biomedical Materials & Devices, 1-7, 2022.

[25]. Dritsas, Elias, and Maria Trigka, "Machine learning techniques for chronic kidney disease risk prediction," Big Data and Cognitive Computing 6.3, 98, 2022.

[26]. Singh, Vijendra, Vijayan K. Asari, and Rajkumar Rajasekaran, "A deep neural network for early detection and prediction of chronic kidney disease," Diagnostics 12.1, 116, 2022.

[27]. R. J. Kate, Ruth M. Perez, Debesh Mazumdar, Kalyan S. Pasupathy, and Vani Nilakantan, "Prediction and detection models for acute kidney injury in hospitalized older adults," BMC medical informatics and decision making 16, 1-11, 2016.

[28]. Davis, Sharon E., Thomas A. Lasko, Guanhua Chen, Edward D. Siew, and Michael E. Matheny, "Calibration drift in regression and machine learning models for acute kidney injury," Journal of the American Medical Informatics Association 24, no. 6, 1052-1061, 2017.

[29]. Cheng, Peng, Lemuel R. Waitman, Yong Hu, and Mei Liu, "Predicting inpatient acute kidney injury over different time horizons: how early and accurate," In AMIA Annual Symposium Proceedings, American Medical Informatics Association, vol. 17, p. 565, 2017.

[30]. Koyner, Jay L., Kyle A. Carey, Dana P. Edelson, and Matthew M. Churpek, "The development of a machine learning inpatient acute kidney injury prediction model," Critical care medicine 46, no. 7, 1070-1077, 2018.

[31]. Lee, Hyung-Chul, Soo Bin Yoon, Seong-Mi Yang, Won Ho Kim, Ho-Geol Ryu, Chul-Woo Jung, Kyung-Suk Suh, and Kook Hyun Lee, "Prediction of acute kidney injury after liver transplantation: machine learning approaches vs. logistic regression model," Journal of clinical medicine 7, no. 11, 428, 2018.

[32]. Park, Namyong, Eunjeong Kang, Minsu Park, Hajeong Lee, Hee-Gyung Kang, Hyung-Jin Yoon, and U. Kang, "Predicting acute kidney injury in cancer patients using heterogeneous and irregular data," PloS one 13, no. 7, e0199839, 2018.

[33]. Adhikari, Lasith, Tezcan Ozrazgat-Baslanti, Matthew Ruppert, R. W. M. A. Madushani, Srajan Paliwal, Haleh Hashemighouchani, Feng Zheng, "Improved predictive models for acute kidney injury with IDEA: Intraoperative Data Embedded Analytics," PloS one 14, no. 4, e0214904, 2019.

[34]. Chiofolo, Caitlyn, Nicolas Chbat, Erina Ghosh, Larry Eshelman, and Kianoush Kashani, "Automated continuous acute kidney injury prediction and surveillance: a random forest model," In Mayo Clinic Proceedings, vol. 94, no. 5, pp. 783-792. Elsevier, 2019.

[35]. M. Flechet, S. Falini, C. Bonetti, F. Guiza, M. Schetz, G. Van den Berghe, G. Meyfroidt, “Machine learning versus physician’s prediction of acute kidney injury in critically ill adults: a prospective evaluation of the AKI predictor,” Crit. Care 23, 282, 2019.

[36]. He, Jianqin, Yong Hu, Xiangzhou Zhang, Lijuan Wu, Lemuel R. Waitman, and Mei Liu, "Multi-perspective predictive modeling for acute kidney injury in general hospital populations using electronic medical records," JAMIA open 2, no. 1, 115-122, 2019.

[37]. J. Parreco, H. Soe-Lin, J.J. Parks, S. Byerly, M. Chatoor, J.L. Buicko, N. Namias, R. Rattan, “Comparing machine learning algorithms for predicting acute kidney injury,” Am. Surg. 85, 725–729, 2019.

[38]. M. Sun, J. Baron, A. Dighe, P. Szolovits, R.G. Wunderink, T. Isakova, Y. Luo, “Early prediction of acute kidney injury in critical care setting using clinical notes and structured multivariate physiological measurements,” Stud. Health Technol. Inform. 264, 368–372, 2019.

[39]. Cronin, Robert M., Jacob P. VanHouten, Edward D. Siew, Svetlana K. Eden, Stephan D. Fihn, Christopher D. Nielson, Josh F. Peterson, "National Veterans Health Administration inpatient risk stratification models for hospital-acquired acute kidney injury," Journal of the American Medical Informatics Association 22, no. 5, 1054-1071, 2015.

[40]. Thottakkara, Paul, Tezcan Ozrazgat-Baslanti, Bradley B. Hupf, Parisa Rashidi, Panos Pardalos, Petar Momcilovic, and Azra Bihorac, "Application of machine learning techniques to high-dimensional clinical data to forecast postoperative complications," PloS one 11, no. 5, e0155705, 2016.

[41]. N.K. Tran, S. Sen, T.L. Palmieri, K. Lima, S. Falwell, J. Wajda, H.H. Rashidi, “Artificial intelligence and machine learning for predicting acute kidney injury in severely burned patients: a proof of concept,” Burns 45, 1350–1358, 2019.

[42]. Z. Zhang, K.M. Ho, Y. Hong, “Machine learning for the prediction of volume responsiveness in patients with oliguric acute kidney injury in critical care,” Crit. Care 23, 112, 2019.

[43]. Ghosh, Pronab, FM Javed Mehedi Shamrat, Shahana Shultana, Saima Afrin, Atqiya Abida Anjum, and Aliza Ahmed Khan, "Optimization of prediction method of chronic kidney disease using machine learning algorithm," In 2020 15th international joint symposium on artificial intelligence and natural language processing (iSAI-NLP), pp. 1-6, IEEE, 2020.

[44]. Gudeti, Bhavya, Shashvi Mishra, Shaveta Malik, Terrance Frederick Fernandez, Amit Kumar Tyagi, and Shabnam Kumari, "A novel approach to predict chronic kidney disease using machine learning algorithms," In 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1630-1635, IEEE, 2020.

[45]. Devika, R., Sai Vaishnavi Avilala, and V. Subramaniyaswamy, "Comparative study of classifier for chronic kidney disease prediction using naive Bayes, KNN, and random forest," 2019 3rd International conference on computing methodologies and communication (ICCMC), IEEE, 2019.

[46]. Baidya, Deepanita, Umme Umaima, Md Nazrul Islam, FM Javed Mehedi Shamrat, Anik Pramanik, and Md Sadekur Rahman, "A deep prediction of chronic kidney disease by employing machine learning method," In 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 1305-1310, IEEE, 2022.

[47]. Basak, Setu, Md Mahbub Alam, Aniruddha Rakshit, Ahmed Al Marouf, and Anup Majumder, "Predicting and Staging Chronic Kidney Disease of Diabetes (Type-2) Patient Using Machine Learning Algorithms," Int. J. Innov. Technol. Explor. Eng 8, 2019.

[48]. Chen, Chi-Jim, Tun-Wen Pai, Hui-Huang Hsu, Chien-Hung Lee, Kuo-Su Chen, and Yung-Chih Chen, "Prediction of chronic kidney disease stages by renal ultrasound imaging," Enterprise Information Systems 14, no. 2, 178-195, 2020.

[49]. Søreide, Kjetil, "Receiver-operating characteristic curve analysis in diagnostic, prognostic and predictive biomarker research," Journal of clinical pathology 62.1, 1-5, 2009.

[50]. Obuchowski, Nancy A., and Jennifer A. Bullen, "Receiver operating characteristic (ROC) curves: a review of methods with applications in diagnostic medicine," Physics in Medicine & Biology 63.7, 07TR01, 2018.

[51]. Rashidi, Hooman H., Nam K. Tran, Elham Vali Betts, Lydia P. Howell, and Ralph Green, "Artificial intelligence and machine learning in pathology: the present landscape of supervised methods," Academic pathology 6, 2374289519873088, 2019.

[52]. Nasteski, Vladimir, "An overview of the supervised machine learning methods," Horizons. b 4, 51-62, 2017.

[53]. Antony, Linta, Sami Azam, Eva Ignatious, Ryana Quadir, Abhijith Reddy Beeravolu, Mirjam Jonkman, and Friso De Boer, "A comprehensive unsupervised framework for chronic kidney disease prediction," IEEE Access 9, 126481-126501, 2021.

[54]. Ojugo, Arnold Adimabua, and Elohor Ekurume, "Predictive intelligent decision support model in forecasting of the diabetes pandemic using a reinforcement deep learning approach," Int. J. Educ. Manag. Eng. 11.2, 40-48, 2021.

[55]. Sharma, Sahil, Vinod Sharma, and Atul Sharma, "Performance based evaluation of various machine learning classification techniques for chronic kidney disease diagnosis," arXiv preprint arXiv: 1606.09581, 2016.

[56]. Nelson, Wayne B, “Accelerated testing: statistical models, test plans, and data analysis,” John Wiley & Sons, 2009.

[57]. Makowski, Dominique, Mattan S. Ben-Shachar, Indrajeet Patil, and Daniel Lüdecke, "Methods and algorithms for correlation analysis in R," Journal of Open Source Software 5, no. 51, 2306, 2020.

[58]. Obilor, Esezi Isaac, and Eric Chikweru Amadi, "Test for significance of Pearson’s correlation coefficient," International Journal of Innovative Mathematics, Statistics & Energy Policies 6.1, 11-23, 2018.

[59]. Lee Rodgers, Joseph, and W. Alan Nicewander, "Thirteen ways to look at the correlation coefficient," The American Statistician 42, no. 1, 59-66, 1988.

[60]. Katiyar, Shruti, and Shruti Jain, "Predictive analysis on diabetes liver and kidney diseases using machine learning," Liver 2045, no. 1, 2020.

[61]. Hodge, Victoria, and Jim Austin, "A survey of outlier detection methodologies," Artificial intelligence review 22, 85-126, 2004.

[62]. Kelleher, John D., Brian Mac Namee, and Aoife D'arcy, “Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies,” MIT press, 2020.

[63]. Almasoud, Marwa, and Tomas E. Ward, "Detection of chronic kidney disease using machine learning algorithms with least number of predictors," International Journal of Soft Computing and Its Applications, v. 10, no. 8, 2019.

[64]. Dash, Ch Sanjeev Kumar, Ajit Kumar Behera, Satchidananda Dehuri, and Ashish Ghosh, "An outliers detection and elimination framework in classification task of data mining," Decision Analytics Journal 6, 100164, 2023.

[65]. Alrashidi, Aishah, and Ali Mustafa Qamar, "Data-Driven Load Forecasting Using Machine Learning and Meteorological Data," Computer Systems Science & Engineering 44, no. 3, 2023.

[66]. Abd Elkader, Seham, Mohammed Elmogy, Shaker El-Sappagh, and Abdel Nasser H. Zaied, "A framework for chronic kidney disease diagnosis based on case based reasoning," International Journal of Advanced Computer Research, v. 8, no. 35, 59-71, 2018.

[67]. Ali, Farman, Shaker El-Sappagh, SM Riazul Islam, Daehan Kwak, Amjad Ali, Muhammad Imran, and Kyung-Sup Kwak, "A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion," Information Fusion 63, 208-222, 2020.

[68]. Yildirim, Pinar, "Chronic kidney disease prediction on imbalanced data by multi-layer perceptron: Chronic kidney disease prediction," 2017 IEEE 41st annual computer software and applications conference (COMPSAC), Vol. 2, IEEE, 2017.

[69]. Mahesh, Batta, "Machine learning algorithms-a review," International Journal of Science and Research (IJSR), [Internet] 9, 381-386, 2020.

[70]. William, S. N., and S. A. Teukolsky, "What is a support vector machine," Nat Biotechnol 24, 1565-7, 2006.

[71]. Breiman, Leo, "Random forests," Machine learning 45, 5-32, 2001.

[72]. Carranza, Coleen, Corjan Nolet, Michiel Pezij, and Martine van der Ploeg, "Root zone soil moisture estimation with Random Forest," Journal of hydrology 593, 125840, 2021.

[73]. Bewick, Viv, Liz Cheek, and Jonathan Ball, "Statistics review 14: Logistic regression," Critical care 9, no. 1, 1-7, 2005.

[74]. Bhattacharyya, Debnath, Bhanu Prakash Doppala, and N. Thirupathi Rao, "Prediction and forecasting of persistent kidney problems using machine learning algorithms," Int. J Current Res Rev 12, 20, 134-139, 2020.

[75]. Nikam, Sagar S, "A comparative study of classification techniques in data mining algorithms," Oriental Journal of Computer Science and Technology 8, no. 1, 13-19, 2015.

[76]. Emon, Minhaz Uddin, Rakibul Islam, Maria Sultana Keya, and Raihana Zannat, "Performance analysis of chronic kidney disease through machine learning approaches," In 2021 6th International Conference on Inventive Computation Technologies (ICICT), pp. 713-719, IEEE, 2021.

[77]. Dolo, Bakary, Faiza Loukil, and Khouloud Boukadi, "Early Detection of Diabetes Mellitus Using Differentially Private SGD in Federated Learning," In 2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA), pp. 1-8, IEEE, 2022.

[78]. Sagi, O., & Rokach, L, “Ensemble learning: A survey,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1249, https://doi.org/10.1002/widm.1249, 2018.

[79]. Dritsas, Elias, and Maria Trigka, "Machine learning techniques for chronic kidney disease risk prediction," Big Data and Cognitive Computing 6.3, 98, 2022.

[80]. Alassaf, Reem A., Khawla A. Alsulaim, Noura Y. Alroomi, Nouf S. Alsharif, Mishael F. Aljubeir, Sunday O. Olatunji, Alaa Y. Alahmadi, Mohammed Imran, Rahma A. Alzahrani, and Nora S. Alturayeif, "Preemptive diagnosis of chronic kidney disease using machine learning techniques," In 2018 international conference on innovations in information technology (IIT), pp. 99-104, IEEE, 2018.

[81]. Ali, Md Shahin, Md Khairul Islam, A. Arjan Das, D. U. S. Duranta, Mst Haque, and Md Habibur Rahman, "A novel approach for best parameters selection and feature engineering to analyze and detect diabetes: Machine learning insights," BioMed Research International-23, 2023.

[82]. Hand, David J., Peter Christen, and Nishadi Kirielle, "F*: an interpretable transformation of the F-measure," Machine Learning 110.3, 451-456, 2021.

[83]. Ben-David, Arie, "About the relationship between ROC curves and Cohen's kappa," Engineering Applications of Artificial Intelligence 21, no. 6, 874-882, 2008.

[84]. Obuchowski, Nancy A., Michael L. Lieber, and Frank H. Wians Jr., "ROC curves in clinical chemistry: uses, misuses, and possible solutions," Clinical chemistry 50.7, 1118-1125, 2004.

[85]. Zhu, Wen, Nancy Zeng, and Ning Wang, "Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations," NESUG proceedings: health care and life sciences, Baltimore, Maryland 19, 67, 2010.

[86]. Mourão-Miranda, Janaina, David R. Hardoon, Tim Hahn, Andre F. Marquand, Steve CR Williams, John Shawe-Taylor, and Michael Brammer, "Patient classification as an outlier detection problem: an application of the one-class support vector machine," Neuroimage 58, no. 3, 793-804, 2011.

APPENDIX

Appendix 1

Implementation code:

```
# Importing Library and Load the Data
import pandas as pd
import numpy as np
data = pd.read_csv('kidney_disease.csv')
data.head (5)
data['classification'].unique()
data['classification'].value_counts()
data.head (5)
data.columns
data.tail(5)
data.describe()
data.corr()
import matplotlib.pyplot as plt
import seaborn as sb
%matplotlib inline

sb.set(rc = {'figure.figsize':(16,10)})
sb.heatmap(data.corr(), annot = True)
data.dtypes
cols=data.select_dtypes(include=['object']).columns
cols
from sklearn.preprocessing import LabelEncoder
cols=data.select_dtypes(include=['object']).columns
# This code will fetch columns whose data type is an object.
le=LabelEncoder()
data[cols]=data[cols].apply(le.fit_transform)
data.head(5)
data.describe()
from sklearn.impute import SimpleImputer

imp = SimpleImputer(missing_values = np.NaN, strategy = 'mean')
```

```
imp.fit(data)
data = pd.DataFrame(data = imp.transform(data))
data.isnull().sum().sort_values(ascending = False).head(13)
data = data.drop([0], axis = 1)
data.head (5)
Q1 = data[1].quantile(0.25)
Q3 = data[1].quantile(0.75)
Q1, Q3
IQR = Q3 - Q1
lowerlimit = Q1 - 1.5 * IQR
upperlimit = Q3 + 1.5 * IQR
data[(data[1] < lowerlimit) | (data[1] > upperlimit)]
def remove(x):
if x < lowerlimit:
x = lowerlimit
elif x > upperlimit:
x = upperlimit
return x
data[1] = data[1].apply(remove)
1, 2, 10, 11, 12, 13, 14, 15, 16, 17, 18
data[18] = data[18].apply(remove)
data[1] = data[1].apply(lambda v: (v - data[1].min()) / (data[1].max() - data[1].min()))
data[2] = data[2].apply(lambda v: (v - data[2].min()) / (data[2].max() - data[2].min()))
data[10] = data[10].apply(lambda v: (v - data[10].min()) / (data[10].max() - data[10].min()))
data[11] = data[11].apply(lambda v: (v - data[11].min()) / (data[11].max() - data[11].min()))
data[25].unique()
data[25].replace({2.0:1.0}, inplace=True)
data[25].unique()
from sklearn.model_selection import train_test_split
from numpy import mean
from numpy import std
"""
X = data.drop([7], axis = 1)
X = data.drop([5], axis = 1)
```

```
X = data.drop([14], axis = 1)
X = data.drop([22], axis = 1)
X = data.drop([8], axis = 1)
"""
X = data.drop([25], axis = 1)
Y = data[25]
#x_train_smt, x_test_smt, y_train_smt, y_test_smt = train_test_split(X, Y, test_size = 0.2,random_state = 109) # 80% training and 20% test
#x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2,random_state = 109) # 80% training and 20% test
from collections import Counter
from imblearn.combine import SMOTETomek
counter = Counter(Y)
print('Before', counter)
smtt = SMOTETomek(random_state = 139)
X,Y = smtt.fit_resample(X,Y)
counter = Counter(Y)
print('After', counter)
data.head (5)
x_train_smt, x_test_smt, y_train_smt, y_test_smt = train_test_split(X, Y, test_size = 0.2,rando
m_state = 109) # 80% training and 20% test

#SVC
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
modelSVC = SVC(probability=True)
modelSVC.fit(x_train_smt, y_train_smt)
print(modelSVC.score(x_test_smt, y_test_smt))

y_pred = modelSVC.predict(x_test_smt)
ac = accuracy_score(y_test_smt, y_pred)
print(ac)
from sklearn.model_selection import cross_val_score
cv_score_for_SVC = cross_val_score(modelSVC, x_train_smt, y_train_smt, cv = 10)
```

```
print(cv_score_for_SVC)
print(mean(cv_score_for_SVC))
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

y_pred = modelSVC.predict(x_test_smt)
print(confusion_matrix(y_test_smt, y_pred))
print(classification_report(y_test_smt, y_pred))
yp=modelSVC.predict(x_test_smt)
from sklearn.metrics import cohen_kappa_score
cmSVC = confusion_matrix(y_test_smt, modelSVC.predict(x_test_smt))

#If we use TP TN FP and FN of below's comment we get a range
#FP = cmSVC.sum(axis=0) - np.diag(cmSVC)
#FN = cmSVC.sum(axis=1) - np.diag(cmSVC)
#TP = np.diag(cmSVC)
#TN = cmSVC.sum() - (FP + FN + TP)

TP = cmSVC[1,1]
TN = cmSVC[0,0]
FP = cmSVC[0,1]
FN = cmSVC[1,0]

# Sensitivity, hit rate, recall, or true positive rate
TPR = TP/float(TP+FN)
# Specificity or true negative rate
TNR = TN/float(TN+FP)
# Precision or positive predictive value
PPV = TP/float(TP+FP)
# Negative predictive value
NPV = TN/float(TN+FN)
# Fall out or false positive rate
FPR = FP/float(FP+TN)
# False negative rate
```

```
FNR = FN/float(TP+FN)
# False discovery rate
FDR = FP/float(TP+FP)
totalSVC=sum(sum(cmSVC))
Accuracy = (TN+TP)/totalSVC
# MCC
val = (TP * TN) - (FP * FN)
MCC_SVC = val / np.sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))
# Cohen Kappa
Y_pred = modelSVC.predict(x_test_smt)
cohen_score = cohen_kappa_score(y_test_smt, Y_pred)
#print("Sensitivity/TPR: " + str(TPR))
print("FPR: " + str(FPR))
print("Specificity/TNR: " + str(TNR))
print("MCC: " + str(MCC_SVC))
print("Cohen Kappa: " + str(cohen_score))
from sklearn.metrics import roc_auc_score
# predict probabilities
pred_prob1 = modelSVC.predict_proba(x_test_smt)
pred_prob2 = modelSVC.predict_proba(x_test_smt)
# auc scores
auc_score1 = roc_auc_score(y_test_smt, pred_prob1[:,1])
auc_score2 = roc_auc_score(y_test_smt, pred_prob2[:,1])
print(auc_score1, auc_score2)

#RFC
from sklearn import ensemble
from sklearn.metrics import accuracy_score

modelRFC = ensemble.RandomForestClassifier(n_estimators = 100) # by default n_estimators
= 100
modelRFC.fit(x_train_smt, y_train_smt)
print(modelRFC.score(x_test_smt, y_test_smt))
y_pred = modelRFC.predict(x_test_smt)
```

```
ac = accuracy_score(y_test_smt, y_pred)
print(ac)
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
y_pred = modelRFC.predict(x_test_smt)
print(confusion_matrix(y_test_smt, y_pred))
print(classification_report(y_test_smt, y_pred))
from sklearn.metrics import cohen_kappa_score
cmRFC = confusion_matrix(y_test_smt, modelRFC.predict(x_test_smt))

TP = cmRFC[1,1]
TN = cmRFC[0,0]
FP = cmRFC[0,1]
FN = cmRFC[1,0]

TPR = TP/float(TP+FN)
TNR = TN/float(TN+FP)
PPV = TP/float(TP+FP)
NPV = TN/float(TN+FN)
FPR = FP/float(FP+TN)
FNR = FN/float(TP+FN)
FDR = FP/float(TP+FP)
totalRFC=sum(sum(cmRFC))
Accuracy = (TN+TP)/totalRFC
val = (TP * TN) - (FP * FN)
MCC_RFC = val / np.sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))
Y_pred = modelRFC.predict(x_test_smt)
cohen_score = cohen_kappa_score(y_test_smt, Y_pred)

print("FPR: " + str(FPR))
print("Specificity/TNR: " + str(TNR))
print("MCC: " + str(MCC_RFC))
print("Cohen Kappa: " + str(cohen_score))
from sklearn.metrics import roc_auc_score
```

```
pred_prob1 = modelRFC.predict_proba(x_test_smt)
pred_prob2 = modelRFC.predict_proba(x_test_smt)

auc_score1 = roc_auc_score(y_test_smt, pred_prob1[:,1])
auc_score2 = roc_auc_score(y_test_smt, pred_prob2[:,1])
print(auc_score1, auc_score2)
#LR
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

modelLR = LogisticRegression()
modelLR.fit(x_train_smt, y_train_smt)
print(modelLR.score(x_test_smt, y_test_smt))



y_pred = modelLR.predict(x_test_smt)
ac = accuracy_score(y_test_smt, y_pred)
print(ac)
from sklearn.model_selection import cross_val_score
cv_score_for_LR = cross_val_score(modelLR, x_train_smt, y_train_smt, cv = 10)
print(mean(cv_score_for_LR))
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

y_pred = modelLR.predict(x_test_smt)
print(confusion_matrix(y_test_smt, y_pred))
print(classification_report(y_test_smt, y_pred))
from sklearn.metrics import cohen_kappa_score
cmLR = confusion_matrix(y_test_smt, modelLR.predict(x_test_smt))

TP = cmLR[1,1]
TN = cmLR[0,0]
FP = cmLR[0,1]
```

```
FN = cmLR[1,0]

TPR = TP/float(TP+FN)
TNR = TN/float(TN+FP)
PPV = TP/float(TP+FP)
NPV = TN/float(TN+FN)
FPR = FP/float(FP+TN)
FNR = FN/float(TP+FN)
FDR = FP/float(TP+FP)
totalLR=sum(sum(cmLR))
Accuracy = (TN+TP)/totalLR
val = (TP * TN) - (FP * FN)
MCC_LR = val / np.sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))
Y_pred = modelLR.predict(x_test_smt)
cohen_score = cohen_kappa_score(y_test_smt, Y_pred)

print("FPR: " + str(FPR))
print("Specificity/TNR: " + str(TNR))
print("MCC: " + str(MCC_LR))
print("Cohen Kappa: " + str(cohen_score))
from sklearn.metrics import roc_auc_score

pred_prob1 = modelLR.predict_proba(x_test_smt)
pred_prob2 = modelLR.predict_proba(x_test_smt)

auc_score1 = roc_auc_score(y_test_smt, pred_prob1[:,1])
auc_score2 = roc_auc_score(y_test_smt, pred_prob2[:,1])
print(auc_score1, auc_score2)

#KNN
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

```
modelKNN = KNeighborsClassifier(n_neighbors = 5, algorithm = 'ball_tree', weights = 'distance', metric = 'minkowski', p = 2)
modelKNN.fit(x_train_smt, y_train_smt)
print(modelKNN.score(x_test_smt, y_test_smt))

y_pred = modelKNN.predict(x_test_smt)
ac = accuracy_score(y_test_smt, y_pred)
print(ac)
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

y_pred = modelKNN.predict(x_test_smt)
print(confusion_matrix(y_test_smt, y_pred))
print(classification_report(y_test_smt, y_pred))
from sklearn.metrics import cohen_kappa_score
cmKNN = confusion_matrix(y_test_smt, modelKNN.predict(x_test_smt))

TP = cmKNN[1,1]
TN = cmKNN[0,0]
FP = cmKNN[0,1]
FN = cmKNN[1,0]

TPR = TP/float(TP+FN)
TNR = TN/float(TN+FP)
PPV = TP/float(TP+FP)
NPV = TN/float(TN+FN)
FPR = FP/float(FP+TN)
FNR = FN/float(TP+FN)
FDR = FP/float(TP+FP)
totalKNN = sum(sum(cmKNN))
Accuracy = (TN+TP)/totalKNN
val = (TP * TN) - (FP * FN)
MCC_KNN = val / np.sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))
Y_pred = modelKNN.predict(x_test_smt)
```

```
cohen_score = cohen_kappa_score(y_test_smt, Y_pred)

print("Sensitivity/TPR: " + str(TPR))
print("FPR: " + str(FPR))
print("Specificity/TNR: " + str(TNR))
print("MCC: " + str(MCC_KNN))
print("Cohen Kappa: " + str(cohen_score))
from sklearn.metrics import roc_auc_score

pred_prob1 = modelKNN.predict_proba(x_test_smt)
pred_prob2 = modelKNN.predict_proba(x_test_smt)

auc_score1 = roc_auc_score(y_test_smt, pred_prob1[:,1])
auc_score2 = roc_auc_score(y_test_smt, pred_prob2[:,1])
print(auc_score1, auc_score2)

#SGD
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import accuracy_score
modelSGDC = SGDClassifier(loss = 'log', penalty = "l1", max_iter = 10)
modelSGDC.fit(x_train_smt, y_train_smt)
print(modelSGDC.score(x_test_smt, y_test_smt))

y_pred = modelSGDC.predict(x_test_smt)
ac = accuracy_score(y_test_smt, y_pred)
print(ac)

from sklearn.model_selection import cross_val_score
cv_score_for_SGDC = cross_val_score(modelSGDC, x_train_smt, y_train_smt, cv = 10)
print(mean(cv_score_for_SGDC))
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

y_pred = modelSGDC.predict(x_test_smt)
```

```
print(confusion_matrix(y_test_smt, y_pred))
print(classification_report(y_test_smt, y_pred))
from sklearn.metrics import cohen_kappa_score
cmSGDC = confusion_matrix(y_test_smt, modelSGDC.predict(x_test_smt))

TP = cmSGDC[1,1]
TN = cmSGDC[0,0]
FP = cmSGDC[0,1]
FN = cmSGDC[1,0]

TPR = TP/float(TP+FN)
TNR = TN/float(TN+FP)
PPV = TP/float(TP+FP)
NPV = TN/float(TN+FN)
FPR = FP/float(FP+TN)
FNR = FN/float(TP+FN)
FDR = FP/float(TP+FP)
totalSGDC=sum(sum(cmSGDC))
Accuracy = (TN+TP)/totalSGDC
val = (TP * TN) - (FP * FN)
MCC_SGDC = val / np.sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))
Y_pred = modelSGDC.predict(x_test_smt)
cohen_score = cohen_kappa_score(y_test_smt, Y_pred)

print("FPR: " + str(FPR))
print("Specificity/TNR: " + str(TNR))
print("MCC: " + str(MCC_SGDC))
print("Cohen Kappa: " + str(cohen_score))
from sklearn.metrics import roc_auc_score

pred_prob1 = modelSGDC.predict_proba(x_test_smt)
pred_prob2 = modelSGDC.predict_proba(x_test_smt)

auc_score1 = roc_auc_score(y_test_smt, pred_prob1[:,1])
```

```
auc_score2 = roc_auc_score(y_test_smt, pred_prob2[:,1])
print(auc_score1, auc_score2)

#Ensemble RSL
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import StackingClassifier

estimators = [
('rf', RandomForestClassifier(n_estimators=10, random_state=42)),
('svc',SVC(random_state = 42)),
('lr', LogisticRegression(solver='liblinear'))
]
clf = StackingClassifier(
estimators=estimators, final_estimator = LogisticRegression(solver='liblinear')
)

clf.fit(x_train_smt, y_train_smt).score(x_test_smt, y_test_smt)
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

y_pred = clf.predict(x_test_smt)
print(confusion_matrix(y_test_smt, y_pred))
print(classification_report(y_test_smt, y_pred))
from sklearn.metrics import cohen_kappa_score
cmABCH = confusion_matrix(y_test_smt, clf.predict(x_test_smt))

TP = cmABCH[1,1]
TN = cmABCH[0,0]
FP = cmABCH[0,1]
FN = cmABCH[1,0]

TPR = TP/float(TP+FN)
TNR = TN/float(TN+FP)
```

```
PPV = TP/float(TP+FP)
NPV = TN/float(TN+FN)
FPR = FP/float(FP+TN)
FNR = FN/float(TP+FN)
FDR = FP/float(TP+FP)
totalABCH = sum(sum(cmABCH))
Accuracy = (TN+TP)/totalABCH

val = (TP * TN) - (FP * FN)
MCC_ABCH = val / np.sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))

Y_pred = clf.predict(x_test_smt)
cohen_score = cohen_kappa_score(y_test_smt, Y_pred)

print("Sensitivity/TPR: " + str(TPR))
print("FPR: " + str(FPR))
print("Specificity/TNR: " + str(TNR))
print("MCC: " + str(MCC_ABCH))
print("Cohen Kappa: " + str(cohen_score))
from sklearn.metrics import roc_auc_score

pred_prob1 = clf.predict_proba(x_test_smt)
pred_prob2 = clf.predict_proba(x_test_smt)

auc_score1 = roc_auc_score(y_test_smt, pred_prob1[:,1])
auc_score2 = roc_auc_score(y_test_smt, pred_prob2[:,1])

print(auc_score1, auc_score2)
from sklearn.metrics import roc_curve

pred_prob1 = modelSVC.predict_proba(x_test_smt)
pred_prob3 = modelRFC.predict_proba(x_test_smt)
pred_prob5 = modelLR.predict_proba(x_test_smt)
pred_prob7 = modelKNN.predict_proba(x_test_smt)
```

```
pred_prob8 = modelSGDC.predict_proba(x_test_smt)
pred_prob10 = clf.predict_proba(x_test_smt)

fpr1, tpr1, thresh1 = roc_curve(y_test_smt, pred_prob1[:,1], pos_label = 1)
fpr3, tpr3, thresh3 = roc_curve(y_test_smt, pred_prob3[:,1], pos_label = 1)
fpr5, tpr5, thresh5 = roc_curve(y_test_smt, pred_prob5[:,1], pos_label = 1)
fpr7, tpr7, thresh7 = roc_curve(y_test_smt, pred_prob7[:,1], pos_label = 1)
fpr8, tpr8, thresh8 = roc_curve(y_test_smt, pred_prob8[:,1], pos_label = 1)
fpr10, tpr10, thresh10 = roc_curve(y_test_smt, pred_prob10[:,1], pos_label = 1)

random_probs = [0 for i in range(len(y_test_smt))]
p_fpr, p_tpr, _ = roc_curve(y_test_smt, random_probs, pos_label=1)
import matplotlib.pyplot as plt
plt.style.use('seaborn')

# plot roc curves
plt.plot(fpr1, tpr1, linestyle = '--', color = 'orange', label = 'SVC')
plt.plot(fpr3, tpr3, linestyle = '--', color = 'red', label = 'RFC')
plt.plot(fpr5, tpr5, linestyle = '--', color = 'yellow', label = 'LR')
plt.plot(fpr7, tpr7, linestyle = '--', color = 'magenta', label = 'KNN')
plt.plot(fpr8, tpr8, linestyle = '--', color = 'cyan', label = 'SGDC')
plt.plot(fpr10, tpr10, linestyle = '--', color = 'teal', label = 'Enemble')
plt.plot(p_fpr, p_tpr, linestyle = '--', color = 'blue')
plt.title('ROC curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive rate')
plt.legend(loc='best')
#plt.savefig('ROC',dpi=300)
plt.show()
```

LIST OF PUBLICATION:

1. M. M. I. Raju, S. Sarker, and M. M. Islam, “Chronic Kidney Disease Prediction using Ensemble Machine Learning,” Journal of Information Hiding and Multimedia Signal Processing (JIH-MSP), Vol. 14, No. 1, pp. 1-9, https://doi.org/10.5281/zenodo.7741968, 2023.