**Comprehensive Performance assessment of Deep Learning Models in Early Prediction and Risk Identification of Chronic Kidney Disease**

**Shamima Akter1, Ahsan Habib2,Md. Ashiqul Islam3, Md. Sagar Hossen4,Wasik Ahmmed Fahim5, Puza Rani Sarkar6, Manik Ahmed,PhD7**

1Department of Bioinformatics and Computational Biology, George Mason University, Fairfax, Virginia, USA.

2 Department of Computer Science and Engineering, Eastern University, Dhaka, Bangladesh.

3,4,5,6 Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh

7Department of Civil Engineering, Virginia Tech, Blacksburg, Virginia, USA

**ABSTRACT**  The incidence of chronic kidney disease (CKD) is rising rapidly around the globe. Asymptomatic CKD is common and guideline-directed monitoring to predict CKD by various factors is underutilized. Computer-aided automated diagnostic (CAD) can play a major role to predict CKD. CAD systems such as deep learning algorithms are pivotal in disease diagnosis due to their high classification accuracy. In this paper, various clinical features of CKD were utilized and seven state-of-the-art deep learning algorithms (ANN, LSTM, GRU, Bidirectional LSTM, Bidirectional GRU, MLP, and Simple RNN) were implemented for the prediction and classification of CKD in the domain of Internet of Medical Things (IoMT) platform. The proposed algorithms were applied based on artificial intelligence by extracting features from pre-processed and fitted CKD datasets. The study measured accuracy, precision, recall, and calculated the loss and validation loss in prediction. Further, the study analyzed computation time and prediction ratio, and AUC to evaluate the model performance along with their statistical significance to compare their performances. While classifying CKD, algorithms such as ANN, Simple RNN, and MLP provided high accuracy of 97%, a good prediction ratio along reduced time. The model outperforms traditional data classification techniques by providing superior predictive ability. Subsequently, the experimental process on the IoMT along with the help of predictive analytics can utilize the advances in deep learning to provide a promising framework to predict CKD and beyond. The study is the first fundamental step toward realizing the potentiality to classify and predict CKD using deep learning models and its associated risk factors.

**INDEX TERMS Artificial Neural Network, Chronic Kidney Disease, Classification, Deep Learning, IoMT**

I. INTRODUCTION

CKD is one of the most crucial health concerns due to its increased prevalence globally[1] and includes conditions damaging the kidneys slowly and reducing the ability to perform the essential functions of the body for a longer time. CKD is associated with complications such as renal failure, high blood pressure, anemia, nerve damage, etc.[2]. An estimated 2.2 million people around the world are plagued by renal failure. For instance, CKD has affected a large portion of the population in other developing countries such as Pakistan, India, Nepal, Bangladesh, Bhutan, Sri Lanka, and Afghanistan [3]. In addition,750,000 Americans every year suffer from CKD every year [4]. It is alleged that multiple risk factors (not limited to) such as the history of renal failure, high blood pressure, or diabetes, etc. are required to monitor each year for any abnormal test results [5]. Few blood tests are commonly used to detect CKD; (i) determine the glomerular filtration rate (eGFR), (ii) verify the concentration of albumin in the blood and urine, (iii) measure the blood urea nitrogen (BUN) index, and creatinine (CR) [6]. These circumstances lead to two major concerns (i) reliability of the screening test and ii) rising cost. First, there is no conclusive evidence that relying on screening tests can not help one suspected patient to prevent the prognosis of CKD, because the disease is highly dependent on epidemiology and other clinical features. Second, CKD has no signs and symptoms in its early stages, diagnostics testing is one of the ways to distinguish whether a patient has the renal disease or not. Once diagnosed, the patients follow various stages of CKD leading to end-stage renal disease (ESRD) which requires kidney transplants or dialysis to save people's lives (Figure 1) [7]–[9]. Kidney transplants or hemodialysis are very costly and many patients in underdeveloping countries cannot afford these treatments [10]. Health care has been digitalized, which has resulted in the creation of vast new data sets, and these are electronic medical record (EMR) systems, health insurance claims data, X-rays, lab reports, etc. Due to these vast available data, the conventional medical facility shows a limited capacity to predict CKD effectively and accurately. Hence, performing predictive health analytics to harness data is imperative.

Predictive tools such as machine learning (ML) and deep learning (DL models/algorithms can be used to overcome the limitations of traditional healthcare management [11]. Application of DL-based diagnosis may reduce unnecessary and invasive procedures to improve the efficacy and Logo, company name

Description automatically generatedsustainability of existing health care practices. Utilizing DL’s knowledge discovery capabilities, such as data mining and classification techniques, it is now possible to handle massive and valuable data to improve medical diagnosis and prognosis in decision making [9]. When health care providers combine this information with other data sources, they can create new solutions with the support of predictive analytics for early CKD diagnosis, associated health risks, and even prescriptive analytics for precision medicine. Early detection of CKD can prevent ESRD progression which is achievable by DL models prediction and subsequently reduce the cost. The researchers used DL models and attained very good performance in classifying diseases such as liver disease [12], heart disease [13], and kidney disease. Moreover, the application of DL algorithms helps to develop a fast-acting, non-invasive, and easily accessible platform that is comprised of various data related to kidney disease. This usage of DL will eventually create a valuable supporting tool for early, accurate and fast diagnosis of CKD. Hence, the current work contributes to the development of health monitoring framework which can be used as an IoMT portal based on DL algorithms.

Figure 1. CKD progression in different stages

The goals of the research are to showcase how CKD can be diagnosed efficiently through prediction and classification by using DL algorithms. To achieve the goal, seven DL algorithms- ANN, LSTM, GRU, Bidirectional LSTM, Bidirectional GRU, MLP, and Simple RNN are proposed in the study. These algorithms were then extensively compared based on their accuracy and error(s) to classify CKD diseases. Subsequently, prediction ratio and computational times were calculated to evaluate the model performances. Further, the statistical significance was carried out to validate the outcome. The developed system and model were applied to the CKD dataset, which is publicly available on the UCI machine learning repository [14]. Therefore, the key contributions of the study paper are as follows:

1. The study involved predicting and diagnosing CKD by using seven deep learning algorithms. These models have not been explored extensively relating to CKD. Subsequently, the paper compares and evaluates the performances of the models against each other.
2. The study identified the risk factors associated with CKD that can prevent disease progression to end-stage, and
3. The study demonstrated the use of the IoMT platform to predict CKD.

The remainder of the paper is structured as follows: Section II examines the related studies performed by researchers using ML and DL algorithms along with gaps in the extant literature. Section III discusses the proposed algorithms along with detailed descriptions. Section IV explains the experimental analysis and results. Further, a detailed discussion along with limitations is presented in Section V. Finally, Section V summarizes the conclusion of the research.

II. RELATED STUDIES

While examining the extant studies, it is evident that predicting CKD has become a key focus among researchers. The studies emphasized the utilization of ML and DL algorithms. However, the prominence of using deep learning models casts interest among researchers in recent years.

To monitor and diagnose chronic diseases, machine learning-related techniques have been used [15]. For example, the authors implemented seven machine learning algorithms including ANN and linear support vector machine (LSVM) to predict CKD. They used the CKD data set from the UCI repository. Three feature selection methods (i) filter (ii) wrapper, and (iii) embedded method were used to extract important features. Further, they obtained the highest accuracy of 98.46% using LSVM[16]. Chen et al. [15] applied three models to the UCI dataset. They used KNN, SVM, and SIMCA (soft independent modeling of class analogy) to calculate the patient's risk. The SVM and KNN models achieved the highest accuracy of 99.7 % [17]. In addition, six classification algorithms; Naive Bayes, MLP (Multilayer Perceptron), SVM, J48, and Decision Tree were used to assess the accuracy and effectiveness of classification of CKD. The results showed that MLP provided 99.75% accuracy [18].

Notable DL algorithms/classifiers along with hybrid versions were observed in the extant literature. The primary focuses are on the fitness of utilizing DL methods and discussing these method’s performance in diagnosing CKD. For instance, researchers used a sensor data set, extracted the features, and classified CKD by applying a Convolutional Neural Network-Support Vector Machine (CNNSVM). The concentration of urea in the saliva sample was measured to detect CKD. The study showed 96.59% prediction accuracy for the proposed algorithm[19]. Another study utilized a deep convolutional neural network (DNN) to distinguish serum potassium levels from 449,380 patients observed at Mayo Clinic’s Rochester, Minnesota, and was consequently confirmed using retrospective data from the Mayo Clinic in Minnesota, Florida, and Arizona. The study used ECG to detect hyperkalemia in CKD patients where the deep-learning model detected hyperkalemia with high sensitivity (90%) with an area under the curve (AUC) between 0.853 and 0.90 [20]. In another study, Heterogeneous Modified Artificial Neural Network (HMANN) was applied to describe the different architectures, colors, and locations of kidney stones. They achieved high accuracy (97.50%) and a substantial reduction of the required time. They used kidney ultrasound images to detect and segment kidney stones [6]. A Deep Neural Network (DNN) classifier was used to predict CKD and its severity level. The model classified CKD with 98.25% accuracy which later was increased to 99.25% by the PSO(Particle Swarm Optimization)feature selection method [21].

Limited study has been observed on the study utilizing the IoMT platform. Notably, a study emphasized an adaptive hybridized deep convolutional neural network (AHDCNN) for the early detection of kidney disease. CNN-based algorithm model was implemented to improve the classification accuracy by reducing the feature dimension. the model showed of 97%Accuracy. The study used a health monitoring framework as part of the IoMT portal. [11]. Another study developed Ensembling Multi-stage Deep Learning Approach (EMSDLA) to assess tumors in the kidney. For kidney and kidney tumors, the average Dice score is 0.96 and 0.74 on 90 unknown test cases [22]. The study claimed that the findings can advance tumor segmentation on the IoMT platform.

Further, the Adaptive Neuro-fuzzy Inference System (ANFIS) was utilized to help determine chronic renal failure. Based on the fuzzy method, ANFIS networks estimated GFR with a high degree of accuracy [23]. Researchers utilized 10 ResNet models to predict eGFR and 10 XGBoost models to classify CKD. The models provided 85.6%accuracy. In another application, researchers applied deep convolutional neural network (CNN) based eras applications with the help of transfer learning and fine-tuning to detect human emotion from the eyes and surroundings areas[24].

Overall, we can assert that most of the researchers largely applied ANN and CNN-based models including modified ANN and CNN to predict CKD, but the application of a wide range of other DL algorithms i.e., Recurrent Neural Network, SimpleRNN, LSTM, and GRU are missing. Alongside, the performance evaluation among various DL models is lacking in the extant literature. The reliance on one model or its hybrid edition does not warrant the model performance and so does its accuracy to predict CKD. The risk factors of CKD need to be detected to prevent CKD progression. Very limited ML studies and DL studies performed risk factor analysis of CKD. Hence, the study envisioned to fill these gaps to apprehend whether proposed advanced DL algorithms work efficiently to diagnose or classify CKD coupled with how their performance can be evaluated. Thus, the study emphasized seven deep learning algorithms along with ANN and MLP to substantiate comparative performance among the models to classify CKD accompanying risk factor analysis.

III. METHODS

A. DATA RETRIEVAL AND DESCRIPTION

The real-time data was collected from the UCI repository [14]. In this dataset, the number of instances is 400. From the test report analysis, 250 patients are affected in CKD and 150 patients are not affected. This dataset contains 25 attributes, where 11 attributes are numeric and the remaining 14 are nominal (Table 1).

Table 1. Data attribute and description

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **Full-Form** | **Type** | **Description/Unit** |
| age | Age | Numerical | Years |
| bp | Blood Pressure | Numerical | mm//Hg |
| sg | Specific Gravity | Nominal | Ranges from (1.005,1.010,1.015,1.020,1.025). |
| al | Albumin | Nominal | Ranges from 1-5.0 |
| su | Sugar | Nominal | Severity indicated by 5 level |
| RBC | Red Blood Cells | Nominal | Normal or abnormal |
| pc | Pus Cell | Nominal | 5-10 normal for males and females |
| PCC | Pus Cell clumps | Nominal | Less than 10 is normal. |
| ba | Bacteria | Nominal | Up to 10,000 is normal. |
| bgr | Blood Glucose Random | Numerical | <140 mg/dL is normal and > 200 mg/dL |
| bu | Blood Urea | Numerical | mg/dl |
| sc | Serum Creatinine | Numerical | Higher-level is risky |
| sod | Sodium | Numerical | mEq/L |
| pot | Potassium | Numerical | mEq/L |
| hemo | Hemoglobin | Numerical | Hemo is gms |
| PCV | Packed Cell Volume | Numerical |  |
| WC | White Blood Cell Count | Numerical | Cell counts in number. |
| RC | Red Blood Cell Count | Numerical | Higher and less than normal increase risk. |
| htn | Hypertension | Nominal | Yes or no (categorical) |
| dm | Diabetes Mellitus | Nominal | < 140 mg/dL is normal. |
| cad | Coronary Artery Disease | Nominal | Higher increases heart attack. Less than 130 mg/dL is free from CA disease |
| appet | Appetite | Nominal | Less is good more causes disorder. |
| pe | Pedal Edema | Nominal | Yes/no), Swelling or puffiness of tissue under the skin. |
| ane | Anemia | Nominal | Yes/no, Lower than hemoglobin indicates Anemia. |
| classification | Class | Nominal | Indicates either CKD or Non-CKD patients. |

B.  RESEARCH METHODOLOGY

Co-relation Based Feature Selection (CFS), (2). Recursive Feature Elimination (RFE), (3) Lasso Regression, (4) Boruta utilized

DATA PRE-PROCESSING

Data preprocessing could be a strategy that is utilized to change over the raw information into a clean dataset. It is athe basic step to train every machine learning classier algorithm.

We performed multiple imputations (MI)- (a) Linear regression to handle missing values in the numerical columns,(b) Logistic regression to handle missing values in the categorical column.

B.1. FEATURE SELECTION

It is essential to remove unnecessary features from the dataset before training DL classifiers. Features-based classification improves performance and reduces model execution time.

Feature selection was performed CFS, RFE, Lasso and Boruta after reading the data in Pandas (python data analyze package) data frame. In addition, WFS issue

**CFS**

CFS is an algorithm that selects features based on attribute rankings. Based on the correlation heuristic assessment function [28]. CFS determines the rank of the attribute subset. The function employs an approach that generates two class labels, one associated with class and one not, and picks only correlated label class characteristics.

**RFE**

RFE is a wrapper-type feature selection method. RFE works by exploring a subset of features by beginning with all features in the training dataset and successfully eliminating features until the required number remains. This is done by fitting the provided machine learning algorithm utilized in the core of the model, ranking features by significance, removing the least important features, and re-fitting the model.

**Lasso**

Lasso Regression is a regularized linear regression in which an L1 penalty is applied. This has the effect of lowering the coefficients for input factors that have no substantial impact on the prediction task. This L2 penalty allows some coefficient values to go to zero, essentially eliminating input variables from the model and providing a kind of automatic feature selection.

**Boruta**

Boruta is a feature selection algorithm. Precisely, it operates as a wrapper algorithm over Random Forest. Boruta uses an all-relevant feature selection approach where it collects all features which are in certain conditions important to the result variable.

After collecting the data from the UCI repository, the adjusted values of the model parameters were defined. Then the dataset was randomly divided into a training set (80%) and a validation or test set (20%). We selected the parameters of the maximum average performance to build the model. Seven deep learning algorithms were implemented and customized to fit the model with the dataset and find the best fit to perform the analysis We applied input and output layers, different activation functions, and different parameters for model compiling and fitting. For ANN, we used hidden layers where the activation function; Rectified Linear Unit (ReLU), and sigmoid were used in input/hidden and output layers respectively. In the case of GRU and bidirectional GRU, 4 layers of DL models which contain 50 units each were used with 20 % dropout for handling and overfitting. Similarly, 2 layers of DL models which contain 50 units each were implemented with 20% dropouts for handling overfitting in LSTM and, Bidirectional LSTM

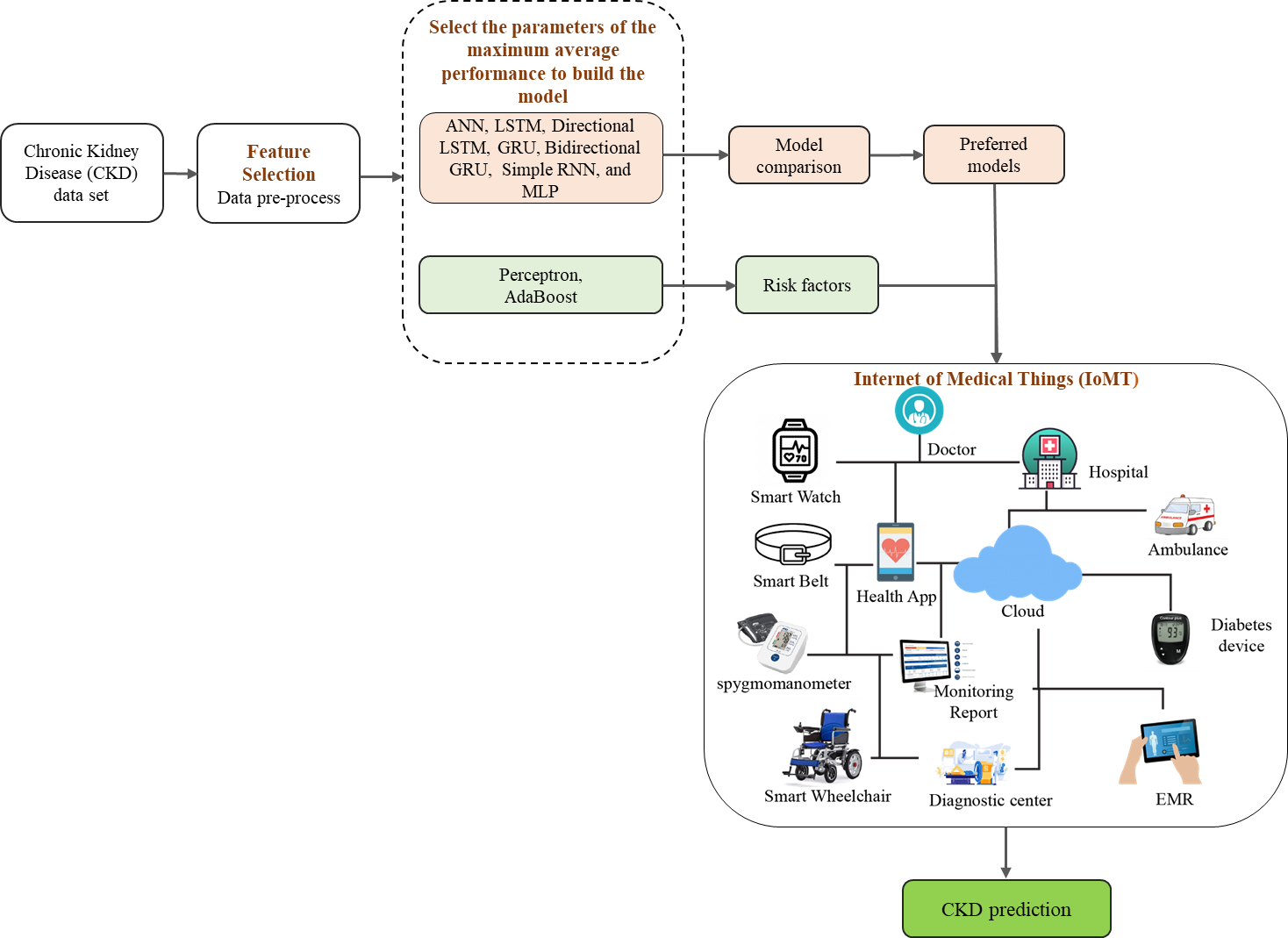


Figure 2 Proposed model

In the case of simple RNN and MLP, we used 2 layers of DL models where 32 units were in the first layer with 20% of dropout. Sigmoid and ReLU (activation function), Adam and SGD (optimizers) were used for model compiling in ANN and MLP respectively. tanh as the activation function to compile and fit parameters, SGD as an optimizer, loss as MSE (mean squared error), and 200 epochs were applied in the remaining models ( LSTM, bidirectional LSTM, GRU, and bidirectional GRU algorithm). ReLU as activation function, RMSPROP as an optimizer, and MSE as the error was implemented for SimpleRNN (Table 2).

Table 2.Models activation functions, optimizer, and loss

|  |  |  |  |
| --- | --- | --- | --- |
| Models | Activation function | Optimizer/Tuning | Loss calculation |
| ANN | Sigmoid and ReLU | Adam | Binary cross-entropy |
| LSTM | tanh | SGD\* | Mean Squared Error(MSE) |
| Bidirectional LSTM | tanh | SGD | MSE |
| GRU | tanh | SGD | MSE |
| Bidirectional GRU | tanh | SGD | MSE |
| SimpleRNN | ReLU | RMSPROP | MSE |
| MLP | Sigmoid and ReLU | SGD | Binary cross-entropy |

\**(SGD) - Stochastic Gradient Descent; Root Mean Squared Propagation RMSPROP*)

We calculated accuracy, Precision, Recall, F1 score, the loss and validation loss of the seven models, and visualized the model performance through the AUC ROC curve. Further, the study evaluates the prediction ratio and computation time of the models. The parameter score of the current parameter combination was used to compute the average performance. We trained our data 200 times in the training set for each model and tested it with the testing dataset. The optimum model was predicated on the testing/verification set to obtain the prediction result. After that, we made comparisons of the models with each other. All processing, visualization, and computation were done on Google Collaboration using python programming. The significance of the comparison among DL models in terms of accuracy was evaluated through Wilcoxon signed-rank test using R and the Deep dominance test in python.

D. MODEL DESCRIPTION

We applied seven deep learning algorithms. Those are ANN, LSTM, GRU, Bidirectional LSTM, Bidirectional GRU, Multi-Layer Perceptron, and Simple RNN. Further, Adaboost and perceptron are using to find out the significant risk factors of CKD. The models are described as follows:

D.1.ANN

Artificial Neural Network (**ANN**) is a computational algorithm developed like a human brain where neuron nodes are interconnected like a web. ANN algorithm can be used for both machine learning and pattern recognition. ANN can learn from past data or example data for classification and prediction. Figure 3 depicts the generic architecture of ANN.

Figure 3.Proposed model

A picture containing pool ball

Description automatically generated

Figure . Sample architecture/process of ANN [26]

There are 3 layers in ANN. These are (1) Input layer, (2) Hidden layer, and (3) Output layer. The general equation of ANN is:

................................................................................. (1)

*a* = f(wT.p+b)……………………………………….. (1.1)

where f is the activation function, wT are the weights and b is the bias term.

D.2. LSTM

Long short-term memory (LSTM) is a deep learning algorithm that resembles recurrent neural network (RNN) where connections between nodes form a directed graph along a chronological sequence [27]. LSTM is an algorithm that can retain information for a long time. Depending on time series data it can classify, predict and process data. LSTM algorithms retain information with the support of cell and memory manipulation with gates. Figure 4 describes the generic architecture of LSTM

A screenshot of a computer

Description automatically generated with low confidence

Figure . Sample architecture/process of LSTM[28]

LSTM has three gates: 1. Input Gate(it), 2. Forget Gate(ft), 3. Output Gate (ot). The gate equations are as follows:

For gates,

………….…………............ (2)

…………………………. (2.1)

……………………….… (2.2)

For Cells,

……………………….... (3)

……..........………………..... (3.1)

…………………………......…… (3.2)

Here,

ct = Cell state at timestamp (t)

c’t= Candidate of cell state at timestamp (t)

D.3. GRU

Gated Recurrent Unit (GRU) is an RNN algorithm that uses the hidden states to transfer information. GRU algorithm uses two vectors: weight (W) and unit (U). These two vectors decide what information should take or not for the output. Without removing it through time, they (two vectors) can be trained to keep information for a long time. Figure 5 shows the generic architecture of GRU The equation of gates is as follows:

A screenshot of a computer

Description automatically generated with low confidence

Figure 6. Sample architecture/process of GRU [29]

Update Gate(zt):

The formula for update gate is,

…………………………… (4)

Here, x\_t is multiplied by its weight W(z) when it is plugged into the network unit. The same process is going for h\_(t-1) were multiplied by weight U(z).

Reset Gate (rt):

The formula for update gate is,

……………………….….. (5)

Here, the formula is the same as the update gate. The main difference in weights and the usage of the gate.

D.4. BIDIRECTIONAL LSTM

When used in sequence classification, bidirectional LSTMs offer an improvement over regular LSTM. Instead of one LSTM, bidirectional LSTMs train two LSTMs on the input sequence. Bidirectional run inputs in two ways past to future and future to past. Figure 6 depicts the generic architecture of Bidirectional LSTM.

Diagram

Description automatically generated

Figure 7. Sample architecture/process of Bidirectional LSTM [30]

First computes the forward hidden sequence. Then compute the background sequence. Finally generating two computes and , we get an output: yt.

…………………… (6)

………….............. (6.1)

………………………….. (6.2)

D.5. BIDIRECTIONAL GRU

A bidirectional GRU can be called BiGRU. GRU means gated recurrent units. It is a sequence processing model which consists of two GRU’s. It is a bidirectional recurrent neural network (inputs in forward and backward direction) with only the input and forgets gates. Figure 7 describes the generic architecture of Bidirectional GRU.

Diagram

Description automatically generated

Figure 8. Sample of the process of Bidirectional GRU [31]

Forward direction: …………………………… (7)

…………...…………… (7.1)

…………… …………. (7.2)

…..………………. (7.3)

……………............ (7.4)

Backward direction: …….......................................... (8)

……………................. (8.1)

…………………........ (8.2)

…………………... (8.3)

……….………….. (8.4)

Output:

…………….. (9)

here zt update gate, rt reset gate ht iis new memory and is reset the memory.

D.6. MLP

Multi-layer perceptron (MLP) algorithm can be used for facilitating supervised learning of binary classifiers through a linear classifier or an algorithm. There are 5 primary components of perceptron: Input, Weights, Bias, Step function, weighted summation. The features must be added to train in the first layer as input. Later, the result of weights and inputs are multiplied. Figure 8 describes the generic architecture of MLP.

Graphical user interface

Description automatically generated

Figure 9. Sample architecture/process of MLP [32]

Bias value-added for shifting output function. The equation of Perceptron is given below:

………………………….. (10)

…………………..……… (10.1)

Here x0 = 1 and w0 = - θ

D.7. SIMPLE RNN

Simple RNN is the collection of neurons where it can work with the variable length of sequences. RNN has been considered as a state model for its feedback loop. The state develops over time for the nature of recurrence relation and the feedback is used in the state with one timestep delay. The delay feedback loop works as a memory causing it stores information between timesteps. Figure 9 describes the generic architecture of SimpleRNN. The recurrence relation over timesteps is given below:

………………………. (11)

Graphical user interface, application, background pattern

Description automatically generated

Figure 10. Sample architecture/process of Simple RNN[33]

Where Sk represents the state at k time.

Xk means input at k time.

Wrec and Wx are the weight parameter and free forward nets. Sk is the final output of the network with k timestep, which is typically calculated as Sk-i⋯Sk+j.

The current state Sk can be calculated from current input Xk or else previous state Sk-1 and it can predict the next state from Sk+1 where the current state is Skad current input is Xk.

D.8. ADABOOST

AdaBoost may be a collective learning process (also referred to as “meta-learning”) that was first shaped to raise the productivity of binary classifiers. AdaBoost uses an iterative tactic to soak up from the errors of weak classifiers and check out them into strong ones. Figure 10 describes the generic architecture of MLP

Graphical user interface

Description automatically generated with medium confidence

Figure 11. Sample architecture/process of AdaBoost [34]

The overall equation for AdaBoost is summarized as

F\_m = the m\_th weak classifier

= corresponding weight

IV. EXPERIMENTS AND RESULTS

In this paper, we have presented the DL model-based prediction of CKD. For classification of the disease, we evaluated the accuracy, precision, Recall, and F1 Score, ROC curve area, loss, and validation loss of the models. The results of the performance of the seven algorithms are shown accrding to four feature selection methods followed by the performance without feature selection. We determined the prediction ratio and computational time of each model for a comprehensive understanding of the models. Finally, statistical significance analyses were performed to evaluate the reliability of the performance. In addition, AdaBoost and perceptron algorithms are applied to find the significant risk factors of CKD.

A. MODEL PERFORMANCE: ACCURACY, PRECISION, RECALL, F1 SCORE

For classification prediction, it is important to explain the concept of a confusion matrix. A confusion matrix is defined by 2×2 matrices, containing 4 attributes namely true positive (TP), true negative (TN), false positive (FP), and false-negative (FN)(Table 3) [21].

Table 3. Confusion matrix

|  |  |  |
| --- | --- | --- |
|  | **Predicted:**1 | **Predicted:**0 |
| **Actual:**1 | True Positive | False Positive |
| **Actual:**0 | False Negative | True Negative |

The most widely used prediction performance parameter is accuracy. It measures the value of classified instances events and is denoted in percentage (%). For greater classification results, the accuracy should be close to 100%, as defined in Eq. (16).

…………………………… (16)

1. **Precision** computes and estimates the number of positive classes representing the truly positive class.
2. **Recall** calculates and predicts the number of positive classes in the data set overall positive instances.
3. **F1-score** is calculated as the harmonic mean of precision and recall for each model.

Precision implies the number of real positives which are correctly classified as positives and is represented in Eq. (17).

……………………………………. (17)

Recall signifies the number of real negatives which are correctly classified as positives and are represented in Eq. (18).

…………………………………..….. (18)

The F-score is a measure of the testing process' accuracy. Precisions and recall sets are used to calculate the average. The equation is expressed as:

………………………. (19)

A1 Without Feature Selection

When we used all the features in the data set and the models are fitted in the training dataset and predicted the CKD, it has been observed that SimpleRNN, MLP and ANN provided highest performance in terms of accuracy, Precision, Recall, AUC,Loss and Validation loss.

|  |  |
| --- | --- |
| ANN | Bidirectional GRU |
| |  |  |  |  | | --- | --- | --- | --- | | N = 80 | Predicted **0** (Not CKD) | Predicted **1** (CKD) |  | | Actual **0** (Not CKD) | TP | FP | **FPR** | | 28 | 0 | 0 | | Actual **1** (CKD) | FN | TN | **FNR** | | 2 | 50 | 0.067 | | |  |  |  |  | | --- | --- | --- | --- | | N = 80 | Predicted **0** (Not CKD) | Predicted **1** (CKD) |  | | Actual **0** (Not CKD) | TP | FP | **FPR** | | 25 | 3 | 0.059 | | Actual **1** (CKD) | FN | TN | **FNR** | | 4 | 48 | 0.138 | |
| LSTM | MLP |
| |  |  |  |  | | --- | --- | --- | --- | | N = 80 | Predicted **0** (Not CKD) | Predicted **1** (CKD) |  | | Actual **0** (Not CKD) | TP | FP | **FPR** | | 20 | 8 | 0.140 | | Actual **1** (CKD) | FN | TN | **FNR** | | 3 | 49 | 0.130 | | |  |  |  |  | | --- | --- | --- | --- | | N = 80 | Predicted **0** (Not CKD) | Predicted **1** (CKD) |  | | Actual **0** (Not CKD) | TP | FP | **FPR** | | 28 | 0 | 0 | | Actual **1** (CKD) | FN | TN | **FNR** | | 2 | 50 | 0.067 | |
| GRU | Simple RNN |
| |  |  |  |  | | --- | --- | --- | --- | | N = 80 | Predicted **0** (Not CKD) | Predicted **1** (CKD) |  | | Actual **0** (Not CKD) | TP | FP | **FPR** | | 26 | 2 | 0.041 | | Actual **1** (CKD) | FN | TN | **FNR** | | 5 | 47 | 0.161 | | |  |  |  |  | | --- | --- | --- | --- | | N = 80 | Predicted **0** (Not CKD) | Predicted **1** (CKD) |  | | Actual **0** (Not CKD) | TP | FP | **FPR** | | 28 | 0 | 0 | | Actual **1** (CKD) | FN | TN | **FNR** | | 3 | 49 | 0.097 | |
| Bidirectional LSTM |  |
| |  |  |  |  | | --- | --- | --- | --- | | N = 80 | Predicted **0** (Not CKD) | Predicted **1** (CKD) |  | | Actual **0** (Not CKD) | TP | FP | **FPR** | | 22 | 6 | 0.109 | | Actual **1** (CKD) | FN | TN | **FNR** | | 3 | 49 | 0.120 | | |

Figure Confusion matrix for 7 DL models

WFS Table 4. Performance table all deep learning models without feature slection

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm Name | Accuracy | Precision | Recall | F1 Score | AUC | Loss | Validation loss | Time |
| ANN | 99% | 98% | 99% | 99% | 0.99 | 0.0175 | 0.0887 | 10.88 |
| LSTM | 85% | 84% | 86% | 84% | 0.86 | 0.1501 | 0.1499 | 29.056 |
| Bidirectional LSTM | 88% | 87% | 90% | 87% | 0.90 | 0.134 | 0.15 | 87.58 |
| GRU | 85% | 85% | 88% | 85% | 0.88 | 0.1363 | 0.1398 | 63.29 |
| Bidirectional GRU | 89% | 88% | 91% | 88% | 0.91 | 0.1437 | 0.1492 | 90.63 |
| Simple RNN | 96% | 96% | 96% | 96% | 0.96 | 0.001 | 0.0186 | 42.19 |
| Multi-Layer Perceptron | 96% | 95% | 97% | 96% | 0.97 | 0.0795 | 0.01317 | 11.06 |

Table 4 provides the experimental results of the proposed seven DLclassifiers with CFS method and represents the different measures such as Accuracy,Precision, Recall, F1 Score, AUC, Loss, and Validation loss (errors) [35]. 80three96-9 ANN showed the highest accuracy of 99%. SimpleRNN and MLP, 858885 89. . ANN provided highest score for precision, Recall, and F1 score. SimpleRNN and MLP also showed high acores for Precision, Recall, and F1 measure.LSTM and GRU showed lower Precision and F! measure compare to other algorithms. While predicting CKD disease, LSTM showed lowest Precision (84%), and F1 score (84%), and Recall (86%and ).

Table 4. Performance table all deep learning models without feature slection

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm Name | Accuracy | Precision | Recall | F1 Score | AUC | Loss | Validation loss | Time |
| ANN | 96% | 95% | 97% | 96% | 0.97 | 0.04 | 0.08 | 10.54 |
| LSTM | 89% | 88% | 91% | 88% | 0.91 | 0.15 | 0.15 | 44.88 |
| Bidirectional LSTM | 86% | 85% | 88% | 86% | 0.88 | 0.15 | 0.15 | 46.00 |
| GRU | 85% | 85% | 88% | 85% | 0.88 | 0.16 | 0.16 | 47.06 |
| Bidirectional GRU | 88% | 87% | 90% | 87% | 0.90 | 0.15 | 0.16 | 51.06 |
| Simple RNN | 97% | 97% | 98% | 97% | 0.98 | 0.01 | 0.02 | 21.68 |
| MLP | 97% | 97% | 98% | 97% | 0.98 | 0.05 | 0.07 | 11.04 |

|  |
| --- |
|  |
|  |
| Chart  Description automatically generated with medium confidence |

A1 CSF

Table 4 provides the experimental results of the proposed seven DL classifiers and represents the different measures such as Accuracy, Precision, Recall, F1 Score, AUC, Loss, and Validation loss (errors) [35]. All of these measurements should have values as close to 100. We considered the classifier that handles the greatest score to be the best classification algorithm. All the models showed 80 to over 90% accuracy, where two models showed over 95 % accuracy. Out of all algorithms, ANN showed highest accuracy of 97%, followed by MLP and SimpleRNN-96% accuracy. Bidirectional LSTM, GRU, and Bidirectional GRU-achieved the accuracy of 85%. Table 4 also shows precision, recall, and F1 score for all DL models. ANN provided the highest value of Precision (97%), Recall (98%), and F1score (97%). The values of Precision, Recall, and F1 scores of Bidirectional LSTM, GRU, and Bidirectional GRU are the same i.e., 85% and 88%, 85% respectively. While predicting CKD disease, these models showt lowest Precision sand F1 measure score.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm Name | Accuracy | Precision | Recall | F1 Score | AUC | Loss | Validation loss | Time |
| ANN | 99% | 98% | 99% | 99% | 0.99 | 0.0175 | 0.0887 | 10.88 |
| LSTM | 85% | 84% | 86% | 84% | 0.86 | 0.1501 | 0.1499 | 29.056 |
| Bidirectional LSTM | 88% | 87% | 90% | 87% | 0.90 | 0.134 | 0.15 | 87.58 |
| GRU | 85% | 85% | 88% | 85% | 0.88 | 0.1363 | 0.1398 | 63.29 |
| Bidirectional GRU | 89% | 88% | 91% | 88% | 0.91 | 0.1437 | 0.1492 | 90.63 |
| Simple RNN | 96% | 96% | 96% | 96% | 0.96 | 0.001 | 0.0186 | 42.19 |
| Multi-Layer Perceptron | 96% | 95% | 97% | 96% | 0.97 | 0.0795 | 0.01317 | 11.06 |

|  |
| --- |
|  |
|  |
|  |
| Diagram  Description automatically generated with medium confidence |

A2 RFE

Table 4 provides the experimental results of the proposed seven DLclassifiers and represents the different measures such as Accuracy,Precision, Recall, F1 Score, AUC, Loss, and Validation loss (errors) [35]. All of these measurements should have values as close to 100. We considered the classifier that handles the greatest score to be the best classification algorithm. All the models showed 80 to over 90% accuracy, where three models showed 97% accuracy. Out of all algorithms, ANN showed the highest accuracy of 97%, followed by LSTM and Bidirectional LSTM -90% and 91% respectively, GRU-93% and BidirectionalGRU-91%, Perceptron-97%, and Simple RNN-97%. Table 4 shows the overall accuracy, precision, recall, and F1 score for all DL models. ANN showed the same percentage of Precision and F1 scores (97%) and Recall (98%). The values of Precision, Recall, and F1 scores of Simple RNN and MLP are the same as ANN.These three scores are 92% for GRU. Precision, Recall, and F1 score of both Bidirectional GRU and Bidirectional LSTM are the same i.e., 91%, 90%, and 90% respectively. While predicting CKD disease, LSTM showed Precision (90%), Recall (88%), and F1 score (89%).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm Name | Accuracy | Precision | Recall | F1 Score | AUC | Loss | Validation loss | Time |
| ANN | 97% | 97% | 98% | 97% | 0.98 | 0.07 | 0.10 | 10.89 |
| LSTM | 88% | 87% | 90% | 87% | 0.90 | 0.13 | 0.15 | 47.60 |
| Bidirectional LSTM | 85% | 85% | 88% | 85% | 88% | 0.12 | 0.13 | 46.95 |
| GRU | 85% | 85% | 88% | 85% | 0.89 | 0.12 | 0.12 | 46.60 |
| Bidirectional GRU | 85% | 85% | 88% | 85% | 0.88 | 0.13 | 0.13 | 91.44 |
| Simple RNN | 96% | 95% | 97% | 96% | 0.97 | 0.01 | 0.03 | 15.00 |
| MLP | 96% | 95% | 97% | 96% | 0.97 | 0.09 | 0.08 | 9.50 |

|  |
| --- |
|  |
|  |
|  |
| Chart  Description automatically generated |

C Lasso

Table 4 provides the experimental results of the proposed seven DLclassifiers and represents the different measures such as Accuracy,Precision, Recall, F1 Score, AUC, Loss, and Validation loss (errors) [35]. All of these measurements should have values as close to 100. We considered the classifier that handles the greatest score to be the best classification algorithm. All the models showed 80 to over 90% accuracy, where three models showed more than 90 % accuracy. Out of all algorithms, ANN showed the highest accuracy of 97%, followed by Perceptron-96%, and Simple RNN-95%. LSTM Bidirectional LSTM, and GRU showed same score for accuracy (79%),GRU-93% and BidirectionalGRU-86%, Table 4 also shows precision, recall, and F1 score for all DL models. ANN, MLP, and SimpleRNN scored high Precision, Recall and F1 scores. ANN achieved highest Precision, Recall, and F1 measure among all models.The values of Precision, Recall, and F1 scores of LSTM, BidirectionalLSTM, and GRU are the same; 81%, 84%, and 79% respectively and these scores are low compare to all other models.



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm Name | Accuracy | Precision | Recall | F1 Score | AUC | Loss | Validation loss | Time |
| ANN | 97% | 97% | 98% | 97% | 0.98 | 0.04 | 0.07 | 10.90 |
| LSTM | 79% | 81% | 84% | 79% | 0.84 | 0.17 | 0.16 | 24.18 |
| Bidirectional LSTM | 79% | 81% | 84% | 79% | 0.84 | 0.14 | 0.15 | 27.77 |
| GRU | 79% | 81% | 84% | 79% | 0.84 | 0.11 | 0.16 | 30.94 |
| Bidirectional GRU | 86% | 86% | 89% | 86% | 0.89 | 0.13 | 0.15 | 60.88 |
| Simple RNN | 95% | 94% | 96% | 95% | 0.96 | 0.00 | 0.04 | 21.81 |
| MLP | 96% | 95% | 97% | 96% | 0.97 | 0.04 | 0.12 | 11.04 |

|  |
| --- |
|  |
|  |
|  |
| Chart  Description automatically generated |

D. Boruta

Table 4 provides the experimental results of the proposed seven DLclassifiers with Boruta method and represents the different measures such as Accuracy,Precision, Recall, F1 Score, AUC, Loss, and Validation loss (errors) [35]. All the models showed 80 to over 90% accuracy, where three models showed 96-99% accuracy. Out of all algorithms, ANN showed the highest accuracy of 99%. SimpleRNN and MLP showed the accuracy of 97% and 96% respectively, followed by LSTM and Bidirectional LSTM -86% and 88% respectively,GRU and Bidirectional GRU-84%. Table 4 shows the overall precision, recall, and F1 score for all DL models. ANN provided highest score for precision, Recall, and F1 score. ANN,SimpleRNN and MLP also showed high acores for Precision, Recall, and F1 measure. GRU and Bidirectional GRU showed lower Precision and F1measure compare to other algorithms. While predicting CKD disease, GRU showed lowest Precision (84%), and F1 score (84%), and Recall (83% ).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm Name | Accuracy | Precision | Recall | F1 Score | AUC | Loss | Validation loss | Time |
| ANN | 99% | 98% | 99% | 99% | 0.99 | 0.02 | 0.02 | 10.83 |
| LSTM | 86% | 86% | 89% | 86% | 0.89 | 0.13 | 0.15 | 85.54 |
| Bidirectional LSTM | 88% | 87% | 90% | 87% | 0.90 | 0.16 | 0.17 | 87.04 |
| GRU | 84% | 84% | 88% | 83% | 0.88 | 0.13 | 0.15 | 0.61 |
| Bidirectional GRU | 84% | 84% | 88% | 83% | 0.88 | 0.13 | 0.15 | 50.18 |
| Simple RNN | 97% | 97% | 98% | 97% | 0.98 | 0.00 | 0.01 | 21.10 |
| MLP | 96% | 95% | 97% | 96% | 0.97 | 0.01 | 0.13 | 0.12 |

|  |
| --- |
|  |
|  |
|  |
| Chart  Description automatically generated |

B. RECEIVER OPERATING CHARACTERISTIC (ROC)/ AREA UNDER CURVE (AUC)

At different threshold settings, the AUC-ROC curve is plotted to assess the performance of classification algorithms. Receiver Operation Characteristics (ROC) denotes the probability and Area Under Curve (AUC) represent the separability measure or degree. Higher AUC (close to 1) means better performance to distinguish whether the patient has a disease or not [36]. Figure 12 shows the AUC curve where the x and y-axis represent FPR (False Positive Rate) and TPR (True Positive Rate) respectively. FPR is applied to calculate the likelihood for a particular test incorrectly by discarding the null hypothesis. It is the proportion of negative instances predicted as positive in the dataset. It is denoted in Eq. (20).

…………………………………………. (20)

TPR or sensitivity is the ratio of the number of positive instances correctly categorized as positive to the total number of correctly classified cases. It is denoted in Eq. (21).

…………………………………………. (21)

Here, ANN, SimpleRNN, and MLP showed the highest AUC of 0.98 (close to 1) indicating very good performance. Bidirectional LSTM and bidirectional GRU provided AUC of 0.90, GRU and LSTM demonstrated AUC of 0.92 and 0.88 respectively. So, the three DL algorithms: ANN, MLP, and SimpleRNN performed superior to other deep learning algorithms.

C. LOSS AND VALIDATION

Loss is defined by the error that occurred during each iteration (epoch) on the training dataset to predict the class of CKD. Loss is calculated for the testing dataset and described as validation loss or val-loss. After completing 200 epochs (which entails the number of iterations), the loss and validation of the seven DL models are shown in Figure 13 (A-G). It is observed that both loss and validation loss is showing a similar pattern for all the models. This indicates a perfect fitting of the data to models. To highlight, SimpleRNN showed minimum error as loss and validation loss among all the models in 4 feature selection method along with the data without feature selection i.e., CFS-0.006,Boruta-0.003, Lasso-0.0036, RFE-0.0123, and WFS-0.001 as loss and CFS-0.02,Boruta-0.013,Lasso-0.044, RFE-0.01, WFS-0.0186 as validation lossor Val\_loss. ANN and MLP showed low score score of loss and validation loss . The largest error as loss and validation loss was achieved by LSTM in WFS, Lasso, and RFE. GRU in CFS,Bidirectional LSTM in Boruta also showed high score for both loss and validation loss. However, the difference between loss and validation loss is very low for the DL models in all selection method suggesting the perfect fiting of data to the models. Thus we can say that models were not overfitted to the train9ng dataset. (Table 4)

|  |  |  |
| --- | --- | --- |
| A | | |
|  | | |
| B |
|  |
| C |
|  |
| D |
| Chart, line chart  Description automatically generated |
| Chart, histogram  Description automatically generatedE |
| F |
| Chart, histogram  Description automatically generated |
| G |
| Chart  Description automatically generated with medium confidence |

Figure 15 (A-G). Error as Loss of training dataset and error as validation loss of testing dataset are shown for seven deep learning models. Here val\_loss denotes the error as validation loss for the testing dataset.

D. PREDICTION RATIO

The algorithm uses a training set of features and the associated result to predict a given outcome (prediction results). To improve prediction learning techniques, the proper selection parameters for the tests of different classifiers need to be determined. Hence prediction ratio determines the ratio of correctly classified instances to incorrectly ones for a given data set. The study divided the whole data set into eight segments containing 50 instances. Figure 15 shows the prediction ratio of the seven DL algorithms. In case of WFS, CFS, and Boruta, it is observed that ANN, MLP, and SimpleRNN showed a high prediction ratio for small dataset.However the prediction ratio of MLP drops to zero with large dataset (300-400 instances).For other models, Prediction ratio was high with small data set, then reduced, at last increased with large dtaset.For Laso and the prediction ratio of MLP and Bidirectional GRU appears in a opposite direction; the ratio dropsawith higher data set while the ratio increases to 100% with large data set. This analysis shows that SimpleRNN and ANN performed best compare to to other models while predicting CKD disease. For the remaining models, the prediction ratio varies concerning different sets of data. In the case of 150 and 200 data sets, the prediction ratio dropped to ~ 60-80% for Bidirectional LSTM, GRU, and LSTM, and increased to around 100% for 400 data set

E. COMPUTATION TIME

Performance is also verified by calculating the computation time of all models. The time for the computation to predict CKD is considered for each model. The proposed ANN, MLP, and Simple RNN models showed low computation time i.e. 28 sec, 31 sec, and 38 sec respectively that indicates high performance. Figure 15 demonstrates the computational time of the implemented seven DL models. GRU took the highest time;107 sec, LSTM, Bidirectional LSTM, and Bidirectional GRU completed the prediction or classification in moderate time: 62 sec, 69 sec, and 86 sec respectively.

Chart, bar chart

Description automatically generated

Figure 17. Computational time of the DL models

F. STATISTICAL TEST OF SIGNIFICANCE

We performed a statistical test of significance to validate the findings that are likely is real, reliable, and not occurred by chance. To achieve so, the study used Wilcoxon signed-rank test and calculated p-values among all models based on accuracy[37]. Table 6 depicts the results of the p-value of pair-wise comparison of the models. To explain, the p-value among ANN and other models (LSTM, Bidirectional LSTM, GRU, Bidirectional GRU) is lower than 0.05 (p-value ~0.005- 0.006) whereas the p-value among ANN and two other models (SimpleRNN and MLP) is greater than 0.05 (p-value ~0.07- 0.15). Similar results were observed for SimpleRNN and MLP. Hence, the observed accuracy of ANN, MLP, and Simple RNN is significantly higher than that of other models to validate the outcome from the previous findings.

Table 5. Wilcoxon signed -rank test results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Models | ANN | LSTM | Bidirectional LSTM | GRU | Bidirectional GRU | Simple RNN | MLP |
| ANN | - | - | - | - | - | - | - |
| LSTM | 0.005 | - | - | - | - | - | - |
| Bidirectional LSTM | 0.006 | 0.551 | - | - | - | - | - |
| GRU | 0.005 | 0.050 | 0.040 | - | - | - | - |
| Bidirectional GRU | 0.006 | 0.605 | 0.156 | 0.223 | - | - | - |
| Simple RNN | 0.152 | 0.005 | 0.005 | 0.006 | 0.006 | - | - |
| MLP | 0.072 | 0.005 | 0.005 | 0.005 | 0.005 | 0.032 | - |

(Wilcoxon test was performed using R, 95% confidence level with the default parameters.)

G. DEEP DOMINANCE TEST

Along with the Wilcoxon signed test, the study performed the deep dominance test to compare the performance of the seven algorithms. The test determines “Almost Stochastic Dominance” which follows a measurement of stochastic dominance (between two

Algorithms [38]. Each algorithm was compared against the other, and deep dominance (ϵ) was measured between 0 to 1.0. 0 corresponds to perfect stochastic dominance of one algorithm (X) over another (Y) and 1 corresponds to perfect stochastic dominance of Y over X. Table 7 shows the results from the Deep Dominance test.

Table 6. Deep Dominance test results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Models | ANN (A) | LSTM (B) | Bidirectional LSTM (C) | GRU (D) | Bidirectional GRU (E) | Simple RNN (F) | MLP (G) |
| ANN(A) | - | - | - | - | - | - | - |
| LSTM (B) | , A^B | - | - | - | - | - | - |
| Bidirectional LSTM(C) | , A^C | , B ***nbt*** C | - | - | - | - | - |
| GRU (D) | , A^D | ,  B^ D | ,  C ***^*** D | - | - | - | - |
| Bidirectional GRU (E) | , A^E | , B ***nbt*** E | ,  C^ E | ,  D **nbt** E | - | - | - |
| Simple RNN (F) | , A^F | ,  B ***nbt*** F | ,  C nbt F | ,  D ***nbt*** F | ,  E ***nbt*** F | - | - |
| MLP (G) | ,  A***^*** G | ,  B ***nbt*** G | ,  C ***nbt*** G | ,  D ***nbt*** G | ,  E ***nbt*** G | ,  E ^ G | - |

**nbt** -not better than

G. RISK FACTOR ANALYSIS

We calculated and identified the risk factors of CKD for the data set through the ranking of the features using Perceptron and AdaBoostClassifier models. Figure 16 shows the risk factors of CKD for both models. After preprocessing and normalizing, both models were fitted to the CKD data, features were ranked based on importance in predicting the class of the disease [39]. Perceptron provided the ranking of 14 features as follows: al, bgr, sod, su, bp, sc, bu, wc, pot, age, rc, pcv, sg, and hemo. Al (albumin) was identified as the most important feature as the calculated value is the highest (i.e.,9.7186). In the case of the Adaboost classifier model, the important features were ranked as follows: al, sc, sod, hemo, pcv, age, sg, wc, bu, rc, pot, bgr, bp, su, and al-albumin was found also as the most important feature in classifying the disease. Adaboost provided the highest value (i.e.18) for al. Thus, in both cases, albumin is the highest-ranked feature and risk factor of CKD.

|  |  |  |
| --- | --- | --- |
| A |  | B |
| Chart  Description automatically generated |  | Chart, funnel chart  Description automatically generated |

Figure 18 (A-B). Risk factor analysis with Perceptron and AdaBoost models, here al, bgr, sod, su, bp,sc denotes for albumin, random blood sugar, sodium, blood pressure, serum creatinine respectively according to the attribute (Table 1).

A picture containing chart

Description automatically generated

# The risk of CKD is increasing rapidly, and consequently more people are suffering and dying due to a lack of proper treatment. Moreover, CKD needs to be identified at an early stage because late diagnosis leads to severe consequences and treatment becomes highly expensive. Perhaps, DL applications are key developments in recent years to combat various medical diagnoses. These approaches can potentially reduce the cost and treatment of CKD; due to the advantage of the built-in feature engineering method for DL algorithms, data is scanned through a faster learning technique and researchers can get efficient results. Thus, the efficacy of deep learning algorithms to predict CKD along with identified risk factors has been shown in this study. We used 7 deep learning algorithms on 400 CKD data for predicting the class of CKD. The proposed methods achieved high accuracy ranging from 90% to 97%. In comparison, ANN, MLP SimpleRNN achieved the highest accuracy in terms of predicting CKD compared to other deep learning models. ANN, MLP, and SimpleRNN showed 97% accuracy, GRU provided accuracy of 93%, and both Bidirectional GRU and Bidirectional LSTM showed 91 % accuracy. The rest of model LSTM showed 90% accuracy to predict CKD class. It is evident that all the algorithms provided minimum error around 0.1 and the error of both types i.e., loss and validation loss decreased with longer time and iterations. The outcome substantiates the best fitting of the data to all the deep learning models. Overall, compared to other algorithms, SimpleRNN showed the minimum error for both the testing and training data i.e., 0.003 and 0.0283 respectively. Bidirectional LSTM provided a maximum error of 0.1264 for the testing dataset. The other key findings are as follows; (1) GRU took the highest computation time (i.e., 107 sec), (2) ANN, MLP, and SimpleRNN showed similar and comparatively high prediction ratios (i.e., 0.98-1.0). These three algorithms took less computation time (28 sec) in predicting CKD (Figure 14). Hence ANN, Simple RNN, and MLP are the best fit for the CKD data. Also, these algorithms can classify the CKD data more accurately and efficiently than the remaining four algorithms. algorithms can classify the CKD data more accurately and efficiently than the remaining four algorithms.

# Further, the findings were validated by statistical significance analysis While comparing with other models, significant p-values (<0.05) were obtained for ANN, SimpleRNN, and MLP (Table 6). This substantiates the findings using the Wilcoxon test that ANN, SimpleRNN, and MLP showed superior performance. In addition, the Deep dominance test showed that ANN is better than all other models where . The values of other models (LSTM, Bidirectional LSTM, GRU, and Bidirectional GRU) is ~1.0 while comparing with Simple RNN and MLP. The findings indicate that fours models are not better than (*nbt*) SimpleRNN and MLP. Overall, both significance tests support the results.

# Further, the study identified the risk factors of CKD. For finding the risk factor the Perceptron and Adaboost classifier models were used. Both models successfully provided albumin as the responsible risk factor of CKD. Albumin usually found in blood and kidneys filters this protein. Thus, albumin is not commonly observed in urine. The presence of albumin in urine indicates that the kidney nephrons are damaging and lose the ability to filter albumin from the body. This increased amount of albumin in urine indicates CKD disease [40]. High-risk individuals having diabetes, hypertension, etc. are generally recommended to check albumin in the urine. The best models and risk factors identified in this study can be implemented in IoMT which will enable remote monitoring of CKD. Therefore, IoMT can be implemented for (1) improved diagnosis and treatment, (2) effective CKD management, and (3) reduced cost.

The study has limitations too. In the presence of relevant data, DL algorithms can build complex systems to obtain accurate decisions. Generally, DL models perform well as long as there is enough data [11]. However, during optimization, overfitting can be prevented by customizing the parameters because the sample size was fairly small in this study  [41]. Another issue is handling the missing value; 10 features have missing

values in the categorical numeric column in the data set and those columns were dropped. Subsequently, the number of attributes became less that possess overfitting of the dataset to the models.

VI. CONCLUSION

This paper proposed a methodology utilizing seven deep learning algorithms to detect CKD and identify risk factors that are crucial for early diagnosis to prevent the prognosis of the disease to end-stage. All the proposed models obtained notable results with more than 90% prediction accuracy with minimum error. Thus, this research examines the efficacy of seven DL algorithms for predicting CKD. When comparing the models, ANN, MLP, and Simple RNN were found to be most powerful, providing an accuracy of 97 % in predicting the disease. In this study, applied DL algorithm MLP and Adaboost identified albumin as the risk factor of CKD. These results can boost the medical community by predicting CKD and the risk factors.

VII. REFERENCES

[1] J. Coresh, “Update on the burden of CKD,” *J. Am. Soc. Nephrol.*, vol. 28, no. 4, pp. 1020–1022, 2017.

[2] R. Kazancioğlu, “Risk factors for chronic kidney disease: an update,” *Kidney Int. Suppl.*, vol. 3, no. 4, pp. 368–371, 2013.

[3] G. Abraham *et al.*, “Chronic kidney disease hotspots in developing countries in South Asia,” *Clin. Kidney J.*, vol. 9, no. 1, pp. 135–141, 2016.

[4] T. J. Hoerger *et al.*, “The future burden of CKD in the United States: a simulation model for the CDC CKD Initiative,” *Am. J. Kidney Dis.*, vol. 65, no. 3, pp. 403–411, 2015.

[5] M. Almasoud and T. E. Ward, “Detection of chronic kidney disease using machine learning algorithms with least number of predictors,” *Int. J. Soft Comput. Its Appl.*, vol. 10, no. 8, 2019.

[6] F. Ma, T. Sun, L. Liu, and H. Jing, “Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network,” *Futur. Gener. Comput. Syst.*, vol. 111, pp. 17–26, 2020.

[7] M. A. Islam, S. Akter, M. S. Hossen, S. A. Keya, S. A. Tisha, and S. Hossain, “Risk Factor Prediction of Chronic Kidney Disease based on Machine Learning Algorithms,” in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 2020, pp. 952–957.

[8] I. U. Ekanayake and D. Herath, “Chronic Kidney Disease Prediction Using Machine Learning Methods,” in *2020 Moratuwa Engineering Research Conference (MERCon)*, 2020, pp. 260–265.

[9] B. Khan, R. Naseem, F. Muhammad, G. Abbas, and S. Kim, “An empirical evaluation of machine learning techniques for chronic kidney disease prophecy,” *IEEE Access*, vol. 8, pp. 55012–55022, 2020.

[10] A. Al Imran, M. N. Amin, and F. T. Johora, “Classification of chronic kidney disease using logistic regression, feedforward neural network and wide & deep learning,” in *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, 2018, pp. 1–6.

[11] G. Chen *et al.*, “Prediction of chronic kidney disease using adaptive hybridized deep convolutional neural network on the internet of medical things platform,” *IEEE Access*, vol. 8, pp. 100497–100508, 2020.

[12] S. Akter, H. U. Shekhar, and S. Akhteruzzaman, “Application of Biochemical Tests and Machine Learning Techniques to Diagnose and Evaluate Liver Disease,” *Adv. Biosci. Biotechnol.*, vol. 12, no. 6, pp. 154–172, 2021.

[13] V. Shorewala, “Early detection of coronary heart disease using ensemble techniques,” *Informatics Med. Unlocked*, p. 100655, 2021.

[14] D. Dua and C. Graff, “UCI Machine Learning Repository,” *University of California, School of Information and Computer Science*, 2019. http://archive.ics.uci.edu/ml.

[15] J. Qin, L. Chen, Y. Liu, C. Liu, C. Feng, and B. Chen, “A machine learning methodology for diagnosing chronic kidney disease,” *IEEE Access*, vol. 8, pp. 20991–21002, 2019.

[16] P. Chittora *et al.*, “Prediction of chronic kidney disease-a machine learning perspective,” *IEEE Access*, vol. 9, pp. 17312–17334, 2021.

[17] Z. Chen, X. Zhang, and Z. Zhang, “Clinical risk assessment of patients with chronic kidney disease by using clinical data and multivariate models,” *Int. Urol. Nephrol.*, vol. 48, no. 12, pp. 2069–2075, 2016.

[18] H. Kriplani, B. Patel, and S. Roy, “Prediction of chronic kidney diseases using deep artificial neural network technique,” in *Computer aided intervention and diagnostics in clinical and medical images*, Springer, 2019, pp. 179–187.

[19] N. Bhaskar and M. Suchetha, “An approach for analysis and prediction of CKD using deep learning architecture,” in *2019 International Conference on Communication and Electronics Systems (ICCES)*, 2019, pp. 1660–1664.

[20] C. D. Galloway *et al.*, “Development and validation of a deep-learning model to screen for hyperkalemia from the electrocardiogram,” *JAMA Cardiol.*, vol. 4, no. 5, pp. 428–436, 2019.

[21] S. K. Lakshmanaprabu, S. N. Mohanty, S. Krishnamoorthy, J. Uthayakumar, and K. Shankar, “Online clinical decision support system using optimal deep neural networks,” *Appl. Soft Comput.*, vol. 81, p. 105487, 2019.

[22] G. Santini, N. Moreau, and M. Rubeaux, “Kidney tumor segmentation using an ensembling multi-stage deep learning approach. A contribution to the KiTS19 challenge,” *arXiv Prepr. arXiv1909.00735*, 2019.

[23] J. Norouzi, A. Yadollahpour, S. A. Mirbagheri, M. M. Mazdeh, and S. A. Hosseini, “Predicting renal failure progression in chronic kidney disease using integrated intelligent fuzzy expert system,” *Comput. Math. Methods Med.*, vol. 2016, 2016.

[24] M. N. R. Shuvo, S. Akter, M. A. Islam, S. Hasan, M. Shamsojjaman, and T. Khatun, “Recognizing Human Emotions from Eyes and Surrounding Features: A Deep Learning Approach.”

[25] Y. Amirgaliyev, S. Shamiluulu, and A. Serek, “Analysis of chronic kidney disease dataset by applying machine learning methods,” in *2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*, 2018, pp. 1–4.

[26] Ben Dickson, “What are artificial neural networks (ANN),” *Tech Talks*, 2019. https://bdtechtalks.com/2019/08/05/what-is-artificial-neural-network-ann/.

[27] U. Naqvi, A. Majid, and S. A. Abbas, “UTSA: Urdu Text Sentiment Analysis Using Deep Learning Methods,” *IEEE Access*, 2021.

[28] R. Dolphin, “LSTM Networks | A Detailed Explanation,” *towards data science*, 2020. https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9.

[29] R. Dey and F. M. Salem, “Gate-variants of gated recurrent unit (GRU) neural networks,” in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, 2017, pp. 1597–1600.

[30] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” *arXiv Prepr. arXiv1508.01991*, 2015.

[31] X. Liu, Y. Wang, X. Wang, H. Xu, C. Li, and X. Xin, “Bi-directional gated recurrent unit neural network based nonlinear equalizer for coherent optical communication system,” *Opt. Express*, vol. 29, no. 4, pp. 5923–5933, 2021.

[32] D. W. Ruck, S. K. Rogers, and M. Kabrisky, “Feature selection using a multilayer perceptron,” *J. Neural Netw. Comput.*, vol. 2, no. 2, pp. 40–48, 1990.

[33] Dishashree Gupta, “Fundamentals of Deep Learning – Introduction to Recurrent Neural Networks,” *Analytics Vidhya*, 2017. https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/.

[34] J. Li, “Boosting algorithm: AdaBoost,” *towards data science*, 2017. https://towardsdatascience.com/boosting-algorithm-adaboost-b6737a9ee60c.

[35] J. Singh *et al.*, “Performance Assessment of Classification Algorithms on Early Detection of Liver Syndrome,” *Artif. Intell. Med.*, vol. 4, no. 1, pp. 42–46, 2018.

[36] N. A. Almansour *et al.*, “Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study,” *Comput. Biol. Med.*, vol. 109, pp. 101–111, 2019.

[37] B. Rosner, R. J. Glynn, and M. Ting Lee, “Incorporation of clustering effects for the Wilcoxon rank sum test: a large‐sample approach,” *Biometrics*, vol. 59, no. 4, pp. 1089–1098, 2003.

[38] R. Dror, S. Shlomov, and R. Reichart, “Deep dominance-how to properly compare deep neural models,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2773–2785.

[39] W. Wang, G. Chakraborty, and B. Chakraborty, “Predicting the risk of chronic kidney disease (ckd) using machine learning algorithm,” *Appl. Sci.*, vol. 11, no. 1, p. 202, 2021.

[40] K. D. Liu *et al.*, “Urine neutrophil gelatinase–associated lipocalin levels do not improve risk prediction of progressive chronic kidney disease,” *Kidney Int.*, vol. 83, no. 5, pp. 909–914, 2013.

[41] M. Elhoseny, K. Shankar, and J. Uthayakumar, “Intelligent diagnostic prediction and classification system for chronic kidney disease,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–14, 2019.

**A person smiling for the camera

Description automatically generated with low confidence**Shamima Akter pursuing her Ph.D. at George Mason University after completing MS in Bioinformatics at Virginia Tech in 2017.From 2017 to 2019, she was a Research Associate at James wager and Maurer lab performing as bioinformatics.Her research interests include artificial intelligence, machine learning in biomedical science i.e., liver disease, chronic kidney disease, mathematical modeling on Alzheimer's diseases, and computational biology. He is the author of three books chapters and more than 10 peer-reviewed articles.

**Ahsan Habib** received his B.S degree in Computer Science & Engineering from Eastern University, Dhaka, Bangladesh. In 2019, he is started working as Software Engineer in Indetechs Software Limited, Dhaka, Bangladesh. His research interests include Artificial Intelligence, Machine learning, Deep learning, and Natural Language Processing. He has been working on bubble tracking since January 2021 using Mask RCNN deep learning technique.

**Md. Ashiqul Islam** was born in Dhaka, Bangladesh in 1998. He received his BSc in Computer Science and Engineering from Daffodil International University.

Recently he is working as a Research Scholar in DIU Journal Analytica R & D Lab. He has hands-on experience in Deep Learning and Machine Learning based techniques. His research interest’s areas are Blockchain Technology, Cryptocurrency, Applied Cryptography, Satellite, and Radar Communication, Deep Learning and Machine Learning Techniques, and IoT. He is the reviewer of two journals. One is “Acta Agriculturae Scandinavica, Section-B, Soil, and Plant Science”, Taylor and Francis and another one are “Bulletin of Electrical Engineering and Informatics”, Elsevier. He is also a Technical Program Committee member of “The Second International Conference on Modern Management based on Big Data (MMBD 2021)”, China.

****

**Md. Sagar Hossen** received BSc degree from the Department of Computer Science & Engineering at Daffodil International University, Bangladesh in 2021. He is currently working as a researcher at the Bangla Insight A.I, Bangladesh. He has authored or coauthored more than 14 scientific publications. His research interests include Artificial Intelligence, Computer vision, Image processing, and Human-computer Interaction, Computer security.

**Wasik Ahmmed Fahim** had completed BSc in Computer Science Engineering Daffodil International University. His ambition mainly focuses on different research projects. He has a sweet spot in machine learning and deep learning. Also, he is a good native application developer. He’s ambitious to work with Artificial Intelligence in the future.



Ms. Puja Rani Sarkar graduated in 2021 with a B.S. in computer science engineering from Daffodil International University in Dhaka, Bangladesh.

She started her career as a researcher in machine learning and deep learning algorithm. Her ambition is to work with different algorithms for predicting the types of human diseases. She is currently working with artificial intelligence for research purposes. Her research interest includes machine learning, deep learning, artificial intelligence in medical science

Manik Ahmed got his BS in Civil Engineering from Bangladesh University Engineering and Technology (BUET) in 1999 in Civil Engineering. Then He worked in the industry for 15 years mostly focusing on automation, engineering technology, and program management. He also engaged in developing different computer applications for project management such as in BIM and Primavera. He had his MS from Virginia Tech, Blacksburg, USA in 2017 and a Ph.D. from the same in 2021 in Civil engineering. Currently, he has been engaged in various research initiatives such as smart city and urban network improvement.