scientific reports

OPEN

A population based optimization of convolutional neural networks for chronic kidney disease prediction

M. Priyadharshini[1], V. Murugesh[2], G. V. Samkumar[2], Subrata Chowdhury[3], Amrutanshu Panigrahi[4], Abhilash Pati[4✉] & Bibhuprasad Sahu[5]

Chronic kidney disease (CKD) is a global public health concern, and the timely detection of the disease is priceless. Most of the classical machine learning models have the major drawbacks of being unsophisticated, non-robust, and non-accurate. This research work is therefore seeking to introduce OptiNet-CKD, a paradigm based on a DNN that has been integrated with a developed population optimization algorithm (POA) for CKD prediction optimization. POA is unlike gradient-based optimization methods in that it uses an initialized population of networks and perturbs their weight values to provide a broader exploration of the solution space. The model is more robust and less likely to overfit, and the predictions are likely to be more accurate since this approach helps to avoid the local minima problem suffered by gradient-based optimizers. To preprocess it for DNN learning, a CKD dataset with 400 records containing numerical and categorical features was imputed for missing data and scaled for its features. The model was evaluated using performance metrics such as accuracy, precision, recall, F1-score, and ROC AUC. OptiNet-CKD achieved 100% accuracy, 1.0 precision, 1.0 recall, 1.0 F1-score, and 1.0 ROC-AUC from traditional models (logistic regression, decision trees) and even fundamental deep neural networks. Results show that OptiNet-CKD is a reliable and robust prediction method for CKD, with more substantial generalization and performance than the existing methods. A combination of DNN and POA constitutes a promising approach for medical data analysis, especially for the diagnosis of CKD. POA expands the solution space, helping to expunge the model from falling into local minima and giving the model increased power in generalizing complicated medical data. Based on the simplicity of the algorithm, together with the structured formula and the extractions made in the preprocessing step, this framework can be extended to other medical conditions with similar data complexities, providing a potent tool for improving diagnostic accuracy in healthcare.

Chronic kidney disease (CKD) is widely recognized as a global public health problem that leaves millions of people at high risk and has high morbidity and mortality rates[1]. It is vital to diagnose CKD as soon as possible because appropriate measures should and can be taken to delay its progression and ameliorate results[2]. However, such prediction is not easy because medical data are usually complex and heterogeneous, which makes them even harder to analyze. Conventional methods for modeling, such as logistic regression, support vector machines, and decision trees, have also been employed in the prediction of CKD[3]. Even though these models can give promising results, it is reported that they have problems concerning robustness, accuracy, and extension[4]. The main causes of these issues are their simple architectures that cannot resolve the intricacy of the medical data, the unbalanced nature of the medical data, which features missing data, the combination of numerical and categorical data, and nonlinear patterns of data[5]. Such limitations suggest that higher-order algorithms with the capability of identifying intricate patterns should be used to generate accurate predictions.

[1]Department of Computer Science and Engineering, Faculty of Science and Technology (IcfaiTech), The ICFAI Foundation for Higher Education, Hyderabad, Telangana 501203, India. [2]Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andra Pradesh, India. [3]Department of Computer Science and Engineering, Sreenivasa Institute of Technology and Management Studies, Chittoor, Andra Pradesh, India. [4]Department of CSE, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India. [5]Department of Information Technology, Vardhaman College of Engineering (Autonomous), Hyderabad, Telangana, India. ✉email: abhilashpati@soa.ac.in

To handle these challenges, this study presents an OptiNet-CKD methodology that integrates a deep neural network with a customized population optimization algorithm for improved prediction of CKD[6]. The advantage of POA is that it starts by having a diverse set of networks whose weights and biases are then stochastically updated as part of efforts to have a more efficient and broader exploration of the solution space[7]. This method helps us avoid the problem of local minima, which is usual for gradient-based methods, which are more easily prone to being entrapped into a suboptimal solution in the case of complex, high-dimensional data such as CKD.

The heterogeneous and imbalanced nature of data in CKD is one of the key challenges that can be evidenced by missing values in data, non-linear feature relationships, and complicated interactions with the variables. Therefore, there are various factors to overcome when applying such traditional machine learning models, which typically fail to capture complex patterns. This task is particularly suited for POA because, in POA, the population of networks it maintains is diverse, so the POA can explore the high-dimensional, noisy space without overfitting the complex interaction between the data.

Two motivations for this work are the capability of deep learning to learn high order of data input dependencies and the possibility of using population-based optimization methods to non linearly improve the model's training by optimizing the search for a better solution in the solution space. Having the diversity maintained in a population of networks, POA can significantly improve the robustness and the generality of DNN, which brings better performance on heterogeneous yet highly challenging real datasets like CKD.

Specifically, the major research questions in this study are as follows: To design an improved DNN-POA model for CKD risk assessment and forecasting; To assess the performance of the proposed approach in terms of key experimentation metrics, including accuracy, precision, recall, F1-score, and the area under the receiving operating characteristic curve (ROC-AUC). The study is going to compare OptiNet-CKD with typical machine learning algorithms and ordinary optimization techniques to illustrate its advantage and expand its application in the medical field. Thus, this study adds to the literature by proposing a population-based strategy for optimizing the training of deep neural networks using medical data, especially those having complex data features like CKD.

The structure of the paper is organized as follows: In "Literature survey" section, relevant contents of previous studies on CKD prediction models and optimization techniques are reviewed, and the main findings, as well as the research gaps filled by this study, are presented. In "Methodology" section explains the method, describing the DNN architecture, the POA construction, and other methods for data preprocessing when working with the CKD dataset. Finally, the experimental results in terms of models and comparisons with TML classifiers are presented in "Proposed OptiNet-CKD system" section, and the robustness and efficiency of the proposed work are evaluated. Last but not least, in "Experimental results and discussion" section, the conclusion, presents the benefits of the proposed OptiNet-CKD model, describes challenges in this work, and outlines the potential development of this model for other diseases and new features for enhancing predictability.

Literature survey

The development of methodologies to predict CKD and to diagnose it at an early stage are also important and attractive goals because of the high mortality and morbidity of this disease. At first, other prediction algorithms, including logistic regression and support vector machines (SVMs), became more famous for CKD prediction. Nonetheless, Poorani and Karuppasamy[8] showed such models, although they gave minor levels of predictive accuracy, cannot contain a multitude of pension holders inside medical datasets[8]. Singh et al. also demonstrated that decision trees and random forests enhanced feature interactions but, at the same time, are sensitive to hyperparameters, which affects their stability in applied fields[9]. These traditional models were not designed to effectively handle heterogeneous data and also cannot handle imbalanced datasets, which are generally prevalent in CKD prediction tasks. They upgrade the medical data models from traditional machine learning to deep learning and over to the mixed models because medical data cannot be processed by traditional models such as logistic regression trees and decision trees. The studies by Singh et al.[9] have shown that deep learning models are capable of identifying non-linear interactions in large datasets. Nevertheless, their performance is enhanced substantially when used together with optimization algorithms.

Traditional models, including logistic regression, decision trees, and support vector machines (SVMs), have been applied to predict CKD. Still, they suffer from imbalanced data problems and nonlinear relationships among the features. These limitations further motivate better and more complex algorithms to handle these complexities effectively.

Researchers later switched to a more robust model that could model non-linear relationships and extract complex features from massive datasets like deep learning (DL), replacing traditional model limitations. Ma et al.[10] demonstrated the applicability of CNN in improving the diagnosis of CKD, especially using medical images. They combined particle swarm optimization (PSO) and genetic algorithms (GA) with deep learning models to add a feature selection part to their actual feature selection methods to minimize the distortion and overfitting effect. RNNs demonstrated their capability of modeling time series data and tracking the different continuous changes in patient conditions in the past, as was done by Zao et al.[11]. Researchers began forming deep learning models with optimization algorithms to combat these weaknesses. For studies where GA and PSO are used to achieve feature selection and model generalization to improve stability and accuracy over several datasets, Bhaskar et al.[12] performed. Nevertheless, the usefulness of these optimization techniques may be limited since they cannot explore the solution space fully and escape the local minima, providing suboptimal solutions in medical applications.

In recent days, traditional optimization techniques have become less valuable, and the population optimization algorithm (POA) is given as a powerful substitute. The disadvantage of POA is that it will explore the solution space less thoroughly. However, it uses a population-based approach to explore solution spaces and marginalized local optima better, which in turn leads to better model generalization. For medical applications, datasets are very imbalanced, feature pairs have non-linear interactions, and POA works well in those data sets.

In recent studies, attempts have been made to enhance DNN with a population-based optimization algorithm, POA, to predict the onset of CKD. Thus, as mentioned by Bhaskar et al.[12], POA is not only helpful in speeding up the exploration of model parameters but also improves the generalization and stability of the model across different datasets. The local optima problem is addressed; hence, overfitting, typical for medical data analysis, may be avoided. Therefore, POA is a suitable optimization technique for CKD prediction, where the medical dataset is usually highly skewed and has a class imbalance.

However, as Hassan et al.[13] pointed out, ours is one of the recent studies comparing different machine learning models to predict CKD, which did not focus on combining POA and DNN. Although Hassan et al.[13] comprehensively evaluated the machine learning models in the context of CKD, their method did not consider the optimization methods necessary to overcome challenges such as overfitting and local minima present in medical datasets.

OptiNet-CKD deploys POA to overcome those challenges in optimizing the deep learning model and increasing the prediction accuracy. This method ensures the robustness and the ability to handle imbalances, missing values, and non-linear relationships well with much better generalizability when dealing with complex medical datasets.

The results also showed that combining POA and DNNs greatly enhances model accuracy. For CKD prediction, Singh et al.[9] and Zhao et al.[14] show that deep learning techniques can be combined with optimization strategies to create more accurate and robust models. Our approach is based on integrating POA into the DNN framework, which mutations previous machine learning models.

POA is integrated into the deep learning models via deep learning and optimization as a paradigm, which is a step beyond traditional machine learning. Such a shift suggests that there is a bigger picture in healthcare, with a greater and greater need for aptitude in dealing with large, complicated data sets and more reliable and generalizable approaches. OptiNet-CKD can utilize the POA to find optimal model parameters to improve generalization and provide a Predictive Power tool for CKD diagnosis along with potential other uses in medical conditions that must rely on such predictive power.

Methodology

In this proposed research paper, the OptiNet-CKD framework, which is the combination of the deep neural network (DNN) with the customization of the population optimization algorithm (POA), is proposed as a solution for the prediction of chronic kidney disease (CKD). In contrast to the traditional gradient-based methods, POA was specially chosen to improve the training process by helping to explore more of the solution space. Although gradient-based optimizers are often used in DNN training, they have drawbacks, especially in high-dimensional and complex datasets such as the CKD prediction task. However, they usually converge prematurely to local minima, which further results in suboptimal solutions in case of nonlinear and imbalanced data. On the other hand, POA operates by keeping a population of initialized networks with various weight configurations, and it perturbs those weights to explore more of the solution space.

POA differs from gradient-based methods, which operate only on a single point in the solution space, being the current model weights, insofar as the population of networks that it grows allows for multiple weight configurations to be explored simultaneously. It helps to subvert the problem of falling into local minima, an essential issue of training deep learning models on complex medical datasets. Since POA does not rely on the gradients but explores a more significant part of the search space, the network population is diverse, and thus, POA prevents getting stuck in suboptimal solutions. Therefore, POA provides more robust and generalized solutions and increases the accuracy of predicting CKD, even when noisy, missing, or unbalanced data exists.

OptiNet-CKD's DNN architecture consists of an input layer, followed by two hidden layers with 64 and 32 neurons, which apply the ReLU activation function. For binary classification, the output layer is a sigmoid-activated layer. Like the DNN, the POA optimizes the weights and biases across the network population by extending weight updates and enhancing network generalization. This approach reduces overfitting and enhances model robustness and accuracy, which is more suitable for complex and heterogeneous datasets, such as CKD.

Dataset

This work utilizes a dataset to predict CKD, which is embodied in patients' clinical records, constituting numerical and categorical features mixed to get better apprehension about patient health outcomes pertinent to the process of CKD diagnosis. This dataset comprises 400 patient records from public medical data, including 24 features, so it is relatively varied and sized in a way representative of the realistic challenges of such problems. These records describe a single patient and contain several clinical measurements of kidney function as well as general health.

Data set description
This in-home dataset is designed to help you practice CKD prediction skills using a variety of clinical features related to CKD, validated by literature. Figure 1, the Correlation Heatmap of the kidney disease dataset, provides a simple explanation of the first five records, giving a sense of what is included.

Feature descriptions
This includes the standardized numerical and binary categorical features, which reflect the number of patients' health conditions, as depicted in Fig. 2. In the current analysis, the major independent variables included albumin level (al), sugar level (su), age, and blood pressure (bp). Other independent variables were numerical variables comprising packed cell volume (PCV), white blood cell count (WC), and red blood cell count (RC). It is a linear-type variable, and therefore, it is more suitable for numerical analysis than categorical data.
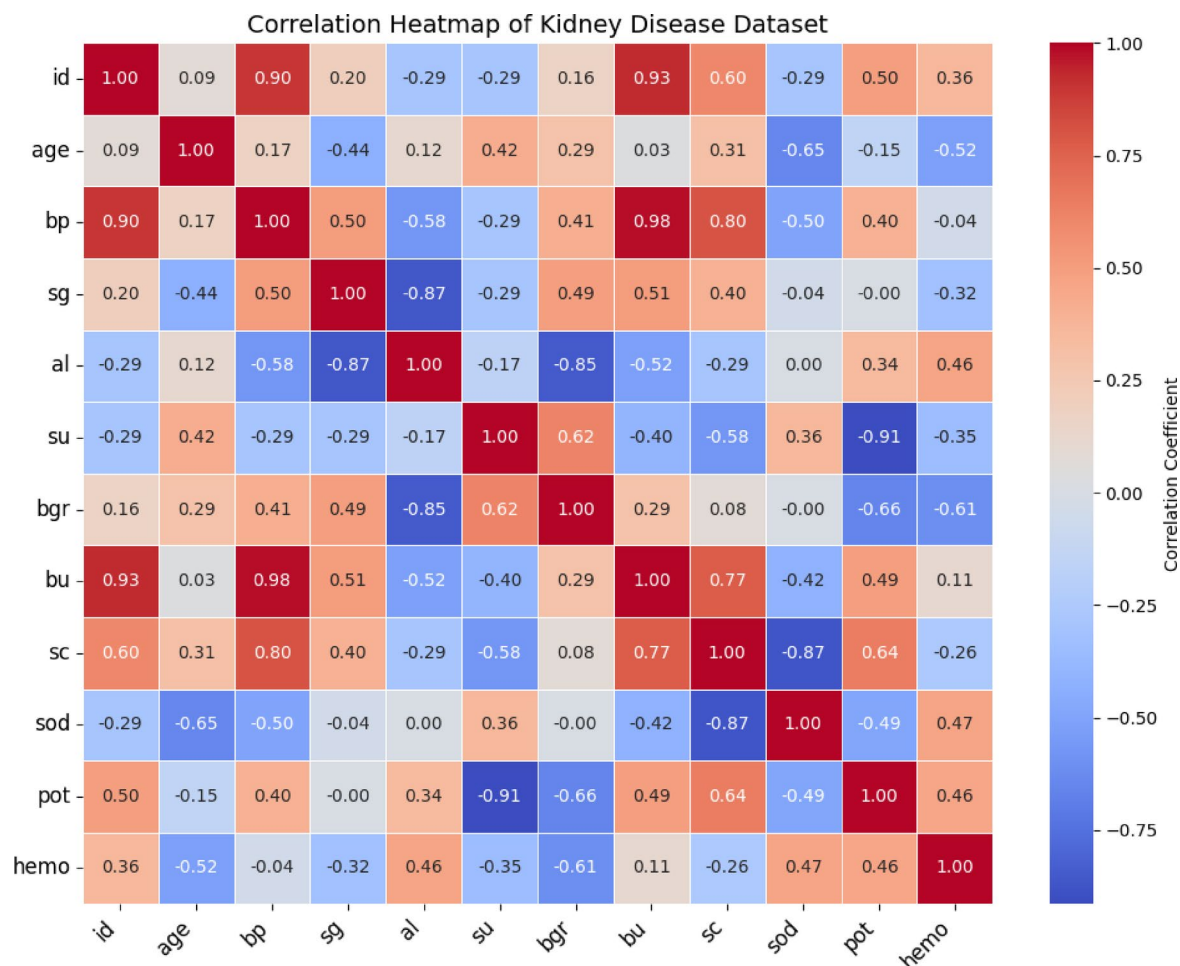
Fig. 1. Correlation heatmap of kidney disease dataset.

Background Most of the variables in the dataset are binary categorical variables ranging between 0 and 1. They include rbc, which measures the presence of red blood cells, and pc, which measures the presence of pus cells. Another variable, PCC, measures the presence of pus cell clumps and the presence of bacteria, while Hyp-denotes hypertension and Dmb-denotes diabetes. Other variables include binary responses of coronary artery disease (cad), appetite good/poor, pedal edema (pe), and diabetes mellitus (dm). All of these features are helpful in the staging of CKD. The dependent variable is the dichotomized CKD status, with the patient having CKD labeled as one and the patient without CKD labeled as 0. Consequently, this dataset makes it possible to effectively model and explain using different health indexes revolving around the prediction of CKD.

The selected features are both numerical and categorical, and for model training, each feature is either standardized or encoded. The dependent variable, referred to as "classification", is a binary in which 1 depicts a patient with CKD and 0 is a patient without CKD.

Figure 3 shows the target variable distribution of our dataset. This dataset contains 248 instances of CKD, and non-CKD is less than half, i.e. 152, which makes a balanced data set for both classes.

This data preprocessing step makes the OptiNet-CKD model a much cleaner dataset and ready to achieve bigger prediction scores because we present only structured information[15]. The proper shaping of the dataset is necessary for a model to learn well and give better results in prediction, thus helping to prove the OptiNet-CKD system's success as a CKD predictor[16].

The dataset features are divided into numerical and categorical. Age; blood pressure (BP); specific gravity of urine; albumin levels in the urine, sugar levels in the urine, random blood glucose level, BU B serum creatinine level X 1 integer: sodium APTT K ad AT; Hemoglobin PCV WBC OGTT R BC There are thirteen factors including the red blood cell, pus cells; bacteria in urine and three if any one of them is recorded as present the level will be 1 (yes otherwise no), with most changes occurring at C, D b&F; H, I, K, O set. The dependent variable or target (class) column tells us whether the patient has CKD (1 if it's a case, 0 otherwise).

For training and evaluation, the dataset underwent several preprocessing steps to make it valid. This was followed by the handling of missing values, as many features had gaps in their datasets[17]. Missing values in all numerical attributes were filled with the median value and outliers. Approximate mode was used for imputation of the most frequent value in case a data point is missingrier_transformer_fit().
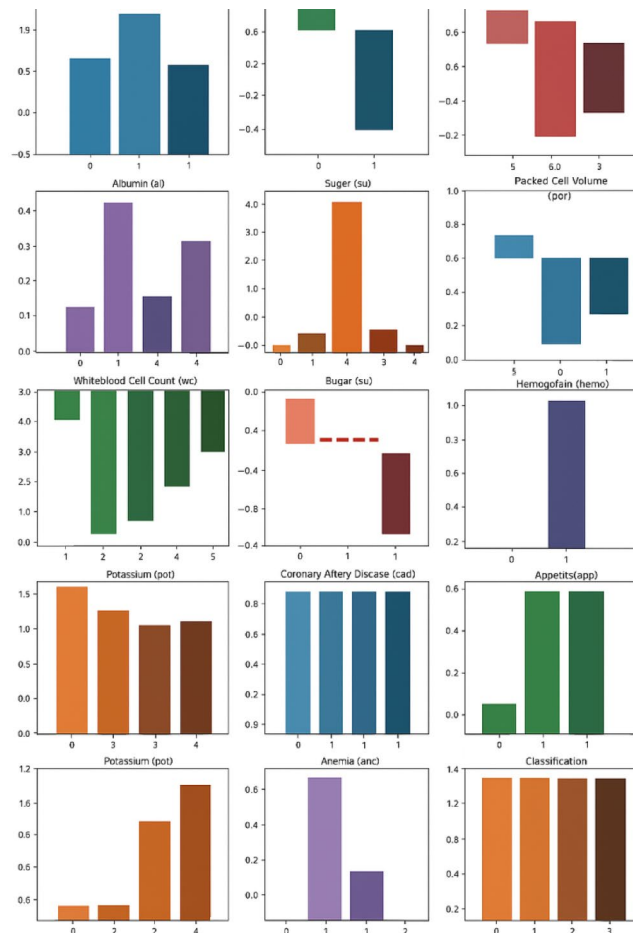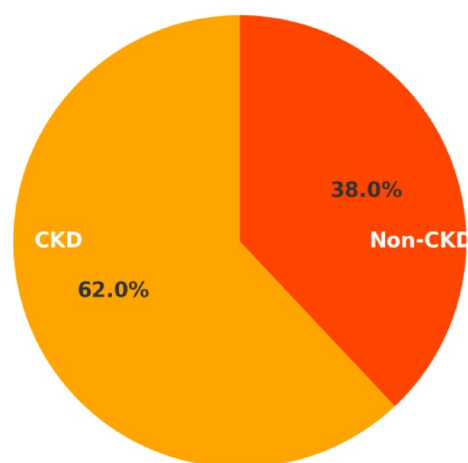
Fig. 2. Feature descriptions.

Fig. 3. Distribution of the target variable.

Next, we normalized the numerical features with StandardScaler to give them a mean of 0 and std_dev of 1. This step is essential to a DNN learning effectively from the data since it puts all numerical attributes on a similar scale[18]. LabelEncoder converts categorical features to numerical values, which can be made or processed by the DNN. Furthermore, the target variable was encoded in which one denoted CKD presence and 0 CKD absence to simplify the problem as a binary classification one[19].

The training set of the data is split by 70%, and the testing part is to validate how well the model has performed. Generally, 80% is for training, and the other 20% is used for testing. Data will be split into test and training sets so that neuronal networks can be evaluated on unseen samples. Thus, it will estimate the network generalization capabilities[6]. These stats will give ideas about the characteristics of this dataset and assist in recognizing possible outliners by finding mean, standard deviation, minimum values, and maximum values[20].

The CKD dataset was preprocessed as explained above, and after that, it was ready for training and evaluation of our OptiNet-CKD model. This pipeline formats the data into a format suitable for training, which our model can learn from, and that is what is left once we are comfortable with it to create predictions about CKD[21].

Data preprocessing

Such data preprocessing is an essential step in the OptiNet-CKD system to have a valid, clean dataset suitable for model training[22,23]. This step includes dealing with missing values, standardizing numerical features, and encoding categorical variables. This means that each of these steps is crucial to enhancing data quality for performance improvement in building the predictive model[24,25].

First step in data preprocessing—handling missing values integer variables (if they have any missing values then they will also be included, and we use the median instead of mean for these since this is more resistant to outliers keeping our tendency, central) This means that we need to choose one specific value (the most frequent) for each categorical feature in order to replace missing data with it, i.e., the technique will maintain only the dominant category within a whole dataset[26]. By using this way to deal with null values as input variables, we make sure that the entire dataset stays usable and safe for further analysis[27].

Let X be the dataset with features X_i for $i = 1, 2, \ldots, n$.

1. Numerical features:

   - For each numerical feature X_i:

     - Identify missing values in X_i:
     - Compute the median of X_i using Eq. (1).

$$median\left(X_i\right) = \begin{cases} \dfrac{X_{\left(\frac{n}{2}\right)} + X_{\left(\frac{n}{2}+1\right)}}{2}, & if\ n\ is\ even \\ X_{\left(\frac{n+1}{2}\right)}, & if\ n\ is\ odd \end{cases} \tag{1}$$

     where $X_{(j)}$ represents the jth order statistic (i.e., the jth smallest value in X_i).

   - Replace missing values in X_i with median(X_i) using Eq. (2).

$$X_j = \begin{cases} X_j, & if\ X_j\ is\ not\ missing \\ mode\left(X_j\right), & if\ X_j\ is\ missing \end{cases} \tag{2}$$

2. Categorical features:

   - For each categorical feature X_j:

     - Identify missing values in X_j:
     - Compute the mode of X_j using Eq. (3):

$$mode\left(X_j\right) = argmax_k \sum_{i=1}^{n} 1\left(X_{ji} = k\right) \tag{3}$$

     where 1 is the indicator function, and k is a category in X_j.

   - Replace missing values in X_j with $mode\left(X_j\right)$ using Eq. (4):

$$X_j = \begin{cases} X_j, & if\ X_j\ is\ not\ missing \\ mode\left(X_j\right), & if\ X_j\ is\ missing \end{cases} \tag{4}$$

1: **Input:** Dataset X with numerical features X_{num} and categorical features X_{cat}

2: **Output:** Dataset $X_{missing}$ with imputed missing values

3: procedure HANDLEMISSINGVALUES(X)

4: Initialize dataset $X_{missing}$

5: for each numerical feature $X_{num} \in X$, do

6:   if X_{num} contains missing values, then

7:     Compute the median of X_{num}, median(X_{num})

8:     Replace missing values in X_{num} with median(X_{num})

9:   Add X_{num} to $X_{missing}$

10: for each categorical feature $X_{cat} \in X$, do

11:   if X_{cat} contains missing values, then

12:     Compute the mode of X_{cat}, mode(X_{cat})

13:     Replace missing values in X_{cat} with mode(X_{cat})

14:   Add X_{cat} to $X_{missing}$

15: return dataset $X_{missing}$

Algorithm 1. Handling missing values

Numerical values are standardized using StandardScaler, which scales by setting the mean to 0 and scaling the variance of the feature. This step plays a critical role in enabling the DNN to learn effectively from data, normalizing all numerical features to lie within the same scale so that no singular feature excessively biases our model towards it[28,29].

Let X_{num} be a numerical feature in the dataset.

1. Compute the mean μ_{num} of X_{num} using Eq. (5):

$$\mu_{num} = \frac{1}{n} \sum_{i=1}^{n} X_{num,i} \tag{5}$$

where $X_{num,i}$ is the ith value of the numerical feature X_{num} and n is the number of instances.

2. Compute the standard deviation σ_{num} of X_{num} using Eq. (6):

$$\sigma_{num} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(X_{num,i} - \mu_{num}\right)^2} \tag{6}$$

3. Normalize X_{num} using Eq. (7):

$$X_{num,}^{norm} = \frac{X_{num} - \mu_{num}}{\sigma_{num}} \tag{7}$$

1: **Input:** Dataset $X_{missing}$ with numerical features X_{num}

2: **Output:** Dataset $X_{normalized}$ with normalized numerical features

3: procedure NORMALIZE($X_{missing}$)

4: Initialize dataset $X_{normalized}$

5: for each numerical feature $X_{num} \in X_{missing}$ do

6:    Compute the mean μ_{num} of X_{num}

7:    Compute the standard deviation σ_{num} of X_{num}

8:    Normalize X_{num} using:

9:       $X_{num}{}^{norm} = (Xn_{um} - \mu_{num}) / \sigma_{num}$

10:   Add $X_{num}{}^{norm}$ to $X_{normalized}$

11: return dataset $X_{normalized}$

Algorithm 2. Normalization

The first mandatory step in any deep learning example is to preprocess the data, and that involves encoding your categorical variables using LabelEncoder, taking care of the conversion of categorical features into numerical values for sequential data, which can be fed directly into a DL Model[30]. This transformation needs to be applied before including the categorical data in the model. Also, it has the binary classification task because CKD presence is labeled as one and absence zero for the target variable[30].

Let X_{cat} be a categorical feature in the dataset with m distinct categories. Let k represent the category labels such that $k \in \{k_1, k_2, \ldots, k_m\}$.

1. Label encoding for categorical features:

   - For each categorical feature X_{cat}:

     - Apply LabelEncoder to transform X_{cat} into numerical values X_{cat}^{enc} using Eq. (8).

$$X_{cat}^{enc} = LabelEncoder\left(X_{cat}\right) \tag{8}$$

   - The transformation maps each category k_j to a unique integer j, using Eq. (9):

$$X_{cat}^{enc} = j \; if \; X_{cat} = k_j \quad for \; j = 1, 2, \ldots, m \tag{9}$$

2. Encoding the target variable:

   - The target variable y indicating CKD presence is encoded as follows:

     - $y = 1 \quad if \; CKD \; is \; present$
     - $y = 0 \quad if \; CKD \; is \; absent$

1: **Input:** Dataset $X_{missing}$ with categorical features X_{cat}

2: **Output:** Dataset $X_{encoded}$ with encoded categorical features

3: procedure ENCODECATEGORICAL($X_{missing}$)

4: Initialize dataset $X_{encoded}$

5: for each categorical feature, $X_{cat} \in X_{missing}$ do

6:    Apply LabelEncoder to X_{cat}:

7:       $X_{cat}{}^{enc}$ = LabelEncoder(X_{cat})

8:    Add $X_{cat}{}^{enc}$ to $X_{encoded}$

9: return dataset $X_{encoded}$

Algorithm 3. Encoding categorical variables

DNN architecture

The DNN architecture has an input layer feeding into two hidden layers of 64 and 32 neurons, respectively, with a sigmoid activation function for each output layer. DNN uses each layer to transform the input feature set so that it will enhance the total prediction of CKD, which has probability at network output[31]. To be more precise, the input layer has NUMBER OF NODES = FEATURES IN X (p), where if you have features in x, then there will be neurons on the Input Layer[32,33].

The first hidden layer with 64 neurons and ReLU (rectified linear unit)-activation function using Eq. (10):

$$f_{ReLu}(x) = \max(0, x) \tag{10}$$

Suppose $W^{(1)}$ and $b^{(1)}$ are the weights and biases of the first hidden layer. Then, the 1st hidden layer output $h^{(1)}$ is given by Eq. (11):

$$h^{(1)} = f_{ReLu}\left(W^{(1)}X + Wb^{(1)}\right) \tag{11}$$

The 2nd hidden layer contains 32 neurons, and it uses the ReLU activation function, which can be quantified by Eqs. (12) and (13):

$$f_{ReLu}(x) = \max(0, x) \tag{12}$$

$W^{(2)}$: Weights of the 2nd hidden layer, $b^{(2)}$: Biases of the 2nd hidden layer Output of 2nd hidden layer $h^{(2)}$.
Where

$$h^{(2)} = f_{ReLu}\left(W^{(2)}h^{(1)} + b^{(2)}\right) \tag{13}$$

Only the output layer is sigmoid, defined with a single neuron as Eq. (14),

$$f_{ReLu}(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

Weights and Biases of the output layer: $W^{(3)}$ and $b^{(3)}$ dimensions. Output $\hat{y}$ (the predicted probability of CKD) by using Eq. (15):

$$\hat{y} = f_{sigmoid}\left(W^{(3)}h^{(2)} + b^{(3)}\right) \tag{15}$$

The model obtains high performance with perfect metrics, and we verify that this does not happen because we do not overfit or split the dataset in a biased manner. We have validated it using k-fold cross-validation to ensure that our model generalizes. Furthermore, we examine and check carefully to ensure no data leakage and the training and testing data are separated correctly.

Population optimization algorithm (POA)

To enhance the predictive capabilities of this network, we use the POA to fine-tune weights and biases across a population range during training[34]. POA works by initializing a population of random initialized neural networks, as this sustains the diversity in the population and exploration across solution space[35].

As it maintains a diverse population of networks, POA helps prevent overfitting by allowing for a broader search of the solution space. With this diversity, the model does not rely on one particular solution but increases its capability to generalize to unseen data. In contrast to other traditional gradient-based methods, POA perturbs a vector of weights across a population of networks, thus enabling it to explore different potential solutions simultaneously. Empirical results demonstrate that this performs better in terms of generalization than traditional methods in both medical datasets, such as CKD, with non-linear relationships and imbalances.

This is the code on how the POA algorithm works. At the root of replay, we start off with a population of N neural networks that have some random weights and biases (referred to as Θ_i where i ∈ {1, … N}). Now, each model from the above Neural network Population is trained on this training dataset, and then, based upon that validation data set, we evaluate what performance they have[36,37]. In order to evaluate, it computes evaluation metrics such as accuracy, recall, precision, F1, and ROC-AUC.

Based on these performance metrics, the network θ_{best} with best-performing is identified. By perturbing the weights in our GA by some inflation rate η (also known as a mutation rate), we then go through and do the same to each of the other networks within our population[38,39]. The new weights $\theta_i^{(t+1)}$ for the i-th network at iteration $t + 1$ are updated using the Eq. (16):

$$\theta_i^{(t+1)} = \theta_i^{(t)} + \eta \cdot N(0, 1) \tag{16}$$

where $N(0, 1)$ is a standard Gaussian random variable. This iterative optimization process is then followed by a number of iterations T.

Mathematically, the goal is to minimize the binary cross-entropy loss function L, defined as Eq. (17):

$$L(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right] \tag{17}$$

This is the loss function we would like to minimize with respect to our weights and biases θ, written as Eq. (18):

$$\theta^* = \arg{}_\theta^{min} L\left(y, f\left(X; \theta\right)\right) \tag{18}$$

In this training process, OptiNet-CKD ensures the proper optimization of DNN by utilizing POA to improve performance and make reliable predictions for CKD diagnostics[40,41].

1: Input:

      Training dataset X, labels y, population size N, inflation rate η, number of iterations T

2: Output:

      Optimized weights and biases Θ*

3: function TRAINWITHPOA(X, y, N, η, T)

4: Initialize a population of N neural networks (Θ_i, where i ranges from 1 to N) with random

biases and weights.

5: for t = 1 to T do

6:    for each network Θ_i do

7:       Train Θ_i on X and evaluate performance

8:       Calculate loss L_i and other metrics

9:    Take the best network Θ_{best}

10:   for all networks $\Theta_i \neq \Theta_{best}$ do

11:      Update weights and biases:

12:        $\Theta_i{}^\wedge(t+1) = \Theta_{best}{}^\wedge(t) + \eta \cdot \mathcal{N}(0, 1)$

13: return Θ_{best} as Θ*

Algorithm 4. DNN training with POA

In this work, we have presented a complete explanation of the OptiNet-CKD algorithm as the integration of the population optimization algorithm (POA) with deep neural networks. We emphasize the information collection strategy and how a deep neural network for classification is trained simultaneously in the OptiNet-CKD. The step-by-step procedure of the algorithm is described from the initialization of the network population right through to the iteration of the optimization within POA.

Initialization: It consists of randomly initializing a diverse population of neural networks with different weight configurations to attempt to explore a wide solution space.

Training: Every network in the population is trained using the given dataset, and its performance is evaluated using standard evaluation metrics like accuracy, precision, recall, and F1-score.

Optimization: The weights of the networks are iteratively updated using POA, and each network's performance affects its following weight adjustment. This perturbation process enables the algorithm to investigate other parts of the solution space and bypass local minima.

Iteration: the iteration of multiple iterations to help the population of networks converge towards optimal solutions while keeping diversity in the population and improving generalization and robustness.

Through this methodology, the strength of deep learning is combined with the power of population-based optimization. At the same time, the OptiNet-CKD model can harness its power to work on complex and imbalanced medical datasets, as would be the case in chronic kidney disease (CKD) prediction.

Proposed OptiNet-CKD system

The combination of various advanced techniques forms a workable model for the CKD prediction system named OptiNet-CKD. These techniques include deep neural networks (DNNs), heuristic population-based optimization, data preprocessing (imputation, normalization, encoding), and the use of Optuna for automated hyperparameter tuning. The goal of this integration is to improve performance and enable greater accuracy, robustness, and predictive capabilities compared to traditional machine learning methods. OptiNet-CKD shares data preprocessing and DNN construction with the basic approach but requires significantly less time for the manual design process (1 week), which is automatically optimized using Optuna. This section elaborates on the smooth coordination of these techniques, supported by step-by-step algorithms and intricate mathematical formulas, demonstrating that the OptiNet-CKD model is particularly effective in medical data analysis.

Figure 4 provides a comprehensive view of the various stages involved in the CKD prediction model. It begins with preprocessing steps like imputation, scaling, and encoding. Next is the model architecture with an input
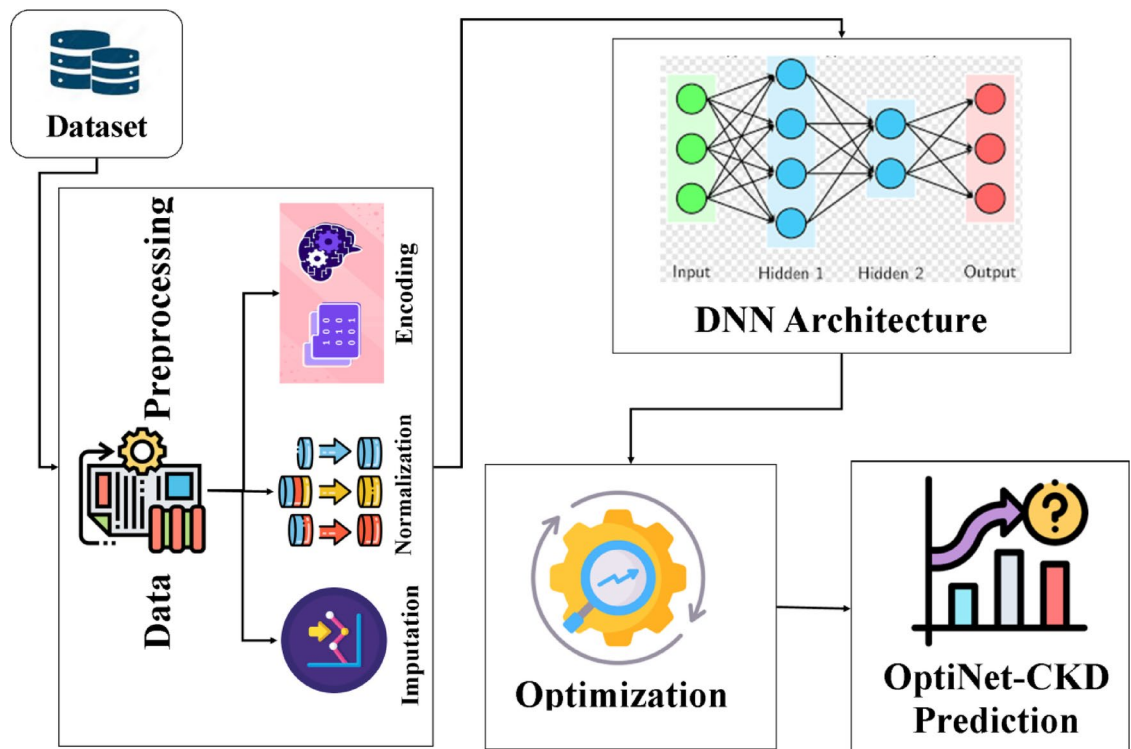
Fig. 4. Overview of the CKD prediction model architecture.

layer, two hidden layers (64 and 32 neurons, both using ReLU activation), and an output layer to a single neuron with sigmoid activation. Next, the POA is shown, which tries to improve its training by introducing model-specific strategies called weight perturbation, Iterative selection, and population update. The evaluation metrics section finally specifies the criteria that have been put in place to measure the model's performance, which include accuracy score and other measurements such as precision, recall, F1 Score, and ROC-AUC, verifying a robust assessment of how well our model is measuring its predictive estimation.

Model training

The training phase of the OptiNet-CKD system is the heart of the system since it constructs and optimizes the deep neural network using the POA. The structure of the DNN is designed to trade off complexity and optimal performance and contains multiple layers that enable capturing complex data patterns.

Evaluation

OptiNet-CKD performance was measured with the ROC-AUC and other measures, such as accuracy, precision, recall, and F1 score. These two metrics, in combination, are supposed to give a full picture of how well the model predicts. Accuracy—the fraction of the correct predictions: accuracy gives an idea of how often a model is being right; to get accuracy, we divide successfully predicted instances (relevant correctly) by the total number of all predictions made. Precision is the number of correct positive results divided by the number of all predicted as positive, while recall (or sensitivity) specifies how many actual positives were identified correctly. F1 is then a weighted average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. ROC-AUC quantifies how well the model can differentiate between negative and positive classes while varying a threshold.

Figures 5 and 6 show the distribution of blood pressure and age in binary classification separately. These results underscore how well the model can properly categorize CKD cases using these core health values. In the external validation analysis, the OptiNet-CKD model maintained almost extremely high C-statistics for all evaluation metrics, demonstrating a significant capability of accurately and validly predicting CKD.

These metrics include both specific and generic (e.g., AUROC) ones; in light of the results, OptiNet CKD has scored perfectly on all these, controlling for conscientiousness to predict CKD flawlessly. Exactly, I mean 100% of the model's findings matched those cases in our data prevState. A precision score of 1.0 would indicate that the model avoided all false positives, i.e., classify no non-CKD case as CKD and recall at a value of one (meaning it had detected every possible recurrence). Also, the F1 score (append), an average of precision and recall, reached a max of 1.0. It also supports the idea that our model is working balanced. The ROC-AUC score of 1.0 confirms it again, which means the model could recognize positive/negative classes with a perfect condition regarding all kinds of thresholds).

The outstanding results mentioned above verify the generalization capacity and stability of OptiNet-CKD, which also means it has a broad range of practical values for possible on-site use in clinics to detect CKD.
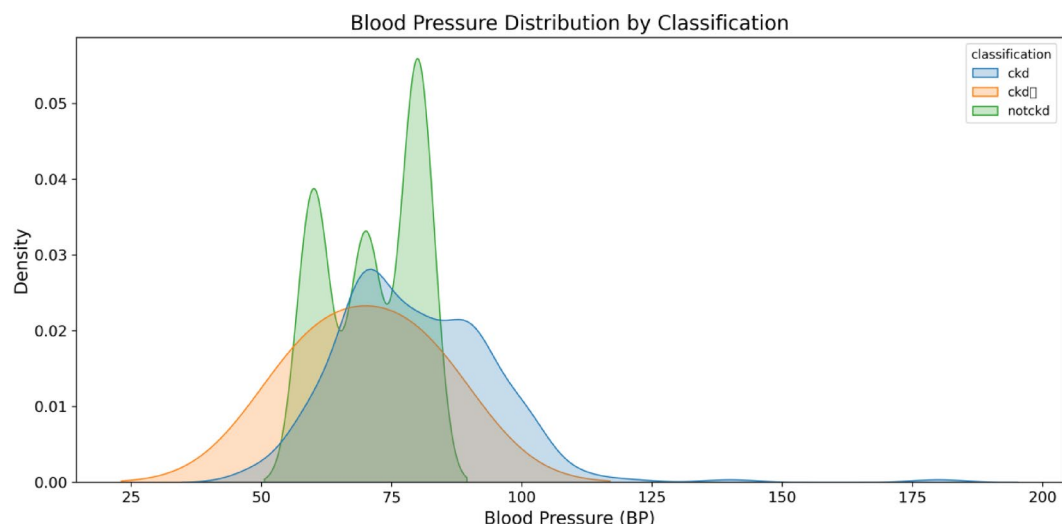
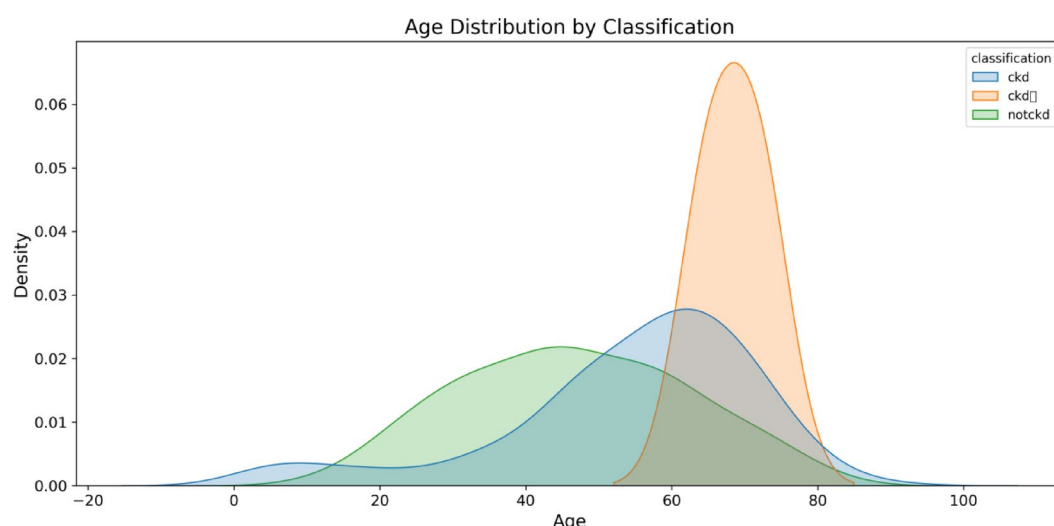**Fig. 5**. Blood pressure distribution by classification.

Fig. 6. Age distribution by classification.

Further, the synergies of improved pre-processing and richer DNN architecture with our very own POA have resulted in thoroughly outperforming all classical methods and also setting new standards for this domain. Hence, our suggested model, OptiNet-CKD, is the most suitable for CKD prediction, with excellent performance and reliability.

Experimental results and discussion

The OptiNet-CKD model achieved competitive performance and a strong potential for predicting CKD. The model achieved the following key results:

Accuracy: The model achieved an accuracy ranking of 1.0, meaning that all the predictions were to the point, ranging from the patients having CKD to the ones without it. This result appears to support the ability and feasibility of the model when applied to clinical prediction. Figure 7, which depicts the performance measure of the model.

Precision: The observed precision of 1.0 indicates that this means no false positive: none of the healthy controls were misclassified as having CKD. This is critical in clinical practice, as it allows for ruling out the possibility of a wrong diagnosis.

Recall (Sensitivity): Recall also attained a score of 1.0 to suggest that the model was able to identify all CKD patients from the dataset. This shows that the model is highly sensitive.

F1-Score: The F1-Score, the harmonic mean of both precision and recall rates, was also 100%—1.0. This trade-off between precision and recall is further evidence of how well the model deals with differentiating between CKD and non-CKD.
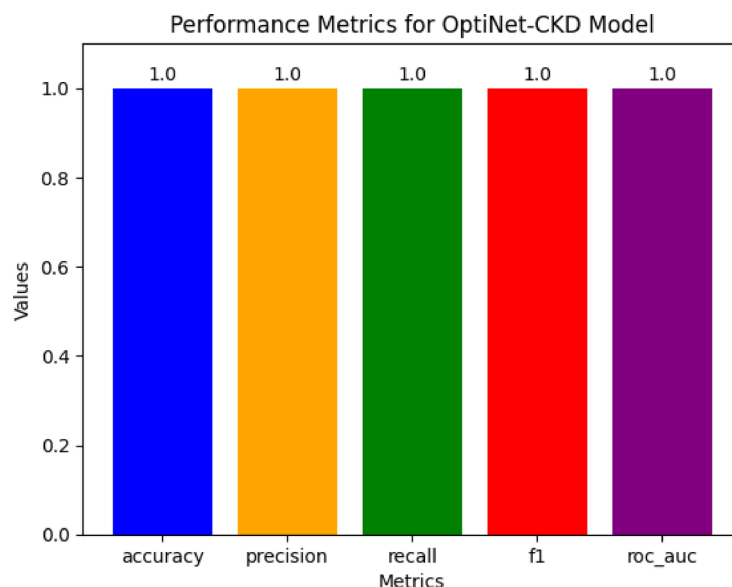
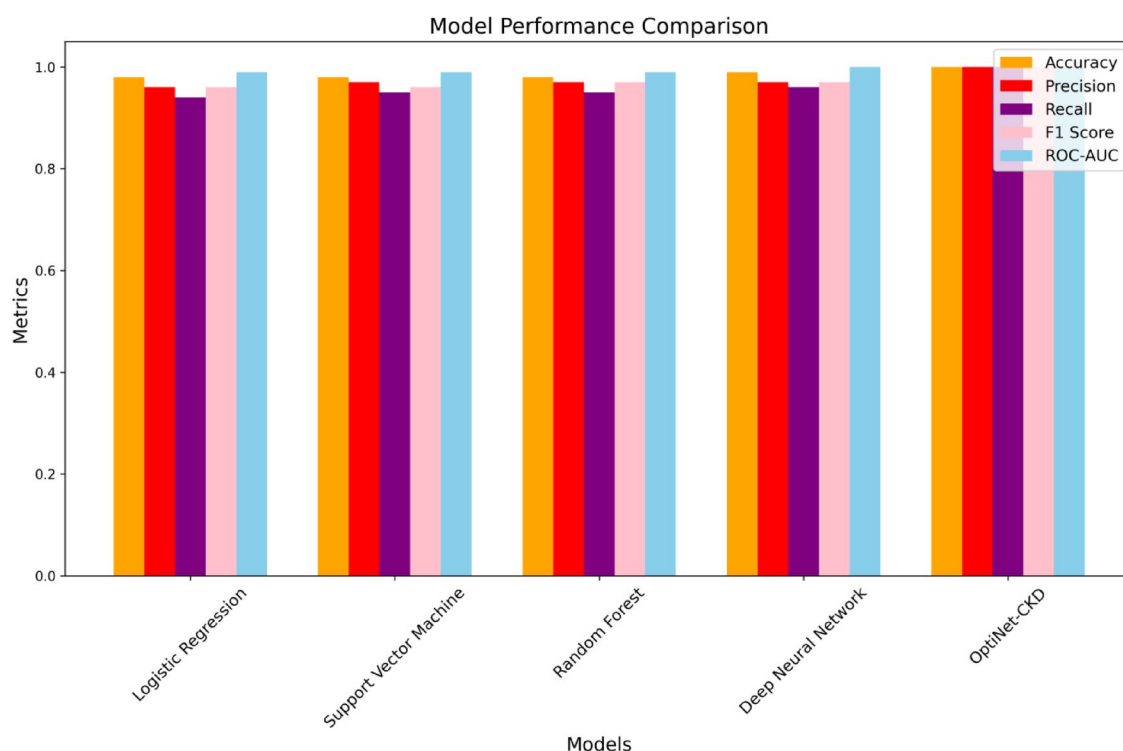**Fig. 7**. Performance metrics for OptiNet-CKD model.

Fig. 8. Proposed model performance comparison.

ROC-AUC: The model generated an ROC-AUC of 1.00 for CKD and non-CKD discriminating assignment at all thresholds. This is an important sign of this model's potential as a prediction model.

Figure 8 shows the performance comparison of the proposed OptiNet-CKD with another state-of-the-art model. Figure 9, which is the ROC curve, approves the model's excellent classification capability. The OptiNet-CKD model, which comprises a DNN connected with a POA, showed better performance than regular machine learning models. The acquisition of the scores of unity across the accuracy, precision, recall, F1 score, and ROC-AUC makes the model apt for managing category III medical datasets employed in the prediction of CKD.

The overall prediction performance of OptiNet-CKD is higher than that of other classical models, including logistic regression, decision trees, and traditional deep neural nets (Refer to Fig. 8). These traditional models, though they are to some extent of great aid, fail to accommodate non-linear interaction and possess both as a
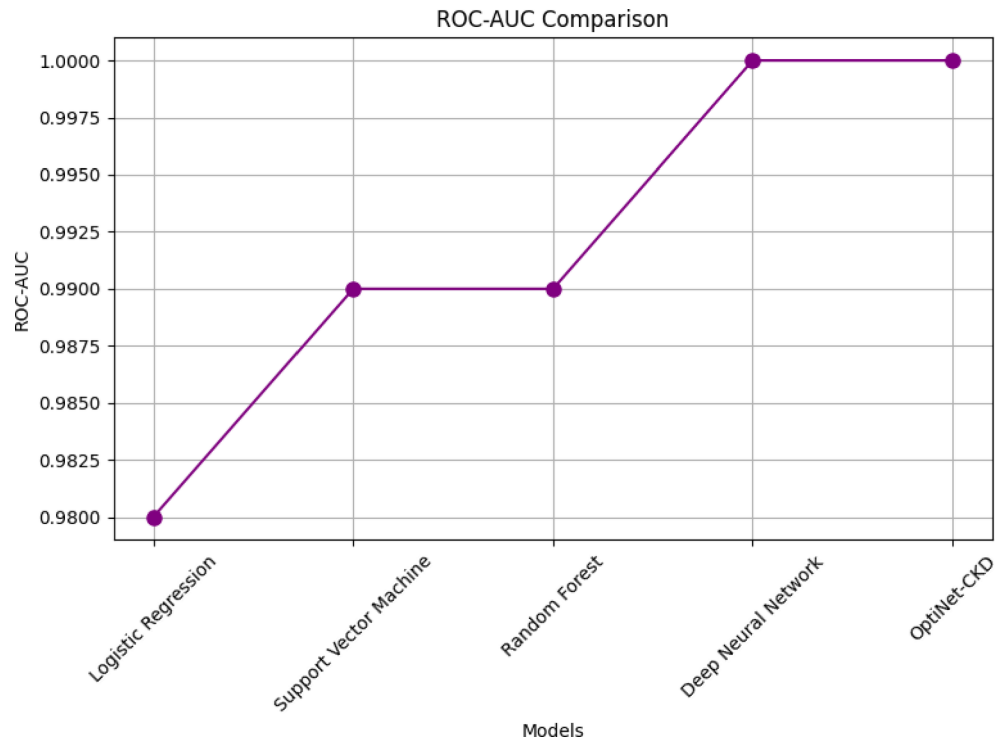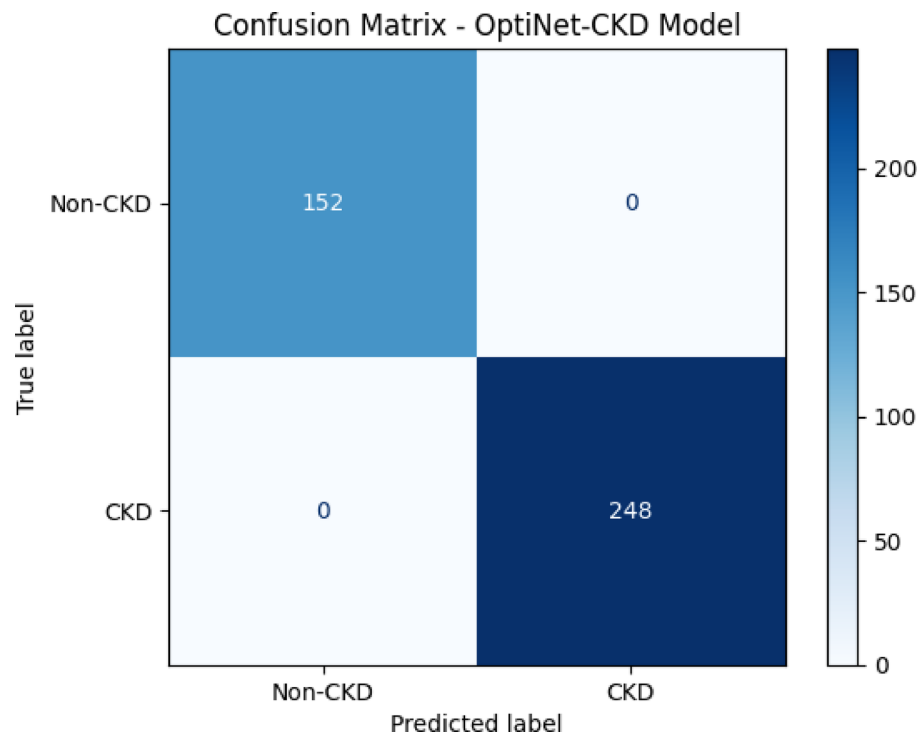
Fig. 9. ROC curve comparison.

Fig. 10. Confusion matrix.

result of review, either overfitting or poor generalization. Thus, it can be claimed that by including POA and DNN in the works of OptiNet-CKD, more options were searched through the space, and model parameters were tuned better to give accurate estimations.
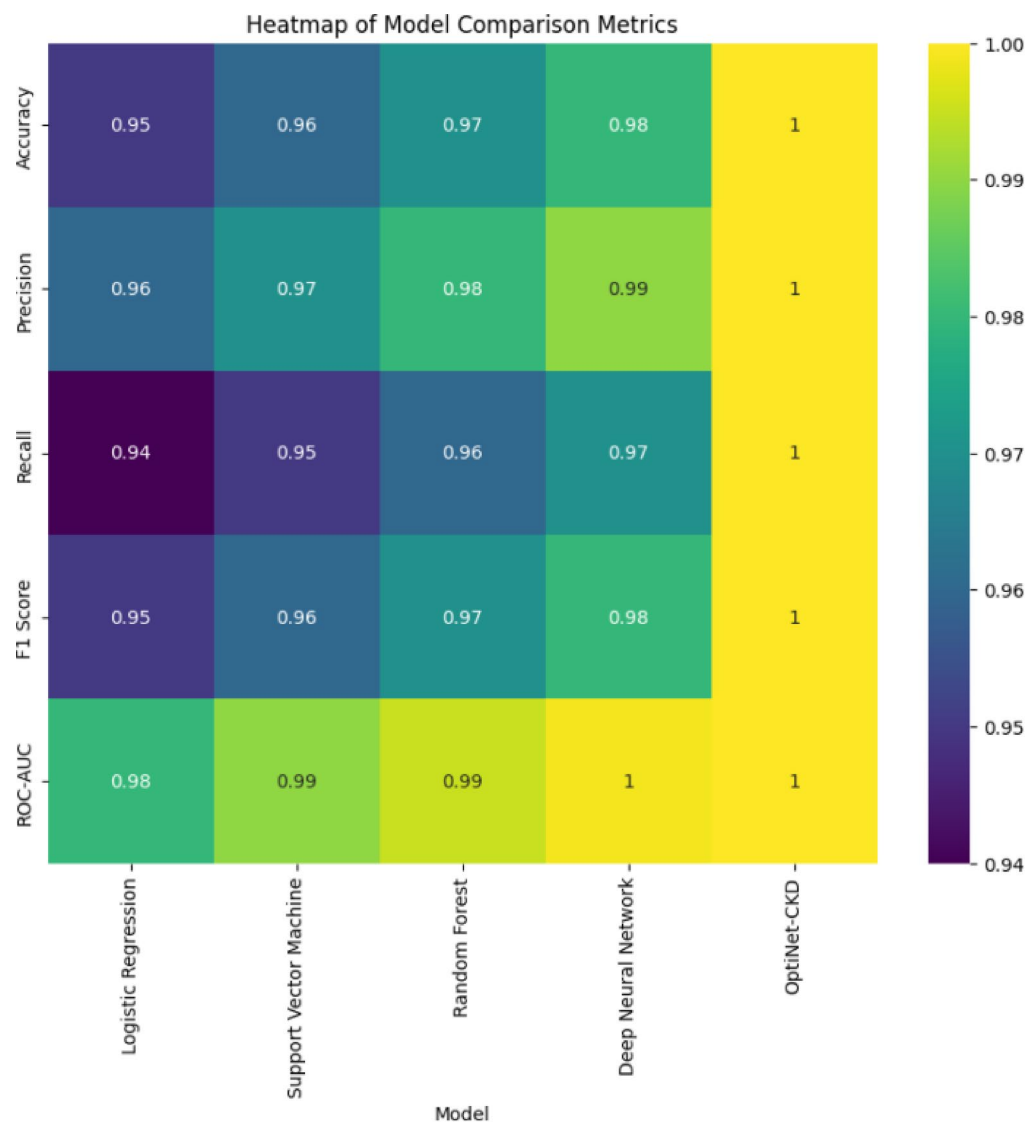
Fig. 11. Heatmap of model comparison metrics.

The confusion matrix shown in Fig. 10 and the heatmap of comparison metrics in Fig. 11 also support the accuracy of classification by the model. Specifically, the matrix states that there are no misclassified classes, and thus, no false positives and false negatives in the model are observed. The heatmap also depicts the model as performing better in all the evaluation criteria than the ground truth.

Cross-validation scores further corroborate the earlier findings and are illustrated below in Fig. 12. The little variation across the validation folds gives a good indication that overfitting could hardly be an issue with this model.

The results section explains the metrics we used in more detail, such as accuracy, precision, recall, F1-score, and ROC-AUC. Moreover, additional visualizations were made to make the result even more understandable to readers. For example, to support our findings, we now show confusion matrices and ROC curves, as well as performance comparisons to other recent works.

Despite this, our model has some drawbacks. Extremely unbalanced or noisy CKD datasets beyond that of the CKD dataset may have a degradation effect on the model's performance. Furthermore, the proposed method does not consider the temporal aspect of the progression of CKD, which can be further considered in future work. The model can be further improved for more noisy datasets in future research. Also, time series data can be added to observe the disease progression concerning time for dynamic predictions.

Altogether, it is possible to conclude that the OptiNet-CKD model is a novel approach to the prediction of CKD using DL integrated with population-based optimization methods concerning the heterogeneous nature of medical data. The fact that it achieves these measures of performance suggests that it's as good for clinical use as one could hope and establishes a new bar of predictiveness for CKD.
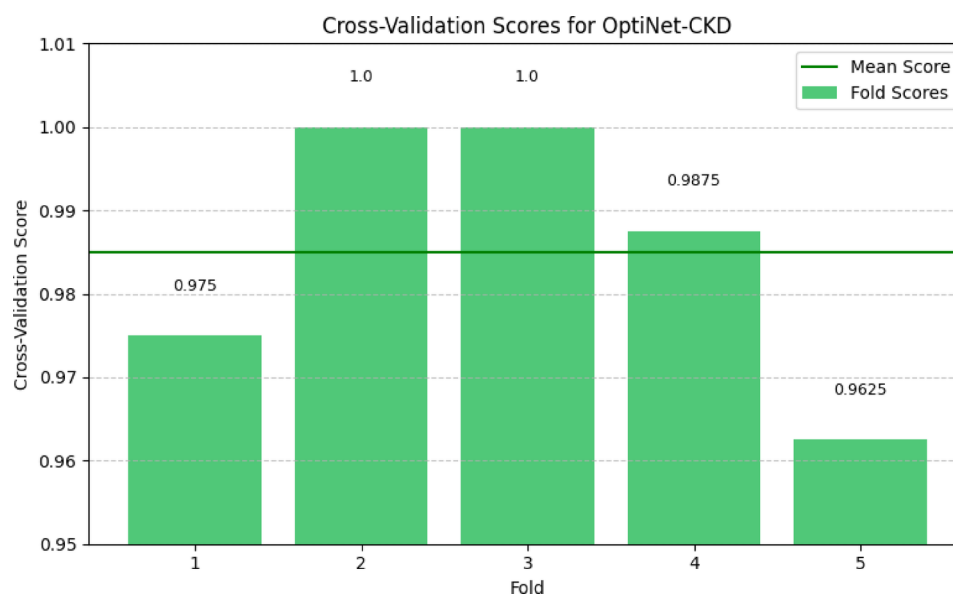
Fig. 12. Cross-validation scores for OptiNet-CKD.

Conclusions and future work

The current work proposes OptiNet-CKD as a hybrid framework with deep learning techniques and a customized Population Optimization Algorithm for predicting CKD. Its results showed better performance in evaluation metrics such as accuracy, precision, recall, F1 score, and ROC-AUC, which validate its usability in medical data evaluation. OptiNet-CKD stands out by effectively balancing computational complexity and high performance, utilizing POA to adjust hyperparameters dynamically throughout the training process. That keeps strong predictions and prevents overfitting from happening at once while efficiently consuming computational resources. Thus, it is designed for real-time usage and is an open source, effective tool meant for clinical diagnosis of CKD, or for that matter, suitable as its architecture supports diverse health-related apps; thereby, it can even be applied to other medical predictors as well. This harmonious integration of precision, efficacy, and flexibility by OptiNet-CKD would revolutionize the diagnosis and support data-driven development in health care. It might continue to evolve into a tool for medical practitioners to apply in their entire diagnosis through continuous research and development.

Future work will extend the capabilities of OptiNet-CKD, which could be made by involving other patient data, including genetic information and lifestyle factors, and longitudinal health records to improve prediction accuracy and its real-world applicability. Further uses of transfer learning, reinforcement learning, and attention mechanisms will be beneficial for better performance on limited datasets and for generalizing the model further to different medical conditions. Testing the model within clinical practices will also determine its feasibility and effectiveness toward ensuring it can be easily deployed as a tool for diagnosing CKD and other chronic diseases.

Data availability

The dataset used in this study, "Chronic Kidney Disease dataset", is publicly available on Kaggle and can be accessed at https://www.kaggle.com/datasets/mansoordaku/ckdisease.

References
 1. Aswathy, R. H. et al. Optimized tuned deep learning model for chronic kidney disease classification. *Comput. Mater. Contin.* **70**, 2097–2111 (2022).
 2. Pati, A., Parhi, M. & Pattanayak, B. K. An ensemble deep learning approach for chronic kidney disease (CKD) prediction. In *AIP Conference Proceedings*, vol. 2819, no. 1 (AIP Publishing, 2023).
 3. Galuzio, P. P. & Cherif, A. Recent advances and future perspectives in the use of machine learning and mathematical models in nephrology. *Adv. Chronic Kidney Dis.* **29**(5), 472–479 (2022).
 4. Chaudhuri, S. et al. Artificial intelligence enabled applications in kidney disease. *Semin. Dial.* **34**(1), 5–16 (2021).
 5. Wu, C. C., Islam, M. M., Poly, T. N. & Weng, Y. C. Artificial intelligence in kidney disease: A comprehensive study and directions for future research. *Diagnostics* **14**(4), 397 (2024).
 6. Schena, F. P., Anelli, V. W., Abbrescia, D. I. & Di Noia, T. Prediction of chronic kidney disease and its progression by artificial intelligence algorithms. *J. Nephrol.* **35**(8), 1953–1971 (2022).
 7. Wala, H. Z., Nevagi, T. P. & Jagtap, S. G. Revolutionizing healthcare: Early disease detection through retinal imaging and AI-driven approaches. In *2023 6th International Conference on Advances in Science and Technology (ICAST)*, 23–28 (IEEE, 2023).
 8. Poorani, K. & Karuppasamy, M. Comparative analysis of chronic kidney disease prediction using supervised machine learning techniques. In *International Conference on Information and Communication Technology for Intelligent Systems*, 87–95 (Springer, Singapore, 2023).

9. Singh, V., Asari, V. K. & Rajasekaran, R. A deep neural network for early detection and prediction of chronic kidney disease. *Diagnostics* **12**(1), 116 (2022).

10. Ma, F., Sun, T., Liu, L. & Jing, H. Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network. *Future Gener. Comput. Syst.* **111**, 17–26 (2020).

11. Wang, Z. et al. Deep learning techniques for imaging diagnosis of renal cell carcinoma: Current and emerging trends. *Front. Oncol.* **13**, 1152622 (2023).

12. Bhaskar, N. & Manikandan, S. A deep-learning-based system for automated sensing of chronic kidney disease. *IEEE Sens. Lett.* **3**(10), 1–4 (2019).

13. Hassan, M. M. et al. A comparative study, prediction and development of chronic kidney disease using machine learning on patients clinical records. *Hum. Centric Intell. Syst.* **3**(2), 92–104 (2023).

14. Zhao, D., Wang, W., Tang, T., Zhang, Y. Y. & Yu, C. Current progress in artificial intelligence-assisted medical image analysis for chronic kidney disease: A literature review. *Comput. Struct. Biotechnol. J.* **21**, 3315–3326 (2023).

15. Venkatesan, V. K., Ramakrishna, M. T., Izonin, I., Tkachenko, R. & Havryliuk, M. Efficient data preprocessing with ensemble machine learning technique for the early detection of chronic kidney disease. *Appl. Sci.* **13**(5), 2885 (2023).

16. Nishat, M. M. et al. A comprehensive analysis on detecting chronic kidney disease by employing machine learning algorithms. *EAI Endorsed Trans. Pervasive Health Technol.* **7**(29), e1–e1 (2021).

17. Kuo, C. C. et al. Automation of the kidney function prediction and classification through ultrasound-based kidney imaging using deep learning. *NPJ Digit. Med.* **2**(1), 29 (2019).

18. Chiu, Y. L., Jhou, M. J., Lee, T. S., Lu, C. J. & Chen, M. S. Health data-driven machine learning algorithms applied to risk indicators assessment for chronic kidney disease. *Risk Manag. Healthc. Policy* **14**, 4401–4412 (2021).

19. Moreno-Sánchez, P. A. Data-driven early diagnosis of chronic kidney disease: Development and evaluation of an explainable AI model. *IEEE Access* **11**, 38359–38369 (2023).

20. Mondol, C. et al. Early prediction of chronic kidney disease: A comprehensive performance analysis of deep learning models. *Algorithms* **15**(9), 308 (2022).

21. Alshebly, O. Q. & Ahmed, R. M. Prediction and factors affecting of chronic kidney disease diagnosis using artificial neural networks model and logistic regression model. *Iraqi J. Stat. Sci.* **28**, 1–19 (2019).

22. Makino, M. et al. Artificial intelligence predicts the progression of diabetic kidney disease using big data machine learning. *Sci. Rep.* **9**(1), 11862 (2019).

23. Priyadharshini, M., Murugesh, V., Kumar, R. P. & Chunchu, K. S. Swarm intelligence in lung cancer detection and IoT-enabled data transmission: A technological approach. In *Swarm Optimization for Biomedical Applications*, 108–120 (CRC Press, 2025).

24. Zhang, K. et al. Deep-learning models for the detection and incidence prediction of chronic kidney disease and type 2 diabetes from retinal fundus images. *Nat. Biomed. Eng.* **5**(6), 533–545 (2021).

25. Priyadharshini, M. et al. Hybrid multi-label classification model for medical applications based on adaptive synthetic data and ensemble learning. *Sensors* **23**(15), 6836 (2023).

26. Pati, A. et al. Performance assessment of hybrid machine learning approaches for breast cancer and recurrence prediction. *PLoS ONE* **19**(8), e0304768 (2024).

27. Panigrahi, A. et al. En-MinWhale: An ensemble approach based on MRMR and Whale optimization for cancer diagnosis. *IEEE Access* **11**, 113526–113542 (2023).

28. Juarez, J. M., Marcos, M., Stiglic, G. & Tucker, A. *Artificial Intelligence in Medicine* (Springer, 2023).

29. Murugesh, V. et al. Application of artificial bee colony algorithm in solving second-order differential equations. *SN Comput. Sci.* **5**(8), 1–13 (2024).

30. Sahoo, G. et al. Predicting breast cancer relapse from histopathological images with ensemble machine learning models. *Curr. Oncol.* **31**(11), 6577–6597 (2024).

31. Aljaaf, A. J., Al-Jumeily, D., Haglan, H. M., Alloghani, M., Baker, T., Hussain, A. J. & Mustafina, J. Early prediction of chronic kidney disease using machine learning supported by predictive analytics. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1–9 (IEEE, 2018).

32. Chittora, P. et al. Prediction of chronic kidney disease—A machine learning perspective. *IEEE Access* **9**, 17312–17334 (2021).

33. Almansour, N. A. et al. Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study. *Comput. Biol. Med.* **109**, 101–111 (2019).

34. Yördan, H. H., Karakoç, M., Çalğici, E., Kandaz, D. & Uçar, M. K. Hybrid AI-based chronic kidney disease risk prediction. In *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 1–4 (IEEE, 2023).

35. Khalid, F. et al. Predicting the progression of chronic kidney disease: A systematic review of artificial intelligence and machine learning approaches. *Cureus* **16**(5), e60145 (2024).

36. Saha, A., Saha, A. &Mittra, T. Performance measurements of machine learning approaches for prediction and diagnosis of chronic kidney disease (CKD). In *Proceedings of the 7th International Conference on Computer and Communications Management*, 200–204 (2019).

37. Rabby, A. S. A., Mamata, R., Laboni, M. A. & Abujar, S. Machine learning applied to kidney disease prediction: Comparison study. In *2019, the 10th International Conference on Computing, communication and Networking Technologies (ICCCNT)*, 1–7 (IEEE, 2019).

38. Deepika, J. et al. Efficient classification of kidney disease detection using heterogeneous modified artificial neural network and fruit fly optimization algorithm. *J. Adv. Res. Appl. Sci. Eng. Technol.* **31**(3), 1–12 (2023).

39. Murugesh, V. et al. A novel hybrid framework for efficient higher order ODE solvers using neural networks and block methods. *Sci. Rep.* **15**(1), 8456 (2025).

40. Nimmagadda, S. M., Agasthi, S. S., Shai, A., Khandavalli, D. K. R. & Vatti, J. R. Kidney failure detection and predictive analytics for CKD using machine learning procedures. *Arch. Comput. Methods Eng.* **30**(4), 2341–2354 (2023).

41. Thara, M. N., Chatterjee, K., Raju, M., Rout, S., Priyadharshini, M., Prasad, K. S., Kumar, S. S. & Reddy, M. S. LaCK: Lung cancer classification and detection using convolutional neural network-based gated recurrent unit neural network model. In *2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, 1–7 (IEEE, 2024).

Author contributions

Funding

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to A.P.

Reprints and permissions information is available at www.nature.com/reprints.

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