Saif *et al.*
Journal of Electrical Systems and Inf Technol (2024) 11:17
https://doi.org/10.1186/s43067-024-00142-4

Journal of Electrical Systems and Information Technology

RESEARCH Open Access

Early prediction of chronic kidney disease based on ensemble of deep learning models and optimizers

Dina Saif[1*], Amany M. Sarhan[1] and Nada M. Elshennawy[1]

*Correspondence:
dina68545@f-eng.tanta.edu.eg

[1] Department of Computers and Control Engineering, Faculty of Engineering, Tanta University, Tanta, Egypt

Abstract

Recent studies have proven that data analytics may assist in predicting events before they occur, which may impact the outcome of current situations. In the medical sector, it has been utilized for predicting the likelihood of getting a health condition such as chronic kidney disease (CKD). This paper aims at developing a CKD prediction framework, which forecasts CKD occurrence over a specific time using deep learning and deep ensemble learning approaches. While a great deal of research focuses on disease detection, few studies contribute to disease prediction before it may occur. However, the performance of previous work was not competitive. This paper tackles the under-explored area of early CKD prediction through a high-performing deep learning and ensemble framework. We bridge the gap between existing detection methods and preventive interventions by: developing and comparing deep learning models like CNN, LSTM, and LSTM-BLSTM for 6–12 month CKD prediction; addressing data imbalance, feature selection, and optimizer optimization; and building an ensemble model combining the best individual models (CNN-Adamax, LSTM-Adam, and LSTM-BLSTM-Adamax). Our framework achieves significantly higher accuracy (98% and 97% for 6 and 12 months) than previous work, paving the way for earlier diagnosis and improved patient outcomes.

Keywords: Chronic kidney disease, CKD, CKD prediction, Health risk prediction, Deep ensemble, Deep learning

Introduction

Chronic kidney disease (CKD) is a kidney disease caused by an inability to adequately filter blood. The basic function of the kidneys is to filter excess water and waste from human blood and eliminate them through urine [1]. In other words, when a person has CKD, waste builds up in their body, causing many harmful symptoms. Kidney damage develops gradually over time and can influence the rest of the human body, leading to serious disorders and might cause death [2].

Therefore, deep learning-based CKD prediction is an important application that predicts the medical condition before it begins, which tremendously contributes to saving people's lives. Many studies have demonstrated that if medical intervention started in the first or second trimester, high-risk problems can be avoided [3]. Disease detection suggests that the patient has the disease already, but disease prediction implies that it may arise in the future. As a result, scientists have attempted to forecast or diagnose kidney disease early. Currently available risk prediction models either do not provide patient specific risk factors or only predict in-hospital mortality rates. Machine learning models were applied to predict and calculate individual patient risk for disease occurrence or mortality. Disease detection implies that the patient already has the disease; whereas, disease prediction implies that it may occur in the future. Thus, scientists have attempted to detect kidney disease early, or predict its occurrence.

Several studies used Support Vector Machines and Artificial Neural Networks, Deep neural networks, an Ensemble algorithm, Extra tree, Random Forest, and Logistic Regression models to detect CKD at an early stage [1, 4–8]. Furthermore, Decision trees, Random Forest, LightGBM, Logistic Regression, and CNN models have been developed to predict CKD six to twelve months in advance [9].

Machine learning proved to be useful for detecting correlations in huge, complicated datasets. The field of precision medicine, in which disease risk is predicted using patient data, is one of the potential uses of machine learning. However, due to the vastly increased quantity of characteristics, developing an appropriate prediction model based on data remains difficult. As a result, feature selection improves the generalizability of machine learning models by extracting only the most "informative" features and removing noisy “non-informative,” irrelevant, and redundant information. This will help the decision makers in the medical filed to give a better decision about the action to be made to treat or even prevent this disease if the features identified could lead to such disease.

There are numerous studies in this field for CKD detection. However, only one study adopted CKD prediction [9] using Taiwan’s National Health Insurance Research Database (NHIRD) [10]. This dataset contains information on insurance claims made by patients between 1997 and 2012 and was used for the study, too. Every patient’s comorbidity or prescription is included in their record. The ICD 9 codes for the comorbidities and the ATC codes for the drugs indicate what they are. Consequently, many challenges emerge after reviewing the literature which motivated our research:

1. CKD data are scarce. Previous studies’ datasets were based on medical tests [4, 7–14]. It does, however, contain a limited number of samples (only 400 samples) [15].
2. Previous research concentrated on detecting the disease after it had already occurred [4, 7–14].
3. Due to the lack of data, research on this field has not been fully explored.
4. Only one study attempted to predict the CKD possible occurrence [9]. It, however, used an imbalanced dataset without providing a solution to the problem. Furthermore, it employed many features, which increased the computational cost. Finally, the performance of this work was low.

As a result, the novelty of this work is to investigate using optimized deep learning models, as well as using an ensemble model, for CKD prediction to enhance the prediction performance. In addition, we use large datasets from Taiwan’s National Health

Insurance Research Database (NHIRD) [16] that contain 90,000 samples as in [9], furthermore, we solve the problem of the imbalance of the dataset**.** As a summary of the contributions made, we list them in the following points:

1. We propose three deep learning predictive models to predict CKD six months and twelve months before disease occurrence, which are:

    1.1. Convolutional neural networks (CNN) model.
    1.2. Long short-term memory (LSTM) model.
    1.3. A combination of long short-term memory and bidirectional long short-term memory (LSTM-BLSTM) model.

2. A comparative evaluation of deep learning optimizers is presented for each model to induce the most powerful optimizer for the CKD dataset.
3. We propose an ensemble model that uses the majority voting technique to combine the three deep learning classifiers (CNN, LSTM, and LSTM-BLSTM), where each is optimized by the best optimizer chosen in stage 2, to improve the classification performance.
4. We train each model for CKD prediction using two public benchmark datasets [10]. The main drawback of these datasets is the imbalance between the two classes, which been addressed using SMOTE (Synthetic Minority Oversampling Technique). The second flaw is the large number of features in the datasets. We remedied it by reducing the number of features using the Random Forest feature selection algorithm.
5. Finally, we assess the predictive models' performance using various metrics to investigate their advantages and disadvantages. To demonstrate the strength of the proposed models, the results are compared to the state-of-the-art work [9] using the same datasets.

This paper is organized as follows. "Related work" section reviews previously developed approaches in CKD detection and prevention. The dataset is presented in "Materials and methodology" section and the proposed models are described in detail. "Proposed models evaluation" section evaluates the proposed predictive models, draws a comparative analysis, and discusses the prediction results. "Conclusion and future work" section concludes this paper.

Related work

Risk detection and prediction for chronic kidney disease

Many existing risk models have been introduced for a variety of diseases to reduce mortality. Given the riskiness of kidney disease to human health, scientists have attempted to detect it early or predict its occurrence in advance. Disease detection implies that the patient already has the disease; whereas, disease prediction implies that it may occur in the future. Consequently, research can be classified into two types: detection and prediction. In aspects of the first type, almost all studies used the same datasets [15], to detect CKD.

Qin et al. [11] used the machine learning models to classify the patients with CKD. The highest accuracy reached 99.75% using random forest developed an intelligent classification technique for CKD called density-based feature selection (DFS) with Ant Colony-based Optimization (D-ACO). This technique tackled the increased number of features in medical data issues by removing redundant features. It also overcomes low interoperability, high computation, and overfitting issues. This technique achieved 95 percent detection accuracy with 14 out of the 24 features.

Jongbo et al. [1] achieved 100% accuracy using an ensemble algorithm that consists of Random Subspace and Bagging. The data are preprocessed, missing values are handled, and the data are eventually normalized. This method was created by combining three base-learners: KNN, Nave Bayes, and Decision Tree. Combining the basis classifiers increased classification performance, according to this study. The suggested model outperformed individual classifiers in the experiments. The random subspace method beat the bagging technique in the majority of situations.

Chittora et al. [12] detected CKD using full or important features. Many techniques were used such as: correlation-based feature selection, wrapper technique feature selection, minority oversampling. Seven types of classifiers were employed including ANN, LSVM, and LR. LSVM attained the maximum accuracy of 98.86% using complete features in the synthetic minority oversampling approach.

Ma et al. [13] proposed an efficient method called the Heterogeneous modified artificial neural network (HMANN), for the detection and diagnosis of chronic kidney disease (CKD). The HMANN model is a hybrid model that combines a support vector machine (SVM) and a multilayer perceptron (MLP) classifier. The SVM is used to classify the presence of cyst or stone in the kidney; while, the MLP classifier is used to diagnose CKD. Overall, the HMANN model is a promising new approach for the detection and diagnosis of CKD. It achieved the highest accuracy on the test set compared to the traditional machine learning algorithms. The HMANN model also uses several techniques to improve its accuracy, such as data augmentation, feature selection, and model regularization.

The accuracy of several machine learning algorithms was examined for diagnosing CKD and discriminating between CKD and non-CKD patients [6]. The authors employed Logistic Regression, SVM, and KNN models to detect CKD where SVM model outperformed the other strategies, with an accuracy of 99.2%.

Machine learning approaches were employed in developing a CKD diagnostic system in [7]. To replace missing data, the mean and mode were applied, and Recursive feature elimination (RFE) was used to choose the most significant features while support vector machine (SVM), random forest (RF), k-nearest neighbors (KNN), and decision tree (DT) are the machine learning methods employed. Among these four classifiers, the random forest (RF) method best the others, obtaining 100% accuracy.

Another research creates a group of deep learning-based clinical decision support systems (EDL-CDSS) for CKD diagnosis [14]. The EDL-CDSS method calls for the creation of Adaptive Synthetic (ADASYN) technology for the outlier detection procedure. Additionally, three models [deep belief network (DBN), kernel extreme learning machine (KELM), and convolutional neural network with gated recurrent unit (CNN-GRU)] are used in an ensemble. The DBN and CNN-GRU models' hyper-parameters are lastly tuned using the quasi-oppositional butterfly optimization algorithm (QOBOA).

Also in 2022, another research aims to create a deep neural network model for predicting chronic kidney disease (CKD) and compare its performance to other machine learning techniques. The hyperparameter optimization is used and the Recursive Feature Elimination to identify key features. This research has achieved 100% on the test set. This is significantly higher than the accuracy of traditional machine learning algorithms.

A deep neural network was proposed by Singh et al. [17]. The average of the related feature was used to replace missing values, and the recursive feature elimination (RFE) technique was used to pick features. Deep neural network DNN, Nave Bayes classifier, KNN, Random Forest, and Logistic regression were used to classify the specified characteristics. In terms of accuracy, DNN surpassed all other models.

In 2023, the proposed deep neural network-based Multi-Layer Perceptron classifier can accurately diagnose CKD, achieving 100% testing accuracy. This model outperforms standard machine learning models used in this research and provides a promising alternative for CKD diagnosis [18].

Disease risk prediction work was proposed in [9]. Over a two-year period, the predictive model was built utilizing comorbidity, demographic, and medication data from patients. Their CNN model got the best AUROC of 0.954 and 0.957 for 12-month and 6-month forecasts, with accuracy of 88% and 89%, respectively. Gout, diabetes mellitus, age, and drugs such as angiotensin and sulfonamides were the most important predictors. Table 1 provides a summary of modern health risk detection and prediction algorithms.

As seen from the summary, all the previous work relied on a small medical dataset, which contains only 400 samples, and based on medical features, they could reach 100% accuracy using this dataset. On the other hand, the work in [9] used another dataset [10] containing comorbidity, demographic, and medication data from patients over two years. It attempted to predict CKD's possible occurrence. It, however, used an imbalanced dataset without providing a solution to the problem. Furthermore, it employed a large number of features, which increased the computational cost. Finally, the performance of this work was low (89 and 88%). Hence, in our work, we use the same dataset [10] and try to increase the performance by solving these issues and developing a robust model.

Ensemble in disease detection

According to a massive amount of research in the machine learning field, two algorithms currently dominate this field: Ensemble and Deep Learning algorithms. Deep learning is the gold standard of machine learning algorithms, and deep ensemble algorithms are a catch-all term for approaches that combine multiple deep learning classifiers to make a decision [19]. Thus, in this research, we use an ensemble algorithm in conjunction with deep learning approaches. Deep learning techniques, on the other hand, are regarded as the most dominant and powerful players in a variety of machine learning challenges. The use of this algorithm improves detection and prediction accuracy by avoiding the drawbacks of traditional learning techniques [20]. Over the last few years, many algorithms that combine ensemble algorithms and deep learning models have been

Table 1 Summary of recent health risk detection and prediction models for CKD

Paper	Detection/ prediction	Dataset/ number of samples	Dataset description	Algorithm	Highest accuracy
Qin-2019 [11]	Detection	Not available	Demographic features, Laboratory results, and ultrasound images	Logistic regression, random forest, support vector machines	99.75%
Jongbo-2020 [1]	Detection	CKD dataset-1/400 [15]	Demographic features: age, sex, race, and ethnicity. Medical features: blood pressure, blood sugar	Ensemble: KNN, NB, DT	100%
Ma-2020 [13]	Detection	Not available	Ultrasound images	SVM, DT, RF, KNN, HMANN	HMANN 98%
Gudeti-2020 [7]	Detection	CKD dataset-1/400 [15]	Demographic features: age, sex, race, and ethnicity. Medical features: blood pressure, blood sugar	SVM, LR and KNN	SVM 99.2%
Chittora-2021 [12]	Detection	CKD dataset-1/400 [15]	Demographic features: age, sex, race, and ethnicity. Medical features: blood pressure, blood sugar	C5.0, CHAID, ANN, LSVM, LR, RT and KNN	LSVM 98.86%
Senan-2021 [8]	Detection	CKD dataset-1/400 [15]	Demographic features: age, sex, race, and ethnicity. Medical features: blood pressure, blood sugar	SVM, RF, KNN, DT	RF 100%
Alsuhibany-2021[14]	Detection	CKD dataset-1/400 [15]	Demographic features: age, sex, race, and ethnicity. Medical features: blood pressure, blood sugar	Ensemble (DBN, KELM, CNN-GRU)	%96.9
Krishnamurthy-2021 [9]	Prediction	CKD dataset-2/90,000 [10]	Comorbidities, medications, age, gender,	CNN, BLSTM, LightGBM, LR, RF, DT	CNN 89% (6 months) 88% (12 months)
Singh-2022 [17]	Detection	CKD dataset-1/400 [15]	Demographic features: age, sex, race, and ethnicity. Medical features: blood pressure, blood sugar	SVM, KNN, LR, RF, Naive Bayes and DNN	DNN 100%
Sawhney-2023 [18]	Detection	Not available	age, blood sugar, red blood cell counts and medical data	SVM, DT, MLP, RF, LR	MLP 100%

developed to improve predictive models' performance. The deep ensemble learning algorithm combines the benefits of both deep learning and ensemble learning to produce a final model with the best generalization performance [21].

The essential logic for ensemble originates from the inclination to collect various points of view and combine them to make a challenging conclusion. Using one of the combination methods [average ensemble (AE), weighted average ensemble (WAE), rank average ensemble (RAE), and majority voting ensemble (MVE)], this notion relies on merging many base-learners to generate a classifier that outperforms them all. Recently, machine learning researchers proved that integrating the outputs of many classifiers increases the performance of a single classifier through hands-on experimental study [18]. Because of its influence on numerous variables, the ensemble approach has been employed in a range of applications, including illness diagnosis and prediction.

Individual classifiers suffer from issues, such as overfitting, class imbalance, concept drift, and the problem of dimensionality, which cause a single classifier prediction to fail [22]. As a result, ensemble learning method has emerged in scientific research to address these issues. Prediction accuracy improves by using this algorithm in different machine learning challenges.

The ensemble learning combines a set k of independent classifiers, c1, c2,..., ck, to give a single output using a combination function f. Given:

- dataset of size n ($D = (x, y)$, $1 \leq \mathrm{i} \leq \mathrm{n}$), and
- features of dimension m, $\mathrm{x}_{\mathrm{i}} \in \mathrm{R}^{\mathrm{m}}$

Equation (1) [19] predicts the output of this approach as:

$$y_i = \varnothing(x_i) = f(c1, c2, ..., ck) \tag{1}$$

Table 2 presents a summary of previous research using the Ensemble technique for the disease detection field.

Materials and methodology

Dataset description

To begin, we are focused in health risk prediction rather than detection. As a result, we chose Taiwan's National Health Insurance Research Database (NHIRD), a public dataset, because it is the only dataset that is concerned with prediction [10]; whereas, the other accessible dataset was dedicated to CKD detection. This dataset was collected by monitoring and recording patients' data for two consecutive years and then classifying them as infected or non-infected with the disease.

The NHIRD dataset includes 965 comorbidities (ICD-9 codes), 537 medications (ATC codes), age, gender, and a CKD class label (0 = no CKD, 1 = CKD). Table 3 shows a sample of the dataset we used. Each feature represents a number, which indicates how many times during the observation period the patient was infected with the disease or took the medication.

Table 2 Literature using ensemble techniques in health risk prediction

Paper	Dataset	Algorithm	Highest accuracy
Raza-2019 [23]	Heart disease dataset–statlog	MVE	88.88%
Atallah-2019 [24]	Heart disease dataset [25]	MVE	90%
Yadav-2019 [26]	Breast cancer Wisconsin (Original)	AE-MVE-WAE	(AE) 0.9998AUC
	Breast cancer Wisconsin (Diagnostic)		(AE) and (RAE) 100% AUC
	Haberman's Survival Dataset		(AE) 0.636
	Heart disease Dataset (Hungarian)		(AE) 0.8994
	Indian liver Patient Database		(AE) 0.7892
	Mammographic Mass Dataset		(AE) 0.8708
	single-proton Emission Computed Tomography (SPECT)		(WAE) 0.8166
	SPECTF heart-imaging Dataset		(RAE) 0.8166
	Statlog (heart) Dataset		(RAE) 0.9272
	Vertebral column Dataset		(AE) and (RAE) 0.9504
Tao Zhou-2021 [27]	The data are available from the author upon request	MVE	99.05%
Chandra-2021 [28]	COVID-chest X-ray [29]	MVE	98.062% Phase-I 91.329% Phase-II
Aurna-2022[30, 31]	Brain tumor	MV	100% training 93% testing
Hireš-2022 [32]	Parkinson's disease	MV	99%

Table 3 A sample of the dataset used in this study

Age	Sex	250	272	A03FA	C09AA	J07BB	C08CA	A10BB
84	1	11	0	1	19	1	0	0
54	1	20	0	3	0	2	11	0
86	1	22	10	16	14	0	0	8
75	0	18	0	1	0	0	0	9
49	1	0	0	0	0	3	0	0
71	1	0	2	3	10	5	10	10

We list the following explanations for the features in the table:

- 250: Diabetes mellitus disease
- 272: Disorders of lipoid metabolism
- A03FA: Propulsive drugs which stimulate gastrointestinal motility.
- C09AA: ACE inhibitors, which block the action of angiotensin-converting enzyme (ACE)
- J07BB: Influenza vaccines
- C08CA: Calcium channel blockers and dihydropyridine derivatives, which work by blocking calcium channels in the heart and blood vessels
- A10BB: Sulfonylureas, which are a class of oral antidiabetic drugs that work by stimulating the pancreas to release more insulin

Our dataset contains too much data to explain in detail here. The dataset includes two sub datasets; the first dataset's certain period is six months; while, the second is 12 months. This dataset is highly imbalanced, with 90,000 patients divided into 18,000 with CKD and 72,000 without CKD. To declare the robustness of our models, we will compare our results with the previous study using the same dataset [9].

Methodology

Our goal in this work was to create CKD prediction models that can handle the problems defined in the introduction section. The prediction problem is treated as a classification problem, with the output of the model being either 0 or 1: (0 indicates that the patient will not develop CKD after the specified period, while 1 indicates that they may develop CKD after the specified period). In this section, we present the architecture of the four proposed predictive models for chronic kidney disease (CKD). Because there has been only one research directed toward solving this problem [9], we intend to use deep learning models to explore different models for the problem. Unfortunately, the previous study did not consider the significant imbalance of the benchmark datasets [9]. Furthermore, a large number of features were trained, which could lead to a variety of issues such as limited interoperability, high computation, and overfitting. Furthermore, using the LightGBM algorithm, the highest accuracy in this study with the same aggregated file was 75.1%. We attempt to find solutions for each of the previous issues.

Figure 1 depicts a block diagram of the methodology used in this study. To begin, the SMOTE (Synthetic Minority Oversampling Technique) is used to deal with the problem of imbalanced dataset. Second, the Random Forest feature selection technique is used to reduce the number of features, and only the most important ones are displayed. Third, after oversampling, the selected features and samples are divided into 80% training and 20% validation. Fourth, for each deep learning classifier, a comparative analysis of deep learning optimizers is performed to identify the most robust one. Fifth, the Ensemble model employs the most robust optimizers. Sixth, our findings are compared to the findings of the only published study with the same objective on the sane dataset [9].

SMOTE (synthetic minority oversampling technique)

A dataset is called “imbalanced” if the classification categories are not roughly equally represented in this dataset. The datasets representing real-world data are frequently composed primarily of “normal” samples, while containing only a small percentage of “abnormal” samples. The predictive accuracy of machine learning algorithms is commonly used to assess their performance. This may not be appropriate, however, when the data are unbalanced and/or the costs of different errors vary significantly. Under-sampling of the majority (normal) class has been proposed as an effective method of increasing a classifier’s sensitivity to the minority class. Oversampling the minority (abnormal) class is another approach to overcome the imbalance of the dataset.

SMOTE is a method of oversampling in which the minority class is oversampled by producing “synthetic” samples rather than oversampling with replacement. This strategy performed well in a variety of applications, including handwritten character recognition and image classification. SMOTE generates additional training data by applying specific processes to actual data. The following is the Pseudo-Code for the algorithm [33] (Fig. 2).

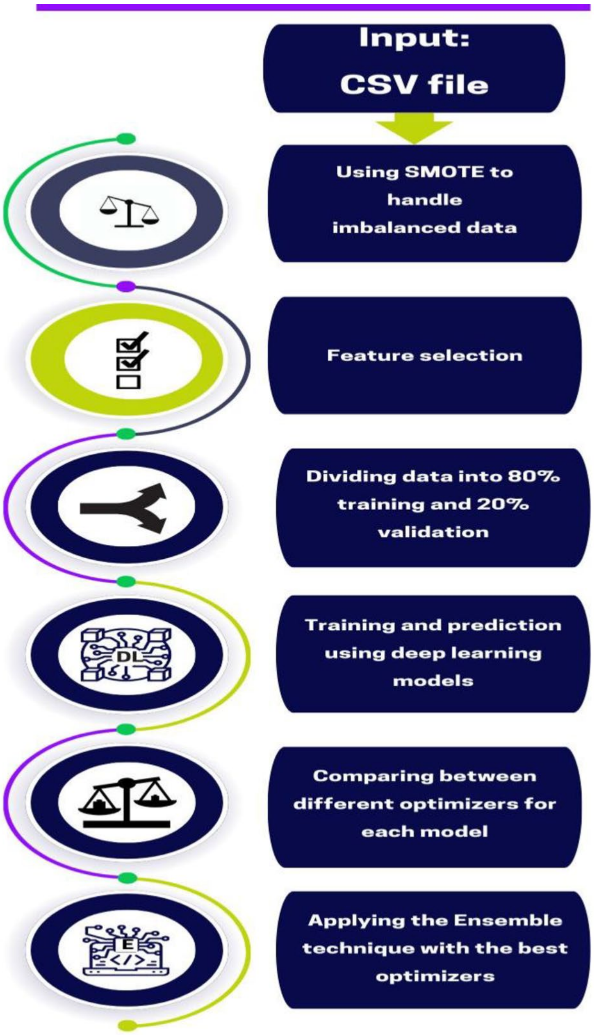
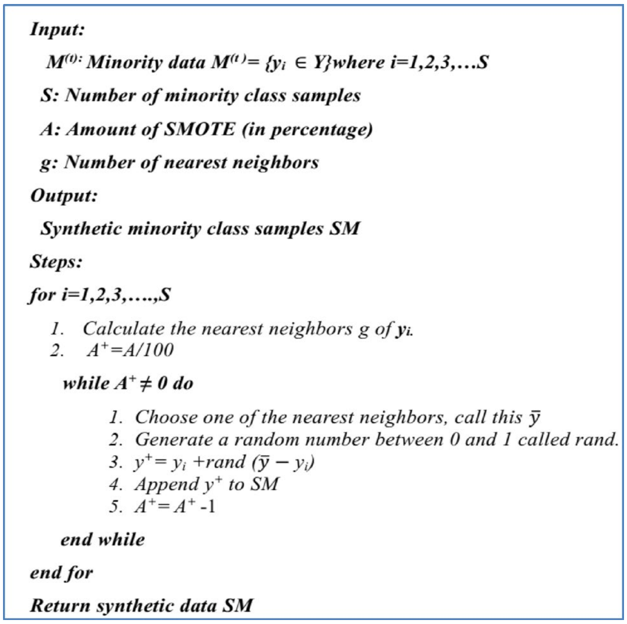

Input:
CSV file
Using SMOTE to handle imbalanced data
Feature selection
Dividing data into 80% training and 20% validation
DL
Training and prediction using deep learning models
Comparing between different optimizers for each model
E
</>
Applying the Ensemble technique with the best optimizers

Fig. 1 Block diagram of the methodology used in this study

In order to prevent a biased model that would perform poorly on positive cases, it was crucial to resolve the data imbalance. The SMOTE technique was chosen to address data imbalance in a CKD dataset with a smaller number of positive cases [9]. SMOTE creates synthetic samples by interpolating between existing minority class samples, increasing the number of minority class samples, and balancing the dataset. However, SMOTE may not be suitable for all datasets, as it may lead to poor performance. Addressing data imbalance was crucial for developing an accurate and reliable deep learning model for early detection and prediction of CKD.

With 18,096 samples in the CKD class and 71,912 samples in the non-CKD class, there is a considerable class imbalance in our example. As a result, the model will seldom anticipate the CKD class. To reduce false negatives, we used SMOTE. Using SMOTE usually results in an increase in the recall parameter. This implies that the number of minority class projections will be increased. After using SMOTE method, the dataset reaches 143,824 individuals, equally split between those with and without



```
Input:
  M^(t): Minority data $M^{(t)} = \{y_i \in Y\}$ where i=1,2,3,…S
  S: Number of minority class samples
  A: Amount of SMOTE (in percentage)
  g: Number of nearest neighbors
Output:
  Synthetic minority class samples SM
Steps:
for i=1,2,3,….,S
    1. Calculate the nearest neighbors g of $y_i$.
    2. $A^+ = A/100$
  while $A^+ \neq 0$ do
        1. Choose one of the nearest neighbors, call this $\bar{y}$
        2. Generate a random number between 0 and 1 called rand.
        3. $y^+ = y_i + rand\,(\bar{y} - y_i)$
        4. Append $y^+$ to SM
        5. $A^+ = A^+ - 1$
  end while
end for
Return synthetic data SM
```

Fig. 2 Pseudocode for SMOTE algorithm

CKD. We set the SMOTE variables as follows: *sampling_strategy = 'minority', random_state = 42, k_neighbors = 5, S = 18,096, and A = 398%.*

Features selection using random forest

The features selection technique is used to provide high-quality data that only contains the most crucial features because the acquired data frequently contains additional features. Additionally, the model's complexity could be decreased, preventing model overfitting [34, 35]. In the random forest technique (RF), one of the crucial criteria for choosing features is their relevance. In our study, we employed a feature selection process to identify the most important features for CKD prediction, as follows:

1. Set up the decision trees where each decision tree in the random forest is sampled with a random put back to create sub-data sets.
2. Create sub-decision trees by ensuring that each decision tree produces a result, and that each sub-decision tree calculates the output result of the sub-data set.
3. The outcome of the voting in the sub-decision tree determines the output result of the random forest.
4. Determine the number of classification errors Ei of out-of-bag data in each sub-decision tree.
5. Disrupt the value of each decision tree's out-of-bag data (X) at random and recalculate the number of classification errors Ex_i.
6. Determine significance and confirm feature selection. Make i equal to 1, 2…, n, where n is the total number of random forest decision trees.
7. Repeat again steps 3 and 4.

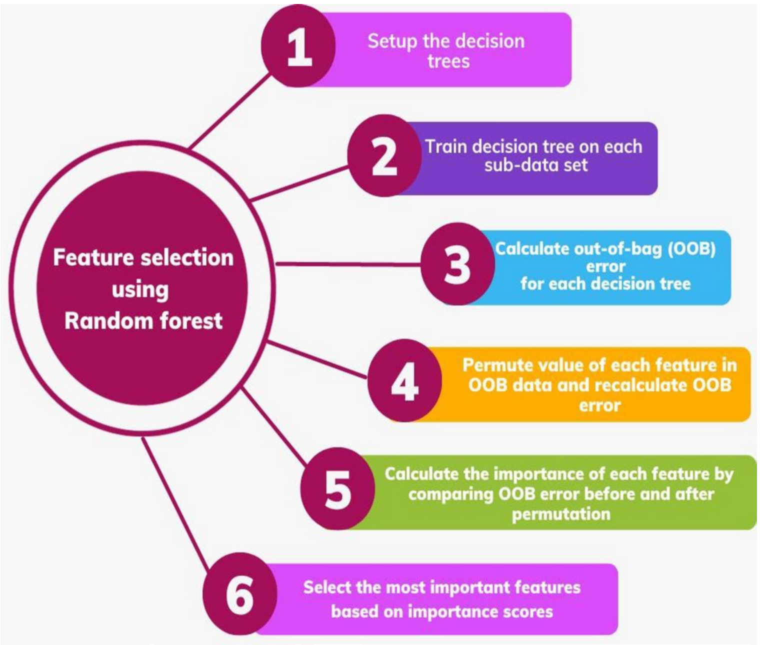

Feature selection using Random forest
1 Setup the decision trees
2 Train decision tree on each sub-data set
3 Calculate out-of-bag (OOB) error for each decision tree
4 Permute value of each feature in OOB data and recalculate OOB error
5 Calculate the importance of each feature by comparing OOB error before and after permutation
6 Select the most important features based on importance scores

Fig. 3 Features selection process using random forest

The following formula expresses the significance of feature [36].

$$lx = \frac{1}{n}\sum_{i}^{N}[\mathrm{Ex_i} - \mathrm{E_i}] \tag{2}$$

Figure 3 represents the features selection process briefly.

Deep learning optimizers

Deep learning is a branch of machine learning that is used to carry out difficult tasks such as health risk prediction and image classification. An activation function, input, output, hidden layer, loss function, and other components make up a deep learning model.

We require both an optimization method as well as an algorithm that maps instances of inputs to outputs. When mapping inputs to outputs, an optimization method determines the value of the parameters that minimizes the error. The effectiveness of the deep learning model is significantly impacted by these optimization methods. They also have an impact on the model's speed training. We must adjust the weights for each epoch during deep learning model training and reduce the loss function. An optimizer is a procedure or method that alters neural network properties like weights and learning rates. As a result, it aids in decreasing total loss and raising precision. The following are the most popular deep learning optimizers.

Deep learning optimizers facilitate the analysis of complex datasets, extract meaningful insights, enhance the interpretability of the results, and increase the model's accuracy. Using the best optimizer with the model aids clinicians and researchers in understanding the underlying factors influencing therapy intensification and improves decision-making regarding therapy intensification.

Stochastic gradient descent (SGD) The effectiveness of SGD algorithms has been demonstrated in the optimization of massive deep learning models. Since the word "stochastic" refers to a procedure connected to a random possibility, only a few samples are randomly selected for each iteration rather than the complete dataset [36]. By altering the network structure after each training stage, SGD seeks to determine the global minimum. Instead of locating the gradient for the entire dataset, this method just decreases the error by approximating the gradient for a randomly selected batch [37].

Adaptive gradient descent (AdaGrad) This optimizer uses several learning rates for every model parameter. It adjusts the learning rate in accordance with how frequently each parameter is updated. The learning rate will decrease with a higher parameter gradient and vice versa [38].

Adaptive delta (Adadelta) This is an extension of the Adagrad optimizer that accumulates earlier gradients over a predetermined time window to ultimately guarantee that learning will continue even after numerous iterations. Adadelta removed the learning rate from the update rule and applied Hessian approximation to verify the update direction in the negative gradient [39].

Adaptive moment estimation (Adam) Adam is an SGD optimization technique that calculates the rates at which each parameter learns to change [40]. The phrase "Adaptive Moments" inspired the name. It combines Momentum and RMSProp. The upgrading method provides a bias correction technique and considers the smooth gradient variant. Adam is invariant to gradient diagonal rescaling, requires less execution memory, and reduces computing costs [41].

Maximum adaptive moment estimation (AdaMax) It is a variation of Adam's adaptive SGD that is based on the infinity norm. The main advantage of AdaMax over SGD is that it is far less sensitive to the choice of hyper-parameters [42]. The second momentum component of the Adam estimate method is fully utilized in the AdaMax equation. This provides a more dependable answer [43].

In our models, optimizers are used to update the model's parameters during the training process to minimize the loss function. The optimization process in our model involves the following steps: In training phase, the process involves initializing model parameters, such as weights and biases, with small random values. Subsequently, during each training iteration, input data are propagated through the network to make predictions, and a loss function quantifies the disparity between predicted and actual target values. Gradients of this loss with respect to the model parameters are then computed through backpropagation, employing the chain rule to propagate errors through the network layers. Finally, an optimizer uses these gradients to iteratively adjust the model's parameters, ultimately minimizing the loss and improving the model's performance over time.

Deep ensemble predictive model (DEM)

Ensemble learning methods are usually used to improve prediction performance when a single classifier is insufficient to achieve a high-performance level. The main idea behind this predictive model is to aggregate a group of different individual classifiers to improve performance by combining a weak classifier with a strong classifier to increase the efficiency of the weak learner.

The study employs (CNN), (LSTM), and (LSTM-BLSTM) models to analyze patient medical data. CNNs are ideal for processing high-dimensional data, such as images and time-series data, by learning local patterns and spatial relationships. LSTMs handle sequential data, capturing temporal patterns and trends providing them a suitable option for forecasting future events based on observations of the past. LSTM-BLSTM captures both forward and backward dependencies in the input sequence, making it more effective in modeling complex temporal relationships. Combining these models can enhance the accuracy of CKD prediction.

In our proposed ensemble model, we combine CNN, LSTM, and LSTM-BLSTM models to produce an effective computational model for CKD prediction based on a majority voting ensemble, as shown in Fig. 4, where each classifier outputs a prediction, which is represented as p1, p2, and p3 in the figure. The majority voting ensemble was chosen due to its robustness and because it is less biased toward the outcome of a particular individual learner. Furthermore, its impressive results in disease detection are documented in the literature [23, 24, 26–28, 30, 32]

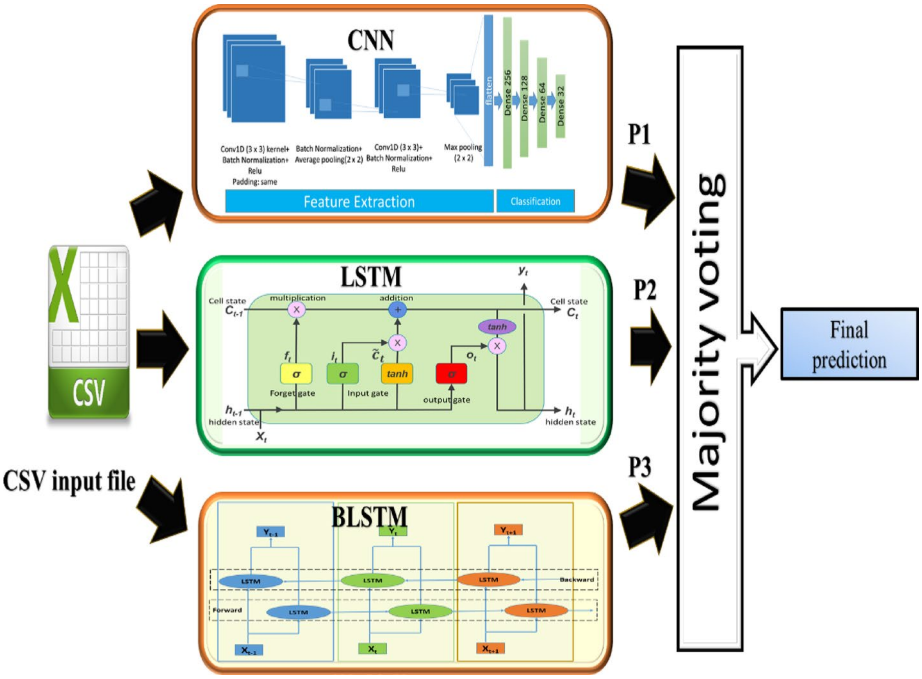


Fig. 4 Structure of the proposed ensemble CKD predictive model

First model in the ensemble: convolutional neural network (CNN)-CKD predictive model The first model in the Ensemble model is based on 1D CNN to generate a fast, generic, and highly accurate CKD predictive model. The 1D convolution is represented by the following equation [32]:

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} \mathrm{conv1}D\left(w_{ik}^{l-1}, s_i^{l-1}\right) \tag{3}$$

where b_k^l is the bias for layer l of the kth neuron, x_k^l is the input for the same layer, s_i^{l-1} is the output of the ith neuron at layer $l-1$, w_{ik}^{l-1} is the kernel (filter) from layer $l-1$ to layer l. The output, y_k^l, can be calculated by passing the input x_k^l through the activation function as follows [32]:

$$y_k^l = f\left(x_k^l\right) \tag{4}$$

The back-propagation algorithm (BP) is then used to reduce the output error. This algorithm works its way backwards from the output layer to the input layer. Consider the output layer (L). The number of classes is represented by N_{L}, and for an input vector p, the target and output vectors are represented by t_i^p and $[y_1^L,\ldots,y_{NL}^L]$, respectively. As a result, the mean-squared error (MSE), E_{p}, can be computed as follows [32]:

$$E_{\mathrm{p}} = \sum_{i=1}^{N_L} \left(y_i^L - t_i^p\right)^2 \tag{5}$$

The derivation is used, and various gradients of the neurons are computed recursively. As a result, the network's weights are updated until the least error is reached.

Second model in the ensemble: long short-term memory (LSTM)-CKD predictive model LSTM is a type of deep learning network model that is frequently used in time-series signals analysis, in addition to single data points as the images. The most significant advantages of this model are [44]: it has a higher accuracy in long-term dependency problems than recurrent neural network (RNN). Furthermore, vanishing gradients problems can be solved using memory blocks using this technique. These blocks are controlled by adaptive multiplicative gates, which retrieve or ignore information based on its importance. The LSTM unit consists of an input gate $\mathrm{I_t}$, an output gate $\mathrm{O_t}$ and a forget gate $\mathrm{F_t}$. The three gates' activations are computed using the following equations [45]:

$$I_t = \sigma(W_i X_t + R_i H_{t-1} + b_i) \tag{6}$$

$$F_t = \sigma\left(W_f X_t + R_f H_{t-1} + b_f\right) \tag{7}$$

$$O_t = \sigma(W_o X_t + R_o H_{t-1} + b_o) \tag{8}$$

The sigmoid activation function and the current input are represented as σ and X_t, respectively. The input weights are denoted as W_i, W_f and W_o while b_i, b_f and b_o are the bias. While the recurrent weights are symbolized as R_i, R_f and R_o. The output of the previous block is represented as H_{t-1}. The modified new memory $\overline{C}_{\mathrm{t}}$ is computed as in Eq. (9)) [45]:

$$\overline{C}_t = \tanh(W_t X_t + R_t H_{t-1} + b_t) \tag{9}$$

where tanh $(\cdot)$ represents the hyperbolic tangent function, R_t and W_t denote the recurrent weight and input weight respectively. The computation of the current memory cell $\mathrm{C_t}$ is illustrated as in Eq. (10) [45]:

$$C_t = F_t \odot C_{t-1} + I_t \odot \overline{C}_t \tag{10}$$

where C_{t-1} represents the previous memory cell, while $\odot$ indicates the element-wise multiplication operation. To calculate the LSTM output $\mathrm{H_{t,}}$ the following equation is used [45]:

$$\mathrm{H_t} = \mathrm{O_t} \odot tanh(\mathrm{C_t}) \tag{11}$$

We use LSTM in our model to avoid the vanishing gradient problem and to build a high-performance computational framework predictive model. The model is made up of an LSTM layer with 500 hidden units. Then, another LSTM layer with 200 hidden units is added. The previous layers are followed by a dense layer of 128 neurons. A dropout is used, followed by a second dense layer of 64 neurons. The dropout is used again to avoid overfitting and improve model performance. Following these layers is a dense layer of thirty-two neurons, which is finally connected to another dense layer for CKD prediction.

Third model in the ensemble: LSTM-BLSTM model As shown in Fig. 4, the third model in the ensemble is a hybrid model that combines LSTM and BLSTM to improve the performance of the ensemble models. The hybrid models used in many applications and achieve high accuracy in many fields [45–47] A Bidirectional LSTM (BLSTM) is an enhanced version of LSTM. BLSTM is made up of two LSTMs that work in opposite directions (forward and backward). The amount of information available to the network has increased because of using this model, and the accuracy has reached high efficiency. The forward direction is represented by h_t^f that denotes the input in ascending order, i.e., t=1, 2, 3… T. The opposite direction is represented by a backward hidden layer called h_t^b, which represents the input in descending order, i.e., t=T…,3,2,1. Finally, y_t is generated by combining h_t^f and h_t^b. The BLSTM model is represented by the following equations [44]:

$$h_t^f = H\left(W_{xh}^f X_t + W_{hh}^f h_{t-1}^f + b_h^f\right) \tag{12}$$

$$h_t^b = H\left(W_{xh}^b X_t + W_{hh}^b h_{t+1}^b + b_h^b\right) \tag{13}$$

$$y_t = W_{hy}^f h_t^f + W_{hy}^b h_t^b + b_y \tag{14}$$

where W is a weight matrix (W_{xh}^{f} is a weight that connects input (x) to the hidden layer (h) in the forward direction, while W_{xh}^{b} is the same but in the backward direction). b_{h}^{f} is a forward direction bias vector; whereas, b_{h}^{b} is a backward direction bias vector, The out is symbolized by y_t [44, 48]. This model is composed of LSTM, BLSTM, flatten, dense 128, dropout, dense 64, dropout, dense 32 which is finally connected to another dense layer for CKD prediction.

Proposed models evaluation

The experiments are carried out using a publicly available dataset [10] that contains two different types of samples. The first sample represents CKD prediction over six months; while, the other sample represents CKD prediction over twelve months. The dataset is divided into 80% training and 20% testing. To check the model's performance, we cutoff 20% of the training data for use as a validation set. A convenient feature in Keras framework called "validation_split" is used to achieve that, which automatically sets aside a portion of the training dataset for validation. Usually, this split is expressed as a ratio or percentage of the training set which represents 20%. The validation data are used to track the model's performance on unseen data and detect potential overfitting as it is trained on the remaining portion of the data.

The models were implemented using Python 3 involving the Keras framework running on Google Colab using a GPU on processor: (Intel(R) Xeon(R) CPU @ 2.20GHz) with 13 GB RAM. The classification process used by the trained deep learning models is applied on the validation dataset. As for the Ensemble model, when a test sample is fed to it, it is first distributed to all individual models. Next, each classifier produces a prediction. After that, the majority voting technique is applied to all base classifiers' results to generate the final prediction.

Performance metrics

To compare the models' performance, four commonly used performance evaluation metrics were used: true negative (TN), true positive (TP), false negative (FN), and false positive (FP). Furthermore, four metrics are used in the evaluation: Recall, Precision, Accuracy, and $F1$_score which are calculated as given in Eqs. (15)–(18). A recall is the number of positive instances predicted from the total number of positive instances; it is also known as sensitivity or true positive rate. Precision, also known as Positive Predictive Value, is the number of instances predicted as positive out of the total number of samples predicted as positive. Accuracy is defined as the number of correctly predicted instances divided by the total number of instances. $F1$-score combines Precision and Recall into a single metric using their harmonic mean. The number of instances predicted as negative out of the total number of negative instances is referred to as specificity [10].

$$\text{Recall}\ (\text{Sensitivity}) = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{17}$$

$$F_1\_\text{score} = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision } + \text{Recall}} \tag{18}$$

where TP denotes true positive or correctly classified positive class, TN denotes true negative or correctly classified negative class, FP denotes false positive or incorrectly classified positive class, and FN denotes false negative or incorrectly classified negative class.

To assess the impact of the proposed deep ensemble approach on prediction results, we ran several experiments on the benchmark datasets and compared the ensemble's performance to all individual models. Finally, we present all experimental results and compare them to previous results in [9].

Experimental results and comparative analysis

This section includes the performance prediction for deep learning models. As shown in Fig. 1, which depicts the flow of the model development process, the process consists of three main steps: data preprocessing, model training, and model evaluation. The first step in the preprocessing phase is to handle the imbalanced data issue using SMOTE technique. In the six-month dataset, there are 90,082 samples total, of which 71,912 are non-CKD samples and 18,096 are CKD samples. The dataset is oversampled using the SMOTE approach, reaching 143,824 divided equally between CKD and non-CKD. In the 12-month dataset, there are 71,271 non-CKD samples, but only 18,025 CKD samples. After using the SMOTE approach, 142,542 samples are obtained, evenly divided between the two classes.

We chose SMOTE for its simplicity and effectiveness in handling imbalanced datasets. It generates synthetic data points within the existing feature space of the minority class, effectively increasing its representation without introducing excessive noise.

The second step in the preprocessing phase is to extract the most informative set of features using RF. RF helps to reduce the model's complexity, which prevents model overfitting. At the end of this stage, the first benchmark dataset involves 284 features (out of 1502 total features); while, the second one involves 291 features (out of 1506 total features). Moreover, to find out what are the most influencing characteristics of this disease, we chose to extract the ten most important features from the main two datasets using RF as shown in Figs. 5, 6 and Table 4.

The third step in the process is to determine the best optimizer to use with each deep learning predictive model. Therefore, all the three proposed deep learning models (CNN, LSTM, and LSTM-BLSTM) were trained on the CKD datasets using the ReLU (Rectified Linear Unit) activation function because it is computationally effective and lessens the likelihood of the gradient vanishing. Each model is trained individually using five optimizers (Adamax, Adam, SGD, Adadelta and Adagrad) to specify the best optimizer

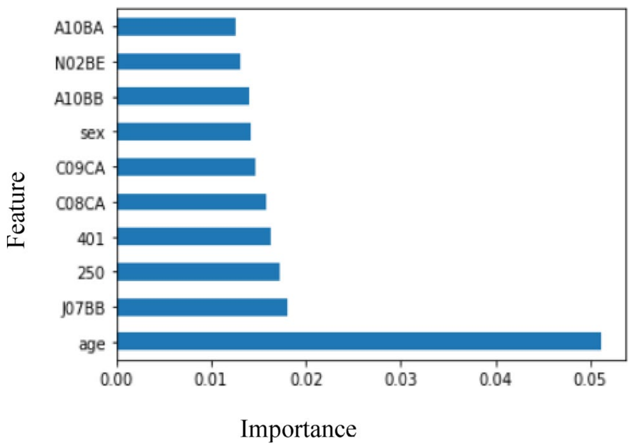


Fig. 5 Most important features for 6 months data

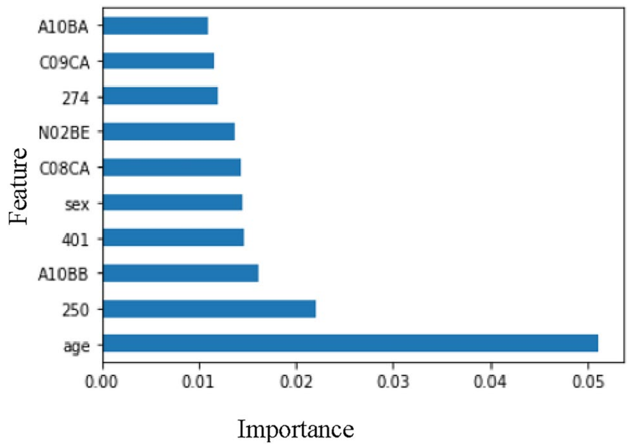


Fig. 6 Most important features for 12 months data

Table 4 Description of the most important features in the CKD dataset produced by random forest

Feature	Description
Age	Patient's age
250	ICD_9 of diabetes
401	ICD_9 of essential hypertension
C08CA	ATC of Dihydropyridine derivatives
C09CA	ATC of angiotensin
Sex	Male or female
A10BB	ATC of Sulfonylureas
N02BE	ATC of Anilids
A10BA	ATC of Biguanides
274	ICD_9 of Gout
J07BB	ATC of INFLUENZA VACCINES

Table 5 Deep learning optimizers' hyperparameter specification

Optimizer	Specification
Adamax	Learning rate = 0.0009, beta1 = 0.9, beta2 = 0.99, epsilon = 1×10^{-8}
Adam	Learning rate = 0.0009, beta1 = 0.9, beta2 = 0.99, epsilon = 1×10^{-8}
SGD	Learning rate = 0.0009, momentum = 0.9, nesterov = False
Adadelta	Learning rate = 0.0009, rho = 0.95, epsilon = 1×10^{-6}
Adagrad	Learning rate = 0.0009, epsilon = 1×10^{-7}

in each deep learning model. We chose five optimizers based on their popularity and effectiveness in deep learning applications for medical data: Adamax and Adam: Adaptive learning rate optimizers known for fast convergence and efficient handling of sparse gradients, common in medical data. SGD (Stochastic Gradient Descent): A well-established optimizer, often used as a baseline for comparison. Adadelta and Adagrad: Adaptive learning rate optimizers focusing on per-parameter learning rates, potentially beneficial for dealing with features with varying scales in medical data. Table 5 provides a summary of the deep learning optimizers' variables. The learning rate is 0.0009 for all optimizers, beta1 = 0.9, beta2 = 0.99, epsilon = $1 \times 10 - 8$ for Adam and Adamax, momentum = 0.9, nesterov = False for SGD. rho = 0.95, epsilon = 1×10^{-6} for Adadelta, while epsilon = 1×10^{-7} for Adagrad. Tables 6 and 7 represent a comparative analysis between each model using each optimizer separately.

To validate our models, we cut off 20% of the training dataset as a validation set. This ensures that the model will generalize well to unseen data. Figures 7, 8, 9, 10, 11 and 12 demonstrate the epoch vs accuracy graph for training and validation for the highest optimizers' accuracy in the first and second datasets, respectively. Each model's input is a CSV file contains the new samples after oversampling processing and feature selection. We load the CSV file first. The input features have been reshaped before applying the model to match the model requirements, using reshape function from NumPy library to perform these operations more efficiently. The datasets are reshaped into 71×4 and 97×3 for six months and twelve months data respectively; while, the output is a binary number that represents the class. The same model structures are used for both benchmark datasets.

The tables indicate that the CNN and LSTM-BLSTM models achieved their best results with Adamax optimizer (95% and 97% accuracy for 6-month and 93% and 96% accuracy for 12-month, respectively); while, the LSTM model achieved its best results with Adam optimizer (96% accuracy for 6-month and 95% accuracy for 12-month) (Figs. 13, 14).

In this paper, the three models, optimized by the best optimizers for model (obtained from this stage), have also been ensembled in the next phase to gain further increase in the performance of deep learning architecture. The ensemble model's structure is shown in Fig. 4. We used the majority voting ensemble (MVE) because it eliminates the drawbacks of other techniques listed earlier and outperforms many other approaches, it is the strategy that is most frequently utilized in the field of

Table 6 Performance evaluation of 6 months data produced by the three proposed individual models using different optimizers

Optimizer	Precision				Sensitivity				F1-score				Accuracy
	CKD	Non-CKD	Macro avg	Weighted avg	CKD	Non-CKD	Macro avg	Weighted avg	CKD	Non-CKD	Macro avg	Weighted avg	
CNN model													
Adamax	**0.93**	**0.96**	**0.95**	**0.95**	**0.96**	**0.93**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**
Adam	0.89	0.93	0.91	0.91	0.93	0.88	0.91	0.91	0.91	0.91	0.91	0.91	0.91
SGD	0.79	0.70	0.75	0.75	0.65	0.83	0.74	0.74	0.71	0.76	0.74	0.74	0.74
Adadelta	0.64	0.82	0.73	0.73	0.89	0.50	0.70	0.69	0.74	0.62	0.68	0.68	0.69
Adagrad	0.71	0.85	0.78	0.78	0.88	0.64	0.76	0.76	0.79	0.73	0.76	0.76	0.76
LSTM model													
Adamax	0.92	0.98	0.95	0.95	0.98	0.92	0.95	0.95	0.95	0.95	0.95	0.95	0.95
Adam	**0.94**	**0.98**	**0.96**	**0.96**	**0.98**	**0.93**	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**
SGD	0.55	0.69	0.62	0.63	0.86	0.32	0.59	0.59	0.67	0.44	0.56	0.56	0.59
Adadelta	0.54	0.70	0.62	0.62	0.89	0.26	0.58	0.57	0.67	0.38	0.53	0.53	0.57
Adagrad	0.68	0.69	0.68	0.68	0.70	0.67	0.68	0.68	0.69	0.68	0.68	0.68	0.68
LSTM-BLSTM model													
Adamax	**0.95**	**0.99**	**0.97**	**0.97**	**0.99**	**0.95**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
Adam	0.95	0.97	0.96	0.96	0.97	0.95	0.96	0.96	0.96	0.96	0.96	0.96	0.96
SGD	0.72	0.60	0.66	0.66	0.45	0.83	0.64	0.64	0.55	0.70	0.62	0.63	0.64
Adadelta	0.60	0.74	0.67	0.67	0.84	0.44	0.64	0.64	0.70	0.55	0.62	0.62	0.64
Adagrad	0.65	0.74	0.70	0.70	0.80	0.58	0.69	0.69	0.72	0.65	0.68	0.68	0.69

The bold, underlined values represent the best optimizer’s performance for each model

Table 7 Performance evaluation of 12 months data by the three proposed individual models using different optimizers

Optimizer	Precision				Sensitivity				F1-score				Accuracy
	CKD	Non-CKD	Macro avg	Weighted avg	CKD	Non-CKD	Macro avg	Weighted avg	CKD	Non-CKD	Macro avg	Weighted avg	
CNN model													
Adamax	**0.91**	**0.95**	**0.93**	**0.93**	**0.95**	**0.91**	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**	**0.93**
Adam	0.90	0.94	0.92	0.92	0.94	0.90	0.92	0.92	0.92	0.92	0.92	0.92	0.92
SGD	0.72	0.81	0.77	0.77	0.84	0.68	0.76	0.76	0.78	0.74	0.76	0.76	0.76
Adadelta	0.64	0.81	0.73	0.73	0.88	0.52	0.70	0.70	0.74	0.63	0.69	0.69	0.70
Adagrad	0.70	0.86	0.78	0.78	0.89	0.62	0.76	0.76	0.78	0.72	0.75	0.75	0.76
LSTM model													
Adamax	0.91	0.96	0.94	0.94	0.96	0.91	0.94	0.94	0.94	0.94	0.94	0.94	0.94
Adam	**0.93**	**0.98**	**0.96**	**0.96**	**0.98**	**0.93**	**0.95**	**0.95**	**0.96**	**0.95**	**0.95**	**0.95**	**0.95**
SGD	0.55	0.69	0.62	0.62	0.85	0.33	0.59	0.59	0.67	0.45	0.56	0.56	0.59
Adadelta	0.53	0.69	0.61	0.61	0.90	0.22	0.56	0.56	0.67	0.34	0.50	0.50	0.56
Adagrad	0.62	0.74	0.68	0.68	0.82	0.52	0.67	0.66	0.71	0.61	0.66	0.66	0.66
LSTM-BLSTM model													
Adamax	**0.94**	**0.98**	**0.96**	**0.96**	**0.99**	**0.94**	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**
Adam	0.94	0.99	0.96	0.96	0.99	0.93	0.96	0.96	0.96	0.96	0.96	0.96	0.96
SGD	0.63	0.72	0.67	0.67	0.77	0.56	0.67	0.66	0.69	0.63	0.66	0.66	0.66
Adadelta	0.58	0.73	0.66	0.66	0.85	0.39	0.62	0.62	0.69	0.51	0.60	0.60	0.62
Adagrad	0.60	0.75	0.67	0.67	0.85	0.44	0.65	0.64	0.70	0.56	0.63	0.63	0.64

The bold, underlined values represent the best optimizer's performance for each model

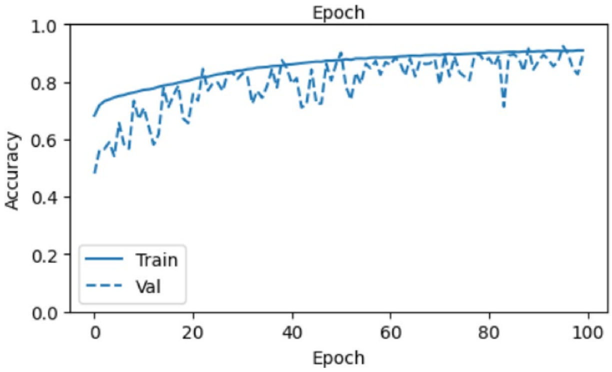


Fig. 7 Epoch vs accuracy graph for CNN_Adamax (6_months data)

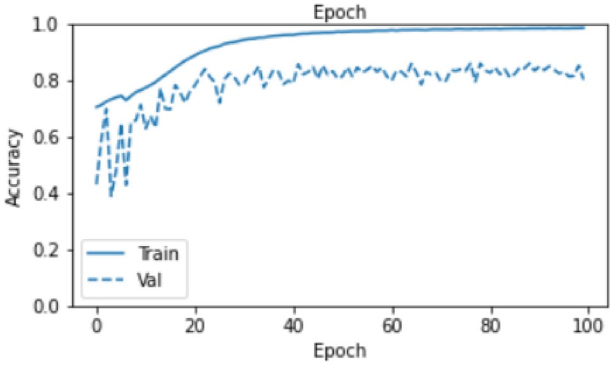


Fig. 8 Epoch vs accuracy graph for LSTM_Adam (6_months data)

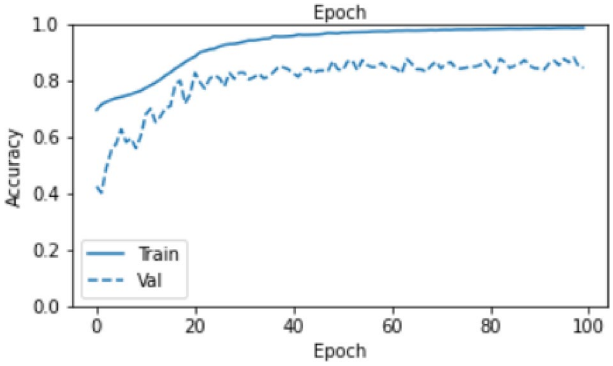


Fig. 9 Epoch vs accuracy graph for LSTM_BLSTM_Adamx (6_months data)

study. Furthermore, we choose it for its Simplicity and interpretability, Robustness to model errors, and Empirical success in CKD prediction. The ensemble-based model's performance is assessed in Table 8.

The ensemble model yielded 98% accuracy for 6-month and 97% accuracy for 12-month, which is better than the three individual models. Additionally, a comparison

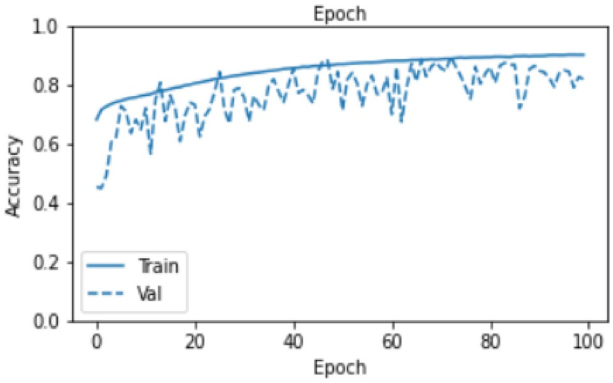


Fig. 10 Epoch vs accuracy graph for CNN_Adamax (12_months data)

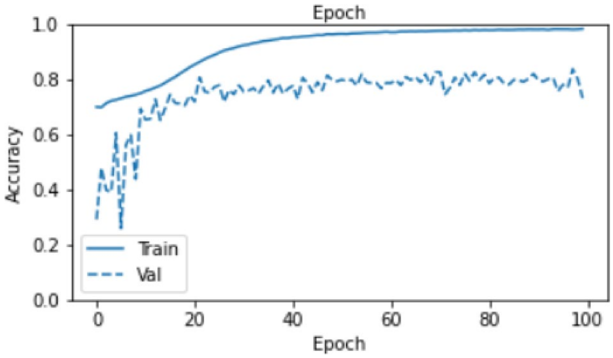


Fig. 11 Epoch vs accuracy graph for LSTM_Adam (12_months data)

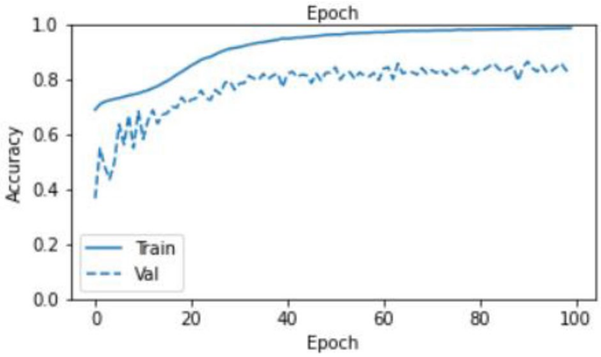


Fig. 12 Epoch vs accuracy graph for LSTM_BLSTM_Adamx (12_months data)

with the outcomes of earlier research is done. Compare our work to the previous study [9] using the same metrics found in their paper. The underline values represent the best accuracy achieved in the compared models. These results show that the ensemble model outperforms the individual models and previous work results in many aspects: sensitivity, precision, specificity, F1- score and accuracy. The proposed model has proven its

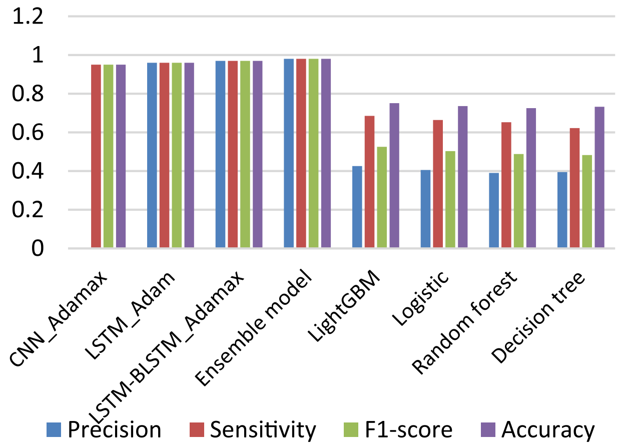


Fig. 13 Performance evaluation 6-month data obtained from the proposed models and the literature

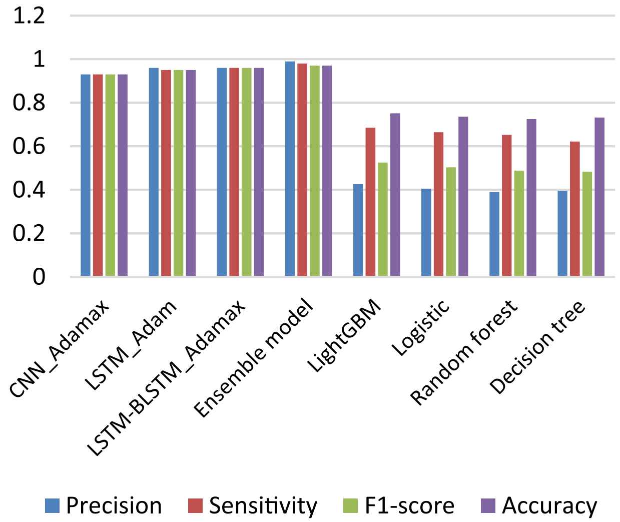


Fig. 14 Performance evaluation 12-month data obtained from the proposed models and the literature

worthiness in all these aspects. On the same datasets for both 6 months and 12 months of data, Figs. 13 and 14 show a graphical depiction of the performance of each proposed model as well as the models in the comparison paper. The figures demonstrate how the model performs better than earlier models.

Results discussion

Our deep learning approaches demonstrated promising performance in CKD prediction. Among individual models, LSTM-BLSTM surpassed others in validation accuracy, $F1$ score, precision, and recall (Table 6, 7). Optimizer choice significantly impacted performance, with Adam and Adamax proving most effective across all architectures.

Table 8 Performance evaluation of 6 and 12-months data for the ensemble model

Model	Precision				Sensitivity				F1-score				Accuracy
	CKD	Non-CKD	Macro avg	Weighted avg	CKD	Non-CKD	Macro avg	Weighted avg	CKD	Non-CKD	Macro avg	Weighted avg	
Ensemble model 6-months data	0.96	0.99	0.98	0.98	0.99	0.96	0.98	0.98	0.98	0.98	0.98	0.98	0.98
Ensemble model 12-months data	0.96	0.99	0.98	0.99	0.96	0.98	0.97	0.98	0.97	0.97	0.97	0.97	0.97

Table 9 Comparison of performance metrics for 6-month data obtained from the proposed models and the literature

Model		Precision	Sensitivity	*F*1-score	Accuracy
Individual classifiers in Ensemble model	CNN_Adamax	0.95	0.95	0.95	0.95
	LSTM_Adam	0.96	0.96	0.96	0.96
	LSTM-BLSTM_Adamax	0.97	0.97	0.97	0.97
Ensemble model		**0.98**	**0.98**	**0.98**	**0.98**
Results for the previous work [9]	LightGBM [9]	0.426	0.685	0.525	0.751
	Logistic [9]	0.405	0.664	0.503	0.736
	Random forest [9]	0.390	0.652	0.488	0.725
	Decision tree [9]	0.395	0.622	0.483	0.732

The bold, underlined values represent the best optimizer's performance for each model

Table 10 Comparison of Performance metrics for 12-month data obtained from the proposed models and the literature

Model		Precision	Sensitivity	*F*1-score	Accuracy
Individual classifiers in Ensemble model	CNN_Adamax	0.93	0.93	0.93	0.93
	LSTM_Adam	0.96	0.95	0.95	0.95
	LSTM-BLSTM_Adamax	0.96	0.96	0.96	0.96
Ensemble model		**0.99**	**0.98**	**0.97**	**0.97**
Results for the previous work [9]	LightGBM [9]	0.426	0.685	0.525	0.751
	Logistic [9]	0.405	0.664	0.503	0.736
	Random forest [9]	0.390	0.652	0.488	0.725
	Decision tree [9]	0.395	0.622	0.483	0.732

The bold, underlined values represent the best optimizer's performance for each model

While Adam outperformed Adamax in LSTM, Adamax yielded the highest CNN accuracy for both datasets. In LSTM-BLSTM, Adamax excelled for six-month data, while Adam matched its performance for twelve-month data. Notably, switching to other optimizers (SGD, Adagrad, Adadelta) led to performance decline, with Adadelta exhibiting the lowest accuracy.

LSTM-BLSTM's superior performance likely stems from its suitability for modeling sequential data like CKD progression. This bidirectional recurrent neural network architecture captures both forward and backward dependencies within features, crucial for understanding long-term effects of comorbidities and medications. Its gating mechanism further enhances ability to learn these long-range dependencies.

To further boost performance, we developed an ensemble model combining the best-performing individual models (CNN-Adamax, LSTM-Adam, and LSTM-BLSTM-Adamax). This ensemble achieved significantly higher accuracy (98% and 97% for 6 and 12 months, respectively) than all other models, albeit with increased computational cost due to higher complexity (Table 8, 9, 10). Importantly, our models outperformed those of a prior study using traditional machine learning techniques on the same datasets (Figs. 13, 14). This improved performance suggests that deep learning models extract more complex feature correlations, leading to more accurate CKD prediction. Additionally, as seen in Figs. 13, 14 and Table 9, 10, the proposed models outperform those of a prior study that employed the same datasets with traditional machine learning techniques. The reason is that the deep learning model revealed more correlation between the features than those revealed by the previous work [9] which led to better prediction of CKD as indicated by the performance measures.

Moreover, to guide the experts on what features to concentrate on when predicting the possible occurrence of CKD, Figs. 5 and 6 highlight the 10 important features of these datasets (as detected by RF algorithm). Age and gender are crucial indicators for prediction. Furthermore, important aspects include dihydropyridine derivatives, and angiotensin, which are used to treat hypertension. Sulfonylureas, treat type 2 diabetes mellitus, and biguanides, an oral medication used to manage mild to moderately severe noninsulin-dependent diabetic mellitus (Type II), in obese or overweight individuals who are often older than 40 years old, have notable characteristics. In terms of feature importance, anilids used to alleviate aches and pains, are in the last four spots. When it comes to diseases, diabetes, gout, and hypertension are regarded as the most common symptoms.

It is known that there are some risk indicators that doctors can use to predict the onset of the disease. There are many features besides the most important ones; whereas, they just represent risk factors for CKD. The model has been trained in extensive and intricate medical datasets, which could make it difficult for doctors to detect the risk factors in the absence of the most important features or analyze all these features manually. If the risk factor is identified by doctors, they will not be able to determine the exact time of the disease onset, while our model can predict. That will contribute significantly to intervening at the right time and saving many patients from this disease. Finally, we have demonstrated through practical experiments the direct impact of these risk factors on the incidence of kidney diseases.

One of the limitations of this research is that the patient's health data must be recorded in the system for two consecutive years to gather the necessary data for decision-making, including the diseases they have contracted and the medications they have taken throughout those years. Undoubtedly, this is not an easy task. This study does not rely on medical analyses, which differ from previous studies. However, the need for such a study based on analyses is essential.

Moreover, the number of features was excessively large, which necessitated the use of a well-known feature selection method. Introducing such a massive amount of data for model training would consume an extremely long time without any actual need for it, given the insignificance of these additional features.

Despite decreasing the number of features, the time required for the proposed model has been a significant obstacle due to the execution of the three models separately. However, on the other hand, accuracy has reached its highest rate.

Conclusion and future work

Recently, machine learning research has shown that combining the output of several individual classifiers can reduce generalization errors and yield better performance in many applications than individual deep learning classifiers. This study focused on predicting CKD before it occurs over a period of time using the Ensemble model. In addition, a comparative evaluation of deep learning optimizers is presented for each individual model to induce the most powerful optimizer for the CKD dataset. In this study, the unbalanced data are handled using SMOTE approach. Random Forest feature selection technique is applied to reduce the number of features. After that, a comprehensive comparison of various deep learning architectures has been conducted. Furthermore, several deep learning optimization methods (Adamax, Adam, SGD, Adadelta, and Adagrad) are used to evaluate how well these models performed.

The Ensemble model is implemented by combining the top three models and optimizers. It was discovered that in terms of validation accuracy, $F1$ score, precision, and recall, the hybrid of LSTM and BLSTM using Adamax optimizer outperformed other optimizers. The Ensemble model, which combines the CNN_Adamax, LSTM_Adam, and LSTM_BLSTM_Adamax models, outperformed all other models by a wide margin, scoring an accuracy score of 98% for the six months dataset and 97% for the twelve months dataset. The research also showed that age, gender, and chronic diseases such as diabetes, high blood pressure, and gout are among the most important causes of chronic kidney disease, in addition to some drugs that treat diabetes, high blood pressure, and relieve pain.

According to the findings of the studies, our ensemble model predicted the incidence of CDK with 98% and 97% accuracy for 6-month and 12-month instances, respectively. This demonstrates the efficiency of the suggested approach in warning medical providers of the probability of a patient having the condition before it occurs (6–12 months). Such information can certainly save lives and lower the death rate of such patients, as well as lower the cost of medical care delivered to those individuals.

As a future step, we plan to test the robustness of our developed models against various datasets based on patient laboratory data collected from local hospitals, medical analysis laboratories, and polyclinics. This can be achieved by collaborating with healthcare providers to assess the feasibility and potential impact of implementing prediction models on medical datasets.

Acknowledgements
Not applicable.

Author contributions
AS contributed to conceptualization of this study, resources, writing and editing the manuscript. DS contributed to conceptualization of this study, formal analysis, methodology, software, validation, and writing—original draft preparation. NME contributed to conceptualization of this study, investigation, methodology, resources, and validation.

Funding
Not applicable.

Availability of data and materials
The datasets analyzed during the current study are available in a public repository [13].

Declarations

Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Received: 2 January 2024 Accepted: 12 March 2024
Published online: 03 April 2024

References

1. Jongbo OA et al (2020) Development of an ensemble approach to chronic kidney disease diagnosis. Sci Afr 8:e00456
2. Elhoseny M, Shankar K, Uthayakumar J (2019) Intelligent diagnostic prediction and classification system for chronic kidney disease. Sci Rep 9(1):9583
3. Kriplani H, Patel B, Roy S (2019) Prediction of chronic kidney diseases using deep artificial neural network technique. In: Computer aided intervention and diagnostics in clinical and medical images. Springer, Berlin
4. Almansour NA et al (2019) Neural network and support vector machine for the prediction of chronic kidney disease: a comparative study. Comput Biol Med 109:101–111
5. Kriplani H, Patel B, Roy S (2019) Prediction of chronic kidney diseases using deep artificial neural network technique. In: Computer aided intervention and diagnostics in clinical and medical images. Springer, Berlin, pp 179-187
6. Ekanayake IU, Herath D (2020) Chronic kidney disease prediction using machine learning methods. In: 2020 Moratuwa engineering research conference (MERCon). IEEE
7. Gudeti B et al (2020) A novel approach to predict chronic kidney disease using machine learning algorithms. In: 2020 4th international conference on electronics, communication and aerospace technology (ICECA). IEEE
8. Senan EM et al (2021) Diagnosis of chronic kidney disease using effective classification algorithms and recursive feature elimination techniques. J Healthc Eng 2021:1004767
9. Krishnamurthy S et al (2021) Machine learning prediction models for chronic kidney disease using national health insurance claim data in Taiwan. In: Healthcare. MDPI
10. *CKD-dataset-2*. Available from: https://osf.io/wbv4p/?show=revision.
11. Qin J et al (2019) A machine learning methodology for diagnosing chronic kidney disease. IEEE Access 8:20991–21002
12. Chittora P et al (2021) Prediction of chronic kidney disease—a machine learning perspective. IEEE Access 9:17312–17334
13. Ma F et al (2020) Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network. Future Gener Comput Syst 111:17–26
14. Alsuhibany SA et al (2021) Ensemble of deep learning based clinical decision support system for chronic kidney disease diagnosis in medical Internet of Things environment. Comput Intell Neurosci 2021:4931450
15. dataset_1, available from: https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease.
16. Lin L-Y et al (2018) Data resource profile: the national health insurance research database (NHIRD). Epidemiol Health 40:e2018062
17. Singh V, Asari VK, Rajasekaran RJD (2022) A deep neural network for early detection and prediction of chronic kidney disease. Diagnostics 12(1):116
18. Sawhney R et al (2023) A comparative assessment of artificial intelligence models used for early prediction and evaluation of chronic kidney disease. Decis Anal J 6:100169
19. Sagi O, Rokach L (2018) Ensemble learning: a survey. Wiley Interdiscip Rev: Data Min Knowl Discov 8(4):e1249
20. Zhang P et al (2018) Learning spatial–spectral–temporal EEG features with recurrent 3D convolutional neural networks for cross-task mental workload assessment. IEEE Trans Neural Syst Rehabil Eng 27(1):31–42
21. Ganaie M, Hu M (2021) Ensemble deep learning: a review. arXiv:2104.02395.
22. Ju C, Bibaut A, van der Laan M (2018) The relative performance of ensemble methods with deep convolutional neural networks for image classification. J Appl Stat 45(15):2800–2818
23. Raza K (2019) Improving the prediction accuracy of heart disease with ensemble learning and majority voting rule. In: U-healthcare monitoring systems. Elsevier, Amsterdam, pp 179–196
24. Atallah R, Al-Mousa A (2019) Heart disease detection using machine learning majority voting ensemble method. In: 2019 2nd international conference on new trends in computing sciences (ICTCS). IEEE
25. Heart disease dataset.
26. Yadav SS, Kadam VJ, Jadhav SM (2019) Comparative analysis of ensemble classifier and single base classifier in medical disease diagnosis. In: International conference on communication and intelligent systems. Springer
27. Zhou T et al (2021) The ensemble deep learning model for novel COVID-19 on CT images. Appl Soft Comput 98:106885
28. Chandra TB et al (2021) Coronavirus disease (COVID-19) detection in chest X-ray images using majority voting based classifier ensemble. Expert Syst Appl 165:113909
29. Monteral J (2020) COVID-Chestxray database
30. Aurna NF et al (2022) A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models. Comput Biol Med 146:105539
31. Neloy M et al (2022) A weighted average ensemble technique to predict heart disease. In: Proceedings of the third international conference on trends in computational and cognitive engineering. Springer, Berlin
32. Hireš M et al (2022) Convolutional neural network ensemble for Parkinson's disease detection from voice recordings. Comput Biol Med 141:105021
33. Chawla NV et al (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
34. Arora N, Kaur PD (2020) A Bolasso based consistent feature selection enabled random forest classification algorithm: an application to credit risk assessment. Appl Soft Comput 86:105936
35. Mursalin M et al (2017) Automated epileptic seizure detection using improved correlation-based feature selection with random forest classifier. Neurocomputing 241:204–214
36. Si Z, Wen S, Dong B (2019) NOMA codebook optimization by batch gradient descent. IEEE Access 7:117274–117281

37. Guo L et al (2020) Application of stochastic gradient descent technique for method of moments. In: 2020 IEEE international conference on computational electromagnetics (ICCEM). IEEE
38. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12(7):2121–2159
39. Zeiler MD (2012) Adadelta: an adaptive learning rate method. arXiv:1212.5701.
40. Yaqub M et al (2020) State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images. Brain Sci 10(7):427
41. Vani S, Rao TM (2019) An experimental approach towards the performance assessment of various optimizers on convolutional neural network. In: 2019 3rd international conference on trends in electronics and informatics (ICOEI). IEEE
42. Yi D, Ahn J, Ji S (2020) An effective optimization method for machine learning based on ADAM. Appl Sci 10(3):1073
43. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
44. Kaushik P et al (2018) EEG-based age and gender prediction using deep BLSTM-LSTM network model. IEEE Sens J 19(7):2634–2641
45. Wang D et al (2023) Forecasting power demand in China with a CNN-LSTM model including multimodal information. Energy 263:126012
46. Xinyang F et al (2023) Attention-BLSTM based quality prediction for complex products. Comput Integr Manuf Syst 29(12):3974
47. Kaushik P, Tripathi I, Roy PP (2023) Motor activity recognition using eeg data and ensemble of stacked BLSTM-LSTM network and transformer model. In: ICASSP 2023–2023 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
48. Rahman M, Watanobe Y, Nakamura K (2021) A bidirectional LSTM language model for code evaluation and repair. Symmetry 13(2):247

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.