Nigerian Journal of Technology (NIJOTECH)
Vol. 41, No. 1, January, 2022, pp.175 –183
www.nijotech.com

Print ISSN: 0331-8443
Electronic ISSN: 2467-8821
http://dx.doi.org/10.4314/njt.v41i1.22

Development of a Graphical User Interface Software for The Prediction of Chronic Kidney Disease

S.C. Nwaneri[1, *] and H.C. Ugo[2]

[1]*Department of Biomedical Engineering, Faculty of Engineering, University of Lagos, Akoka, Lagos, Lagos State, NIGERIA*
[1]*Department of Biomedical Engineering, College of Medicine, University of Lagos, Idi-Araba, Lagos, Lagos State, NIGERIA*
[2]*Department Of Biomedical Engineering, Afe Babalola University, Ado-ekiti, Ekiti State, NIGERIA*

Abstract
Chronic Kidney Disease (CKD) is a severe kidney damage that is difficult to diagnose at the early stages due to the absence of clear symptoms. Late diagnosis of CKD is a common problem in low-income countries and is often associated with lower chances of survival. This study was designed to develop a user-friendly web-based graphical user interface (GUI) software for the prediction of CKD using artificial neural networks (ANNs). The model was developed using Python programming language and trained with 1200 instances of CKD datasets obtained from the University of California Irvine (UCI) machine learning repository. This dataset was split into 80% for training and 20% for testing achieved through an iterative process. A GUI software was developed based on the model using Django, an open-source python web development framework. The model achieved an accuracy of 95.83%, a precision of 100%, a specificity of 100%, and a sensitivity of 89.80%. The GUI software was effectively used to predict CKD and could be of immense benefit as a point of care application for early CKD prediction

Keywords: Artificial Neural Network**,** Chronic Kidney Disease**,** Risk Prediction Model

1.0 INTRODUCTION

Chronic kidney disease (CKD) is a major global health problem associated with high rates of mortality and morbidity. The prevalence of CKD in Low- and Middle-income countries (LMIC) is higher compared to developed countries due to the double burden of communicable and non-communicable diseases present in LMICs [1]. Available statistics suggest that CKD is the sixth fastest cause of death worldwide with over 2.4 million deaths per year and over 19 million disability-adjusted life-years (DALYs) [2 -3]. CKD is defined as kidney damage or glomerular filtration rate (GFR) less than 60 mL/min/ 1.73 m^2 for three (3) months or more [4]. GFR can be estimated from calibrated serum creatinine and estimate equations, such as the Cockcroft-Gault formula [5]. The most common indicators of kidney damage are proteins in blood or urine, blood in the urine, and raised levels of urea or creatinine in the blood. However, kidney damage is confirmed by the persistence of albumin in the urine with albumin-to-creatinine ratio >30 mg/g in two of three spot urine samples [4, 6]. CKD is generally classified into stages 1 - 5, indicating increasing order of severity [7].

The severity of CKD is measured by estimated glomerular filtration rate (eGFR) determined from serum creatinine laboratory tests, race, sex, and age [8]. Patients with stage 4 and 5 CKD are at high risk of end-stage renal disease (ESRD) or death if their condition is not diagnosed on time. Kidney failure is regarded as the most serious outcome of CKD, usually accompanied by complications of reduced kidney function. Some symptoms that indicate the onset of kidney failure are muscle cramps, nausea, appetite losses, swollen feet and ankles, too much urine or not enough urine, and respiratory failure [9]. Complications can occur at any stage of CKD, which often leads to death [10, 11]. Common risk factors of CKD include high blood pressure, old age, heart and blood vessel disease, smoking, diabetes, obesity, abnormal urinary structure, and a family history of kidney disease. Early detection of CKD is necessary to help affected persons commence timely treatment to improve the chances of survival.

The application of data mining and machine learning (ML) techniques in disease prediction has attracted several research interests [12- 20]. These techniques are usually based on mathematical optimization, probabilistic and statistical methods [21].

*Corresponding author **(Tel:** +234 (0) 8035431589)
Email addresses: snwaneri@unilag.edu.ng (S.C. Nwaneri) and hannahugo@abuad.edu.ng (H.C. Ugo)

Most of the algorithms were implemented using supervised learning [12, 14 -20]. Some of these studies focused on the development of a single algorithm for CKD diagnosis [14, 18], while others used two or more ML algorithms [14, 16, 17]. In one of the studies [15], an end-stage kidney disease predictor model was developed based on the ANN ensemble. The ANN classifier influences the results returned by an ensemble of ten networks. However, the study focused on the prediction of the last stage of CKD and analyzed the classification accuracy of 3 supervised ML algorithms; ANN, Decision Tree (DT), and Logical Regression (LR). The algorithms were tested on data from patients undergoing kidney dialysis. ANN performed better with a classification accuracy of 93.852 % compared to DT (78.45%) and LR algorithms (74.74%). In another study [22], a model to predict the risk of cats developing CKD was proposed using a recurrent neural network. The study revealed that model sensitivity decreased when predicting the risk of CKD. A deep learning approach capable of automatically computing the eGFR and CKD status was proposed by [23] which achieved higher accuracy than the predictive accuracy of experienced nephrologists. Furthermore, a data mining approach was employed by [24] to predict the survival of CKD patients. In [25] nine models for predicting CKD progression were evaluated. Elastic Net and XGBoost gave the highest sensitivity (85%), and specificity (0.83) respectively. In [26], a prediction model to determine the progression of stage 1 diabetic kidney disease (DKD) was constructed using artificial intelligence (AI). The application of these models in CKD diagnosis is too complex for medical doctors who need a simple and more user-friendly decision support tool to assist them in CKD diagnosis. This study was designed to develop a user-friendly graphical user interface (GUI) software application based on ANN model for the timely prediction of CKD. This tool will improve the efficiency of CKD diagnosis by nephrologists compared to using conventional approaches.

2.0 METHODS

The present study implemented the ANN-based CKD prediction model using the Python programming language. Python is an open-source, high-level, general-purpose programming language. ANN is well known for its high accuracy in predicting diseases and widely used by most researchers for disease classification [12-15]. Hypertext Markup Language (HTML) was utilized in this study for creating the GUI for the CKD prediction software while JavaScript was used in conjunction with HTML to create the GUI interface for CKD prediction. The workflow for the CKD prediction process is shown in Figure 1.

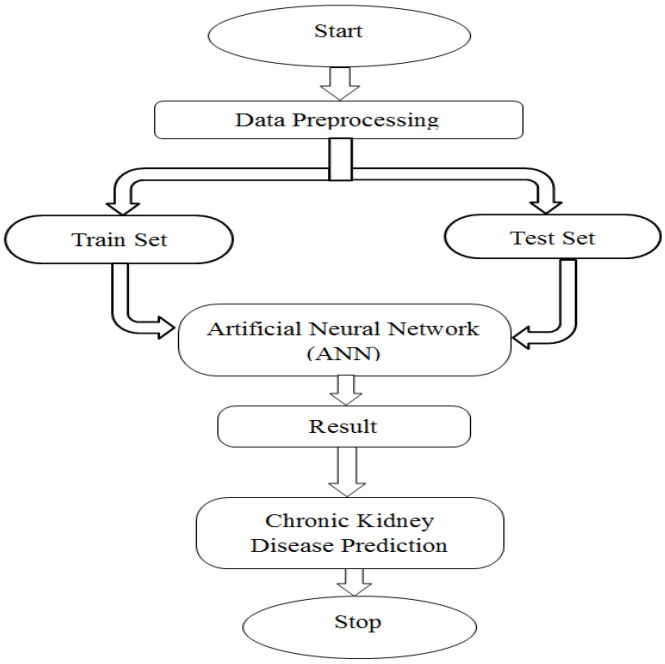

Figure 1: Workflow of CKD prediction system

2.1 Datasets

The dataset used for this research was obtained from the University of California, Irvine (UCI) machine learning repository. A total of 1200 instances of data were generated from the original 400 patients diagnosed with CKD and those without CKD. The dataset contains 24 input features and 1 output feature, which is the class value (target output). It contains 750 instances of CKD and 450 instances of not CKD. Table 1 summarizes the various attributes used in this research, their values, description, and representation in the dataset.

Table 1: Dataset attributes, attribute representation, attribute values, and description

	Attribute	Representation	Attribute Values	Description
1	Age	Age	Numerical	Years
2	Blood pressure	BP	Numerical	mm/Hg
3	Specific gravity	SG	Nominal	1.005,1.010,1.015,1.020, 1.025
4	Albumin	AL	Nominal	0,1,2,3,4,5
5	Sugar	SU	Nominal	0,1,2,3,4,5

	Attribute	Representation	Attribute Values	Description
6	Red blood cell	RBC	Nominal	Normal, Abnormal
7	Pus Cell	PC	Nominal	Normal, Abnormal
8	Pus Cell Clumps	PCC	Nominal	Present, Not Present
9	Bacteria	BA	Nominal	Present, Not Present
10	Blood Glucose Random	BGR	Numerical	mgs/dl
11	Blood Urea	BU	Numerical	mgs/dl
12	Serum Creatinine	SC	Numerical	mgs/dl
13	Sodium	SOD	Numerical	mEq/L
14	Potassium	POT	Numerical	mEq/L
15	Hemoglobin	HEMO	Numerical	Gms
16	Packed Cell Volume	PCV	Numerical	Cells
17	White Blood Cell Count	WC	Numerical	Cells/cumm
18	Red Blood Cell Count	RC	Numerical	Millions/cmm
19	Hypertension	HTN	Nominal	Yes, No
20	Diabetes Mellitus	DM	Nominal	Yes, No
21	Coronary Artery Disease	CAD	Nominal	Yes, No
22	Appetite	APPET	Nominal	Good, Poor
23	Pedal Edema	PE	Nominal	Yes, No
24	Anemia	ANE	Nominal	Yes, No
25	Class	Class	Nominal	CKD, not CKD

2.2 Data Preprocessing

Data pre-processing involves exploration, filtering, and scaling of the dataset. The CKD dataset was explored to determine its shape, data type, and state. A correlation was used to relate the features of the dataset. This is done to determine how all the features are positively or negatively correlated to the target output. Figure 2 revealed various features and their correlation with CKD. Hypertension, albumin levels, diabetes mellitus, blood urea content, and serum creatinine levels are positively correlated with CKD. In contrast, specific gravity, red blood cell count, haemoglobin levels, and packed cell volume are negatively correlated with CKD.

Dataset filtering was performed to clean the dataset of missing values or not a number (NaN). Using the fillna method, the columns with missing values (NaN) are backwardly filled. That is, the missing values in each column are replaced with values directly below them. The next process is to convert non-numeric values to numeric ones. Using LabelEncoder () imported from sklearn.preprocessing, the strings are transformed to integers. The result is a cleaned processed dataset. Dataset scaling was achieved using the min-max scaler method which scales the dataset so that all the input features lie between 0 and 1. This estimator scales and transforms each feature to a given range based on equations 1 – 2.

$$X_{std} = \frac{\left(X - X.\min(axis = 0)\right)}{\left(X.\max(axis = 0) - X.\min(axis = 0)\right)} \quad (1)$$

$$X_{scaled} = X_{std} \times (\max - \min) + \min \quad (2)$$

Where min, max = feature_range.

Afterward, the dataset was split into 80% for training and 20% for testing which was achieved through an iterative process.

2.3 Model Development and Training

This process involves building the neural network architecture and determining the activation functions to be used at the output of each layer of the network. The ANN model is built using the Keras libraries in python. Dense is used to assign the number of layers for the network. The ANN model consists of 25 neurons in the input layer and one neuron at the output layer. Input dimension is defined at the first layer which shows how many parameters will be taken as input. The Rectified Linear Unit (ReLU) and the sigmoid activation functions are used as activation functions in the input and output layers respectively as shown in equations (3) and (4):

$$a = f\left(\sum_{i=0}^{N} W_i X_i\right) \quad (3)$$

The output of the neural network will be:

$$y = f(w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + \ldots\ldots + w_nx_n + b) \quad (4)$$

where y indicates the output, w is the weight, x is the input, and b is the bias. The network structure of the model can be described as a multilayer neural network consisting of one input layer with twenty-five (25) nodes, a hidden layer comprising fifty-six nodes (56), and an output layer. Figure 3 shows the network architecture of the proposed network

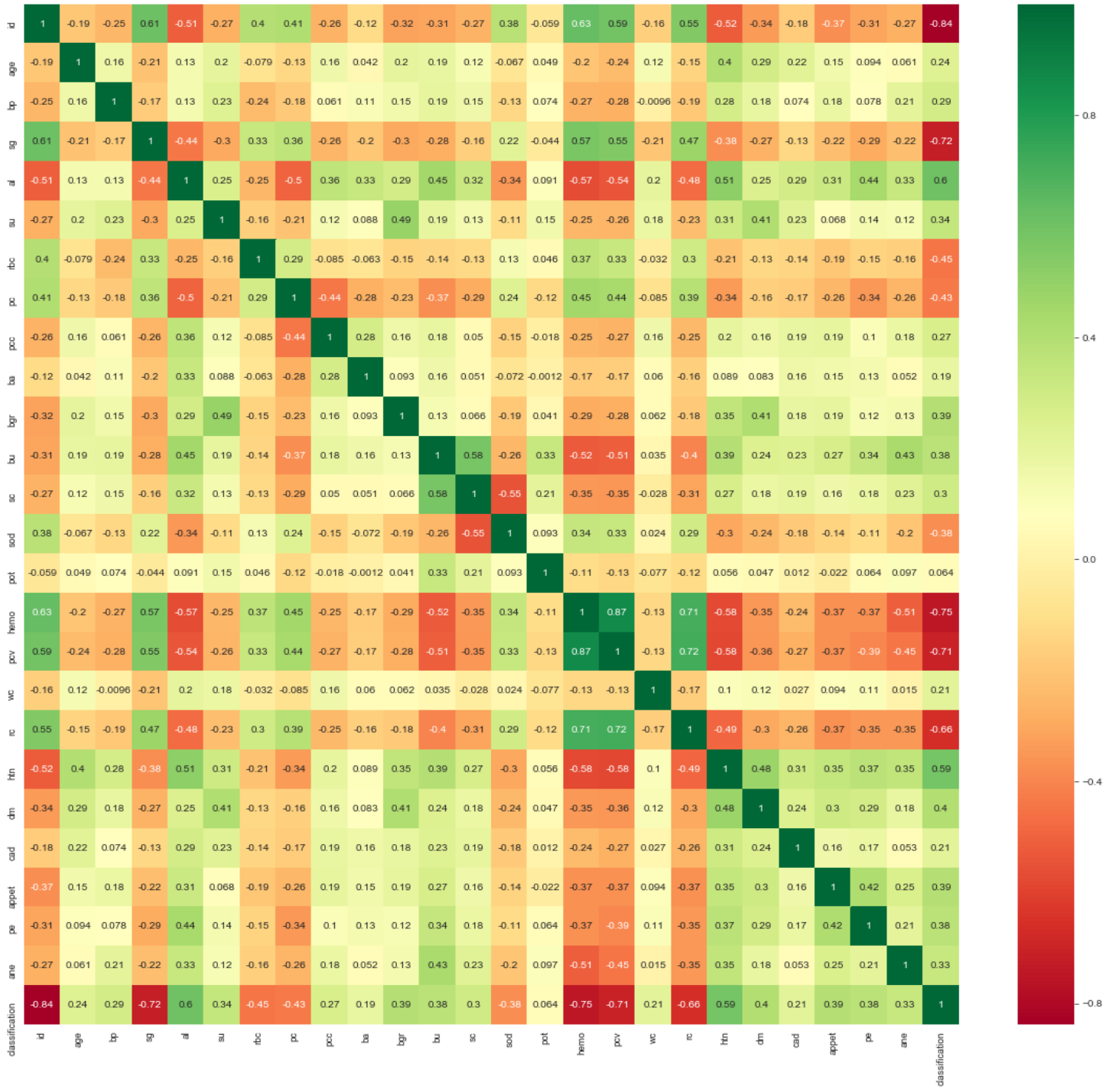

Figure 2: Correlation matrix of the features

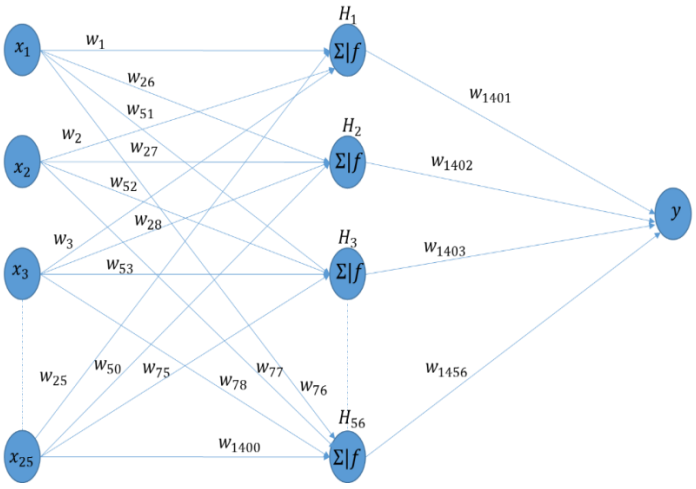

Figure 3: Network architecture consisting of the input, hidden, and output layers

The following are equations on the ANN model used in this study. The hidden layer has fifty-six (56) nodes denoted by H_1 – H_{56}, and twenty-five (25) inputs, x_1 – x_{25}, with their corresponding weights, w_1 -w_{25}.
For node 1:

$$H_1 = X_1w_1 + X_2w_2 + X_3w_3 + \cdots + X_{25}w_{25} \quad (5)$$

ReLu Function is stated mathematically as:

$$F(x) = max\,(0, x) \quad (6)$$

Output of H_1 after the ReLu is passed = max (0, H_1)

$$H_2 = X_1w_{26} + X_2w_{27} + X_3w_{28} + \cdots + X_{25}w_{50} \quad (7)$$

Output of H_2 = max (0, H_2)

$$H_3 = X_1w_{51} + X_2w_{52} + X_3w_{53} + \cdots + X_{25}w_{75} \quad (8)$$

Output of H_3= max (0, H_3)
The operation continues in a similar way till H_{56} (56th node). For the 56th node:

$$H_{56} = X_1w_{76} + X_2w_{77} + X_3w_{78} + \cdots + X_{25}w_{1400} \quad (9)$$

Output of $H_{56} = \max(0, H_{56})$
The output layer has fifty-six (56) input nodes and one (1) output node

$$Y = H_1w_{1401} + H_2w_{1402} + H_3w_{1403} + \cdots + H_{56}w_{1456} \quad (10)$$

Sigmoid Function, S

$$S = \frac{1}{1 - e^{-H}};\ H = Y \quad (11)$$

Output, Y

$$Y = \frac{1}{1 - e^{-Y}}$$

Compiling the model involves specifying the loss function, optimizer, and metrics. The loss function used for this work is the binary loss classification (binary cross-entropy) because the outcome of the output layer is binary classification. Binary loss classification is used along with the sigmoid activation function to fire the output. Optimization technique is needed to reduce the cost function. The optimizer used in this study is the stochastic gradient descent (SGD). The model was trained with an epoch size of 500. The trained model was saved as model.py so that it could be imported to Visual Studio to create the python application. The performance of the model was evaluated by computing the accuracy, sensitivity, specificity, precision, and F-measure as shown in equations (12) to (16) respectively.

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (12)$$

$$\text{Sensitivity} = \frac{T_P}{T_P + F_N} \quad (13)$$

$$\text{Specificity} = \frac{T_N}{T_N + F_P} \quad (14)$$

$$\text{Precision} = \frac{T_P}{T_P + F_P} \quad (15)$$

The F-measure is calculated as:

$$F = 2 \times \frac{\text{Precision X Sensitivity}}{\text{Precision} + \text{Sensitivity}} \quad (16)$$

2.4 Merging the Neural Network Model with the GUI

Deploying the trained classification model involves the following steps: creating a Django project, creating a Django app, connecting the neural network model, creating the GUI, deploying the app to the Heroku server. Django is a high-level open-source project that supports the implementation of python tools and the most popular packages. Running this command creates a folder

with the name of the project (CKD) which contains all the necessary files required to run and manage Django projects. The app is created and added to the installed Apps. The neural network model is added in a file inside the prediction app created earlier. Pickle is imported to save and load the model. Afterward, a function was created to call the new parameters as a NumPy array, and a prediction was made using the saved model. Finally, the neural network model was connected to the Django database model (models.py) by overriding the save method. The landing page for the GUI was generated using HTML, JS, and CSS. To achieve this, templates and static folders were created inside the CKD folder. The template folder bears the HTML files. HTML describes the template of the GUI, specifying its layout, size of the text box for the input variables, the minimum value required for each field for the variable input, and the data type required for each field. Furthermore, CSS and JS files were generated and placed in the static folder. CSS handles the style, fonts, and color of the interface while JS is used in conjunction with the HTML to handle the texts on the user interface. Views for the pages were created and connected to individual templates and URLs (Uniform Resource Locator). The next step was to migrate the database and start the development server. The application location was specified through the command prompt (anaconda prompt) to use TensorFlow backend to deploy the application. On the web browser, using http://localhost:8000/ the GUI application can be assessed locally.

2.5 Deploying the app to Heroku server

The application is further improved by hosting it on Heroku. Heroku is a Platform-as-a-Service (PaaS) cloud infrastructure that provides hosting services for multiple programming languages, including python [27]. To prepare the app, few configuration files were added and some packages were installed. A Procfile was added in the project root directory to define process types and used to absolutely declare what command should be executed to start the app. A runtime.txt file is added in the project root directory as it contains the python version used for the project. Other packages were installed to enable the Django app to run on the server. The database configuration was updated in the settings.py file. The app is created in Heroku from the terminal (command prompt). Afterward, the app domain name (ckdpredictor) was added to ALLOWED_HOSTS in settings.py. This will enable the server to open the app on the web browser. Also, Git was initialized and the app was connected to Heroku Git remote repository. Finally, the project is pushed to the remote repository (deploying the app to Heroku) using this line of code: git push Heroku master and run to check for errors. Then, the database is migrated using: Heroku run python manage.py migrate. The site can be accessed using the link:

Table 2: Performance Evaluation Results

Performance Metrics	Value (%)
Accuracy	95.83
Sensitivity	89.80
Specificity	100.00
Precision	100.00
F-measure	94.63

http://ckdpredictor.herokuapp.com.

3.0 RESULTS

This section explains the result of model training, validation, and integration with the GUI. Figures 4 and 5 represent a plot of the model accuracy and loss. These plots illustrate the current state of the model at each step of the training algorithm. In Figure 4, the accuracy increased slightly and maintained a constant value from 0 to 98 epochs. It started increasing again at 98 to 100 epochs. It is observed that at 100 epochs, the accuracy is above 95%. Also, the accuracy stopped increasing and became constant after 300 epochs which indicates that model has stopped learning.

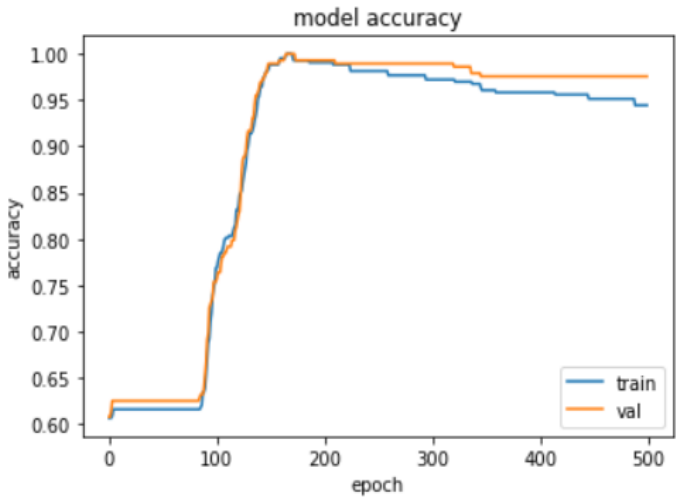

Figure 4: The plot of accuracy versus epoch of the training model

In Figure 5, the training model loss decreased constantly from 100 epochs to 500 epochs. It is observed that at 100 epochs the model loss is 55%. After 200 epochs, the training model loss decreased to 45% and continues to decrease to 400 epochs. Between 400 and 500 epochs the training loss did not decrease which means that the model has stopped learning.

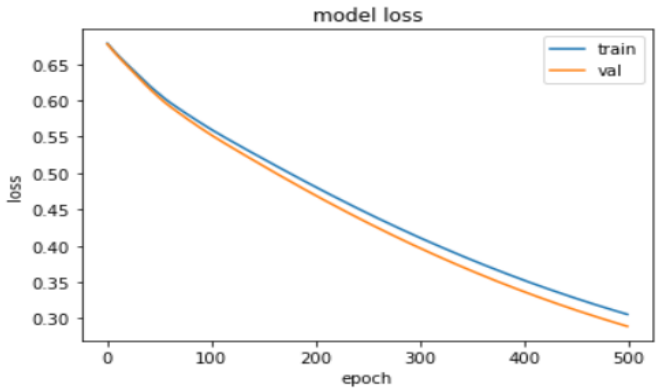

Figure 5: Loss versus epoch of the training model

Results of the model evaluation showing the accuracy, sensitivity, specificity, precision, and F-score are shown in Table 2.

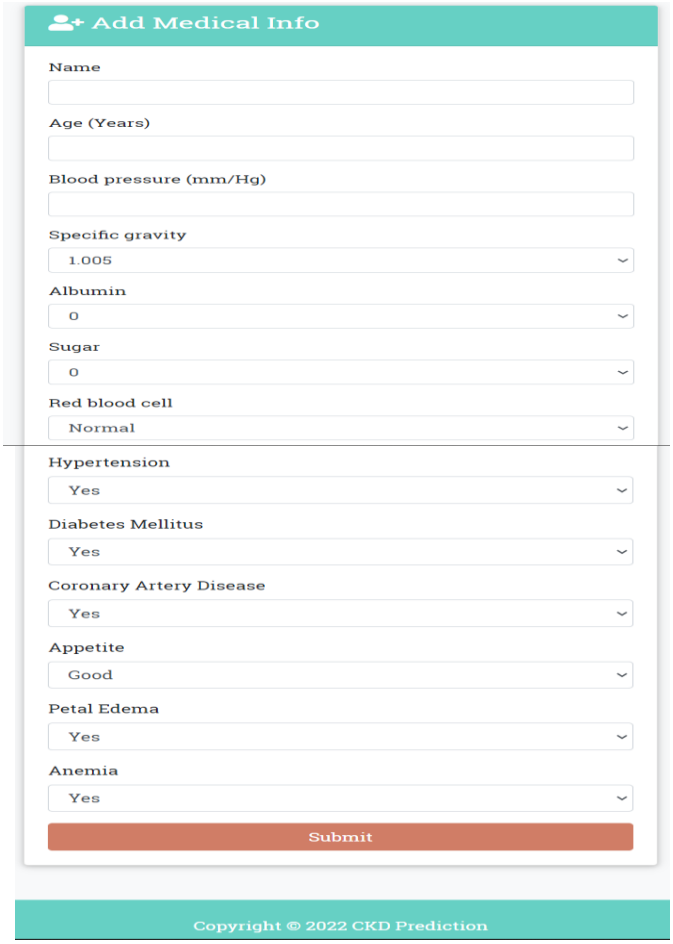

Figure 6: GUI of the CKD software

4.0 DISCUSSIONS

In this paper, the authors first developed an ANN model for the prediction of CKD and subsequently developed user-friendly GUI software based on the model. Given the high prevalence of CKD in LMICs [1], and the shortage of nephrologists, a decision support tool for CKD prediction will be of immense benefit to physicians and their patients.

ANNs can build complex models and make accurate decisions when trained with the required data. Previous studies had developed ML models to determine if a patient had CKD or not [12 -17]. With the large data size used in this research, the model achieved a reasonable accuracy of 96% which is consistent with related models in literature [28]. Although the ANN model developed by [28] achieved a higher performance accuracy of 98% which is marginally higher than our model, the performance accuracy of our model was higher than the models proposed by [25] and [29]. In [30], the performance accuracy of their ANN model for CKD was 97%. The results observed from these recent models are not far from the prediction accuracy of the model obtained in this research. Hence, this research model predicts accurately with high performance.

The novelty of this research is the development of an online user-friendly tool designed to enable individuals determine if they have CKD or not. This online tool can be accessed easily anytime using the link: http://ckdpredictor.herokuapp.com, to facilitate the timely prediction of CKD. The software includes a user interface that accepts all the 24 user attributes shown in Table 1 which include age, blood pressure, specific gravity, albumin, sugar, red blood cell, pus cell, pus cell clumps, bacteria, random blood glucose, blood urea, serum creatinine, sodium, potassium, haemoglobin, packed cell volume, white blood cell count, red blood cell count, hypertension, diabetes, coronary artery disease, appetite, pedal anaemia, and edema. This tool will also enable health practitioners to make accurate decisions when verifying the findings of their diagnosis in a relatively short time thus improving the efficiency of CKD diagnosis compared to the traditional methods of CKD diagnosis.

5.0 CONCLUSIONS

In this study, a Graphical User Interface Software for the Prediction of CKD was developed based on the ANN prediction model. The model was implemented using python programming language. The model was trained using secondary data obtained from UCI machine learning repository and it achieved high training and testing accuracies. A GUI was built to integrate the model and a web browser using Django, an open source python to create a web application. The application was also deployed on Heroku, a platform that hosts web applications. This model could aid patients and clinicians in better decision-making about the best course of action in patients approaching CKD. The non-invasive software can

be used as a point-of-care application by medical practitioners to facilitate the early prediction of CKD in clinical practice and for timely commencement of treatment. The main limitation of this study is the size and attributes of the secondary data used for model training and validation. Also, the generalizability to patients from other regions was not examined.

Ethical Approval

The research work was approved by the Lagos University Teaching Hospital Health Research Ethics Committee (LUTHHREC). The ethical approval number is ADM/DCST/HREC/APP/3259. No direct use of human or animal subjects was required for this research.

REFERENCES

[1] Chukwuonye, I..I. Ogah, O.S. Anyabolu, E.N., Ohagwu, K.A., Nwabuko, O. C., Onwuchekwa, U. et al. "Prevalence of chronic kidney disease in Nigeria: a systematic review of population-based studies", *International Journal of Nephrology and Renovascular Disease*, 11, 2018, pp. 165 –172.

[2] Kassebaum, N.J. Arora, M. Barber, R.M. Bhutta, Z.A. Brown, J. Carter, "A. GBD 2015 DALYs and HALE Collaborators. Global, regional, and national disability-adjusted life-years (DALYs) for 315 diseases and injuries and healthy life expectancy (HALE), 1990-2015: a systematic analysis for the Global Burden of Disease Study 2015", *Lancet*, 388(10053), 2016, pp. 1603–1658.

[3] WKD: Kidney Health for Everyone and Everywhere. http://www.worldkidneyday.org(2019). Accessed on March 14, 2019.

[4] Inker L.A. Astor, B.C., and H, F.C. "KDOQI US commentary on the 2012 KDIGO clinical practice guideline for the evaluation and management of CKD", *American Journal of Kidney Diseases*, 63(5), 2014, pp. 713–735.

[5] NKF. National Kidney Foundation: "K/DOQI Clinical Practice Guidelines for Chronic Kidney Disease: Evaluation, Classification, and Stratification", *American Journal of Kidney Diseases*, 39(1), 2002, pp. S1–S266.

[6] WHO. The World Health Report *2003*. Shaping the future. World Health Organisation: Geneva, 2003.

[7] Bhattacharya, M. Jurkovitz, C., and Shatkay, H. "Identifying patterns of associated-conditions through topic models of Electronic Medical Records", *Proceedings of the IEEE International Conference on BIBM*. 2016, pp. 466-469.

[8] Levin, A., Stevens, P.E., Bilous, R.W., and Coresh, J. "Kidney Disease: Improving Global Outcomes (KDIGO) CKD Work Group. KDIGO 2012 Clinical Practice Guideline for the Evaluation and Management of Chronic Kidney Disease", *Kidney International Supplements*, 3, 2013, pp. 1-150.

[9] Singh, VibhavPrakash, SubodhSrivastava, and Srivastava, R. "An efficient image retrieval based on the fusion of fast features and query image classification", *International Journal of Rough Sets and Data Analysis*, 4(1), 2017, pp. 19-37.

[10] James, M.T. Hemmelgarn, B.R. Wiebe, N. "Alberta Kidney Disease Network. Glomerular filtration rate, proteinuria, and the incidence and consequences of acute kidney injury: a cohort study", *Lancet*, Vol. 376, Number. 9758, 2010, pp. 2096 – 2103.

[11] Levey, A.S., and Coresh, J. "Chronic kidney disease", *Lancet*, 379(9811), 2012, pp. 165 -180.

[12] Nwaneri, S.C., Nwoye, E.O., Irurhe, N.K., and Babatunde, A.M. "Application of Artificial Neural Network in Breast cancer Classification: A Comparative Study", University of Lagos *Journal of Basic Medical Sciences*, 2(1), 2014, pp. 32 – 38.

[13] Kaur, H. and Kumari, V. (2018) 'Predictive modeling and analytics for diabetes using a machine learning approach, *Applied Computing, and Informatics*, 12(4), pp. 2210-8327.

[14] Abhishek, Thakur, G.S., and Gupta, D. "Proposing Efficient Neural Network Training Model for Kidney Stone Diagnosis", *International Journal of Computer Science and Information Technology*, 3(3), 2012, pp. 3900-3904.

[15] Noia, T.D. Ostuni, V.C. Pesce, F., Binetti, G. Naso, D. and Schena, F.P. "An end-stage kidney disease predictor based on an artificial neural networks ensemble", *Expert Systems with Applications*, 40, 2013, pp. 4438–4445.

[16] Lakshmi, K.R., Nagesh, Y. VeeraKrishna, M. "Performance comparison of three data mining techniques for predicting kidney dialysis survivability", *International Journal of Advances in Engineering and Technology*, 2014, pp. 242-254.

[17] Sharma, S. Sharma, V., and Sharma, A. "Performance-based evaluation of various machine learning classification techniques for chronic kidney disease diagnosis", *International Journal of Modern Comput Science*, 2016, pp. 11-16.

[18] Uddin, S. Khan, A. Hossain, M.E. and Moni, M.A. "Comparing Different Supervised

Machine Learning Algorithm for Disease Prediction", *BMC Medical Informatics and Decision Making*, 19(281), https://doi.org/10.1186/s12911-019-1004-8, 2019.

[19] Hung, C.Y. Chen, W.C. Lai, P.T. Lin, C.H. Lee, C.C. "Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database", *Engineering in Medicine and Biology Society (EMBC*), 2017 39th Annual International Conference of the IEEE, 1, pp. 3110–3113.

[20] Hussain L, Ahmed, A. Saeed, S. Rathore, S. Awan, I.A. Shah, S.A. Majid, A. Idris, A. and Awan, A.A. "Prostate cancer detection using machine learning techniques by employing a combination of features extracting strategies", *Cancer Biomarkers*. 21(2), 2018, pp. 393–413.

[21] Mitchell, T.M. "Machine Learning WCB".: McGraw-Hill Boston, MA, 1997.

[22] Bradley, R. Taokopoulos, Kim, M. Kokkinos, Y., Panagiotakos, T. Kennedy, J. De Meyer, G., Watson, P. and Elliot J. "Predicting early risk of chronic kidney disease in cats using routine clinical laboratory tests and machine learning", *Journal of Veterinary Internal Medicine*, 33(6), 2019, pp. 2644 – 2656.

[23] Kuo, C.C. Chang, C.M. Liu, K.T. and Lin, W.K. et al. "Automation of the kidney function prediction and classification through ultrasound-based kidney imaging using deep learning", *NPJ Digital Medicine*, 2019, DOI: 10.1038/s41746-019-0104-2.

[24] Chaurasia, V., Pal, S., and Tiwari, B. B. Chronic kidney disease: a predictive model using decision tree. International Journal of Engineering Research and Technology, 2018, pp. 1781-1794.

[25] Xiao, J., Ding, R., Xu, X., Guan, H., and Feng, X. "Comparison and development of machine learning tools in the prediction of chronic kidney disease progression"0, *Journal of Translational Medicine*. 17, 2019, pp. 119.

[26] Makino, M. Yoshimoto, R. Ono, M. Itoko, T. et al., "Artificial Intelligence Predicts the Progression of Diabetic Kidney Disease Using Big Data Machine Learning", Scientific Reports, 9, 2019, pp. 11862.

[27] Heroku. [Online] https://www.heroku.com/home (Accessed February 22, 2020).

[28] Kadam, V.R., Soujanya, K.L., and Singh, P. "Disease prediction using Deep Learning based on patient treatment history", *International Journal of Recent Technologies and Engineering*, 7(6), 2019, pp. 2277-3878.

[29] Chakrapani, Raj, S., and Singh, V. "Detection of Chronic Kidney Disease using Artificial Neural Network", *International Journal of Applied Engineering Research*, 10(14), 2019, pp. 0973-4562.

[30] Kriplani, H. Patel, B. and Roy, S. "Prediction of Chronic Kidney Diseases Using Deep Artificial Neural Network Technique", *Springer Nature Switzerland AG*, 2019, pp. 179-187.