

Received October 25, 2021, accepted November 16, 2021, date of publication November 19, 2021, date of current version December 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3129491

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 HOSSEN³, WASIK AHMMED FAHIM³, PUZA RANI SARKAR³, AND MANIK AHMED⁴

¹Department of Bioinformatics and Computational Biology, George Mason University, Fairfax, VA 22030, USA

²Department of Computer Science and Engineering, Eastern University, Dhaka 1345, Bangladesh

³Department of Computer Science and Engineering, Daffodil International University, Dhaka 1207, Bangladesh

⁴Department of Civil Engineering, Virginia Tech, Blacksburg, VA 24061, USA

Corresponding author: Shamima Akter (sakter5@gmu.edu)

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. The proposed algorithms were applied based on artificial intelligence by extracting and evaluating features using five different approaches from pre-processed and fitted CKD datasets. In this study, we have 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 statistical significance to compare their performances. While classifying CKD, algorithms such as ANN, Simple RNN, and MLP provided high accuracy of 99%, 96%, 97% respectively, and a good prediction ratio along with reduced time. The model outperforms traditional data classification techniques by providing superior predictive ability. Subsequently, the study proposed the integration of best performing DL models in the IoMT. This proposal will assist predictive analytics to advance CKD prediction by using deep learning more efficiently and effectively. The study is the first fundamental step toward a comprehensive performance assessment 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.

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 developing countries such as

Pakistan, India, Nepal, Bangladesh, Bhutan, Sri Lanka, and Afghanistan [3]. In addition, 750,000 Americans 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 help

The associate editor coordinating the review of this manuscript and approving it for publication was Joanna Kołodziej¹.

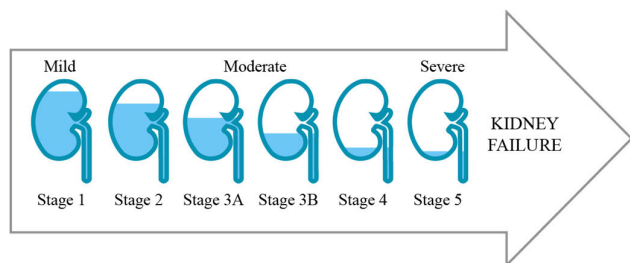


FIGURE 1. CKD progression in different stages.

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 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 under developing 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-ray, 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 sustainability 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 chronic 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.

Given that deep learning offers novel designs and better performance in many domains, we firmly think that deep

learning has much to contribute to the field of CKD. The study shows a detailed explanation of the technical specifics of deep learning (DL) architectures along with a comprehensive performance assessment of the DL methods used to predict CKD. Five sets of feature selection/ranking tools have been utilized and compared to incentivize the application of DL methods. Further, the use of statistical analysis proved the outcome more reliable and effective. In addition, the study shed some insights into the application to the development of a health monitoring framework that can be used as an IoMT portal based on DL algorithms.

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 are as follows:

- i) The study involved in comprehensive performance assessment of seven DL models including simple RNN, Bi GRU, and Bi LSTM, and GRU in predicting and diagnosing CKD.
- ii) Current data modalities were evaluated through feature selection to establish the utility of feature selection in DL models. These approaches have not been explored extensively earlier relating to predicting CKD.
- iii) The study identified the risk factors associated with CKD that can prevent disease progression to the end-stage.
- iv) Two statistical significance tests were further carried out to establish the reliability of the performance assessment.

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 VI summarizes the conclusion of the research.

II. RELATED STUDIES

While examining the extant studies, it is evident that predicting CKD has become a prime interest 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 methods' 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.

Overall, we can assert that most of the researchers essentially 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 and DL studies performed a 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. RESEARCH METHOD

Figure 2 explains the research process/model along with IoMT framework. The details are as below:

A. DATA RETRIEVAL, DESCRIPTION, AND CONCERN

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. As a result, each class has 62.5% with CKD and 37.5% without CKD. This dataset contains 25 attributes, where 11 attributes are numeric and the remaining 14 are nominal (Table 1).

This is a small dataset with a small imbalance issue. Subsequently, some concerns exist with the dataset, which might be an overfitting or generalization problem, imbalance, and the noise of the data. The researcher attempted various strategies/techniques to handle these issues. P. Yang *et al.* concluded that the ensemble technique is better than a single classifier because it is better at reducing the chance of overfitting [24]. Three feature selection methods (Filter, Wrapper method, and embedded methods) were adopted for feature selection while utilizing machine learning algorithms [16].

B. DATA PRE-PROCESSING

Missing data in the medical data set to cause a threat utilizing deep learning. It is generally believed that every attribute

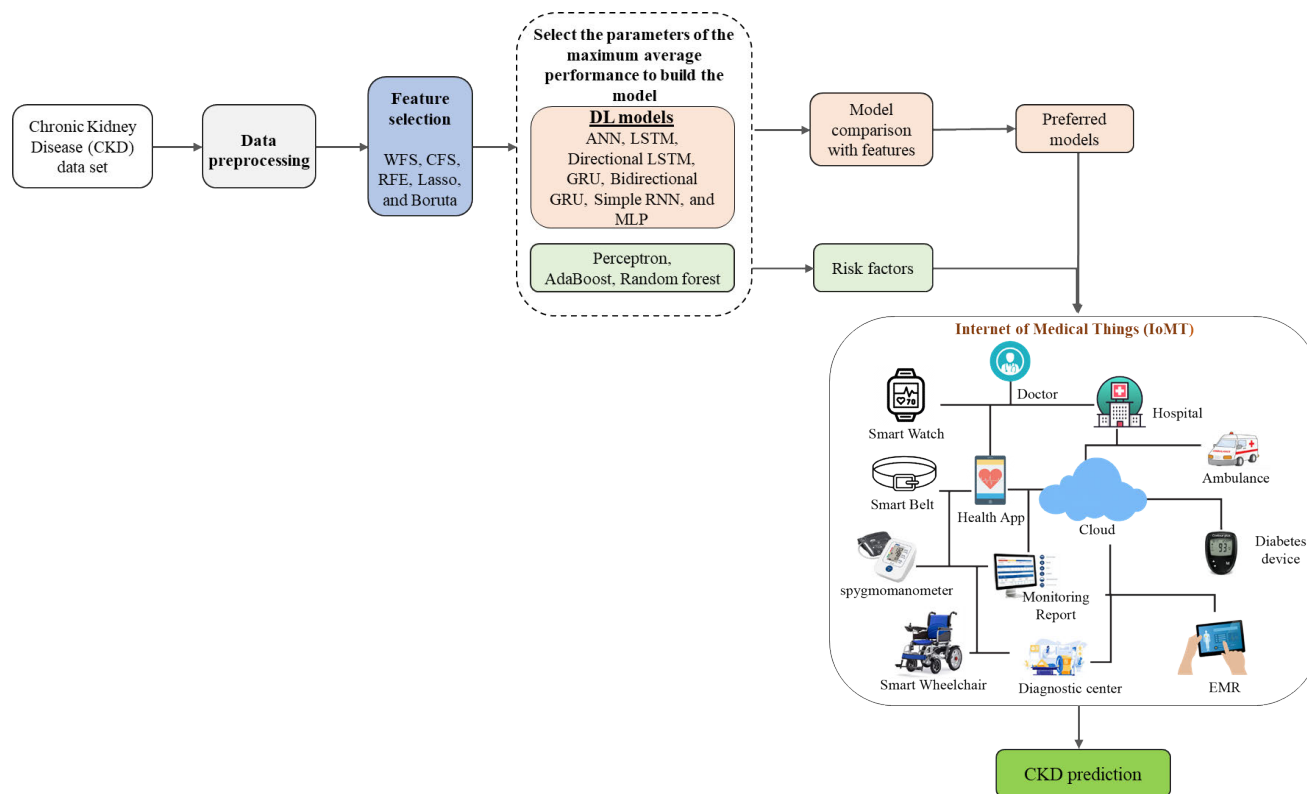


FIGURE 2. Proposed method.

in the medical data poses a significant impact on health assessment. Hence, data pre-processing is the strategy that is utilized to change over the raw data into a clean dataset. It is the basic step to train every DL model/classifier algorithm.

For categorical string data columns, preprocessing was done by converting them into categorical numeric data columns. The categorical numerical data is defined as 0 (negative assertion) or 1 (positive assertion). For example, in the data set, column “pc” was described as “normal” or “abnormal” which were replaced to 1 and 0 respectively; normal was defined as 1 and abnormal was defined as 0.

We performed multiple imputations (MI) to fill the missing values. The imputation process was based on linear regression for predicting continuous variables, and logistic regression for categorical variables. In multiple imputations (MI), missing values in the dataset are replaced by n times, where n is usually a small number (from 3 to 10). We applied MI for 10 iterations to generate 10 different datasets. To narrow down the data to a subset with a plausible range of values, we choose the dataset that had the nearest means and standard deviations for its variables to the original dataset. Subsequently, the missing values for the entire data were filled.

C. FEATURE SELECTION

It is essential to remove unnecessary features from the dataset before training DL classifiers. Among feature selection methods, there were Co-relation based Feature Selection

(CFS), (2) Recursive Feature Elimination (RFE), (3) Lasso Regression, (4) Boruta Feature selection method. The details of the five feature selection approaches/methods are explained as follows:

1) WITHOUT FEATURE SELECTION (WFS)

We considered all features after filling all missing values in the dataset and defined them as WFS datasets.

2) WRAPPER METHOD—CFS

CFS conducts attribute rankings based on the correlation heuristic assessment function [28]. The function employs an approach that generates two class labels, one associated with class and one not

3) RFE

RFE is a wrapper-type feature selection method. RFE works by exploring a various subset of features training dataset and successfully eliminating features until the required number remains. RFE utilizes the core of the model, ranks features by significance, removes the least important features, and re-fits the model.

4) EMBEDDED METHOD—LASSO

Lasso Regression is a linear regression that lowers the coefficients for input factors that have no substantial impact on the prediction task. Later, Lasso allows some coefficient values of features to go to zero, essentially eliminating input

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 | gms (grams per litre) |
| 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. |

variables from the model and providing automatic feature selection.

5) 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.

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 used to find out the significant risk factors of CKD. The models are described as follows:

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

TABLE 2. The selection of features using five methods.

| Feature selection methods | Input feature | Selected features |
|---------------------------|---------------|--|
| WFS | All | 24 features considered |
| CFS | All | 'htn', 'sg', 'rbc', 'dm', 'rc', 'al', 'appet', 'pe', 'ane', 'pc', and 'bp' |
| RFE | All | 'Bu', 'sc', 'sod', 'pot', 'hemo', 'pcv', 'rc', 'rbc', 'pc', 'pcc', 'ba', 'htn', 'dm', and 'cad' |
| Lasso | All | 'sg', 'al', 'hemo', 'pcv', 'htn', and 'dm' |
| Boruta | All | 'Age', 'bp', 'sg', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'hemo', 'pcv', 'wc', 'rc', 'rbc', 'pc', 'htn', 'dm', 'appet', and 'pe'. |

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.

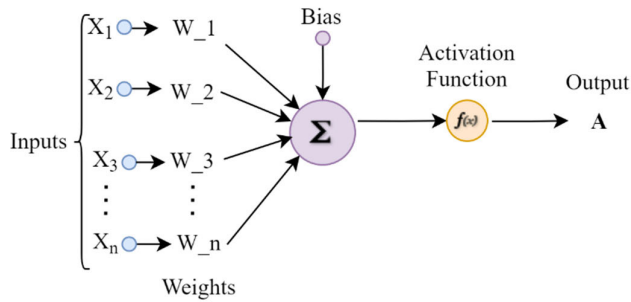


FIGURE 3. Sample architecture/process of ANN [25].

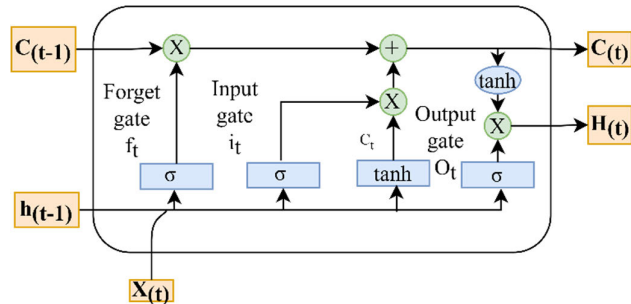


FIGURE 4. Sample architecture/process of LSTM [27].

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

$$\begin{aligned}
 a &= f(\text{net}) = f(n) = f(w^T \cdot p + b) \\
 &= f\left(\sum_{i=1}^R w_{R,i}^T \cdot p_i + b\right) \quad (1) \\
 a &= f(w^T \cdot p + b) \quad (1.1)
 \end{aligned}$$

where f is the activation function, w^T are the weights and b is the bias term.

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 [26]. 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

LSTM has three gates: 1. Input Gate (i_t), 2. Forget Gate (f_t), 3. Output Gate (o_t). The gate equations are as follows:

For gates,

$$i_t = \sigma(w_i [h_t = 1, x_t] + b_i) \quad (2)$$

$$f_t = \sigma(w_f [h_t = 1, x_t] + b_f) \quad (2.1)$$

$$o_t = \sigma(w_o [h_t = 1, x_t] + b_o) \quad (2.2)$$

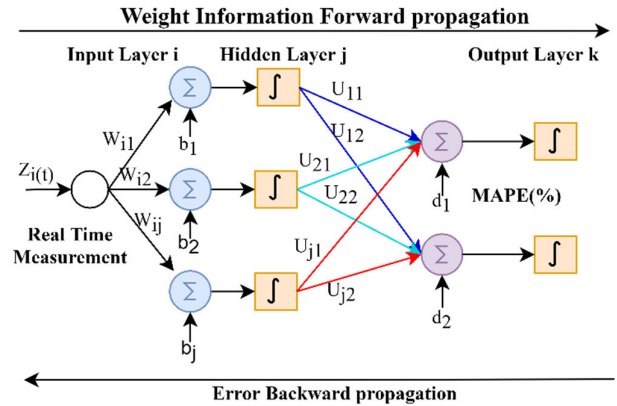


FIGURE 5. Sample architecture/process of GRU [28].

For Cells,

$$c'_t = \tanh(w_c [h_{t-1}, x_t] + b_c) \quad (3)$$

$$c_t = f_t * c_{t-1} + i_t * c'_t \quad (3.1)$$

$$h_t = o_t * \tanh(c_t) \quad (3.2)$$

Here,

c_t = Cell state at timestamp (t)

c'_t = Candidate of cell state at timestamp (t)

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:

Update Gate (z_t):

The formula for update gate is,

$$z_t = \sigma(W^{(z)} x_t + U^{(z)} h_{t-1}) \quad (4)$$

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 (r_t):

The formula for update gate is,

$$r_t = \sigma(W^{(r)} x_t + U^{(r)} h_{t-1}) \quad (5)$$

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

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.

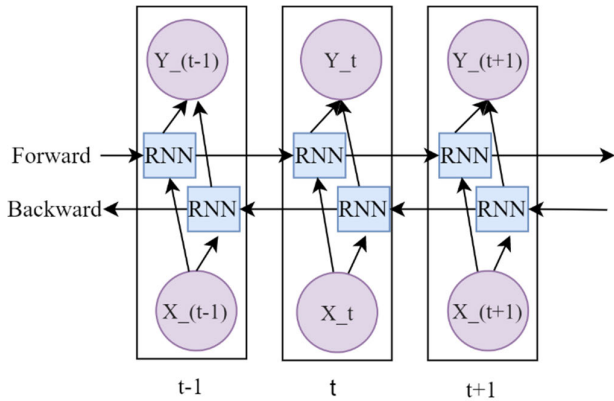


FIGURE 6. Sample architecture of bidirectional LSTM [29].

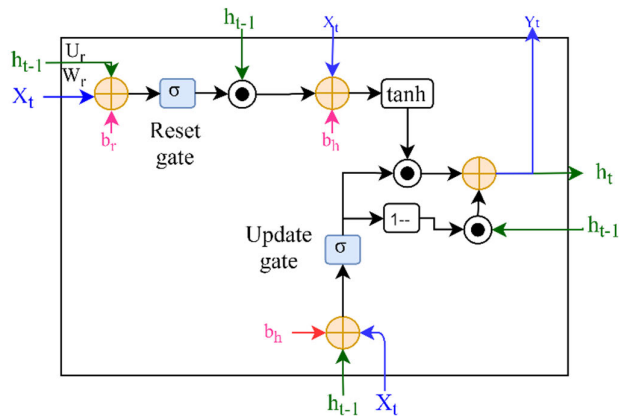


FIGURE 7. Sample of the process of bidirectional GRU [30].

First computes the forward hidden sequence \vec{h}_t . Then compute the background sequence \overleftarrow{h}_t . Finally generating two computes \vec{h}_t and \overleftarrow{h}_t , we get an output: y_t .

$$\vec{h}_t = H \left(W_{x\vec{h}} x_t + W_{h\vec{h}} \vec{h}_{t-1} + b_{\vec{h}} \right) \quad (6)$$

$$\overleftarrow{h}_t = H \left(W_{x\overleftarrow{h}} x_t + W_{h\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}} \right) \quad (6.1)$$

$$y_t = W_{h\vec{y}} \vec{h}_t + W_{h\overleftarrow{y}} \overleftarrow{h}_t + b_y \quad (6.2)$$

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.

$$\text{Forward direction: } \vec{h}_t^{(i)} \quad (7)$$

$$\vec{z}^{(i)} = \sigma \left(\vec{W}_{(i)}^{(z)} x_t^i + U_{(i)}^{(z)} h_{(t-1)}^{(i)} \right) \quad (7.1)$$

$$\vec{r}^{(i)} = \sigma \left(\vec{W}_{(i)}^{(r)} x_t^i + U_{(i)}^{(r)} h_{(t-1)}^{(i)} \right) \quad (7.2)$$

$$\vec{h}^{(i)} = \tanh \left(\vec{W}_{(i)} x_t + r_t^o U_{(i)} \vec{h}_{t-1} \right) \quad (7.3)$$

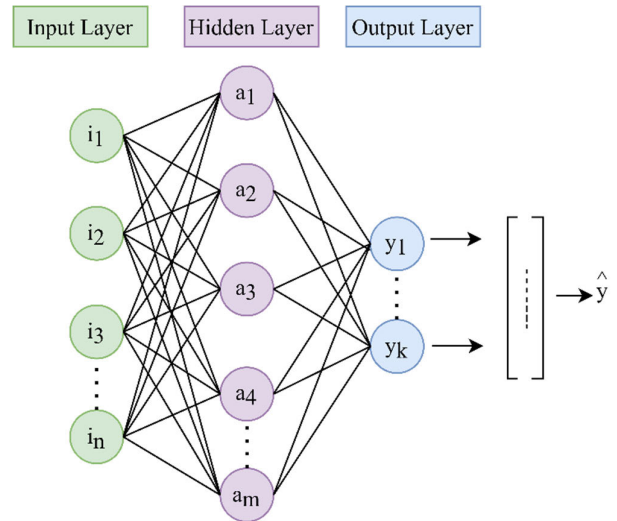


FIGURE 8. Sample architecture/process of MLP [31].

$$\vec{h}^{(i)} = z_t^{(i)o} h_{t-1}^{(i)} + \left(1 - z_t^{(i)} \right)^o \tilde{h}_t^{(i)} \quad (7.4)$$

$$\text{Backward direction: } \overleftarrow{h}_t^{(i)} \quad (8)$$

$$\vec{z}^{(i)} = \sigma \left(\vec{W}_{(i)}^{(z)} x_t^i + U_{(i)}^{(z)} h_{(t-1)}^{(i)} \right) \quad (8.1)$$

$$\vec{r}^{(i)} = \sigma \left(\vec{W}_{(i)}^{(r)} x_t^i + U_{(i)}^{(r)} h_{(t-1)}^{(i)} \right) \quad (8.2)$$

$$\vec{h}^{(i)} = \tanh \left(\vec{W}_{(i)} x_t + r_t^o U_{(i)} \vec{h}_{t-1} \right) \quad (8.3)$$

$$\overleftarrow{h}^{(i)} = z_t^{(i)o} h_{t-1}^{(i)} + \left(1 - z_t^{(i)} \right)^o \tilde{h}_t^{(i)} \quad (8.4)$$

Output:

$$y_t = \text{softmax} \left(U \left[\begin{matrix} \overleftarrow{h}_t^{(top)} & \overrightarrow{h}_t^{(top)} \end{matrix} \right] + a \right) \quad (9)$$

here z_t update gate, r_t reset gate h_{ti} is new memory and \tilde{h}_t is reset the memory.

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.

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

$$y = 1 \quad \text{if } \sum_{i=0}^n w_i * x_i \geq 0 \quad (10)$$

$$y = 0 \quad \text{if } \sum_{i=0}^n w_i * x_i < 0 \quad (10.1)$$

Here $x_0 = 1$ and $w_0 = -\theta$

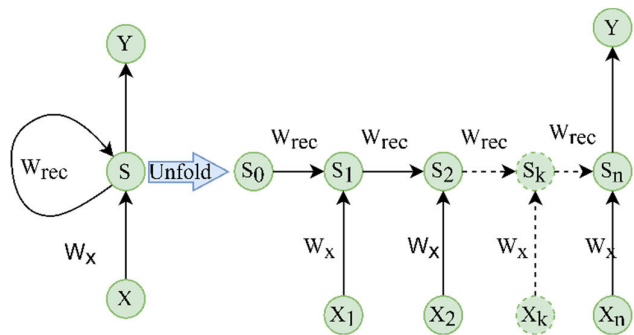


FIGURE 9. Sample architecture/process of simple RNN [32].

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 overtime 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 to store information between timesteps. Figure 9 describes the generic architecture of SimpleRNN. The recurrence relation over time steps is given below:

$$S_k = f(S_{k-1} \cdot W_{rec} + X_k \cdot W_x) \tag{11}$$

where S_k represents the state at k time.

X_k means input at k time.

W_{rec} and W_x are the weight parameter and free forward nets. S_k is the final output of the network with k timestep, which is typically calculated as $S_{k-1} \dots S_{k+j}$.

The current state S_k can be calculated from current input X_k or else previous state S_{k-1} and it can predict the next state from S_{k+1} where the current state is S_k and current input is X_k .

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 AdaBoost.

The overall equation for AdaBoost is summarized as

$$F(x) = \text{sign} \left(\sum_{m=1}^M \theta_m f_m(x) \right) \tag{12}$$

F_m = the m_th weak classifier

θ_m = corresponding weight

9) RANDOM FOREST

RF is a large-dataset aggregate classifier and regression classification method that generates decision trees from a randomly selected subset of training data and returns an output class (i.e., which is the output of individual trees [30]. Even though RF can easily maintain thousands of input attributes, there is no need to reduce variables during analysis.

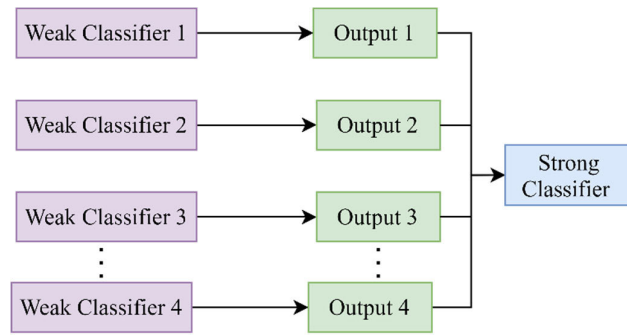


FIGURE 10. Sample architecture/process of AdaBoost [33].

The estimation of variables important in classification is provided by RF.

10) MODEL EXECUTION PROCESS

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 several 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.

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, MSE as loss (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 3).

We calculated accuracy, Precision, Recall, F1 score, the loss and validation loss of the seven models, and visualized the model performance through 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

TABLE 3. 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)

TABLE 4. Confusion matrix.

| | Predicted:0 | Predicted:1 |
|----------|---------------------|---------------------|
| Actual:0 | True Negative (TN) | False Positive (FP) |
| Actual:1 | False Negative (FN) | True Positive (TP) |

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.

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 according 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

1) CONFUSION MATRIX

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 4) [21].

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. (13).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

Precision computes and estimates the number of positive classes representing the truly positive class.

- Recall** calculates and predicts the number of positive classes in the data set overall positive instances.
- 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. (14).

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

Recall signifies the number of real negatives which are correctly classified as positives and are represented in Eq. (15).

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

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:

$$f1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (16)$$

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. (17).

$$FPR = \frac{FP}{FP + TN} \quad (17)$$

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. (18).

$$TPR = \frac{TP}{FN + TP} \quad (18)$$

$$FNR = \frac{FN}{TP + FN} \quad (19)$$

FNR is the miss rate of the model describing positive instances classified as negative (Eq. 19) and this rate is expected to be as close to zero. FPR indicates negative instances classified as positive. Both FPR and FNR rates should be as close to zero for a good performance. Figure 11 shows the confusion matrix for each model while taking all the attributes (WFS) of the data set. This matrix shows the TP, FP, FN, TN of each model. We calculated the False Positive Rate (FPR), and True Positive Rate (TPR) of each model as it is important to know the FPR and the FNR rate in each model. Also, Figure 11 shows the confusion matrix with FPR and FNR for seven DL models with the WFS dataset. ANN showed the lowest FNR and GRU showed the highest FNR of 0.231. MLP and SimpleRNN also provided low FNR;0.058,.038 respectively. The Confusion matrix of the seven algorithms using the other four feature selection methods is presented in the supplementary (Figure S1-S4).

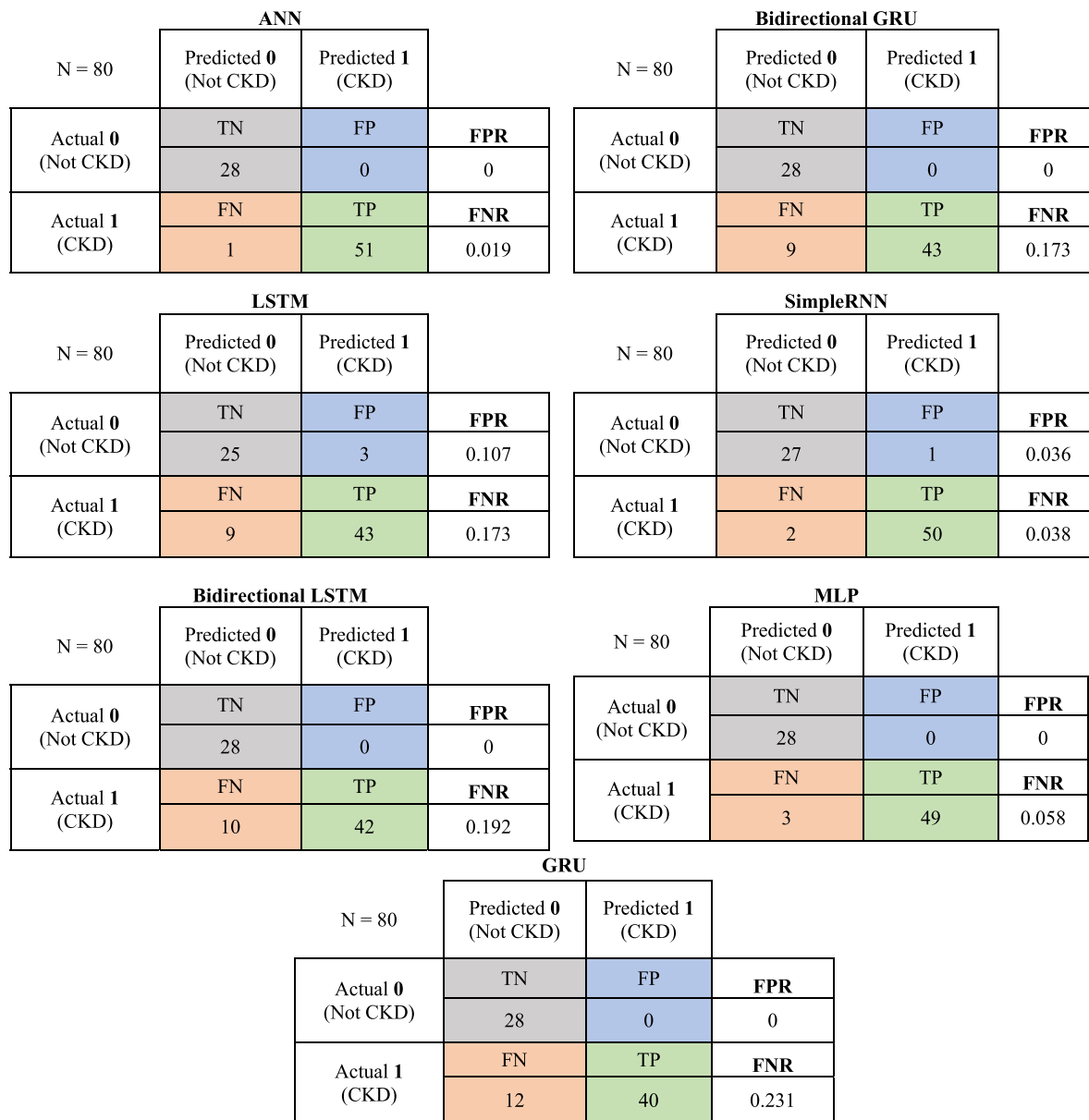


FIGURE 11. Confusion matrix for 7 DL models (WFS).

2) ACCURACY, PRECISION, RECALL, F1 SCORE

We used all the features, and selected features in the data set through different methods, the models were fitted in the training dataset, tested the testing dataset, and predicted the CKD, it has been observed that SimpleRNN, MLP, and ANN provided the highest performance in terms of Accuracy, Precision, Recall, AUC, Loss, and Validation loss for all applied FS and WFS method among all DL models. 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.

a: WITHOUT FEATURE SELECTION (WFS)

Table 5 provides the experimental results of the proposed seven DL model with the WFS method and represents the

different measures such as Accuracy, Precision, Recall, F1 Score, AUC, Loss, and Validation loss (errors) [34]. 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 an accuracy of 97%, followed by LSTM and Bidirectional LSTM-85% and 88% respectively, GRU-85% and Bidirectional GRU-89%. ANN provided the highest score for precision, Recall, and F1 score. SimpleRNN and MLP also showed high scores for Precision, Recall, and F1 measures. LSTM and GRU showed lower Precision and F1 measures compare to other algorithms. While predicting CKD disease, LSTM showed the lowest Precision (84%), and F1 score (84%), and Recall (86%) and (Figure 12).

TABLE 5. Performance comparison of DL models (WFS).

| 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 |

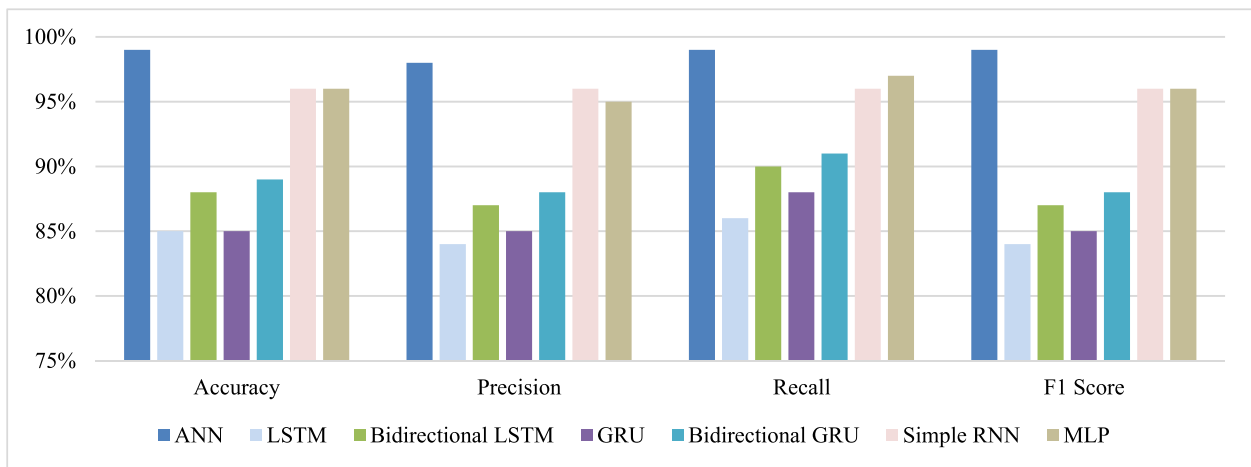


FIGURE 12. Graphical representation of performance comparison (WFS).

TABLE 6. Performance comparison of DL models (CFS).

| 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 |

b: CFS

Table 6 provides the experimental results of the proposed seven DL classifiers with CFS data and represents the different measures such as Accuracy, Precision, Recall, F1 Score, AUC, Loss, and Validation loss (errors) [34]. All the models showed 80 to over 90% accuracy, where the two models showed over 95% accuracy. Out of all algorithms, ANN showed the highest accuracy of 97%, followed by MLP and SimpleRNN-96% accuracy. Bidirectional LSTM, GRU, and Bidirectional GRU achieved an accuracy of 85%. ANN provided the highest value of Precision (97%), Recall (98%),

and F1 score (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 show the lowest Precision and F1 measure score (Figure 13).

c: RFE

Table 7 provides the experimental results of the proposed seven DL models 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

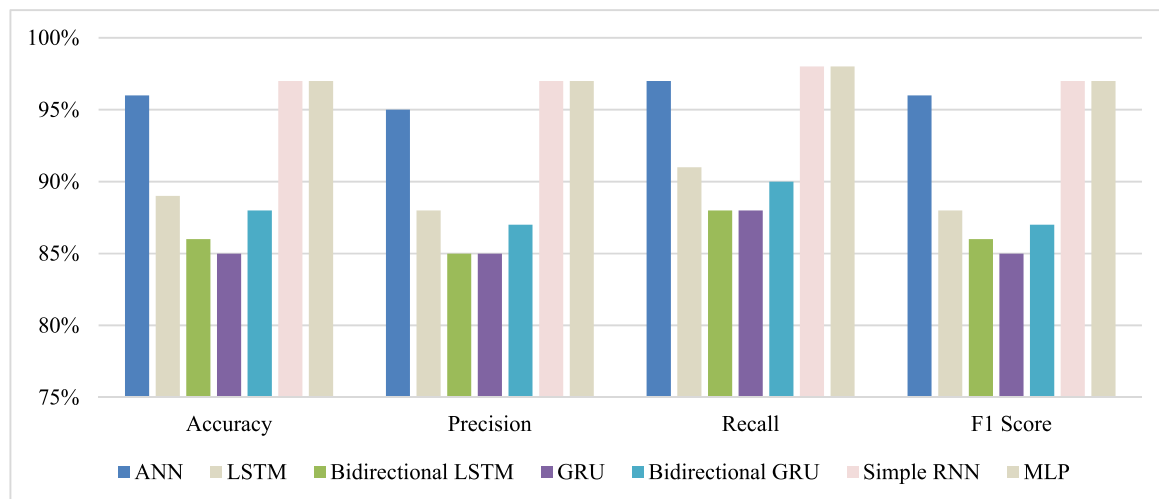


FIGURE 13. Graphical representation of performance comparison (CFS).

TABLE 7. Performance comparison of DL models (RFE).

| 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 |

over 90% accuracy, where three models showed over 95%. Out of all algorithms, ANN showed the highest accuracy of 97%, followed by 96% accuracy of both SimpleRNN and MLP. GRU, BidirectionalGRU, and Bidirectional LSTM achieved the same accuracy(85%). Also, Table 7 shows 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. Precision, Recall, and F1 score are the same for GRU, Bidirectional GRU, and Bidirectional LSTM i.e., 85%, 88%, and 85% respectively. While predicting CKD disease, LSTM showed Precision (87%), Recall (90%), and F1 score (87%) (Figure 14).

d: LASSO

Table 8 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) [34]. 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 the same score for accuracy (79%), the accuracy of BidirectionalGRU was 86%. ANN, MLP, and SimpleRNN scored high Precision, Recall, and F1 scores. ANN achieved the highest Precision, Recall, and F1 measure among all models. The values of Precision, Recall, and F1 scores of LSTM, Bidirectional LSTM, and GRU are the same; 81%, 84%, and 79% respectively, and these scores are low compared to all other models (Figure 15).

e: BORUTA

Table 9 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) [34]. 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%. ANN provided the highest score for precision, Recall, and F1 score.

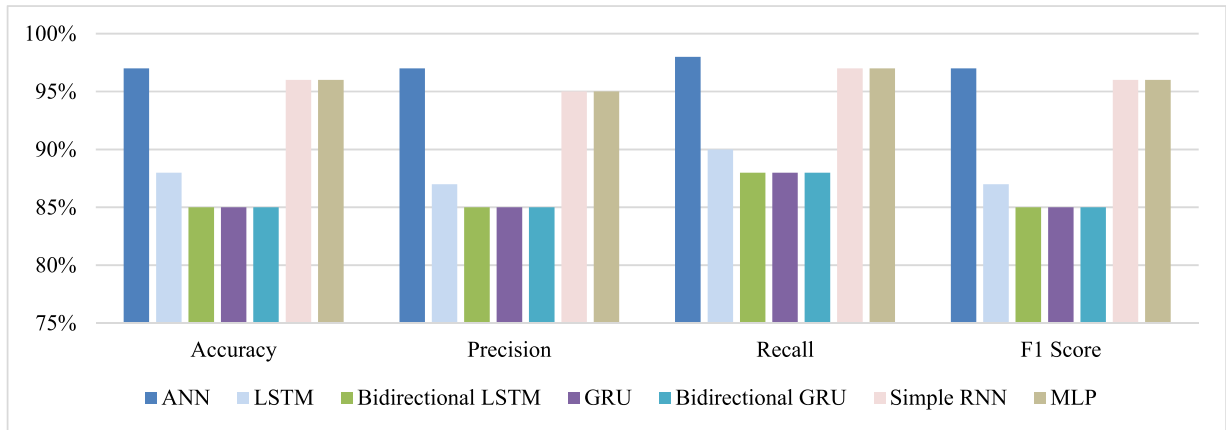


FIGURE 14. Graphical representation of performance comparison (RFE).

TABLE 8. Performance comparison of DL models (Lasso).

| 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 |

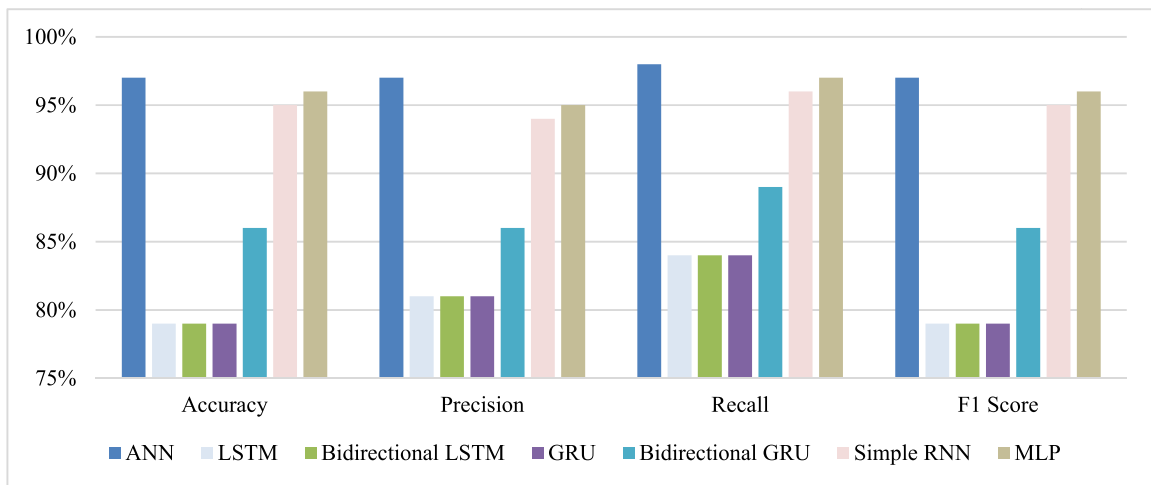


FIGURE 15. Graphical representation of performance comparison (Lasso).

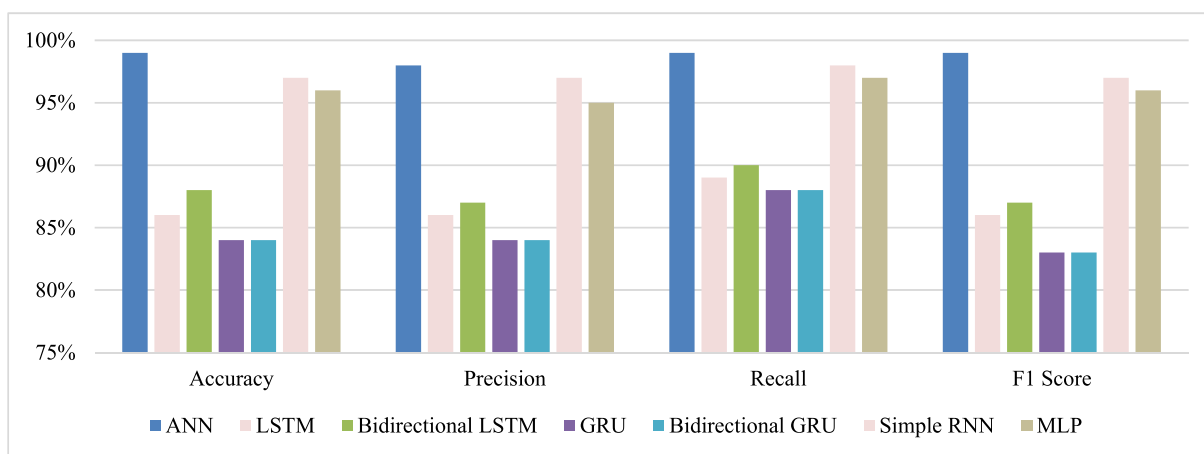
ANN, SimpleRNN, and MLP also showed high scores for Precision, Recall, and F1 measure. GRU and Bidirectional GRU showed lower Precision and F1 measure compared to other algorithms. While predicting CKD disease, GRU showed the lowest Precision (84%), and F1 score (84%), and Recall (83%) (Figure 16).

B. RECEIVER OPERATING CHARACTERISTIC (ROC)/REA 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

TABLE 9. Performance comparison of DL models (Boruta).

| 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 | 0.01 | 0.01 | 21.10 |
| MLP | 96% | 95% | 97% | 96% | 0.97 | 0.01 | 0.13 | 0.12 |

**FIGURE 16.** Graphical representation of performance comparison (Boruta).

measure or degree. Higher AUC (close to 1) means better performance to distinguish whether the patient has a disease or not [35]. Figure 17 shows the AUC curve where the x and y-axis represent FPR (False Positive Rate) and TPR (True Positive Rate) respectively. Here, ANN, showed the highest AUC in all features selected methods and WFS where the value is from 0.97 to 0.99 (close to 1) indicating very good performance. SimpleRNN, and MLP also showed high AUC scores compared to other models. For Lass method LSTM, GRU, Bidirectional LSTM, and bidirectional GRU provided the same AUC of 0.84 which is the lowest in the analysis.

So, the three DL algorithms: ANN, MLP, and SimpleRNN performed superior to other deep learning algorithms in all methods along with WFs considering all features.

C. LOSS AND VALIDATION LOSS

Loss is defined by the error that occurred during each iteration (epoch) on the training dataset to predict the class of CKD. Loss or error is also 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 Loss were measured for each model. Loss and validation loss of the seven DL models is shown in Figure 18 (A-G) for WFS.

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 four feature selection methods 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 loss or Val_loss. ANN and MLP showed low scores 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 scores for both loss and validation loss. However, the difference between loss and validation loss was very low for the DL models in all selection methods suggesting the perfect fitting of data to the models. Thus, we can conclude that models were not overfitted to the training dataset (Table 5). The Loss and Validation Loss results of the seven algorithms using the other four feature selection methods are presented in the supplementary (Figure S5 – Figure S8).

D. PREDICTION RATIO

The algorithm uses a training set of features and the associated result to predict a given outcome (prediction results).

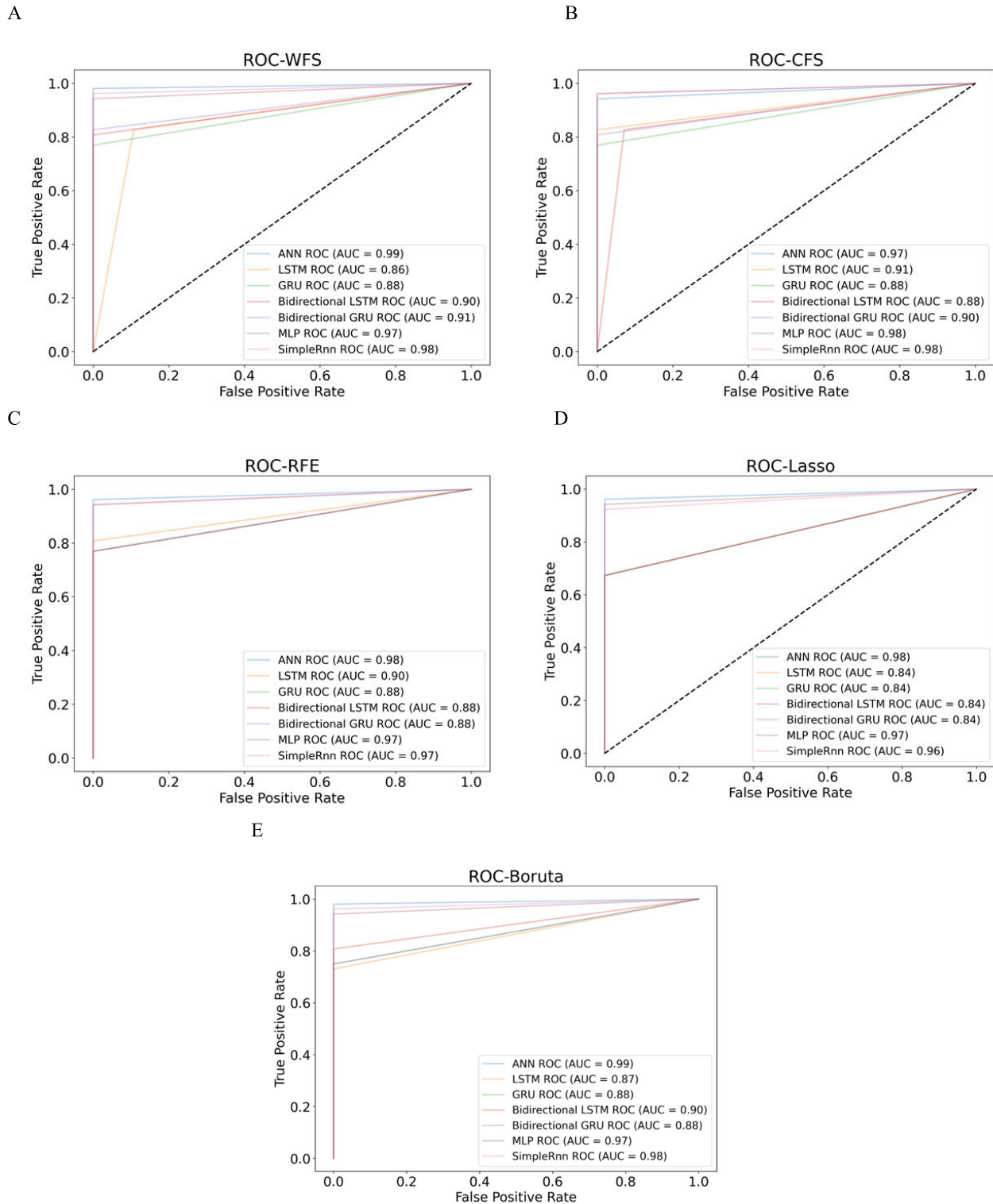


FIGURE 17. ROC curves of DL models.

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 incorrect ones for a given data set. The study divided the whole data set into

eight segments containing 50 instances. Figure 19 shows the prediction ratio of the seven DL algorithms. In the case of WFS, CFS, and Boruta, it is observed that ANN, MLP, and SimpleRNN showed a high prediction ratio for small datasets. However, the prediction ratio of MLP drops to zero with

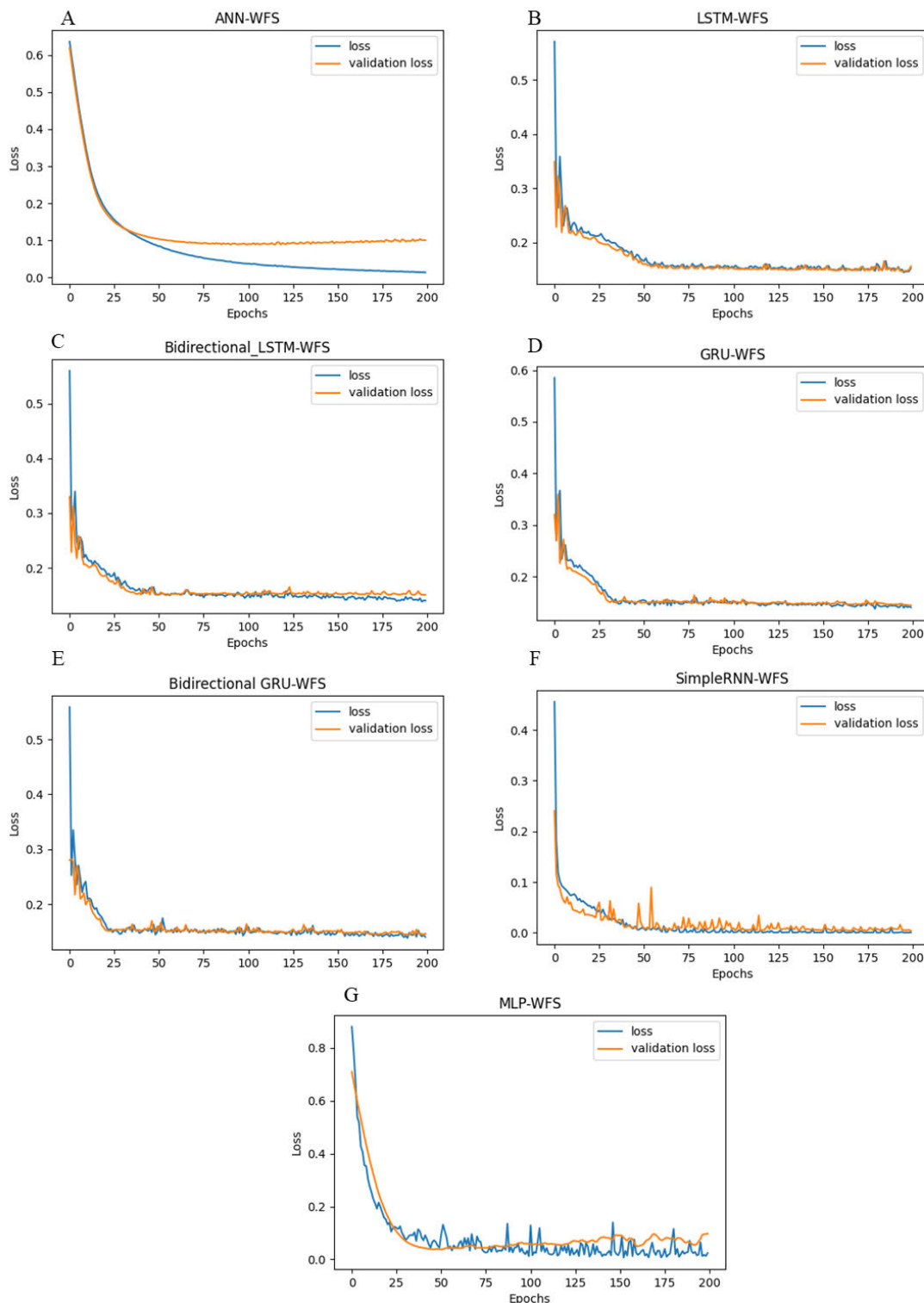


FIGURE 18. (A-G). Error as loss of training dataset and error as validation loss of testing dataset are shown for seven deep learning models using WFS. Here val_loss denotes the error as validation loss for the testing dataset.

a large dataset (300-400 instances). For other models, the Prediction ratio was high with small data set, then reduced, at last increased with a large dataset. For Laso and the prediction ratio of MLP and Bidirectional GRU appears in

the opposite direction; the ratio drops a with higher data set while the ratio increases to 100% with large data set. This analysis shows that SimpleRNN and ANN performed best compared to other models while predicting CKD

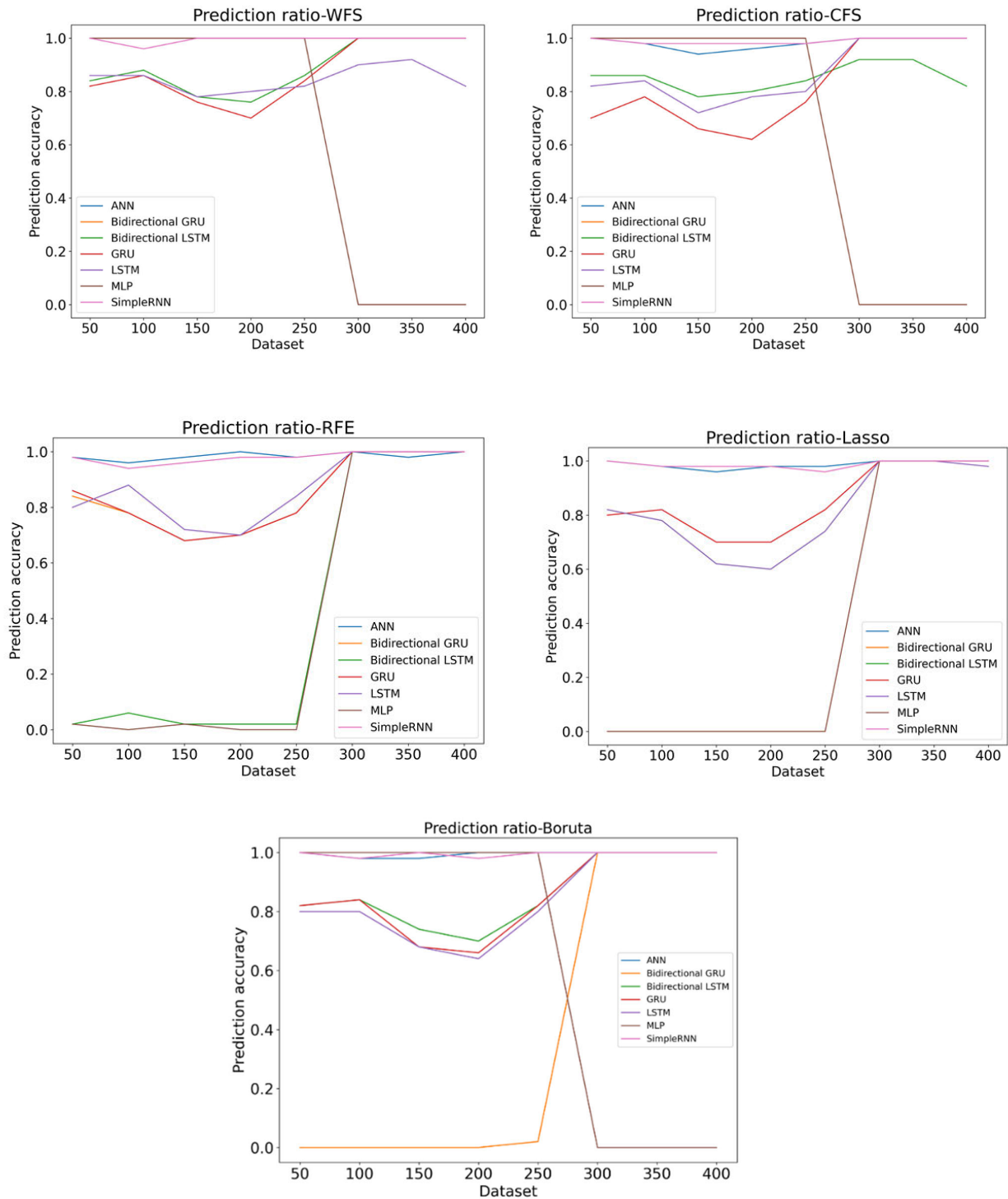


FIGURE 19. Comparison of prediction ratio of DL models.

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 sets.

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 with four feature selection methods described earlier and also without

TABLE 10. Wilcoxon signed-rank test results (WFS).

| Models | ANN | LSTM | Bidirectional LSTM | GRU | Bidirectional GRU | Simple RNN | MLP |
|--------------------|-------|-------|--------------------|-------|-------------------|------------|-----|
| ANN | - | - | - | - | - | - | - |
| LSTM | 0.004 | - | - | - | - | - | - |
| Bidirectional LSTM | 0.003 | 1.000 | - | - | - | - | - |
| GRU | 0.003 | 0.005 | 0.004 | - | - | - | - |
| Bidirectional GRU | 0.004 | 0.943 | 0.832 | 0.005 | - | - | - |
| Simple RNN | 0.235 | 0.004 | 0.004 | 0.004 | 0.004 | - | - |
| MLP | 0.066 | 0.004 | 0.003 | 0.004 | 0.004 | 0.013 | - |

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

feature selection(WFS). The proposed ANN, MLP, and Simple RNN models showed low computation time i.e. 10 sec, 11-13sec, and 21 sec respectively suggesting high performance. Figure 20 demonstrates the computational time of the implemented seven DL models comprehensively with the method. Bidirectional GRU took the highest time; 91 sec with RFE and 90.63 sec with WFS. In Boruta, LSTM, Bidirectional LSTM completed the prediction or classification with more than 80 sec and GRU took ~65 sec (Figure 20).

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 for all feature methods along with WFS based on accuracy [36]. Table 10 depicts the results of the p-value for pair-wise comparison of the models of the WFS dataset. To explain, the p-value among ANN and other models (LSTM, Bidirectional LSTM, GRU, Bidirectional GRU) is lower than 0.05 (p-value ~0.003- 0.004) 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. The Wilcoxon test results of the seven algorithms using the other four feature selection methods are presented in the supplementary (Table S1- Table S4).

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 [37]). Each algorithm was compared against the other, and deep dominance (ϵ) was

measured between 0 to 1.0. 0 that corresponds to perfect stochastic dominance of one algorithm (X) over another (Y) and 1 corresponds to perfect stochastic dominance of Y over X. Table 11 shows the results from the Deep Dominance test. The Deep Dominance test results of the seven algorithms using the other four feature selection methods are presented in the supplementary (Table S 5 – Table S8).

H. RISK FACTOR ANALYSIS

We identified the risk factors of CKD by using Perceptron and AdaBoost Classifier models. Figure 21 shows the risk factors of CKD using both models considering all the features i.e. WFS). 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 [38]. Perceptron provided the full ranking of the features (absolute values of the features were considered) and Adaboost classifier model ranked 15 important features which are as follows: “hemo,” “sc,” “sg,” “age,” “bu,” “rc,” “bgr,” “pcv,” “pe,” “dm,” “wc,” “pot,” “al,” “bp,” and “htn.” Random forest identified top 15 features, and these are hemo, sg, pcv,sc, al Rc, htn, bgr, dm, bu, sod, su, pot, pe, bp. While reviewing both models concerning top 15 features, it is observed that 9 features such as “hemo,” ‘sg,’ ‘pe,’ ‘al,’ ‘dm,’ ‘htn,’ ‘bp,’ ‘sc,’ ‘pcv’ and depicted in Figure 21. The importance and interrelation of these 9 factors towards CKD progression have been substantiated in the medical science and health domains [39].

I. IoMT PLATFORM

Hardware plays a critical role in IoMT. Various hardware can be used to collect real-time data from CKD patients. Data can be collected from a range of medical devices such as ECG (Electrocardiogram) to provide heart monitoring data, fitness tracker to provide data like stress, breathing, Oxygen level, and glucometer to provide diabetes in blood, and blood monitoring devices will provide the pressure of blood (Figure 2). All these devices are interconnected via

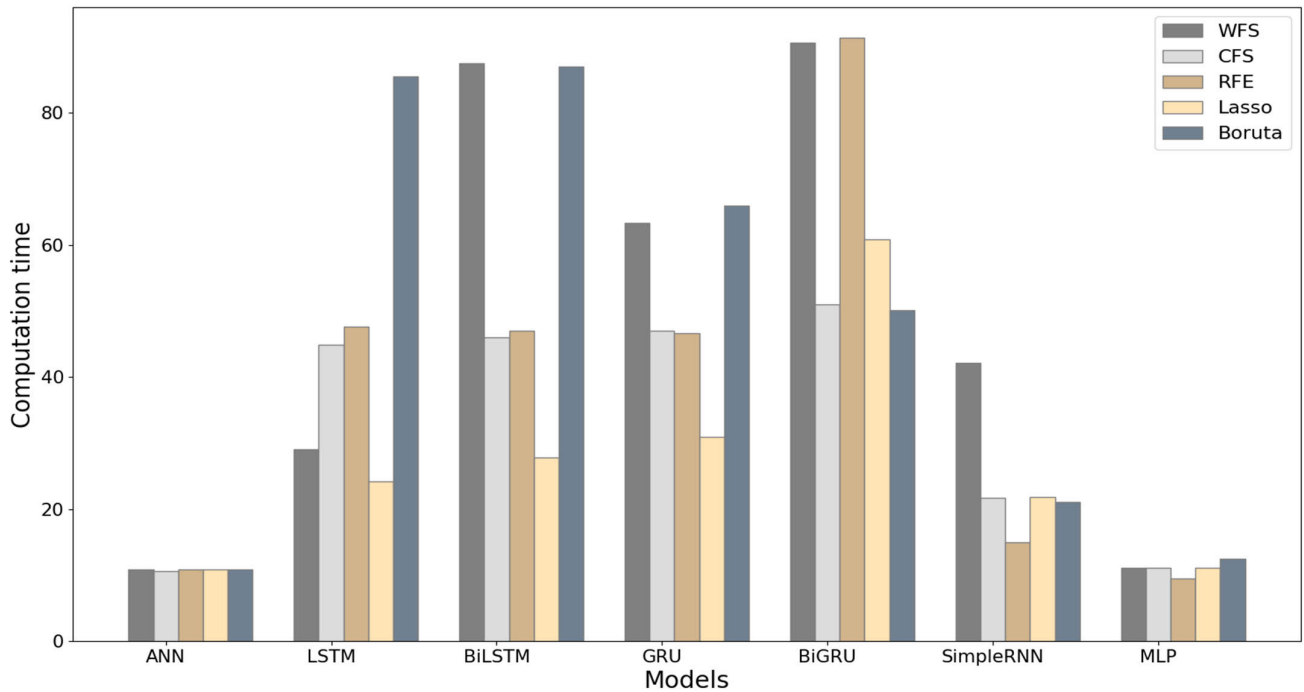


FIGURE 20. Comparison of computation time of DL models.

TABLE 11. Deep dominance test results (WFS).

| Models | ANN (A) | LSTM (B) | Bidirectional LSTM (C) | GRU (D) | Bidirectional GRU (E) | Simple RNN (F) | MLP (G) |
|-----------------------|----------------------------|--------------------------------------|----------------------------------|----------------------------------|----------------------------------|--------------------------------|---------|
| ANN(A) | - | - | - | - | - | - | - |
| LSTM (B) | $\epsilon = 0, A \wedge B$ | - | - | - | - | - | - |
| Bidirectional LSTM(C) | $\epsilon = 0, A \wedge C$ | $\epsilon = 0.710, B \text{ nbt } C$ | - | - | - | - | - |
| GRU (D) | $\epsilon = 0, A \wedge D$ | $\epsilon = 0, B \wedge D$ | $\epsilon = 0, C \wedge D$ | - | - | - | - |
| Bidirectional GRU (E) | $\epsilon = 0, A \wedge E$ | $\epsilon = 0.671, B \text{ nbt } E$ | $\epsilon = 0.269, C \wedge E$ | $\epsilon = 1, D \text{ nbt } E$ | - | - | - |
| Simple RNN (F) | $\epsilon = 0, A \wedge F$ | $\epsilon = 1, B \text{ nbt } F$ | $\epsilon = 1, C \text{ nbt } F$ | $\epsilon = 1, D \text{ nbt } F$ | $\epsilon = 1, E \text{ nbt } F$ | - | - |
| MLP (G) | $\epsilon = 0, A \wedge G$ | $\epsilon = 1, B \text{ nbt } G$ | $\epsilon = 1, C \text{ nbt } G$ | $\epsilon = 1, D \text{ nbt } G$ | $\epsilon = 1, E \text{ nbt } G$ | $\epsilon = .0348, E \wedge G$ | - |

nbt -not better than

the internet and are responsible for data transmission to the cloud through connectivity technologies such as networks and gateways. In many instances, data are uploaded from time to time through an API (Application Programming Interface) key for a secured connection. These API keys will allow the specific devices to access and store data on the IoT platform. Subsequently, deep learning models provide health care professionals with data analytics, reporting, and device control opportunities through software solutions. The approach to utilize IoMT components to manage CKD data and adopt a replicable application using deep learning algorithms to predict CKD is proposed in the study.

V. DISCUSSION

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

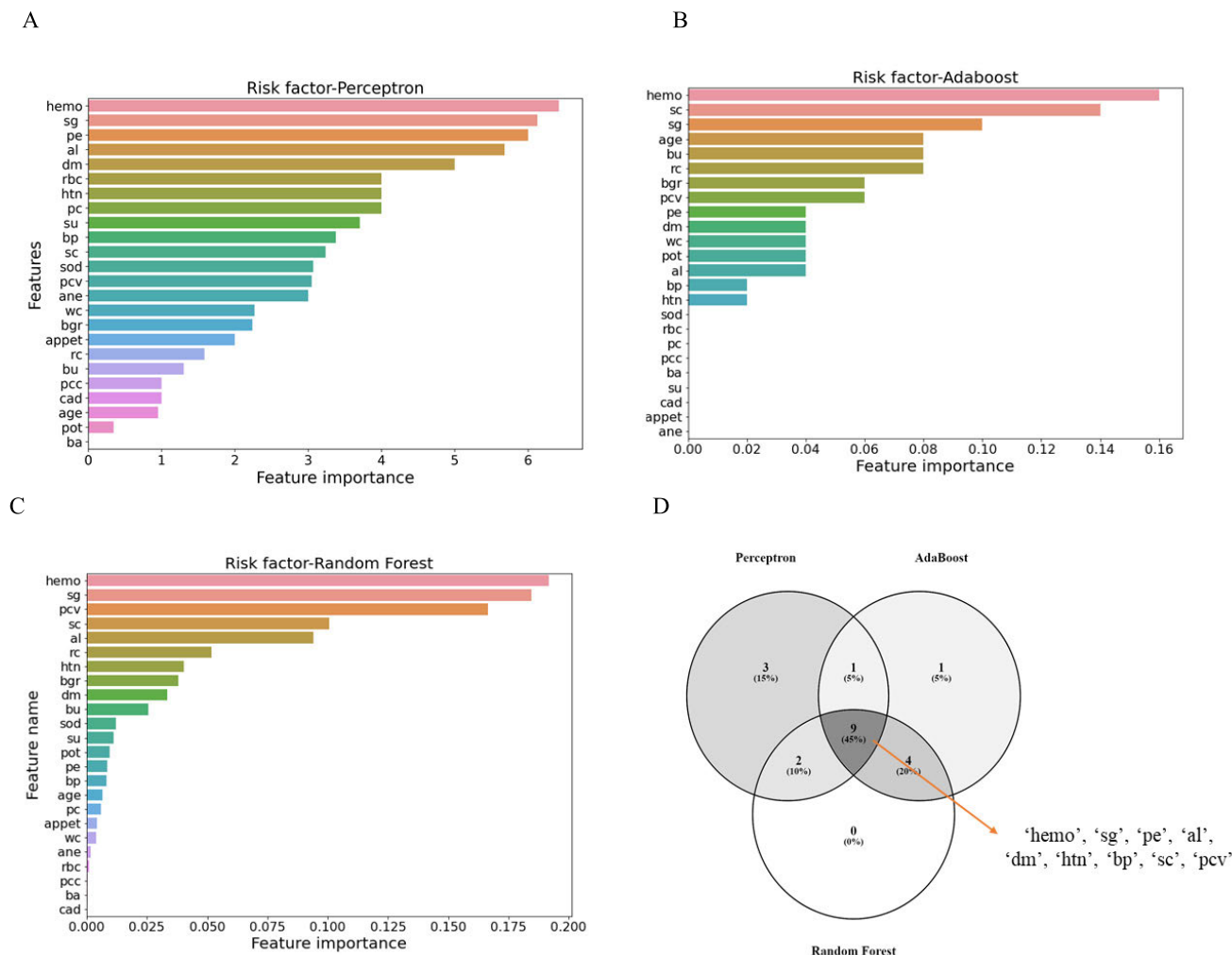


FIGURE 21. (A–D) Risk factor analysis with perceptron, AdaBoost and random forest. Figure 21 (D) the factors those fall in the three models.

efficacy of deep learning algorithms to predict CKD along with identified risk factors has been shown in this study. To demonstrate a comprehensive performance assessment, we used seven deep learning algorithms on 400 CKD data for predicting the class of CKD. The proposed methods achieved high accuracy around 99%. In comparison, ANN, MLP SimpleRNN achieved the highest accuracy in terms of predicting CKD compared to other deep learning models that include feature selection techniques- WFS, CSF, RFE, Lasso, and Boruta. Overall, ANN (96%-99%), MLP (96%-97%), and SimpleRNN (95%-97%) showed accuracy in all the five feature selection processes. GRU provided an accuracy of 85% in RFE, and both Bidirectional GRU and Bidirectional LSTM showed 85% accuracy in CSF. LSTM showed 89% and 88% accuracy to predict CKD class in CFS and RFE respectively. 85% accuracy was observed in the case of GRU both in CSF and RFE.

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, with all features (WFS) compared to other algorithms, SimpleRNN showed the minimum error for both the testing and training data i.e., 0.001 and 0.02 respectively. LSTM provided a maximum error of 0.15 for the testing dataset. The other key findings are as follows; (1) Bidirectional GRU took the highest computation time (i.e., 91 sec), (2) ANN and MLP took less time (~10 sec and ~11 sec) with feature selection methods including WFS (Figure 20). (3) ANN and SimpleRNN showed similar and comparatively high prediction ratios (i.e., 0.98-1.0). 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.

The importance of medical significance on features selection plays a major role in understanding the performance. As indicated five feature selection techniques are adopted in the study. These methods generated different data set containing different features. The result of feature selection provided different attributes shown in Table 2. Further, using these feature selections methods, the average accuracy among the DL models lies between 85-99%. But it does not warrant

true performance. Few features such as bp (blood pressure), sc (serum creatine), bgr (random blood sugar) were not selected in Lasso. Similarly, su (serum urea), bp (blood pressure), and pe (paddle edema) were not selected in RFE. However, it is well documented the direct and indirect influence of these parameters on CKD progression. Hence identifying interconnection between different functions and selecting target attributes are imperative. In this essence, without feature selection methods such as WFS does not eliminate any feature and showed superior performance in terms of Accuracy, Precision, Recall, and prediction ratio while using seven DL models, Therefore the study strongly recommended that without feature selection can potentially be considered in the DL models to predict CKD.

Further, the findings were validated by statistical significance analysis for all five feature selection processes. While comparing with other models, significant p-values (<0.05) were obtained for ANN, SimpleRNN, and MLP (Table 10). This substantiates the findings using the Wilcoxon test that ANN, SimpleRNN, and MLP showed superior performance in the case of WFS. In addition, the Deep dominance test showed that ANN is better than all other models where ϵ is 0. The ϵ values of other models (LSTM, Bidirectional LSTM, GRU, and Bidirectional GRU) is ~ 1.0 while compared with Simple RNN and MLP. The findings indicate that four models are not better than (*nbt*) SimpleRNN and MLP. Overall, both significance tests resemble a similar outcome in the rest of the four feature selections.

Further, the study identified the risk factors of CKD. Perceptron, Adaboost, and Random forest classifier models were used. These models successfully provided nine features as the considerable risk factors of CKD (Figure 21). For example, In CKD, the kidneys fail to produce enough erythropoietin (EPO), a hormone required by the body to produce red blood cells which have direct relation with the hemoglobin. Less red blood cells indicate a lower level of hemoglobin. Hence, rc (red blood count) and heme (Hemoglobin) are crucial parameters for the diagnosis 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. The dataset is small, which could cause the results to be unreliable. It is difficult to find another dataset with more attributes containing higher instances. More specifically, data collection dynamically from IoMT platform is even more difficult. However, during optimization, overfitting was prevented by customizing the

parameters to calculate the error between testing and training datasets. In this study, we successfully applied several input layers, hidden layers, activation functions, and optimizers in all DL models. These actions resulted in a very low difference between loss and validation loss (Figure 18 and Table 5). Hence it can be concluded that the models were not overfitted with the training dataset despite having 400 datasets.

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. The study demonstrates a holistic performance assessment of deep learning algorithms on CKD. The study explains the following contribution to the body of the knowledge- (1) the study adopted a scientific data processing approach to identify the missing values in the CKD dataset. We employed linear regression and logistic regression for numerical and categorical data to fill up the missing values respectively. (2) five state-of-the-art feature selection processes were adopted to select features and subsequent comparative performance of the seven algorithms so that the utility of inclusion and not the inclusion of feature selection application in the deep learning were explained (3) adding the statistical significance (combination of Wilcoxon rank test, and deep dominance test for cross-checking the results to compare the performances) established the reliability of the outcome, finally (4) four DL models simple RNN, Bi GRU, and Bi LSTM, and GRU are applied to predict CKD for the first time. Thus, this research examined the efficacy of seven DL algorithms for predicting CKD. When comparing the models, ANN, MLP, and Simple RNN showed superior performance, providing an accuracy of 97% in predicting the disease. In this study, applied DL algorithm MLP Adaboost, and RF identified nine factors or attributes as the risk factors of CKD. These results can boost the medical community by predicting CKD and the risk factors. Based on the research, we believe that deep learning approaches could be effectively used to translate large amounts of clinical and/or biomedical data of CKD into improved human health.

APPENDIX

Figures S1–S8 and Tables S1–S8 are available as Supplemental files.

REFERENCES

- [1] J. Coresh, "Update on the burden of CKD," *J. Amer. Soc. Nephrol.*, vol. 28, no. 4, pp. 1020–1022, Apr. 2017.
- [2] R. Kazancıoğlu, "Risk factors for chronic kidney disease: An update," *Kidney Int. Supplements*, vol. 3, no. 4, pp. 368–371, Dec. 2013.
- [3] G. Abraham, S. Varughese, T. Thandavan, A. Iyengar, E. Fernando, S. A. J. Naqvi, R. Sheriff, H. Ur-Rashid, N. Gopalakrishnan, and R. K. Kafle, "Chronic kidney disease hotspots in developing countries in South Asia," *Clin. Kidney J.*, vol. 9, no. 1, pp. 135–141, Feb. 2016.
- [4] T. J. Hoerger, S. A. Simpson, B. O. Yarnoff, M. E. Pavkov, N. R. Burrows, S. H. Saydah, D. E. Williams, and X. Zhuo, "The future burden of CKD in the United States: A simulation model for the CDC CKD initiative," *Amer. J. Kidney Diseases*, vol. 65, no. 3, pp. 403–411, Mar. 2015.

- [5] M. Almasoud and T. E. Ward, "Detection of chronic kidney disease using machine learning algorithms with least number of predictors," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 8, pp. 1–9, 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," *Future Gener. Comput. Syst.*, vol. 111, pp. 17–26, Oct. 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 *Proc. 3rd Int. Conf. Intell. Sustain. Syst. (ICISS)*, Dec. 2020, pp. 952–957.
- [8] I. U. Ekanayake and D. Herath, "Chronic kidney disease prediction using machine learning methods," in *Proc. Moratuwa Eng. Res. Conf. (MERCOn)*, Jul. 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. A. Imran, M. N. Amin, and F. T. Johora, "Classification of chronic kidney disease using logistic regression, feedforward neural network and wide & deep learning," in *Proc. Int. Conf. Innov. Eng. Technol. (ICIET)*, Dec. 2018, pp. 1–6.
- [11] G. Chen, C. Ding, Y. Li, X. Hu, X. Li, L. Ren, X. Ding, P. Tian, and W. Xue, "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," *Informat. Med. Unlocked*, vol. 26, Jun. 2021, Art. no. 100655.
- [14] D. Dua and C. Graff. (Accessed: Jul. 7, 2021). UCI Machine Learning Repository. University of California, School of Information and Computer Science. Accessed: 2019. [Online]. Available: <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, 2020.
- [16] P. Chittora, S. Chaurasia, P. Chakrabarti, G. Kumawat, T. Chakrabarti, Z. Leonowicz, M. Jasiński, Ł. Jasiński, R. Gono, E. Jasińska, and V. Bolshev, "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, Dec. 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 *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2019, pp. 1660–1664.
- [20] C. D. Galloway, A. V. Valys, J. B. Shreibati, D. L. Treiman, F. L. Petterson, V. P. Gundotra, D. E. Albert, Z. I. Attia, R. E. Carter, S. J. Asirvatham, and M. J. Ackerman, "Development and validation of a deep-learning model to screen for hyperkalemia from the electrocardiogram," *J. Amer. Med. Assoc. 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, Aug. 2019, Art. no. 105487.
- [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," 2019, *arXiv:1909.00735*.
- [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, Oct. 2016, Art. no. 6080814.
- [24] P. Yang, Y. Hwa Yang, B. B. Zhou, and A. Y. Zomaya, "A review of ensemble methods in bioinformatics," *Current Bioinf.*, vol. 5, no. 4, pp. 296–308, Dec. 2010.
- [25] B. Dickson. (2019). *What are Artificial Neural Networks (ANN)*. [Online]. Available: <https://bdtechtalks.com/2019/08/05/what-is-artificial-neural-network-ann/>
- [26] U. Naqvi, A. Majid, and S. A. Abbas, "UTSA: Urdu text sentiment analysis using deep learning methods," *IEEE Access*, vol. 9, pp. 114085–114094, 2021.
- [27] R. Dolphin. (2020). *LSTM Networks | A Detailed Explanation*. [Online]. Available: <https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9>
- [28] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1597–1600.
- [29] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*.
- [30] 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. Exp.*, vol. 29, no. 4, pp. 5923–5933, 2021.
- [31] 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.
- [32] D. Gupta. (2017). *Fundamentals of Deep Learning—Introduction to Recurrent Neural Networks*. [Online]. Available: <https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/>
- [33] J. Li. (2017). *Boosting Algorithm: AdaBoost*. [Online]. Available: <https://towardsdatascience.com/boosting-algorithm-adaboost-b6737a9ee60c>
- [34] J. Singh, B. Khan, M. A. Shah, K. Wakil, A. Khan, W. Alosaimi, M. I. Uddin, and B. Alouffi, "Performance assessment of classification algorithms on early detection of liver syndrome," *Artif. Intell. Med.*, vol. 4, no. 1, pp. 42–46, 2018.
- [35] N. A. Almansour, H. F. Syed, N. R. Khayat, R. K. Altheeb, R. E. Juri, J. Alhiyafi, S. Alrashed, and S. O. Olatunji, "Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study," *Comput. Biol. Med.*, vol. 109, pp. 101–111, Jun. 2019.
- [36] 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, Dec. 2003.
- [37] R. Dror, S. Shlomov, and R. Reichart, "Deep dominance—How to properly compare deep neural models," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2773–2785.
- [38] 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, Dec. 2020.
- [39] (2021). *National Kidney Foundation*. [Online]. Available: <https://www.kidney.org/>
- [40] K. D. Liu, W. Yang, A. H. Anderson, H. I. Feldman, S. Demirjian, T. Hamano, J. He, J. Lash, E. Lustigova, S. E. Rosas, M. S. Simonson, K. Tao, and C.-Y. Hsu, "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, May 2013.



SHAMIMA AKTER received the B.S. degree in biochemistry and molecular biology from the University of Dhaka, Bangladesh, and the M.S. degree in genomics and bioinformatics from Virginia Tech, in 2017, with machine learning research focus. Currently, she is pursuing the Ph.D. degree in bioinformatics and computational biology with George Mason University. From 2017 to 2019, she was a Research Associate in biomedical science at Virginia Tech to perform bioinformatics that

include Python, R, and Linux programming-based complex data analysis and machine learning-based virology/microbiology projects. She also performed networks analysis, different machine learning projects both in Python and R, and some deep learning projects in Python. Her research interests include artificial intelligence and machine learning in biomedical science i.e., liver disease, chronic kidney disease, mathematical modeling on Alzheimer's diseases, and computational biology.



AHSAN HABIB received the B.S. degree in computer science and engineering from Eastern University, Dhaka, Bangladesh. Since 2019, he has been actively engaged in research both in academia and professional settings. His expertise is in Python (Django), Flask, Celery, AI/ML, and Ruby (ROR). He has more than two years of professional experience in the industry, and always follows best practices of writing clean, testable, manageable code. He has been working on bubble and droplet tracking using Mask RCNN deep learning technique since January 2021. His research interests include artificial intelligence (AI), machine learning (ML), deep learning, OpenCV, and natural language processing.



MD. ASHIQUL ISLAM was born in Dhaka, Bangladesh, in 1998. He received the B.Sc. degree in computer science and engineering from Daffodil International University (DIU). Currently, he is working as a Research Scholar at the DIU Journal Analytica Research and Development Laboratory. He has hands-on experience in deep learning and machine learning-based techniques. His research interests include blockchain technology, cryptocurrency, applied cryptography, satellite, and radar communication, deep learning and machine learning techniques, and the IoT. He is the Reviewer of two journals: *Acta Agriculturae Scandinavica, Section B—Soil & Plant Science* (Taylor & Francis) and *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 HOSSSEN received the B.Sc. degree from the Department of Computer Science and Engineering, Daffodil International University, Bangladesh, in 2021. He has experience working on machine learning and deep learning programming. He is a passionate researcher and also a good problem solver. He can do data analysis and data visualization with programming. He has authored or coauthored more than 14 scientific publications. His research interests include artificial intelligence, computer vision, image processing, human-computer interaction, and computer security.



WASIK AHMMED FAHIM received the B.Sc. degree in computer science engineering from 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.



PUZA RANI SARKAR received the B.S. degree in computer science engineering from Daffodil International University, Dhaka, Bangladesh, in 2021. 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 interests include machine learning, deep learning, and artificial intelligence in medical science.



MANIK AHMED received the B.S. degree in civil engineering from the Bangladesh University of Engineering and Technology (BUET), in 1999, and the M.S. and Ph.D. degrees in civil engineering from Virginia Tech, Blacksburg, USA, in 2017 and 2021, respectively. He then 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 BIM and Primavera. Currently, he has been engaged in various research initiatives, such as smart city, urban networks improvement, and intelligent transportation systems using different machine learning and deep learning applications.

...