American Academic Scientific Research Journal for Engineering, Technology, and Sciences
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402
https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

Diagnostic Assessment and Interpretation of Clinical Kidney Dataset Using Machine Learning

Quadri Ghazal[a]*, Peter Oluwasayo Adigun[b], Nelson Abimbola Ayuba Azeez[c]

[a]*Office of Cybersecurity Compliance, National Health Service, Greater Glasgow & Clyde *NHS GGC (, Digital Services Unit, Scotland, UK*

[b]*Department of Computer Science, New Mexico Highlands University, 1005 Diamond St, Las Vegas, New Mexico, USA*

[c]*Department of Physics, University of Abuja, Abuja, Federal Capital Territory, Nigeria*

[a]*Email: quadri.ghazal@nhs.scot*

[b]*Email: poadigun@nmhu.edu*

[c]*Email: azeez.abimbola2019@uniabuja.edu.ng*

Abstract

One of the vast applications of machine learning is in medical science. The number of individuals who face the medical condition of kidney failure is critical. One of the problems is the inadequate interpretation of medical diagnostics. This study employed machine learning to proffer solutions to this challenge. The design of an ML model that can detect, classify, and interpret kidney test data was achieved using different machine learning algorithms. This was carried out by training an ML model using a kidney test dataset and different machine learning algorithms. The different machine learning algorithms applied were Random Forest, Naïve-Bayes, Support Vector Machine, and Kernel Support Vector Machine, to establish the most efficient algorithm and therefore indicate that machine learning can classify and interpret kidney test datasets. Random Forest algorithms have the best test accuracy, which is 100%, Support Vector Machine and Kernel Support Vector Machine algorithms have a close accuracy score close to that of Random Forest, with the test accuracy of 95.83% and train accuracy of 93.2%, both having the same accuracy value. The Naïve-Bayes algorithm has the lowest accuracy score of 93% for both the tested and trained data. This finding established the integrated application of artificial intelligence and, therefore, indicates that machine learning can classify and categorize the chronic kidney disease dataset.

Keywords: Kidney; machine learning; random forest; Navies-Bayes; support vector machine; Kernel SVM.

Received: 5/25/2026
Accepted: 7/25/2026
Published: 8/5/2026

---

* Corresponding author.

1. Introduction

The basis of the scientific approach towards problems and challenges is to design a machine learning (ML) model that identifies the problem and proffers solutions to the problem entity, which, as a result of this process, knowledge and new technologies are developed. The application of machine learning (ML) as of today has wide coverage in every field and professional setting[1-5]. This application was born from an in-depth understanding of preceding techs and the fabrication of new ones from outstanding research made by prominent authors. Contemporarily, the application of machine learning in medical science and the medical profession is on the rise. This is due to the fact that identification, decision-making, diagnosis, imaging, and prognosis are what machine learning can now do effortlessly. This ability of machine learning (ML) has enabled this study to design and model a machine learning algorithm that can detect and classify kidney clinical datasets for kidney failure.

Machine learning application has made many contributions to healthcare. This contribution has been reported by prominent authors, and their publication has established findings and machine learning designs [1, 6-8]. Saleem & Chishti established that machine learning algorithms (MLAs) are being used to screen and diagnose disease, prognosticate, and predict therapeutic responses. These have been possible due to hundreds of new algorithms that have been developed to improve clinical decision-making and patient outcomes, although there are still some gaps to fill [9]. Machine learning (ML) has been regarded as the study of tools and methods for identifying patterns in data [9]. The appropriate application of ML to data has been able to transform patient risk stratification broadly in the field of medicine, and especially in infectious diseases [9]. According to the online medical database, kidney failure is a condition in which one or both of the kidneys no longer work on their own [10, 11]. A common term used for this is chronic kidney disease. Chronic kidney disease (CKD) has been indicated as a leading medical problem, and the global estimated prevalence of CKD is 13.4%, and also, patients with kidney failure needing renal replacement therapy are estimated between 5 and 7 million [12]. In the United States, acute kidney failure affects about 3 per 1,000 people a year. Chronic kidney failure affects about 1 in 1,000 people, with 3 per 10,000 people newly developing the condition each year [11]. Acute kidney failure is often reversible, while chronic kidney failure often is not [13].

Kidney failure (KF) or renal failure is the deterioration of one or both of the kidneys, which can be acute or chronic. Chronic kidney failure is the most severe stage of kidney disease, and it is fatal without treatment [10]. Furthermore, kidney failure is a medical end-stage kidney disease, a medical condition in which the kidneys can no longer adequately filter waste products from the blood, functioning at less than 15% of normal levels [14]. Kidney failure is categorized as either acute kidney failure, which develops rapidly and may resolve, or chronic kidney failure, which develops slowly and can often be irreversible [13]. Causes include diabetes, high blood pressure, and acute kidney injuries. Symptoms include fatigue, nausea and vomiting, swelling, changes in how often you go to the bathroom, and brain fog. Treatment includes dialysis or a kidney transplant [10, 11].

Traditional diagnostic methods rely on laboratory tests and clinical indicators, which might not always detect subtle changes in kidney function at an early stage. Chronic kidney disease (CKD) remains a prevalent health issue worldwide, with a significant impact on both individual health and healthcare systems. Machine learning models have shown promise in analyzing diverse datasets, including genetic, clinical, and demographic

information, providing a more comprehensive assessment of risk factors and disease progression markers for early CKD detection. The utilization of machine learning in clinical decision-making for kidney failure detection offers the potential for personalized medicine approaches.

The clinical record, information, and data are valuable assets for machine learning. Based on the elements, namely symptoms, complications, and type of kidney disease, the ML model can be trained to predict, decide, or classify, depending on the choice of the programmer. The diagnosis and symptoms are presented in the form of data or information. These data have patterns that can be recognized by training an ML model using machine learning algorithms. This enables the ML machine to be able to decide, detect, and diagnose by classifying the clinical dataset as either chronic kidney disease (CKD) or "No chronic kidney disease" (NOTCKD).

With appropriate treatment, many with CKD can or are likely to continue living [14]. Also, with the influence of assistive systems like ML models, detection and early identification will be possible. Also, this study has employed an ML model to classify and indicate if a patient has chronic kidney disease or not.

This research will assist in contributing to early detection of CKD, which will crucially improve in preventing its progression to kidney failure and associated complications. Hence, there is a pressing need to explore and implement more accurate and efficient tools for early detection and prediction of kidney failure. Since there have been many studies on the application of machine learning in kidney failure diagnostics, this research aims to design a model that can predict and optimize clinical decisions of kidney failure diagnostics.

This research is to train and test a large sampling dataset of kidney disease conditions using machine learning. The dataset comprises various clinical data of kidney disease patients. The dataset is secondary data from an online database, *Kaggle*. This study aims to model and implement a machine learning system capable of making clinical decisions for the early detection of kidney failure.

The objectives of the research are:

- To train the system for classifying and detecting the dataset using the MLA.
- To test and verify the system for the simulation of different kidney disease clinical data.
- To retest and establish the efficiency of the ML model.

2. Methods

2.1 Description of the Study

This practical study presents a design of a machine learning (ML) model trained to make clinical decisions in detecting and diagnosing a clinical dataset of patients with kidney diagnostics. The model was programmed using supervised learning (SL), and four major machine learning algorithms (MLAs) were implemented. The machine learning algorithms implemented are Random Forest (RF), Support Vector Machine (SVM), Kernel SVM (KSVM), and Naïve Bayes Classifier (NBC)

2.2 Data Source and Characteristics

A secondary dataset was used for the ML modeling. The dataset was sourced from *Kaggle* [15, 16]. The dataset originates from the UCI Machine Learning Repository, authored by Yadav, Nitesh, and was prepared clinically to diagnose patients' kidney disease (KD) conditions [15].The dataset is taken over 2 months in India. It has 400 rows with 25 features. The dataset comprises several medical predictor variables and a single target variable (Class). Predictor variables include Blood Pressure (Bp), Albumin (Al), Sugar (Su), Red Blood Cell (Rbc), Serum Creatinine (Sc), Specific Gravity (Sg), Blood Urea (Bu), Sodium (Sod), Potassium (Pot), Hemoglobin (Hemo), White Blood Cell Count (Wbcc), Red Blood Cell Count (Rbcc), Hypertension (Htn), and the Predicted Class.

2.3 Modeling Software

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation[1, 17]. This programming language is used to code, model, train, and simulate the background radiation.

2.4 Preprocessing

The dataset was preprocessed for training and testing. Recoding, categorization, and other data manipulation, which include the column for the diagnosis of the kidney was coded as a binary value (0 and 1). Then, the dataset is trained using the library function of the science kit tool. Machine learning algorithms, namely Random Forest (RF), Naive-Bayes Classifier (NBC), Support Vector Machine (SVM), and kernel SVM (KSVM), were used in training and testing the model. A confusion matrix was also initialized for evaluation purposes of the model and assessment of the performance metrics of each machine learning algorithm. Each algorithm was used to train and test the diagnosis of the kidney dataset, in which the machine is trained using different algorithms, and each model was tested and evaluated, and performance metrics were also computed.

2.5 Data Importation

The Python programming library functions were initialized as depicted in “input 1” below. The library functions initialized were *pandas as pd, numpy as np, matlplotlib.pyplot as plt, seaborn as sns,* and other library functions.

Input 1: Initialization of library functions

In [1]

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn import model_selection

from sklearn.model_selection import KFold
```

Input 2 illustrates the importation of the kidney disease dataset. This operation was carried out using the code depicted below.

Input 2: Importation of the kidney disease dataset

In [2]

```
3. dataset = pd.read_csv ("new_model.csv")
4. dataset
```

Also, the data description was programmed as depicted in "input 3". This was done to get the statistics of the data. The columns were reduced to five because the dataset is very large and it cannot contain the document size.

Input 3: Descriptive statistics of the kidney disease dataset

In [3]

```
5. columns_to_describe = ['Bp', 'Sg', 'Al', 'Su', 'Class']
6. description = dataset[columns_to_describe].describe()
7. print(description)
```

Output 3 illustrates the descriptive statistics of the kidney disease dataset. This includes the mean, count, percentile, range, and standard deviation.

Input 4 is the code used to execute the information of the kidney disease dataset.

Input 4: Information code of the kidney disease dataset

In [4]

```
8. dataset.info()
```

2.6 Data Visualization and Optimization

The visualization of the kidney disease dataset is essential because it easily reveals the trend and illustrates the variation between the various classes (CKD or NOTCKD) of kidney failure. “Input 5 is the code used to generate the countplot, which is a bar chart”.

Input 5: Code for visualization of the kidney disease dataset

In [5]

```
sns.countplot(x='Class', data=dataset, label='count')

plt.show()
```

The class in the dataset was dropped in (Input 6) to begin to train the machine using different algorithms, namely, SVM, KSVM, RF, and NB.

Input 6: Code for isolation of the class column in the dataset

In [6]

```
X = dataset.drop(('Class'), axis = 1)

X
```

Class in the dataset was configured as the dependent variable, while the various predictors of CKD were configured as the independent variables. The next block shows the result of the summary of the kidney disease dataset.

Input 7: Code to recall the transformed data for the class of kidney failure

In [7]

```
y = dataset["Class"]

y
```

2.7 Data Training and Testing

The "*sklearn library*" in the Python program is a science kit tool, which was used to train and test the kidney disease dataset. The "*sklearn library*" also includes sub-libraries. These libraries were initialized to use MLA for training the machine so that classification prediction is performed.

Also, a confusion matrix is part of the library initialized in order to evaluate the performance of the classification model through the calculation of performance metrics. Inputs 8 – 13 were Python codes and "*sklearn library function*" used for training and testing the predicting machine.

In input 8, the "*sklearn.model_selection*" is used to split arrays or case matrices into random subsets for train and testing the dataset, respectively.

Input 8: Initialization of the library functions for training and testing

In [8]

```
from sklearn.model_selection import train_test_split
```

Thirty percent of the dataset was used for training the machine, and the remaining 70% was reserved for testing and prediction purposes. The random state was set to 42 to maintain consistent results.

Input 9: Selection and splitting of the kidney disease dataset for training and testing

In [9]

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Input 10: Training of the data of the independent variables

In [10]

```
X_train
```

Input 11: Training of the data of the dependent variable.

In [11]

```
y_train
```

Input 12: Testing of the data of the independent variables

In [12]

```
X_test
```

Input 13: Testing of the data of the dependent variable

In [13]

```
y_test
```

1. Random Forest Data Modeling

Part of the sub-library functions of "*sklearn*", which is "*sklearn.ensemble*", is a module that includes two averaging algorithms based on randomized decision trees. The random forest (RF) is used to create a set of decision trees from a randomly selected subset of the training set.

The random forest classifier creates a set of decision trees from a randomly selected subset of the training set. The *Max_depth* hyperparameter is used in setting it to 5, which will limit the depth of the decision trees used in the model, helping to prevent the model from overfitting the training data and improving the generalization performance on the new and unseen data.

Input 14: Initialization of the random forest algorithm for training the data

In [14]

```
from sklearn.ensemble import RandomForestClassifier

RFC = RandomForestClassifier (n_estimators=100, criterion='entropy', max_depth=5, random_state=42)

RFC.fit(X_train, y_train)
```

Input 15: Code for prediction of the trained data

In [15]

```
y_predict_train = RFC.predict(X_train)

y_predict_train
```

Input 16: Code to generate the performance metrics of the trained data

In [16]

```
from sklearn.metrics import classification_report
```

```
print(classification_report(y_train, y_predict_train))
```

Input 17: Code for the prediction of the tested data

In [17]

```
from sklearn.metrics import classification_report, confusion_matrix

y_predict_test = RFC.predict(X_test)

cm = confusion_matrix (y_test, y_predict_test)

sns.heatmap(cm,annot = True)
```

Input 18: Code to generate the performance metrics of the tested data

In [18]

```
print(classification_report(y_test, y_predict_test))
```

2. Naive-Bayes Data Modeling

The Naïve-Bayes models were a group of extremely fast and simple classification algorithms that are often suitable for very high-dimensional datasets.

Input 19: Initialization of the Naïve-Bayes algorithm for training the data

```
from sklearn.naive_bayes import GaussianNB

GB_classifier = GaussianNB()

GB_classifier.fit(X_train, y_train)
```

Input 20: Code for the prediction of the trained data

In [20]

```
from sklearn.metrics import classification_report, confusion_matrix



y_predict_train = GB_classifier.predict(X_train)
```

```
y_predict_train
```

Input 21: Code to generate the performance metrics of the trained data

In [21]

```
print(classification_report(y_train, y_predict_train))
```

Input 22: Code for the prediction of the tested data

In [22]

```
from sklearn.naive_bayes import GaussianNB

GB_classifier = GaussianNB()

GB_classifier.fit(X_test, y_test)
```

Input 23: Code for the computation of the confusion matrix of the tested data

In [23]

```
from sklearn.metrics import classification_report, confusion_matrix



y_predict_test = GB_classifier.predict(X_test)

cm = confusion_matrix(y_test,y_predict_test)

sns.heatmap(cm,annot=True)
```

Input 24: Code to generate the performance metrics of the tested data

In [24]

```
print(classification_report(y_train, y_predict_train))
```

3. Support Vector Machine (SVM) Data Modeling

The SVM algorithm is a supervised learning algorithm used for outlier detection, regression, and classification that are both powerful and adaptable.

Input 25: Initialization of the SVM algorithm for training the data

In [25]

```
from sklearn.svm import SVC

classifier = SVC(kernel = 'linear')

classifier.fit(X_train, y_train)
```

Input 26: Code for the prediction of the tested data

In [26]

```
print(classifier.predict(X_test))
```

Input 27: Code to generate the accuracy score of the tested data

In [27]

```
from sklearn.metrics import accuracy_score

accuracy = (accuracy_score(y_test, y_predict_test)*100)

accuracy
```

Input 28: Code to generate the accuracy score of the trained data

In [28]

```
from sklearn.metrics import confusion_matrix, accuracy_score

cm = confusion_matrix(y_train,y_predict_train)

print(cm)

accuracy_score(y_train, y_predict_train)*100
```

4. Kernel SVM Data Modeling

The kernel SVM (KSVM) algorithm is a machine learning algorithm that transforms the data into the required form using the kernel trick.

Input 29: Initialization of the Kernel SVM algorithm for training the data

In [29]

```
from sklearn.svm import SVC

classifier = SVC(kernel = 'rbf')

classifier.fit(X_train, y_train)
```

Input 30: Code for the prediction of the tested data

In [30]

```
print(classifier.predict(X_test))
```

Input 31: Code to generate the accuracy score of the tested data

In [31]

```
from sklearn.metrics import accuracy_score

print(accuracy_score(y_test, y_predict_test)*100)
```

Input 32: Code to generate the accuracy score trained of the dataset

In [32]

```
from sklearn.metrics import confusion_matrix, accuracy_score

cm = confusion_matrix(y_train,y_predict_train)

print(cm)

accuracy_score(y_train, y_predict_train)*100
```

5. Retest of the Model

The RFA is once again used for retesting the trained data. Below is the successful operation of the random forest algorithm model in classifying the class of kidney failure.

Input 33: Code to predict the class of kidney failure

In [33]

```
print(RFC.predict(X_test))
```

3. Results

The output of the machine learning algorithm (MLA) and the set of codes are presented. The output results are presented in the form of tables and graphs.

Output 2: CKD dataset

Out [2]

	Bp	Sg	Al	Sun	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn	Class
0	80.0	1.020	1.0	0.0	1.0	36.0	1.2	137.53	4.63	15.4	7800.0	5.20	1.0	1
1	50.0	1.020	4.0	0.0	1.0	18.0	0.8	137.53	4.63	11.3	6000.0	4.71	0.0	1
2	80.0	1.010	2.0	3.0	1.0	53.0	1.8	137.53	4.63	9.6	7500.0	4.71	0.0	1
3	70.0	1.005	4.0	0.0	1.0	56.0	3.8	111.00	2.50	11.2	6700.0	3.90	1.0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
397	80.0	1.020	0.0	0.0	1.0	26.0	0.6	137.00	4.40	15.8	6600.0	5.40	0.0	0
398	60.0	1.025	0.0	0.0	1.0	50.0	1.0	135.00	4.90	14.2	7200.0	5.90	0.0	0
399	80.0	1.025	0.0	0.0	1.0	18.0	1.1	141.00	3.50	15.8	6800.0	6.10	0.0	0

Figure1

The "output 3" presents the data description of the CKD dataset. The parameters of the data description include mean, standard deviation (std), minimum (min), and maximum (max) value, and percentiles (25%, 50%, & 75%).

Output 3: Table of descriptive statistics of the chronic kidney disease dataset

Out [3]

```
                Bp          Sg          Al          Su       Class
count  400.000000  400.000000  400.000000  400.000000  400.000000
mean    76.455000    1.017712    1.015000    0.395000    0.625000
std     13.476536    0.005434    1.272329    1.040038    0.484729
```

```
min     50.000000    1.005000    0.000000    0.000000    0.000000

25%     70.000000    1.015000    0.000000    0.000000    0.000000

50%     78.000000    1.020000    1.000000    0.000000    1.000000

75%     80.000000    1.020000    2.000000    0.000000    1.000000

max    180.000000    1.025000    5.000000    5.000000    1.000000
```

The information of the CKD dataset is depicted below in “output 4”. The following contains the cells' information, entries, and the type of data in each cell of the table.

Output 4: Generated information of the chronic kidney disease dataset

Out [4]

```
<class 'pandas.core.frame.DataFrame'>

RangeIndex: 400 entries, 0 to 399

Data columns (total 14 columns):

 #   Column  Non-Null Count  Dtype

---  ------  --------------  -----

 0   Bp      400 non-null    float64

 1   Sg      400 non-null    float64

 2   Al      400 non-null    float64

 3   Su      400 non-null    float64

    ..      ... ...  ...    ......

 9   Hemo    400 non-null    float64

 10  Wbcc    400 non-null    float64

 11  Rbcc    400 non-null    float64

 12  Htn     400 non-null    float64
```

```
 13  Class   400 non-null    int64

dtypes: float64(13), int64(1)

memory usage: 43.9 KB
```

The output generated below (output 5) shows the data visualization of CKD dataset which illustrates the trend of various classes of kidney failure.

Output 5: Visualization of the classes of kidney failure

Out [5]

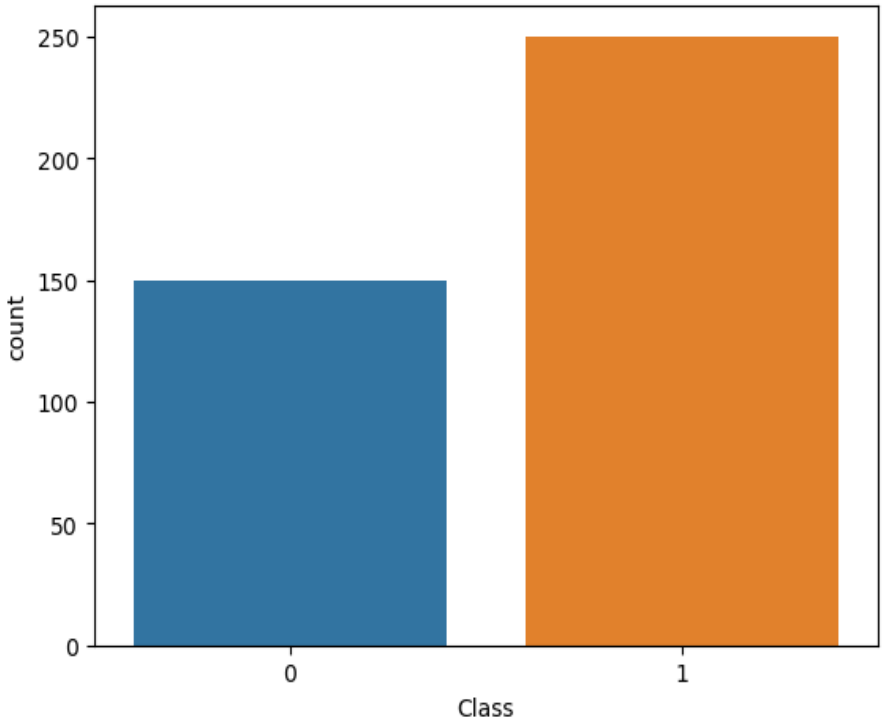

Figure 2: Illustration of the classes of kidney failure

Outputs 6 and 7 depict the isolation of other parameters other than the outcome of kidney failure diagnostics.

Output 6: Isolation of the class column of the CKD dataset

Out [6]

	Bp	Sg	Al	Su	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn
0	80.0	1.020	1.0	0.0	1.0	36.0	1.2	137.53	4.63	15.4	7800.0	5.20	1.0

	Bp	Sg	Al	Su	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn
1	50.0	1.020	4.0	0.0	1.0	18.0	0.8	137.53	4.63	11.3	6000.0	4.71	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	80.0	1.020	0.0	0.0	1.0	49.0	0.5	150.00	4.90	15.7	6700.0	4.90	0.0
399	80.0	1.025	0.0	0.0	1.0	18.0	1.1	141.00	3.50	15.8	6800.0	6.10	0.0

400 rows × 13 columns

Output 7: Summary of transformed data of the class of kidney failure

Out [7]

```
0    1
1    1
2    1
3    1
4    1
    ..
395  0
396  0
397  0
398  0
399  0
Name: Class, Length: 400, dtype: int64
```

The data training and testing involved the use of machine learning algorithms. The class or the kidney failure diagnostics was trained as the dependent variable, whereas other parameters were trained as the independent variables.

Output 10: Trained data of the independent variables

Out [10]

	Bp	Sg	Al	Su	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn
157	70.0	1.025	3.0	0.0	1.0	42.0	1.7	136.00	4.70	12.6	7900.0	3.90	1.0
109	70.0	1.020	1.0	0.0	1.0	50.1	1.9	137.53	4.63	11.7	8406.0	4.71	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
348	80.0	1.020	0.0	0.0	1.0	19.0	0.5	147.00	3.50	13.6	7300.0	6.40	0.0
102	60.0	1.010	0.0	0.0	1.0	32.0	2.1	141.00	4.20	13.9	7000.0	4.71	0.0

280 rows x 13 columns

Figure 3

Output 11: Trained data of the dependent variable

Out [11]

157 1

109 1

17 1

347 0

24 1

..

71 1

106 1

270 0

348 0

102 1

Name: Class, Length: 280, dtype: int64

All other parameters are trained as the independent variables. 70% of the dataset was tested.

Output 12: Tested data of the independent variables

Out [12]

	Bp	Sg	Al	Su	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn
209	70.0	1.020	0.0	0.0	1.0	57.0	3.07	137.53	4.63	11.50	6900.0	4.71	0.0
280	80.0	1.020	1.0	0.0	1.0	33.0	0.90	144.00	4.50	13.30	8100.0	5.20	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
285	70.0	1.020	0.0	0.0	1.0	19.0	0.70	135.00	3.90	16.00	5300.0	5.90	0.0
305	80.0	1.020	0.0	0.0	1.0	25.0	0.80	138.00	5.00	17.10	9100.0	5.20	0.0
281	80.0	1.025	0.0	0.0	1.0	50.0	1.20	147.00	5.00	15.50	9100.0	6.00	0.0

120 rows × 13 columns

Figure 4

Output 13: Tested data of the dependent variable

Out [13]

```
209    1
280    0
33     1
210    1
93     1
     ..
60     1
79     1
285    0
305    0
```

```
281    0

Name: Class, Length: 120, dtype: int64
```

Random forest MLA is used to model the CKD dataset. The result from the performance metrics is detailed for the trained data. The precision (Output 16) for the prediction of CKD is 100%, and NOTCKD is 100%, which indicates that all predicted instances are positives. The recall (Output 16) for the prediction of CKD is 100%, and NOTCKD is 100%, which indicates that for both classes, all actual positive instances are correctly identified. The F1 score (Output 16) for the prediction of CKD is 100%, and NOTCKD is 100%, which indicates a perfect precision and recall for both classes. Also, the accuracy (Output 16) for the prediction of the class of chronic kidney failure is 100%, which indicates that the Random Forest Algorithm has achieved a perfect accuracy of 100%.

Output 15: Prediction of the trained data

Out [15]

```
array([1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1,
       0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1,
       1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0,
       1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0,
       1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1], dtype=int64)
```

Output 16: Performance metrics of the trained data

Out [16]

	precision	recall	f1-score	support
0	1.00	1.00	1.00	106
1	1.00	1.00	1.00	174
accuracy			1.00	280
macro avg	1.00	1.00	1.00	280
weighted avg	1.00	1.00	1.00	280

The result from the performance metrics is detailed for the tested data. The precision (Output 18) for the prediction of CKD is 100%, and NOTCKD is 100%, which indicates that all predicted instances are positives. The recall (Output 18) for the prediction of CKD is 100%, and NOTCKD is 100%, which indicates that for both classes, all actual positive instances are correctly identified. The F1 score (Output 18) for the prediction of CKD is 100%, and NOTCKD is 100%, which indicates a perfect precision and recall for both classes. Also, the accuracy (Output 18) for the prediction of the class of chronic kidney failure is 100%, which indicates that the Random Forest Algorithm has achieved a perfect accuracy of 100%. The confusion matrix (Output 17) also confirmed the precision and accuracy of the RF algorithm, which depicts true negatives (44), true positives (76), and false negatives and positives are zero.

Output 17: Prediction of the confusion matrix of the tested data

Out [17]

<Axes: >

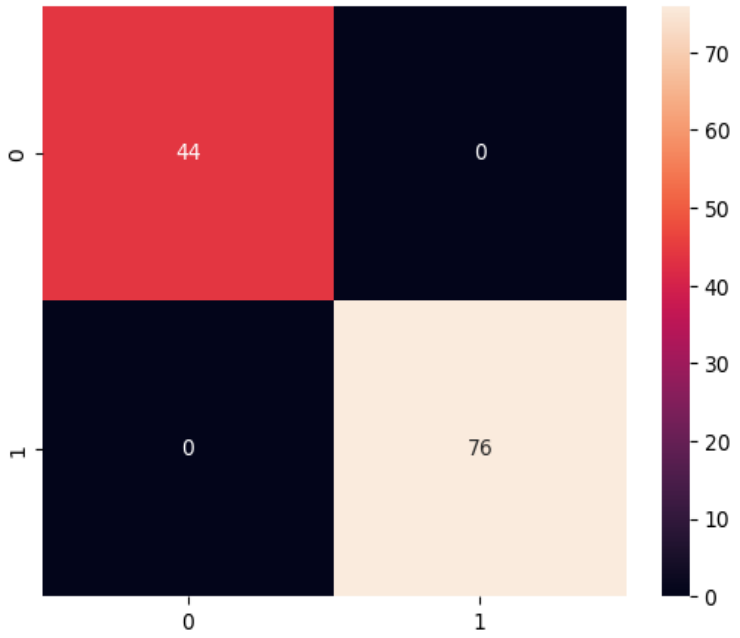

Figure 5: Confusion matrix chart of the tested data by RFA

Output 18: Performance metrics of the tested data

Out [18]

	precision	recall	f1-score	support
0	1.00	1.00	1.00	44
1	1.00	1.00	1.00	76
accuracy			1.00	120
macro avg	1.00	1.00	1.00	120
weighted avg	1.00	1.00	1.00	120

Naïve-Bayes MLA is used to model the CKD dataset. The result from the performance metrics is detailed for the trained data. The precision (Output 21) for the prediction of CKD is 100%, and NOTCKD is 85%, which indicates that all predicted instances are positives. The recall (Output 21) for the prediction of CKD is 89%, and NOTCKD is 100%, which indicates that for both classes, all actual positive instances are correctly identified. The F1 score (Output 21) for the prediction of CKD is 94%, and NOTCKD is 92%, which indicates a perfect precision and

recall for both classes. Also, the accuracy (Output 21) for the prediction of the class of chronic kidney failure is 93%, which indicates that the Naïve-Bayes Algorithm has achieved a perfect accuracy of 100%.

Output 20: Prediction of the trained data

Out [20]

```
array([1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1,
       0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1,
       0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1,
       1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0,
       1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0,
       1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1,
       1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0], dtype=int64)
```

Output 21: Performance metrics of the trained data

Out [21]

	precision	recall	f1-score	support
0	0.85	1.00	0.92	106
1	1.00	0.89	0.94	174

accuracy			0.93	280
macro avg	0.92	0.95	0.93	280
weighted avg	0.94	0.93	0.93	280

The result from the performance metrics is detailed for the tested data. The precision (Output 24) for predicting CKD is 100%, and for NOTCKD it is 85%, indicating that all predicted instances are positives. The recall (Output 24) for predicting CKD is 89%, and for NOTCKD it is 100%, indicating that for both classes, all actual positive instances are correctly identified. The F1 score (Output 24) for predicting CKD is 94%, and for NOTCKD is 92%, indicating perfect precision and recall for both classes. Also, the accuracy (Output 24) for the prediction of the class of chronic kidney failure is 93%, which indicates that the Naïve-Bayes Algorithm has achieved a perfect accuracy of 93%. The confusion matrix (Output 23) also confirmed the precision and accuracy of the NB algorithm, which depicts true negatives (44), true positives (71), false positives (5), and false negative is zero.

Output 23: Computation of the confusion matrix of the tested data

Out [23]

```
<Axes: >
```

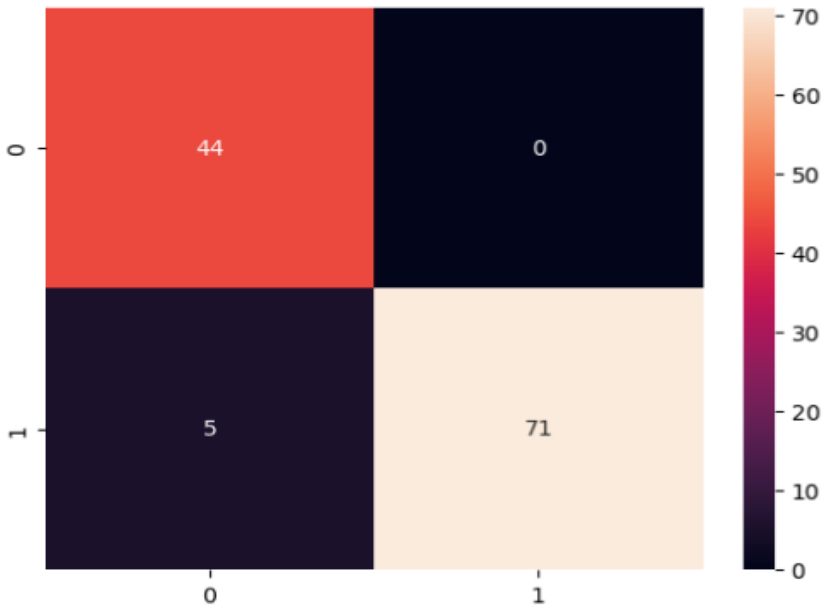

Figure 6: Confusion matrix chart of the tested data by NBC

Output 24: Performance metrics of the tested data

Out [24]

	precision	recall	f1-score	support
0	0.85	1.00	0.92	106
1	1.00	0.89	0.94	174
accuracy			0.93	280
macro avg	0.92	0.95	0.93	280
weighted avg	0.94	0.93	0.93	280

The SVM algorithm is used in the prediction of the trained and tested data. The accuracy of the tested data (Output 27) for the prediction of the class of chronic kidney failure is 95.83%, and the accuracy of the trained data (Output 28) is 93.2%.

Output 26: Prediction of the tested data

Out [26]

```
[1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 1 0 1 0 1 1 1 1
 0 1 1 0 1 1 1 1 0 1 1 1 1 0 0 0 1 0 1 1 1 1 1 1 0 0 1 1 1 0 0 0 1 0 1 1 1
 0 1 1 0 1 0 1 1 1 0 0 1 1 0 1 1 1 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0
 0 0 0 0 1 1 0 0 0]
```

Output 27: Accuracy score of the tested data

out [27]

```
95.83333333333334
```

Output 28: Accuracy score of the trained data

Out [28]

```
[[106   0]

 [ 19 155]]

93.21428571428572
```

The kernel SVM algorithm is used in the prediction of the trained and tested data. The accuracy of the tested data (Output 31) for the prediction of the class of chronic kidney failure is 95.83%, and the accuracy of the trained data (Output 32) is 93.2%.

Output 30: Prediction of the tested data

Out [30]

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

 1 1 1 1 1 1 1 1 1]
```

Output 31: Accuracy test of the tested data

Out [31]

```
95.83333333333334
```

Output 32: Accuracy of the trained dataset

Out [32]

```
[[106   0]

 [ 19 155]]

93.21428571428572
```

Here, the accuracy of the RF algorithm is retested.

Output 33: Accuracy test of the class of kidney failure

Out [33]

```
[1 0 1 1 1 1 0 1 0 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 0 1 1 0 1 0 1 0 1 1 1 1

 0 1 1 0 1 1 1 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 0 0 1 1 1 0 0 0 1 0 1 1 1

 0 1 1 0 1 0 1 1 1 0 0 1 1 0 1 1 1 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1

 0 0 0 0 1 1 0 0 0]
```

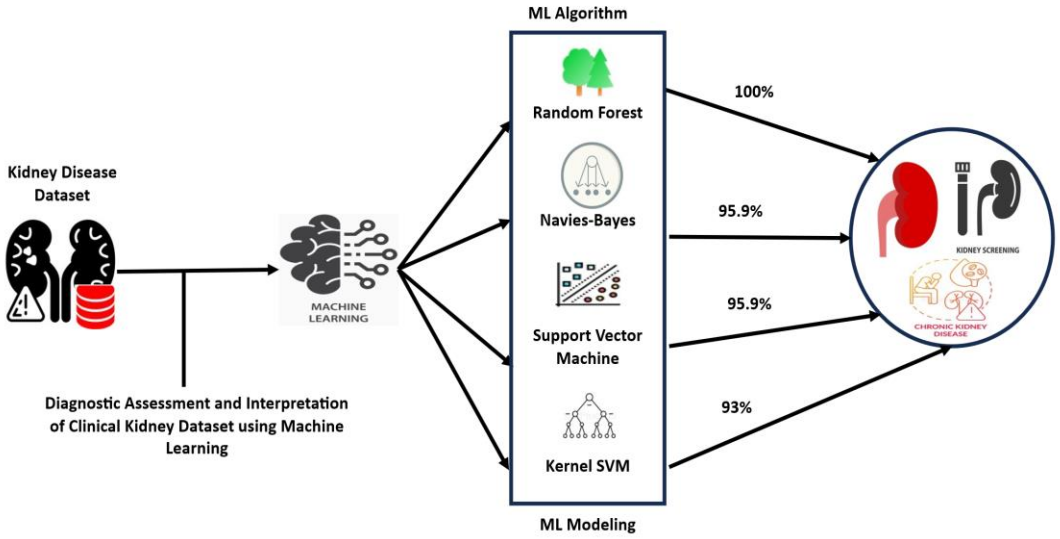

Figure 7: Graphical result of the ML modeling of the kidney disease dataset

4. Discussions

The application of machine learning (ML) in the diagnostic assessment and interpretation of clinical kidney datasets represents an advancement in clinical diagnostics and healthcare analytics. Alternatively, ML algorithms provide an opportunity to enhance diagnostic accuracy, efficiency, and predictive capability by learning complex patterns within large datasets that may not be immediately apparent to clinicians [18]. This study highlights the potential of ML to enhance diagnostic accuracy and clinical decision-making. The dataset incorporated key predictors of kidney disease, including diabetes, high blood pressure, and other clinical variables, enabling the algorithms to classify patients as either having chronic kidney disease (CKD) or not.

The findings from the study, using different machine learning algorithms for modelling and predicting kidney or clinical datasets, have contributed to the application of artificial intelligence, especially machine learning, in the contemporary world. Based on the machine algorithms, namely random forest (RF), Naive-Bayes (NB), support vector machine (SVM), and kernel SVM (KSVM), the kidney failure predictor dataset, which comprises various predictors such as diabetes, high blood pressure, etc., has been achieved and established at various accuracies. RF

has the best accuracy so far, the classification of kidney failure as either chronic kidney disease (CKD) or not CKD.

The findings reveal notable differences in performance across the algorithms. Random forest achieved the highest accuracy, with both training and testing accuracies reaching 100%. This suggests that RF is particularly effective in handling complex, multidimensional datasets and capturing non-linear relationships among predictors. In comparison, Naive Bayes achieved a training accuracy of 93.0% and a testing accuracy of 93%, while SVM and KSVM recorded training accuracies of 93.2% and testing accuracies of 95.83%. These results indicate that while all algorithms performed well, RF consistently outperformed the others in terms of predictive reliability.

Additionally, this study offers a vital application in clinical studies and medical physics. This study contributes to medical diagnostics, prognostic assessment, and AI-enabled tools in medicine, as established by Iqbal, and his colleagues [19] who mentioned that AI-based assistance to pathologists and physicians are substantial way forward towards prediction for disease risk, diagnosis, prognosis, and treatments. Also, this research is comparable with Liu, and his colleagues [20] that were able to apply to the preceding research of random forest, a machine learning prediction model, to create a simulation called the plasmode simulation. They made use of random forest (RF) algorithms to identify and indicate relevant predictor variables out of a large set of data, which enables the use of external information for variable selection in order to enhance model interpretability and variable selection accuracy, thereby improving prediction quality. Their research examined the applicability of external information from prior variable selection studies that used traditional statistical modeling approaches. 200,000 individuals from pharmacoepidemiologic studies were used in the research. Beyond overall accuracy, evaluation metrics such as recall, precision, and F1 score were used to validate the validity of the models. These measures are critical in clinical applications, as they ensure that predictions are not only correct but also sensitive to identifying true cases of CKD and specific enough to avoid misclassification [21]. The superior performance of RF across these metrics underscores its suitability for clinical diagnostic tasks, where both sensitivity and specificity are essential.

Clinically, these findings demonstrate the potential of ML models to support early detection and classification of kidney disease, thereby enabling timely interventions and personalized treatment strategies. The ability of ML algorithms to process large datasets and uncover hidden patterns offers significant advantages over traditional diagnostic methods. However, challenges remain, including the need for larger and more diverse datasets to avoid bias, the importance of model interpretability for clinician trust, and the ethical considerations surrounding patient data privacy.

5. Conclusion

So far, the ML algorithms have demonstrated their capability and accuracy in classifying the kidney disease dataset as either CKD or not-CKD. In general, the training accuracy of random forest is 100% has the most preferred result, while NBA is just 93.0%, SVM is 93.2%, and KSVM is also 93.2%. Therefore, the testing accuracies for the machine learning algorithms are RF is 100%, NB is 93.0%, SVM is 95.83%, and KSVM is 95.83%. In conclusion, the random forest machine learning algorithm has the best testing accuracy out of the four machine learning algorithms used to build the model. Therefore, random forest is highly recommended for

classification analysis modelling.

6. Limitations of the Research

1. Data quality and completeness: Missing values, inconsistent clinical records, and potential measurement errors in laboratory parameters could affect model accuracy and reliability.
2. Feature selection bias: The choice of input features may introduce bias. Important clinical variables not included in the dataset could influence kidney disease diagnosis and progression but remain unaccounted for.
3. Clinical integration challenges: Translating machine learning predictions into real-world clinical workflows requires careful integration with electronic health records, clinician training, and regulatory approval, which were beyond the scope of this study.
4. Ethical and privacy concerns: Patient data used in training models raises concerns about privacy, data security, and compliance with health regulations. These issues were not fully addressed in this research.

7. Recommendations

The following recommendations are outlined for future research and further studies. These recommendations are:

1. The study should implement real-time model monitoring to track performance and identify any anomalies or shifts in patient data, as regular monitoring ensures timely updates and maintains model effectiveness.

2. Application of ensemble methods that combine multiple machine learning models for more robust predictions. This approach can enhance accuracy and provide more reliable early detection of kidney failure.

3. Use of interpretable machine learning techniques that provide clear explanations for model predictions because transparent models foster trust among clinicians and encourage adoption in clinical decision-making workflows.

References

[1] P. Adigun, T. Oyekanmi, and A. Adeniyi, "Simulation Prediction of Background Radiation Using Machine Learning," *ASRJETS*, vol. 101, no. 1, pp. 71–96, 02/03 2025. [Online]. Available: https://asrjetsjournal.org/American_Scientific_Journal/article/view/11432.

[2] P. O. Adigun, A. A. Adeniyi, and T. T. Oyekanmi, "Detection and Interpretation of X-Ray Scans for the Presence of Pneumonia Using Convolutional Neural Network," *American Academic Scientific Research Journal for Engineering, Technology, and Sciences*, Original vol. 101, no. 1, pp. 97–108, 2025. [Online]. Available: https://core.ac.uk/download/pdf/640473726.pdf. Yes.

[3] T. Oyekanmi, P. Adigun, A. Adeniyi, and N. A. Azeez, "Deep Learning-Based Diagnosis of Brain Cancer Using Convolutional Neural Networks On MRI Scans: A Comparative Study of Model Architectures and Tumor Classification Accuracy," *American Scientific Research Journal for Engineering, Technology, and Sciences,* vol. 103, pp. 147–163, 04/10 2025. [Online]. Available: https://asrjetsjournal.org/American_Scientific_Journal/article/view/12059.

[4] A. Ayodeji Adedotun, O. Tobi Titus, O. K. Vitoria, A. Peter Oluwasayo, and A. Nelson Abimbola Ayuba, "Applications of Artificial Intelligence Models in Teletherapy: A Review of Efficacy, and Ethical Implications," *American Scientific Research Journal for Engineering, Technology, and Sciences,* vol. 103, no. 1, pp. 373–391, %11/%26 2025. [Online]. Available: https://asrjetsjournal.org/American_Scientific_Journal/article/view/12139.

[5] O. Tobi Titus, A. Peter Oluwasayo, and A. Ayodeji Adedotun, "Design and Evaluation of a Convolutional Neural Network Model for Automated Detection of Diabetic Retinopathy Using Retinal Fundus Photographs," *American Scientific Research Journal for Engineering, Technology, and Sciences,* vol. 103, no. 1, pp. 313–329, %11/%01 2025. [Online]. Available: https://asrjetsjournal.org/American_Scientific_Journal/article/view/12111.

[6] T. J. Saleem and M. A. Chishti, "Exploring the applications of machine learning in healthcare," *International Journal of Sensors Wireless Communications and Control,* vol. 10, no. 4, pp. 458–472, 2020.

[7] P.-H. C. Chen, Y. Liu, and L. Peng, "How to develop machine learning models for healthcare," *Nature materials,* vol. 18, no. 5, pp. 410–414, 2019.

[8] I. Scott, S. Carter, and E. Coiera, "Clinician checklist for assessing suitability of machine learning applications in healthcare," *BMJ Health & Care Informatics,* vol. 28, no. 1, 2021.

[9] J. Wiens and E. S. Shenoy, "Machine learning for healthcare: on the verge of a major shift in healthcare epidemiology," *Clinical Infectious Diseases,* vol. 66, no. 1, pp. 149–153, 2018.

[10] Cleveland Clinic, "Kidney Failure," in *Cleveland Clinic,* ed, 2023.

[11] A. Biggers, "Everything you need to know about kidney failure," in *Healthline,* ed, 2022.

[12] J.-C. Lv and L.-X. Zhang, "Prevalence and Disease Burden of Chronic Kidney Disease," in *Renal Fibrosis: Mechanisms and Therapies*, B.-C. Liu, H.-Y. Lan, and L.-L. Lv Eds. Singapore: Springer Singapore, 2019, pp. 3–15.

[13] M. T. Liao, C. C. Sung, K. C. Hung, C. C. Wu, L. Lo, and K. C. Lu, "Insulin resistance in patients with chronic kidney disease," (in eng), *J Biomed Biotechnol,* vol. 2012, p. 691369, 2012, doi: 10.1155/2012/691369.

[14] NIDDKD, "Kidney Failure," in *National Institute of Diabetes and Digestive and Kidney Diseases*, ed, 2023.

[15] A. Singh. "Kidney Disease Dataset." Kaggle. https://www.kaggle.com/datasets/akshayksingh/kidney-disease-dataset/data (accessed 20th May 2023, 2023).

[16] N. Yadav. "Chronic Kidney Disease Prediction (98% Accuracy)." Kaggle. https://www.kaggle.com/code/niteshyadav3103/chronic-kidney-disease-prediction-98-accuracy (accessed 20th December 2025, 2025).

[17] D. Kuhlman, "A Python Book: Beginning Python, Adanced." [Online]. Available: https://www.davekuhlman.org/python_book_01.pdf

[18] J. A. Nichols, H. W. Herbert Chan, and M. A. B. Baker, "Machine learning: applications of artificial intelligence to imaging and diagnosis," *Biophysical Reviews,* vol. 11, no. 1, pp. 111–118, 2019/02/01 2019, doi: 10.1007/s12551-018-0449-9.

[19] M. J. Iqbal *et al.*, "Clinical applications of artificial intelligence and machine learning in cancer diagnosis: looking into the future," *Cancer Cell International,* vol. 21, no. 1, p. 270, 2021/05/21 2021, doi: 10.1186/s12935-021-01981-1.

[20] Y. Liu *et al.*, "A photonic integrated circuit–based erbium-doped amplifier," *Science,* vol. 376, no. 6599, pp. 1309–1313, 2022.

[21] F. Furizal, A. Ma'arif, and D. Rifaldi, "Application of Machine Learning in Healthcare and Medicine: A Review," *Journal of Robotics and Control (JRC),* vol. 4, p. 2023, 09/30 2023, doi: 10.18196/jrc.v4i5.19640.