CSCI417

Project: Chronic Kidney Disease Detection

By:

Esraa Abdallah Abdelwahed	**19105371**
Elzahraa Saied Abdullah	**19106429**
Maram Reda Zoughieb	**19105793**
Rana Rizk Mahrous	**19105575**

Submitted in partial fulfillment of the requirements for the Machine Intelligence Project

Jan 08, 2023

Table of Contents

I: Abstract

In this project, we have covered chronic kidney disease (CKD). CKD is among the significant contributor to morbidity and mortality from non-communicable diseases that can affected high percentage global population. Early and accurate detection of the stages of CKD is believed to be vital to minimize impacts of patient’s health complications such as hemoglobin, hypertension, anemia (low blood count), diabetes_mellitus, coronary_artery_disease, appetite, and many other health complications. Various researches have been carried out using machine learning techniques on the detection of CKD at the premature stage. In this project, two types of data have been used, tubular and image data. The prediction models used on these data include Decision Tree (DT), Support Vector Machine (SVM), and k-nearest neighbors (KNN) for tubular data and a convolutional neural network (CNN), for data images. Evaluation of the models was done using a convolution matrix. Data ensembling has been done for getting the mean of all models’ accuracy. Furthermore, the next parts will go into greater details.

II: Problem Statement

Chronic kidney disease (CKD) is a major cause of morbidity and mortality from noncommunicable illnesses, affecting 10-15% of the global population. Early and correct diagnosis of CKD stages is thought to be critical for minimizing the effects of patient health issues. Several studies on the diagnosis of CKD at an early stage have been conducted utilizing machine learning approaches. Their primary focus was not on predicting certain stages. Both binary and multi classification for stage prediction were used in this project.

Keywords: Chronic Kidney Disease (CKD), Machine Learning, convolution matrix Decision Tree (DT), Support Vector Machine (SVM), k-nearest neighbors (KNN), convolutional neural network (CNN)

II: Introduction

Chronic kidney disease (CKD) is a non-communicable disease that has significantly increased patient admission rates, morbidity, and death throughout the world. It is rapidly spreading and rising to the top of the list of leading causes of mortality worldwide. According to a survey from 1990 to 2013, the annual global death toll from CKD increased by 90%, ranking it as the 13th most common cause of death worldwide. There are 850 million people throughout the world who are expected to have renal disease due to various factors. According to the report of world kidney day of 2019, at least 2.4 million people die every year due to kidney-related disease. it's currently the sixth fastest-growing cause of death worldwide, and it is becoming a difficult public health issue. As kidney failure is a worldwide issue, the Renal Replacement Therapy (RRT) cost for total kidney failure is very expensive. Most impoverished nations do not offer treatment. As a result, renal failure and associated complications are extremely difficult to manage in developing nations due to the lack of facilities, specialists, and expensive treatment options. Therefore, it is crucial to find CKD early in order to save costs and enhance the effectiveness of treatments.

Engineers and medical researchers are trying to develop machine-learning algorithms and models that can detect chronic kidney disease at an early stage. The issue is that the size and complexity of the data produced by the health industry make data analysis challenging. However, by applying data mining technology, we can transform this data into a format that can later be used by machine learning algorithms. A combination of age and existing medical conditions can be used to assess the severity of kidney disease but requires more accurate information about the risk to the kidney is required to make clinical decisions about diagnosis and treatment.

When it comes to making predictions based on previous data using classification and regression techniques, machine learning captures a significant portion of artificial intelligence. Based on various data sets, the use of machine learning techniques to predict CKD has been explored. Among them, the datasets have been collected from **Kaggle** and **HiRID.** First dataset type used is tubular and has 400 instances with 26 attributes. Second dataset type is images were divided into 4 categories Cyst, Normal, Tumor and Stone. Similar to most of the related work, this work considers the mentioned benchmark dataset. In this project, Decision Tree (DT), Support Vector Machine (SVM), k-nearest neighbors (KNN) for tubular data, a convolutional neural network (CNN), and Support Vector Machine from CNN for data images have been used to detect CKD. Using the convolution matrix, Accuracy, recall, and F1 score have been obtained for each model. In addition, the ensemble method has been used to get the mean of all models' accuracy to get the optimal one.

III: Related Work

Different machine-learning techniques have been used for effective classification of chronic kidney disease from patients' data.

Alsuhibany et al. presented ensemble of deep learning based clinical decision support systems (EDL-CDSS) for CKD diagnosis in the IoT environment. The presented technique involves Adaptive Synthetic (ADASYN) technique for outlier detection process and employed ensemble of three models, namely, deep belief network (DBN), kernel extreme learning machine (KELM), and convolutional neural network with gated recurrent unit (CNN-GRU).

Salekin and Stankovic did evaluation of classifiers such as K-NN, RF and ANN on a dataset of 400. Wrapper feature selection were implemented, and five features were selected for model construction in the study. The highest classification accuracy is 98% by RF and an RMSE of 0.11. S.

Tekale et al. worked on "Prediction of Chronic Kidney Disease Using Machine Learning Algorithm" with a dataset consists of 400 instances and 14 features. They have used decision tree and support vector machine. The dataset has been preprocessed and the number of features has been reduced from 25 to 14. SVM is stated as a better model with an accuracy of 96.75%.

Yashfi proposed to predict the risk of CKD using machine learning algorithms by analyzing the data of CKD patients. Random Forest and Artificial Neural Network have been used. They have extracted 20 out of 25 features and applied RF and ANN. RF has been identified with the highest accuracy of 97.12%.

Rady and Anwar carried out the comparison of Probabilistic Neural Networks (PNN), Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Radial Basis Function (RBF) algorithms to predict kidney disease stages. the researchers conducted their research on a small size dataset and few numbers of features. The result of this paper shows that the Probabilistic Neural Networks algorithm gives the highest overall classification accuracy percentage of 96.7%.

Poonia et al. employed Various machine learning algorithms, including K-nearest neighbors algorithm (KNN), Artificial neural networks (ANN), support vector machines (SVM), Naive Bayes (NB), and Logistic Regression as well as Re-cursive Feature Elimination (RFE) and Chi-Square test feature-selection techniques. A publicly available dataset of healthy and kidney disease patients was used to build and analyze prediction models.

Priyanka et al. carried out chronic kidney disease prediction through naive bayes. They have tested using other algorithms such as KNN (K-Nearest Neighbor Algorithm), SVM (Support Vector Machines), Decision tree, and ANN (Artificial Neural Network) and they have got Naïve Bayes with better accuracy of 94.6% when compared to other algorithms.

The above reviews indicate that several studies have been conducted on chronic kidney disease prediction using machine-learning techniques. There are various parameters that play important role in improving model performance like dataset size, quality of dataset, and the time dataset collected.

Table1: Summary of some related works

No.	Author	Technique Applied	Claimed Outcome	Drawback
1.	Salekin and Stankovic	K-NN, RF, and NN, Wrapper approach and Embedded approach	Detection F1-score of RF 99.8	Small dataset size with missing values was used; Severity level prediction was not included
2.	Tekale et al.	DT and SVM	DT and SVM with an accuracy of 91.75 and 96.75 respectively	Dataset size need to increased, Severity level prediction was not included. Only to classifiers result compared
3.	Priyanka et al.	NB, KNN, SVM, DT, and ANN. NB.	NB, KNN, SVM, DT, and ANN. NB accuracy is 94.6%	Small size dataset. No stages prediction. Feature extraction was not carried out and classification accuracy needs improvement
4.	Yashfi	RF and ANN	RF and ANN with an accuracy of 97.12% and 94.5%	Small size dataset and no stages prediction

IV. Methodology

IV.I: Data source and description

In this project, we have two datasets, tubular and images data. the datasets have been collected from **Kaggle** and **HiRID.** First dataset type used is tubular and has 400 instances with 26 attributes where 11 are float64, 1 is int64, and 14 object. The attributes in the dataset include age, blood_pressure, specific_gravity, albumin, sugar, red_blood_cells, pus_cell,pus_cell_clumps,bacteria, blood_glucose_random, blood_urea, serum_creatinine, sodium,potassium, hemoglobin, packed_cell_volume, white_blood_cell_count, red_blood_cell_count,hypertension, diabetes_mellitus, coronary_artery_disease, appetite, peda_edema, anemia, class. Second dataset type is images containend 12,446 unique data within it in which the cyst contains 3,709, normal 5,077, stone 1,377, and tumor 2,283.

IV.II: Data Preprocessing

Preprocessing the data before it is fed into classifiers is vital part of developing machine-learning model. The dataset for this project contains missing values that needs to be handled appropriately.

Converting necessary columns to numerical type: This data has categorical columns, packed_cell_volume, white_blood_cell_count, and red_blood_cell_count, which should be converted to numerical as it would be useful after that for other processing operations.

Replacement incorrect values: These data columns, diabetes_mellitus, coronary_artery_disease, and class, have incorrect values which are replaced with correct values.

Handling Missing Values: data is not always available (or missed) due to equipment malfunction, inconsistent with other recorded data, and thus deleted. In this project, handling the missing data is done by using random sampling for higher null values and mean/mode sampling for lower null values.

Feature Encoding: In this data, since all columns have two classes, we can use the label encoder that performs the conversion of these labels of all categorical data into a numerical format.

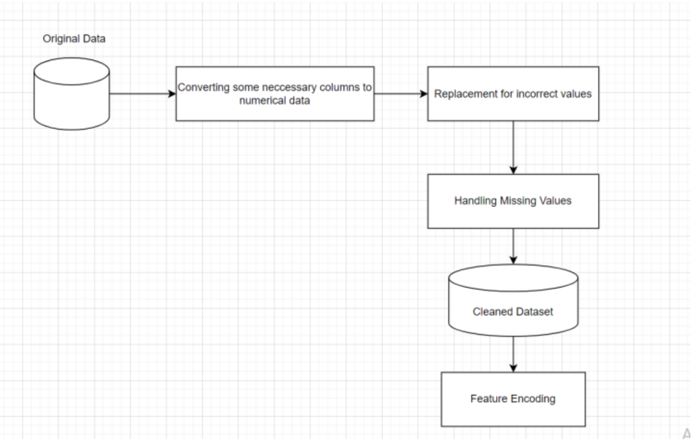

Original Data
Converting some neccessary columns to numerical data
Replacement for incorrect values
Handling Missing Values
Cleaned Dataset
Feature Encoding

Fig 1: Data Preprocessing

IV.III: Data Processing

Splitting the data into X, Y to prepare the data to get into the models as X has all columns except the last column (Class), and Y has the class column. Splitting the x and y into training and testing data (X_train, X_test, y_train, y_test) with the test_size = 0.2 and random sample = 42.

IV.IIII: Models Training

In this work, 5 classification models were considered in training on two types of datasets. The models have been used are Decision Tree Classifier (DT), Support Vector Machine (SVM) k-Nearest Neighbors (KNN) classification for tubular data , and Convolutional Neural Network (CNN), Support Vector Machine from CNN for data images. Tubular dataset is divided into 80% for training and 20% for testing, and images data is divided into 80% for training, 0.10% for validation and 0.10% for testing.

a. Decision Tree classifier

Decision tree is a supervised learning-based predictive modeling tool. A decision Tree solves the problem of machine learning by transforming the data into a tree representation through sorted feature values. To predict a class label for a record in decision trees, we need to start from its root till the leaf nodes. In a decision tree, each leaf node represents a class label for the instances that belong, and each node represents features in an instance to be classified. As a predictive model that maps observations about an item to determine the target value of instances, this model splits the dataset based on the condition using a tree structure. Decision Trees are a popular choice as they are easy to understand since they are in visual form. Moreover, decision trees don’t require data normalization as they process numerical and categorical data that don’t have to be transformed with other methods.

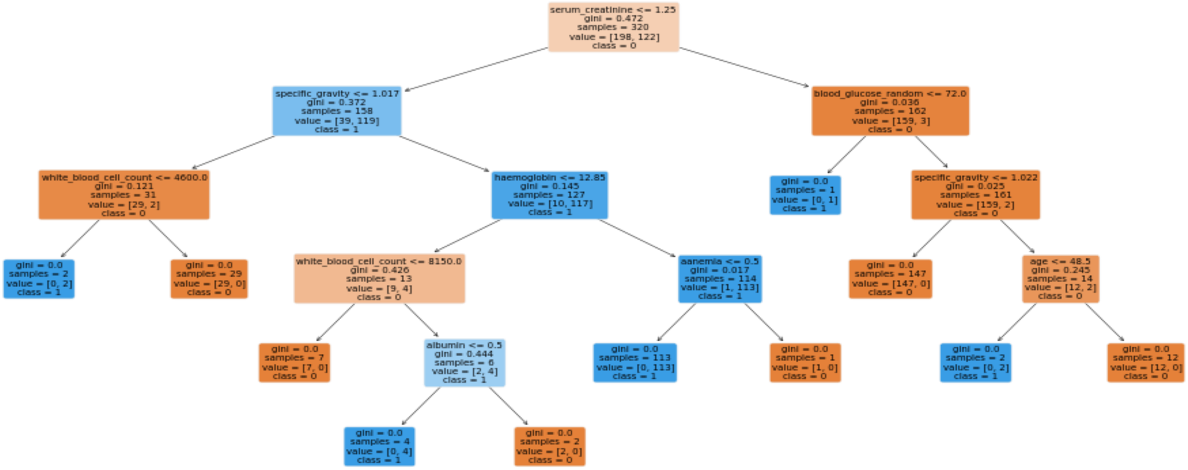


Fig2: Decision making in binary class of chronic kidney disease

b. Support Vector Machine Classifier

Support Vector Machine is one of the prominent and convenient supervised machine-learning algorithm that can be used for classification in this project, for learning and prediction. an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a no probabilistic binary linear classifier. SVM works by mapping data to a high-dimensional feature space so that data points can be classified, even when the data are not otherwise linearly separable. SVMs have been mainly proposed to deal with binary classification, but nowadays many researchers have tried to apply it to multiclass classification because there is a huge amount of data to be classified.

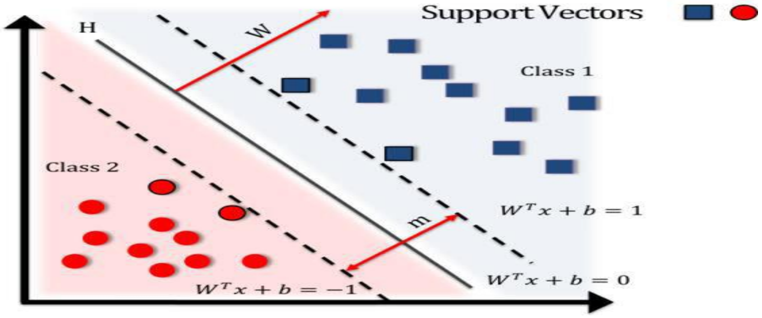


Fig3: SVM Block Diagram

c. K-Nearest Neighbor Classifier

In the project, the K-Nearest Neighbor algorithm (K-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the K closest training examples in the feature space. K-NN is a type of instance-based learning. In K-NN Classification, the output is a class membership. Classification is done by a majority vote of neighbors. The limitation of the K-NN algorithm is it's sensitive to the local configuration of the data. The process of transforming the input data to a set of features is known as Feature extraction. In Feature space, extraction is taken place on raw data before applying the algorithm.

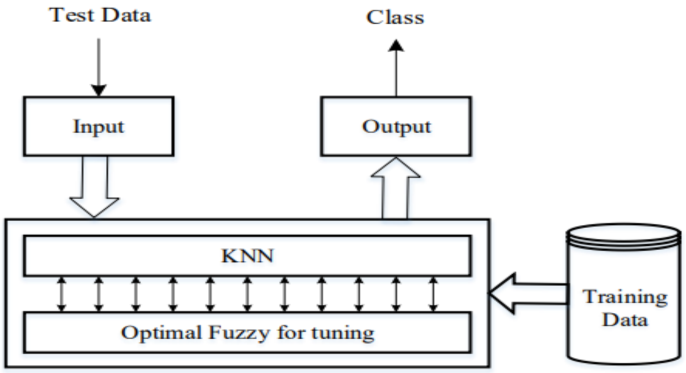


Fig4: KNN Block Diagram

d. Convolution Neural Network (CNN):

We implemented normal CNN with all the steps required :

1_ Convolution and the activation Function is Rectified linear unit.

2_ Max Pooling to get the max pixels and ignore the unnecessary features and prevent the overfitting.

3_ flattening to change the matrix of the pooled map into a vector.

4_ Fully connected layer : The features will be propagated through the layers forward and back until it finds the neurons / features that actually detect each class and discard the others where the output of the final hidden layer will be 4 to match the 4 classes.

The softmax function is also used as an activation function, and Cross Entropy is used as a loss function to calculate the error in the fully connected layers as it is used for multi class classification.

Model Fitting:

The model will be fitted with the train and validation dataset where we used 2 epochs to train the data.

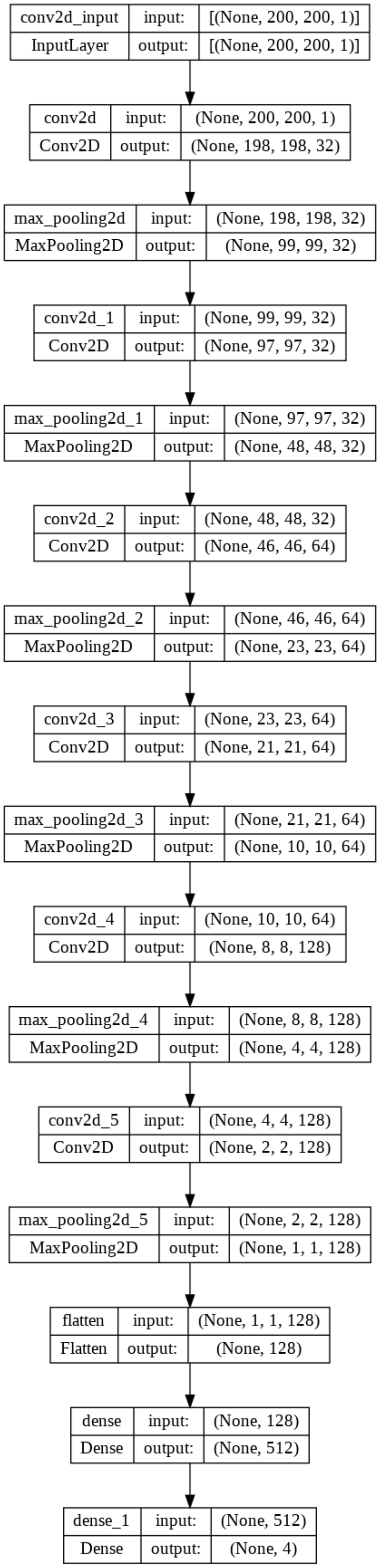
Accuracy, Precision , recall and F1 score will be calculated to measure the behavior of the model on the training and validation dataset.

Model Testing :

The model will be tested with the test dataset to get the predictions.

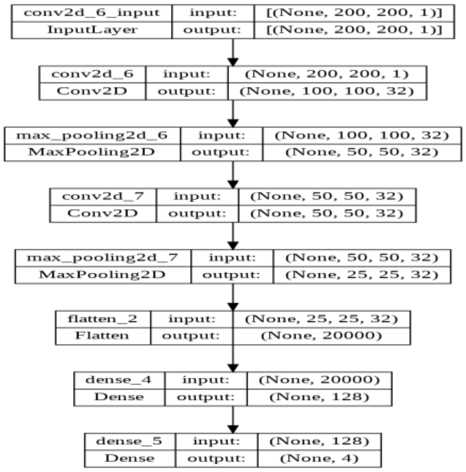
An evaluation function will be used to compare between the actual and predicted output.

Then the final accuracy of the whole model will be calculated.


conv2d_input | InputLayer | input: [(None, 200, 200, 1)] | output: [(None, 200, 200, 1)]
conv2d | Conv2D | input: (None, 200, 200, 1) | output: (None, 198, 198, 32)
max_pooling2d | MaxPooling2D | input: (None, 198, 198, 32) | output: (None, 99, 99, 32)
conv2d_1 | Conv2D | input: (None, 99, 99, 32) | output: (None, 97, 97, 32)
max_pooling2d_1 | MaxPooling2D | input: (None, 97, 97, 32) | output: (None, 48, 48, 32)
conv2d_2 | Conv2D | input: (None, 48, 48, 32) | output: (None, 46, 46, 64)
max_pooling2d_2 | MaxPooling2D | input: (None, 46, 46, 64) | output: (None, 23, 23, 64)
conv2d_3 | Conv2D | input: (None, 23, 23, 64) | output: (None, 21, 21, 64)
max_pooling2d_3 | MaxPooling2D | input: (None, 21, 21, 64) | output: (None, 10, 10, 64)
conv2d_4 | Conv2D | input: (None, 10, 10, 64) | output: (None, 8, 8, 128)
max_pooling2d_4 | MaxPooling2D | input: (None, 8, 8, 128) | output: (None, 4, 4, 128)
conv2d_5 | Conv2D | input: (None, 4, 4, 128) | output: (None, 2, 2, 128)
max_pooling2d_5 | MaxPooling2D | input: (None, 2, 2, 128) | output: (None, 1, 1, 128)
flatten | Flatten | input: (None, 1, 1, 128) | output: (None, 128)
dense | Dense | input: (None, 128) | output: (None, 512)
dense_1 | Dense | input: (None, 512) | output: (None, 4)

e. Support Vector Machine from CNN

We implemented normal CNN but in output layer, there’s a parameter called “kernel_regularizer” and inside this regularizer, we used l2 norm, passed softmax as activation function, and use “squared hinge” as loss function, and that’s what we did in the final output layer.



```
conv2d_6_input      input:   [(None, 200, 200, 1)]
InputLayer          output:  [(None, 200, 200, 1)]

conv2d_6            input:   (None, 200, 200, 1)
Conv2D              output:  (None, 100, 100, 32)

max_pooling2d_6     input:   (None, 100, 100, 32)
MaxPooling2D        output:  (None, 50, 50, 32)

conv2d_7            input:   (None, 50, 50, 32)
Conv2D              output:  (None, 50, 50, 32)

max_pooling2d_7     input:   (None, 50, 50, 32)
MaxPooling2D        output:  (None, 25, 25, 32)

flatten_2           input:   (None, 25, 25, 32)
Flatten             output:  (None, 20000)

dense_4             input:   (None, 20000)
Dense               output:  (None, 128)

dense_5             input:   (None, 128)
Dense               output:  (None, 4)
```

Fig5: CNN to SVM Diagram

f. Ensemble learning

Ensemble Learning is utilized in machine learning to obtain more accurate predictions than individual models by combining the outputs of several single classification models. Voting classification is one of ensemble methods that has been used in this project. In the case of Voting, we focus on the hard method, every individual classifier votes for a class, and the majority wins. In statistical terms, the predicted target label of the ensemble is the mode of the distribution of individually predicted labels.

- Model Averaging

We used “model averaging” to ensemble our two deep learning models. We put the predictions of the two models in one array and then sum the prediction of the first model with the second one, finally we calculated the accuracy, and we observed that the accuracy is equal to the maximum accuracy of two models

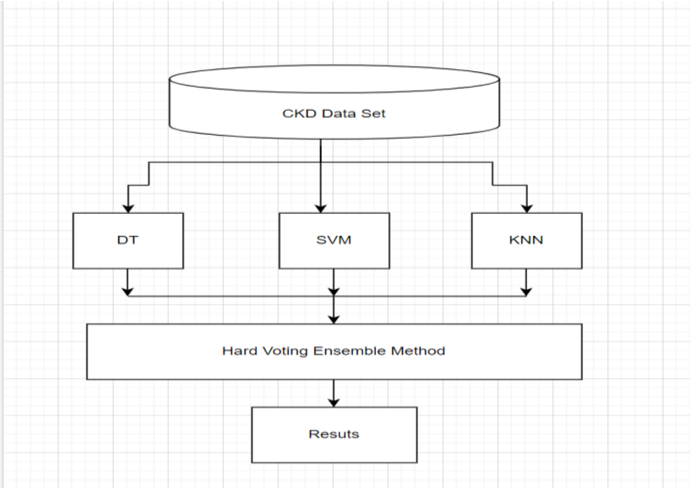


Fig6: Ensemble Diagram for tubular data

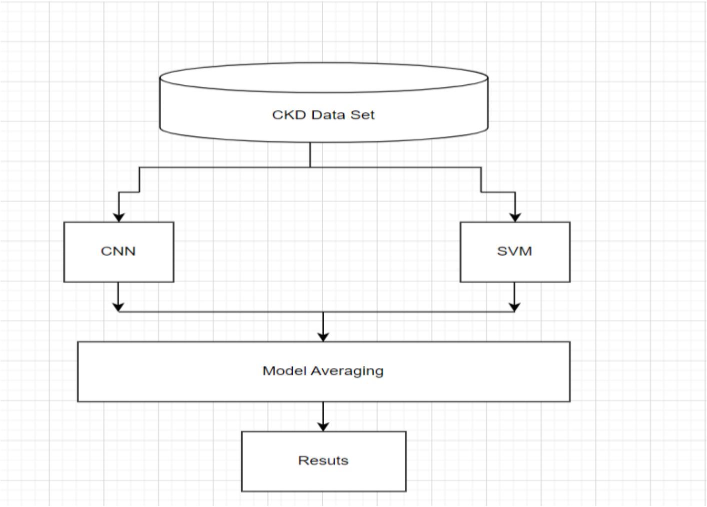


Fig7: Ensemble Diagram for images data

V. Model Architecture

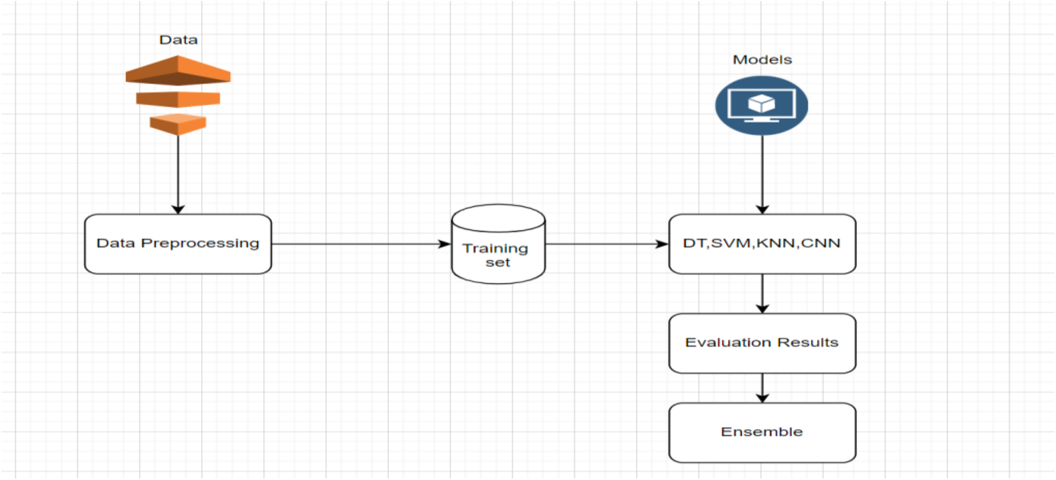


Fig8: Models Architecture

VI. Results and Discussion

VI.I Decision Tree Classifier Results

Confusion Matrix has been generated by DT model for the test data with class (values: CKD: 0, Not CKD: 1) . The confusion Matrix and classification report clearly show important parameters: Precision, Recall, F1_Support, and Accuracy in each class.

Accuracy:

Accuracy is the measure of how close or near the predicted value is to the actual value. The equation of accuracy is shown

Accuracy = (TP + TN) / (TP + FP + TN + FN) *100%

Precision

Precision measure the true values correctly predicted from the total predicted values in the actual class. The equation of precision is shown

$$Precision = TP /(TP + FP) *100\%$$

Recall

Recall measures the rate of positive values that are correctly classified. The equation of recall is shown

$$Recall = TP /(TP + FN) *100\%$$

F-Score

F-Score is also called F1-score is the harmonic mean between recall and precision. The equation of F1_Score is shown

$$F1\_score = 2 \ast (Precision \ast Recall/ Precision + Recall) *100\%$$

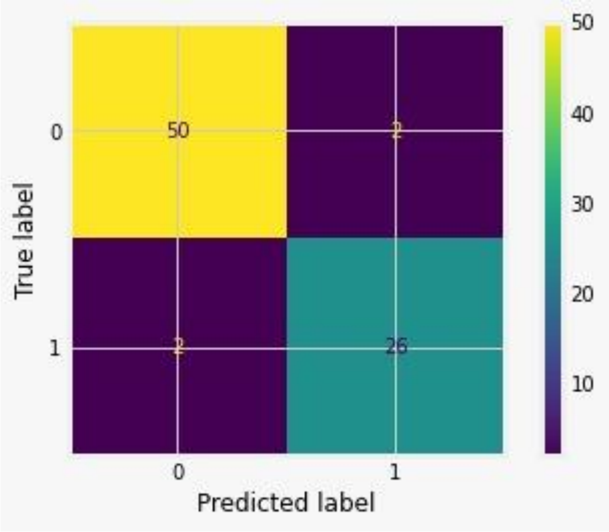
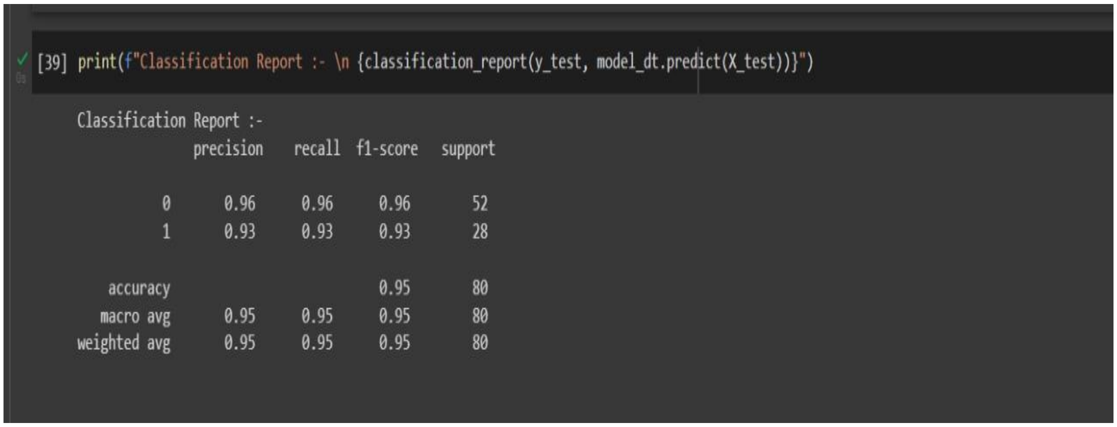


Fig9: Confusion Matrix of DT



```
[39] print(f"Classification Report :- \n {classification_report(y_test, model_dt.predict(X_test))}")

    Classification Report :-
                   precision    recall  f1-score   support

               0       0.96      0.96      0.96        52
               1       0.93      0.93      0.93        28

        accuracy                           0.95        80
       macro avg       0.95      0.95      0.95        80
    weighted avg       0.95      0.95      0.95        80
```

Fig10: Classification Report of DT

As shown in fig3 and fig4, the accuracy of DT model is 0.95(95%), precision for (class:0) is 0.96(96%) and for (class:1) is 0.93(93%), recall for (class:0) is 0.96 (96%) and for(class:1) is 0.93(93%), f1_score for (class:0) is 0.96(96%) and for (class:1) is 0.93(93%).

VI.II SVM Classifier Results

Confusion Matrix has been generated by SVM model for the test data with class (values: CKD: 0, Not CKD: 1) .

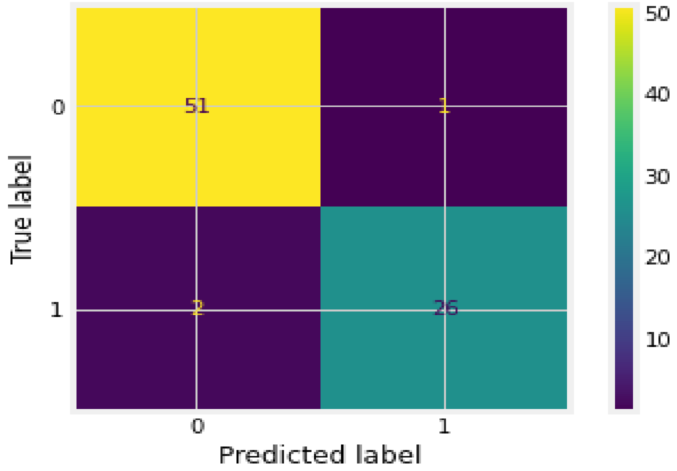
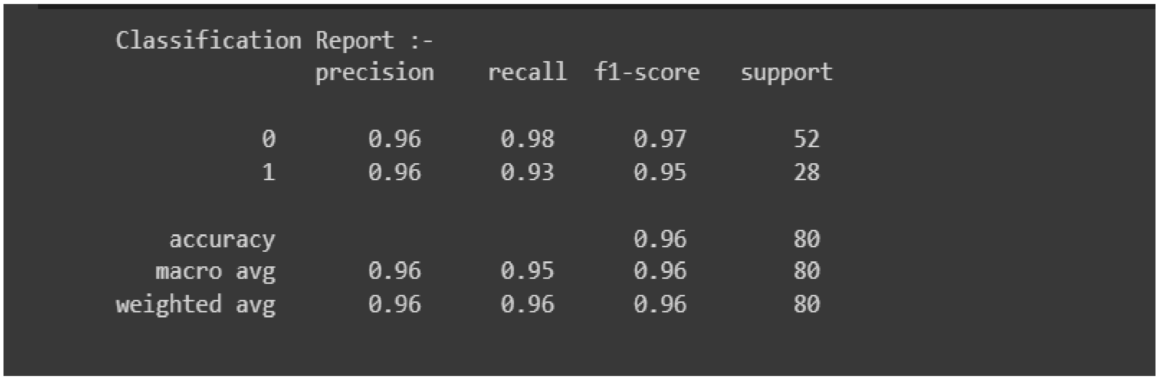


Fig11: Confusion Matrix of SVM



```
Classification Report :-
              precision    recall  f1-score   support

           0       0.96      0.98      0.97        52
           1       0.96      0.93      0.95        28

    accuracy                           0.96        80
   macro avg       0.96      0.95      0.96        80
weighted avg       0.96      0.96      0.96        80
```

Fig12: Classification Report of SVM

As shown in fig5 and fig6, the accuracy of SVM model is 0.96(96%), precision for (class:0) is 0.96(96%) and for (class:1) is 0.96(96%), recall for (class:0) is 0.98 (98%) and for(class:1) is 0.93(93%), f1_score for (class:0) is 0.97(97%) and for (class:1) is 0.95(95%).

VI.III KNN Classifier Results

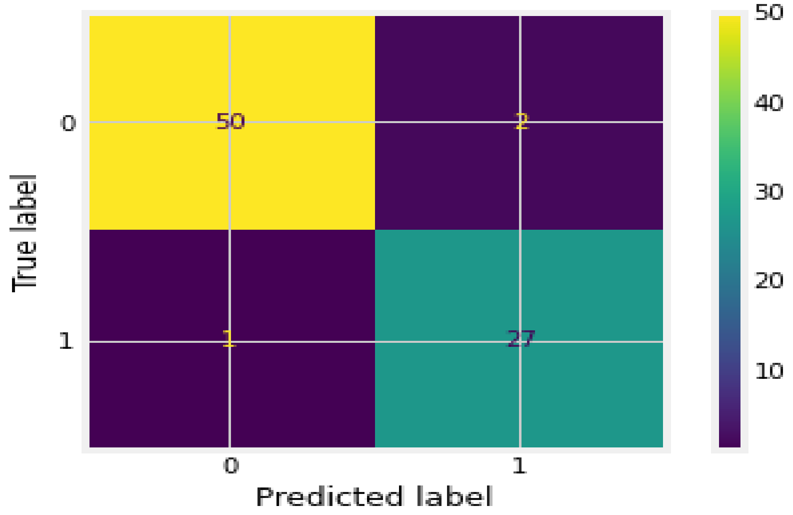
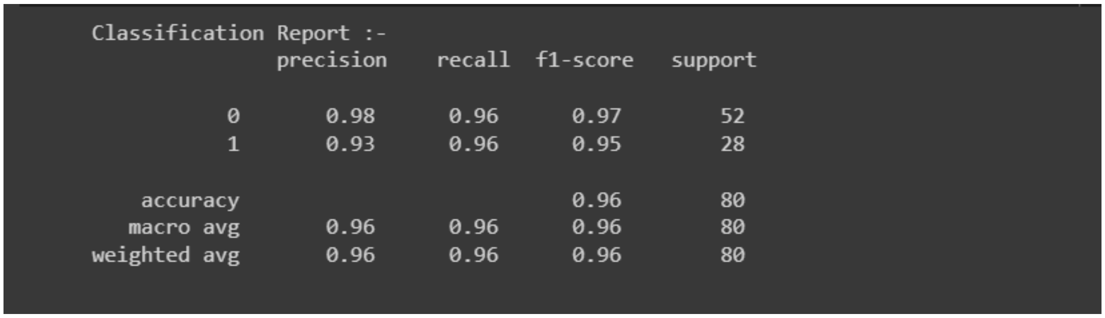


Fig13: Confusion Matrix of KNN



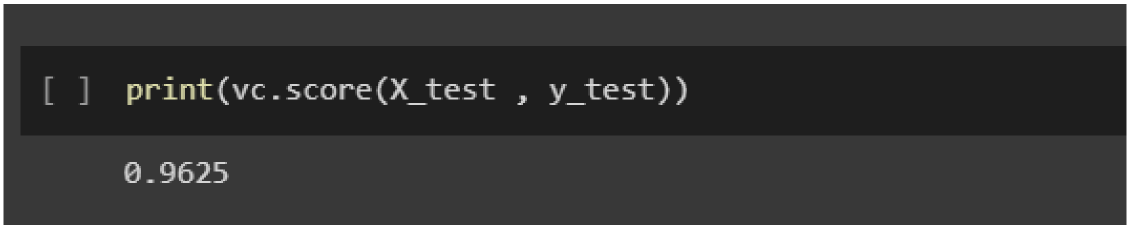
```
Classification Report :-
              precision    recall  f1-score   support

           0       0.98      0.96      0.97        52
           1       0.93      0.96      0.95        28

    accuracy                           0.96        80
   macro avg       0.96      0.96      0.96        80
weighted avg       0.96      0.96      0.96        80
```

Fig14: Classification Report of KNN

As shown in fig13 and fig14, the accuracy of KNN model is 0.96(96%), precision for (class:0) is 0.98(98%) and for (class:1) is 0.93(93%), recall for (class:0) is 0.96 (96%) and for(class:1) is 0.96(96%), f1_score for (class:0) is 0.97(97%) and for (class:1) is 0.95(95%).

VI.IIII Voting Ensemble Results

After using the Voting hard method, the type which takes the majority, it gives an accuracy of 0.**0.9625%** for test data for all models used for tubular data used in this project.

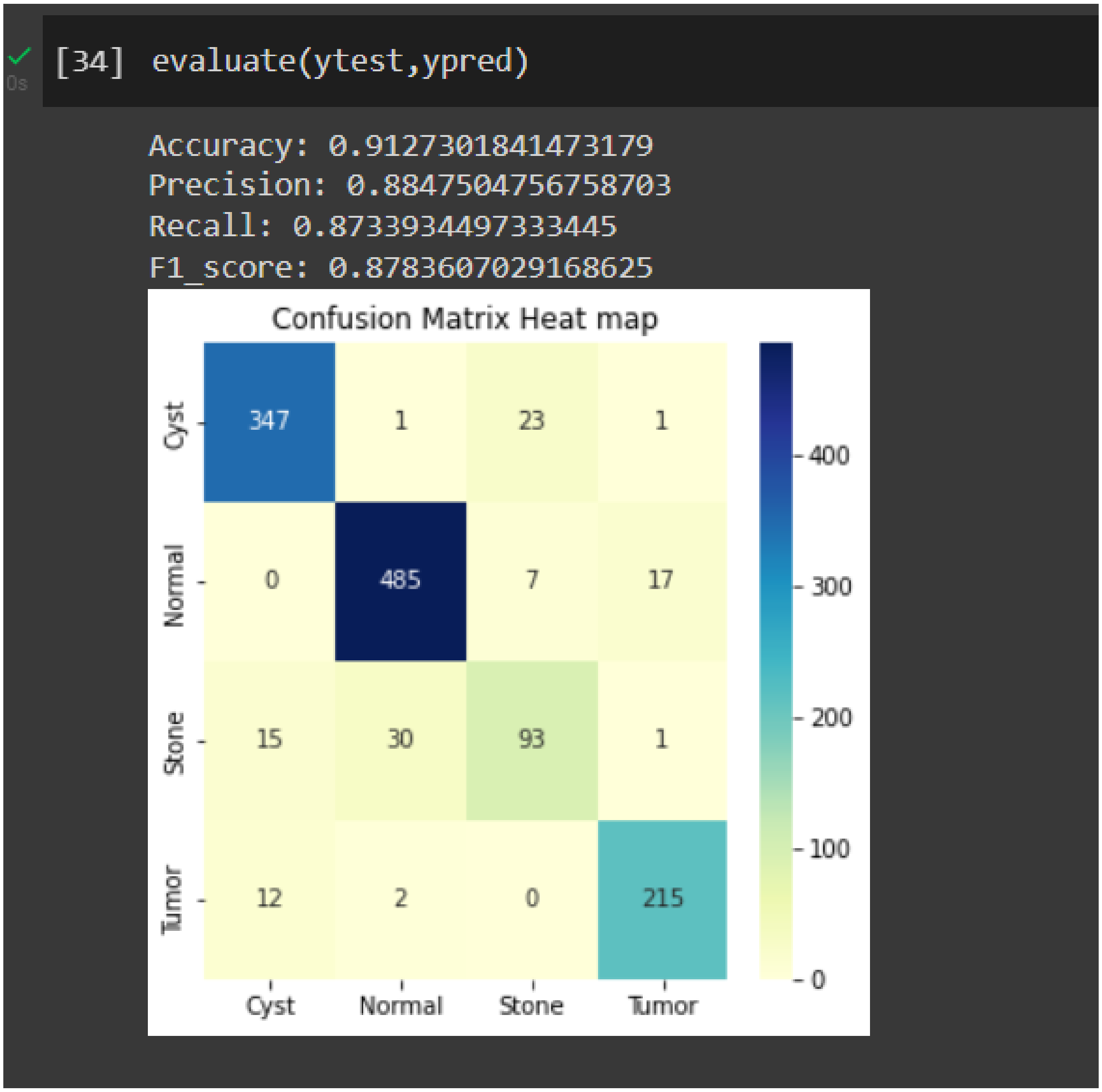


```
[ ]  print(vc.score(X_test , y_test))

     0.9625
```

Fig15: Ensemble result for test data

VI.IIIII CNN :

Confusion Matrix has been generated for the test data with classes (Tumor, Normal, Stone, Cyst). The confusion matrix clearly, says that instances have been classified accurately with test accuracy is 0.91 %.



```
[34]  evaluate(ytest,ypred)

      Accuracy: 0.9127301841473179
      Precision: 0.8847504756758703
      Recall: 0.8733934497333445
      F1_score: 0.8783607029168625
```

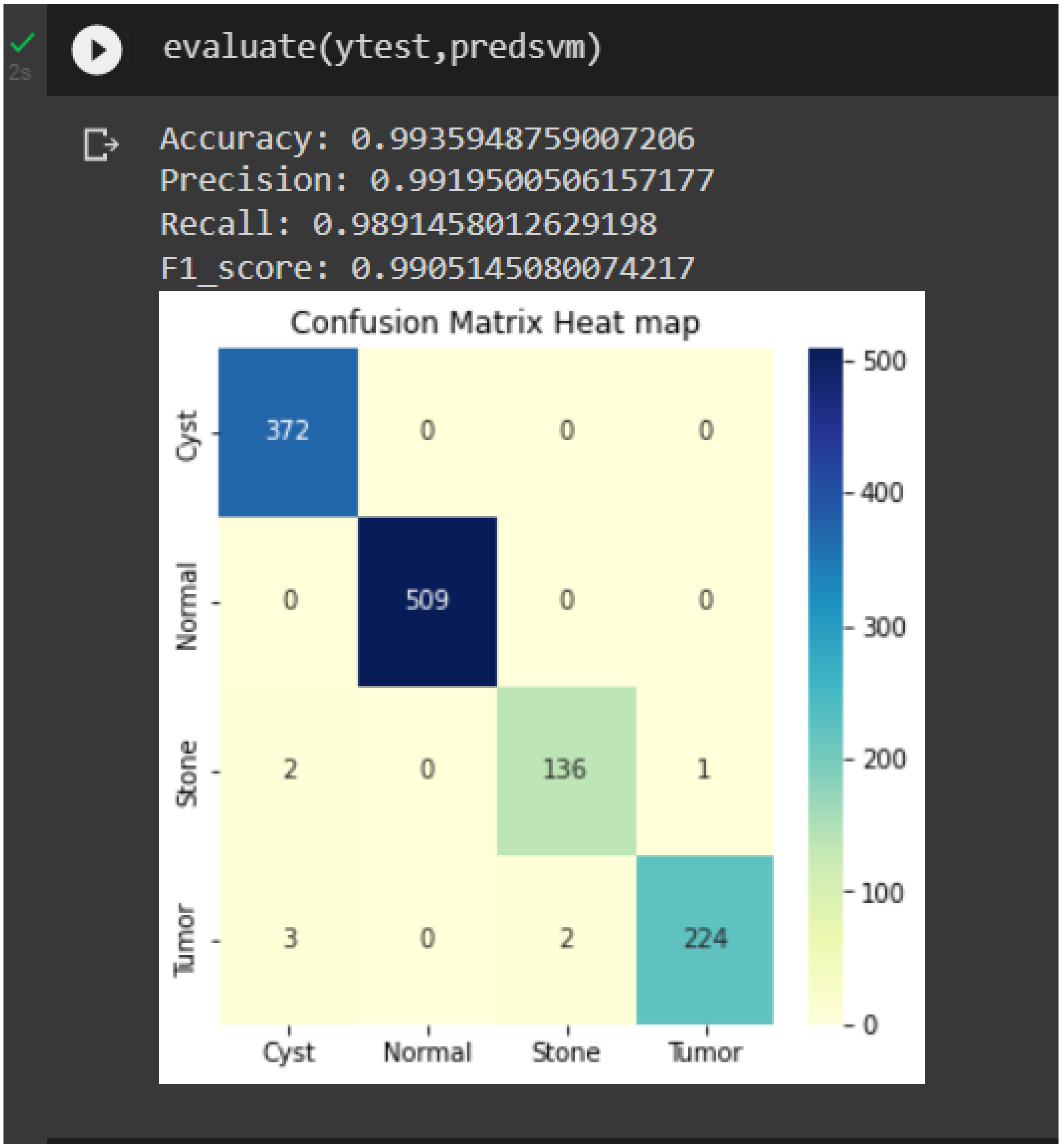
Confusion Matrix Heat map

	Cyst	Normal	Stone	Tumor
Cyst	347	1	23	1
Normal	0	485	7	17
Stone	15	30	93	1
Tumor	12	2	0	215

Fig16: Confusion Matrix of CNN

VI.IIIIII CNN to SVM Results:

Confusion Matrix has been generated for the test data with classes (Tumor, Normal, Stone, Cyst). The confusion matrix clearly, says that instances have been classified accurately with train accuracy of 0.98%, test accuracy of 0.99359% .



```
evaluate(ytest,predsvm)
```

```
Accuracy: 0.9935948759007206
Precision: 0.9919500506157177
Recall: 0.9891458012629198
F1_score: 0.9905145080074217
```

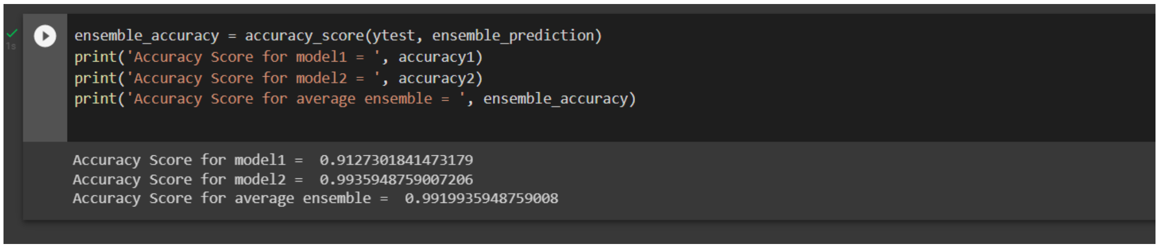
Confusion Matrix Heat map

	Cyst	Normal	Stone	Tumor
Cyst	372	0	0	0
Normal	0	509	0	0
Stone	2	0	136	1
Tumor	3	0	2	224

Fig17: Confusion Matrix of CNN to SVM

VI.IIIIII Model Averaging Ensemble Results:

After using the Model Averaging method, it gives an accuracy of 0.99% for test data for two models used for image data used in this project.



```
ensemble_accuracy = accuracy_score(ytest, ensemble_prediction)
print('Accuracy Score for model1 = ', accuracy1)
print('Accuracy Score for model2 = ', accuracy2)
print('Accuracy Score for average ensemble = ', ensemble_accuracy)
```

```
Accuracy Score for model1 =  0.9127301841473179
Accuracy Score for model2 =  0.9935948759007206
Accuracy Score for average ensemble =  0.9919935948759008
```

Fig18 : Model Averaging Ensemble Results

VII. Conclusion

Chronic kidney disease is a condition characterized by progressive loss of kidney function over time. It is a silent disease, as most sufferers have no symptoms. Early diagnosis and treatment of CKD is a serious task for the medical community that resorts to ML theory to design an efficient solution to this challenge.

In the present work, a methodology based on supervised learning is described, which aims to create efficient models for predicting the risk of CKD occurrence by mainly focusing on ensemble learning-based models. Moreover, we evaluated Decision Trees (DT), Support Vector Machine (SVM), and k-nearest neighbors (KNN). which achieved good performance for tubular data. We also evaluated Convolution Neural Network ,and SVM with CNN and they both performed well with images.

VIII. References

- Images Data retrieved from: HiRID Benchmark (Kidney Function) | Papers With Code
- Tubular Data retrieved from: https://www.kaggle.com/datasets/mansoordaku/ckdisease?select=kidney_disease.csv
- Dibaba Adeba Debal and Tilahun Melak Sitote, Chronic kidney disease prediction using machine learning techniques, Debal and Sitote Journal of Big Data (2022) 9:109
- Salekin A, Stankovic J. Detection of Chronic Kidney Disease and Selecting Important Predictive Attributes. In: Proc. - 2016 IEEE Int. Conf. Healthc. Informatics, ICHI 2016, pp. 262–270, 2016.

- Saurabh Pal, Chronic Kidney Disease Prediction Using Machine Learning Techniques, AUG 2022 ,Biomedical Materials & Devices https://doi.org/10.1007/s44174-022-00027-y
- Tekale S, Shingavi P, Wandhekar S, Chatorikar A. Prediction of chronic kidney disease using machine learning algorithm. Disease. 2018;7(10):92–6.
- Priyanka K, Science BC. Chronic kidney disease prediction based on naive Bayes technique. 2019. p. 1653–9.
- Yashf SY. Risk Prediction Of Chronic Kidney Disease Using Machine Learning Algorithms. 2020,https://doi.org/10.1186/s40537-022-00657-5.

VIIII. Contribution

Name	ID	Work
Esraa Abdullah	**19105371**	• Introduction • Methodology • Related Works • Diagram blocks • Results
Elzahraa Saeed	**19106429**	• Abstract • Introduction • Related Works • Methodology • Data Pre-processing • DT model implementation • DT graphs • Diagram blocks

		• Ensemble • Results
Rana Rizk	**19105575**	• Problem Statement • Introduction • Methodology • Diagram blocks • CNN to SVM implementation • CNN to SVM graphs • Ensemble • Results
Maram Zoughieb	**19105793**	• Introduction • Methodology • Related Works • SVM,KNN implementation • SVM,KNN graphs • Diagram blocks • Ensemble • Results • Conclusion