scientific reports

Check for updates

OPEN

Early prediction of CKD from time series data using adaptive PSO optimized echo state networks

Thangadurai Anbazhagan✉ & Balamurugan Rangaswamy✉

Chronic Kidney Disease (CKD) is a significant problem in today's healthcare since it is challenging to detect until it has improved significantly, which increases medical expenses. If CKD was detected early, the patient might qualify for more effective treatment and prevent the disease from spreading further. Presently, existing methods that effectively detect CKD cannot detect symptoms early on. This problem motivates researchers to work on a predictive model that successfully detects disease symptoms in the early stages. This study introduces a novel Adaptive Particle Swarm Optimization (APSO)-optimized Echo State Network (ESN) model designed to overcome key limitations of existing methods. ESNs, while effective in processing temporal sequences, are highly sensitive to hyperparameter settings such as spectral radius, input scaling, and sparsity, which directly impact stability, memory retention, and predictive Classification Accuracy (CA). To address this, APSO optimizes these hyperparameters dynamically, ensuring a balanced trade-off between stability and computational efficiency. Moreover, Random Matrix Theory (RMT) is integrated into APSO to regulate the spectral radius, enhancing the ESN's capability to handle long-term dependencies while maintaining stability in training. This investigation exploited the Medical Information Mart for Intensive Care-III (MIMIC-III) dataset to train the model they developed. The proposed method employs this data collection to analyze the highly complex temporal sequences signifying CKD is present. The hyperparameters of the ESN, such as the range of the spectral region and the input data sizing, can be optimized in real-time with APSO by applying Random Matrix Theory (RMT). Compared with different recognized models, such as conventional ESN and standard M, the recommended APSO + ESN proved to have higher CA in medical investigations. The APSO + ESN improved the subsequent highest-performing model by 2% in recall and 3% in precision and attained a CA of 99.6%.

Keywords Chronic kidney disease, Adaptive particle swarm optimization, Echo state network, Deep learning, Random matrix theory, Accuracy

The primary medical problem worldwide, Chronic Kidney Disease (CKD), causes a gradual decrease in Kidney Function (KF). It is important to detect this disease early to prevent severe kidney failure from spreading. However, the lack of symptoms of the disease makes early detection particularly complicated. Clinical tests, tests for blood sugar, laboratory tests for urine, and ultrasounds have long served as the backbone of CKD detection techniques. All such methods perform well for detecting kidney function metrics, but they rely on static proof and cannot select minor but significant changes that develop at the starting point of CKD progression. Despite its significance, early detection remains elusive due to the disease's progression's complex, temporal nature and current diagnostic methods' limitations. This research addresses the capability of causing death in CKD detection, which is essential for the dynamic, time-sensitive analysis of kidney function progression that existing methods, including traditional clinical tests and modern Machine Learning (ML) models, fail to capture effectively.

New possibilities to enhance CKD diagnosis are now available because ML is designed to address the errors of traditional models. ML is superior to conventional models in sensitivity and specificity when used with big data to identify CKD procedures and predictors. Because of static datasets without the disease's chronic life, early ML proved accurate in detecting CKD. Drugs, food, and related diseases all serve an integral part in the unpredictable nature of CKD symptoms across life. Early detection and accurate prediction of CKD progression are complicated by models developed from static data trials, which fail to detect significant temporal patterns. A more accurate development of CKD can be detected by sequence time series data that monitor the values of KF over a long time. Therefore, ML developed on identical time series data must comprehend the input data's time-

Department of Electrical and Electronics Engineering, K.S.Rangasamy College of Technology, Tiruchengode 637215, Tamil Nadu, India. ✉email: thangaduraieie@gmail.com; drnrbals@gmail.com

dependent relations and variations. Several applications have shown the effective performance of Recurrent Neural Networks (RNN) and derivatives like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) when analyzing time-series data. Since RNN function more accurately in early CKD detection, training the models on such datasets is challenging, cost-effective, and resource-intensive due to problems like dropping and increasing gradients.

Reservoir Computing (RC) is a prominent RNN developed to deal more effectively with time series data. The RC employs multiple techniques. One noteworthy model distinct from RNN and their different forms (LSTM + GRU) is the application of Echo State Networks (ESN). ESN features a static storage with random relations of neurons. The ESN assesses the input data and detects the temporal dependencies by training its neural relations, minimizing computation cost. Variable optimization, mainly for the spectral radius and input scaling, is essential for the ESN to function correctly, although they typically face growing and vanishing gradient issues. The spectral radius variable defines the reliability and memory size of the network's components; changing this value impacts the ESN's potential for storing and applying data from input sequences.

The input data scale factor determines the network's adaptation to and ability to learn from input data. By addressing massive time-series data, maximizing those variables becomes important to extract the best probable result for the ESN.

The primary objective of this research work is to solve earlier challenges of researching chronic data by improving ESN's vital parameters. In this work, researchers address these limitations by proposing a novel Adaptive Particle Swarm Optimization (APSO)-optimized ESN enhanced by Random Matrix Theory (RMT). The APSO algorithm dynamically tunes ESN hyperparameters, including spectral radius and input scaling, ensuring the model is optimized for stability, memory retention, and adaptability. RMT is integrated into the APSO to achieve an optimal spectral model within the ESN sub-reservoirs, enabling the model to process complex, time-sensitive data efficiently. This dual revolution addresses the vanishing gradient problem and stability-memory trade-off inherent in traditional ESN, significantly improving the model's predictive capabilities. The findings from the Medical Information Mart for Intensive Care-III (MIMIC-III) sample specify that this improves the Classification Accuracy (CA), precision, and recall of CKD diagnoses compared to other traditional methods.

Literature review

This work successfully investigates the research studies that assess several distinct Deep Learning (DL) for predicting CKD development, as the summary provided below exposes.

Several DL likes—Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional LSTM, Bidirectional GRU, MultiLayer Perceptron (MLP), and Basic RNN—were tested by[1] in the study they conducted for their objective of diagnosing CKD. In summary of the results, the ANN, RNN, and MLP were reliable, with CA values from 96 to 99% based on the evaluated test. The study participants were successful in an attempt to implement practical methods into the Internet of Medical Things (IoMT) in an initiative to enhance the CA of the prediction. From the existing research studies[2], designed a *state-of-the-art* Deep Neural Network (DNN) that may diagnose CKD earlier. Feature selection (FS) is what researchers used to train the ML. The method collected significant features such as hemoglobin, precise gravity, and plasma cholesterol. During the procedure of the tests in which they analyzed the algorithms, they built for Support Vector Machine (SVM), k-nearest Neighbours (k-NN), Logistic Regression (LR), Random Forest (RF), and Naive Bayes (NB), researcher experiential that their discrete model was highly reliable.

A Deep Ensemble Model (DEM) combining CNN + LSTM + LSTM-BLSTM classifiers was developed by[3] to predict the start of CKD is 6 or 12 months in advance. The model succeeded over independent methods with a prediction CA of 0.993 and 0.992 for the appropriate periods. For the objective of CKD classification, a hybrid CNN + LSTM was feasible. Its 99.17% CA demonstrated the RNN + CNN[4] on a 2-class CKD dataset, which was superior to other standard models. The "Adaptive Hybridized Deep Convolutional Neural Network (AHDCNN)" is a learning model that was primarily developed by[5] to deal with the problem related to early CKD. To conduct the research, the examiners applied CNN for classifications and Gabor filters to eliminate feature sets from the data they had acknowledged. When analyzed with data from the IoMT setting, the model they designed indicated an enhanced level of achievement in terms of CA.

Authors[6] advised the application of an IoMT to detect and classify CKD accurately. The researcher used the Backpropagation algorithm to perform the recognition study when working with a Heterogeneous Modified Artificial Neural Network (HMANN). These networks comprised SVM, MLP, and additional models. By training the computerized model with multiple sets of ultrasound images, the authors could determine its limits and indicate that their methods are higher than those of different classification and segmentation CA models. The researchers[7] invented the SVM-integrated Correlational Neural Network (CorrNN) technique to provide a model for detecting signs of CKD that proves more efficient computation. An improved CA of 98.67% was shown by the model that was invented about different available prediction approaches using conventional CNN. Furthermore, the method demanded less computational time.

The study by[8] evaluates CNN (EfficientNetB0-B3) and Vision Transformer (ViT) approaches for cervical cancer detection. It explores fusion approaches combining CNN and ViT features to improve diagnostic CA, highlighting the potential of hybrid models for improved medical imaging results.

Similarly[9], developed an ensemble of DL to predict CKD beginning, signifying combining multiple DL for improved prediction CA.

This research introduced[10] a hybrid CNN + SVM to measure Electronic Health Records (EHR) for early CKD prediction. The model demonstrated superior performance compared to traditional ML. Authors[11] applied several MLs, including SVM and RF, to time-series data for early CKD detection. The FS was employed to enhance model performance, achieving a CA of 95%.

Additionally, a comparative analysis by[12] evaluated numerous ML for CKD risk prediction, emphasizing the importance of FS in improving model performance.

The following works have been conducted focusing on RNN like LSTM, GRU, and Bi-LSTM for time-series data:

In the study[13], argued for a more successful RNN of learning to detect precise CKD diagnosis and monitoring. And took advantage of an augmented LSTM to perform a finding on CKD. Attention and residual mechanisms have been implemented to enhance the LSTM. The context of this study showed that the expanded LSTM attained an improved level of CA relative to additional approaches. In order to predict the advancement of CKD from stages II/III to IV/V[14], deployed an RNN that had been developed on chronic EHR. The study Area Under the Receiver Operating Characteristics (AUROC) of 0.957 was achieved using the method the researchers applied, which was developed from predictable glomerular filtration rate (eGFR) data. The AUROC increased to 0.967 when new health-related data was factored in, depending on the experiment's test[15]. It integrates S-MTL with Simple RNN and MLP. Significant health risks have been predictable, namely eGFR and creatinine. As validated on the University of California Irvine machine learning repository (UCI-ML) dataset, the model they developed accurately predicts all five stages of CKD with a CA of 99.2–99.8%. Moreover, they rendered their model accessible online as a smartphone application.

A study by[16] also applied RNN to detect CKD at its initial stages, employing a novel hybrid FS and robust data preparation pipeline to enhance prediction CA.

The application of a 1-D CorrNN and Bi-LSTM in Salivary Analysis (SA) for preliminary identification of CKD has been studied by[17]. Their experimental DL test with non-invasive SA achieved a 98.08% CA rate. Applying Transformer-RNN autoencoders, TRACE, a complete prediction model, was developed by[18]. They used all-purpose clinical data to train algorithms for deploying data loss and complicated risk factors. Their simulation achieved an Area Under the Precision-Recall Curve (AUPRC) of 0.5708 when evaluated with different traditional methods. The implementation of CNN + RNN for feature-less CKD prediction has been studied by[19]. When researchers tested the RNN + CNN using several assessments, they found that the RNN was executed more successfully than the one that used CNN[12]. introduced a novel DL combining FS with the determined diagnosis of better CKD. The study authors applied an advanced FS framework described as the Local Search with Nearest Neighbour (LSNN) improvement method to diagnose the most helpful features compared to the high-dimensional data presented as data. The next stage in establishing an efficient method for the diagnosis of CKD was to develop a more modern DL, which they termed an Integrated Deep Belief Network (IDBN).

The following are Echo State Network (ESN)-Focused Approaches:

The ESN has more scope than the obvious CDSS, based on an investigation performed by[20]. The electroencephalogram (EEG) data contributed by the end-user was employed as data for the design of an ESN for ischemic stroke diagnosis. In the tests they conducted, the algorithm they designed attained a CA level of 99.6%, and the researchers used global and local explanation techniques to render their results clear. Normalized Echo State Networks (NESN) have been used by[21] in their study's attempt to classify CKD by using electrocardiogram (ECG) data as the source. It was owing to what followed that the medical experts could diagnose coronary artery disease. They proved better than the more traditional techniques, attaining a rate of CA of 99.8% and an overall F1-score of 0.995.

The Authors[22] invented a multi-sourced data-driven approach to enhancing End-Stage Renal Disease (ESRD) result of prediction, utilizing ESNs to integrate diverse clinical and claims data for improved prediction.

In the most recent research, which has been recommended[23], succeeded in effectively optimizing the hyperparameter variables of the ESN for prediction by the use of Bayesian Optimization Algorithms (BOAs). The aim was to start the most reliable ESN feasible so that they could provide accurate mandate predictions. After performing a study on real-life data sets, they revealed the evidence that their suggested approach proved efficient[24]. reported an ESN that had been enhanced via Binary Grey Wolf Optimization (BGWO) to examine the CA of prediction throughout an array of input data sources. The following methodology is preferable to traditional methods across numerous computer-generated and real datasets. This is evident in contrast to other traditional methods. The Grasshopper Optimization Algorithm (GOA) was applied in the novel method for ESN improvement that was implemented[25].

The predicted result of the RUL was the goal of the algorithm's development and implementation. The study results demonstrate that the ESN, which includes computational improvements, succeeds higher than additional variables when weighed against other parameters[26]. contributed to designing and setting up a framework for optimizing ESN. The Better Fruit Fly Optimization Algorithm (IFOA) built the system. The improved system was developed to predict the traffic rate in the start phases. The algorithm presented a greater level of CA as it tried to predict the flow of traffic for the five minutes that followed. FS is dealt with by the Binary Improved Gravitational Search Algorithm (BIGSA), which is used by the BIGSA-ESN[27]. This system could address the challenge of redundant features in ESN. Using this method, the contributors achieved more excellent CA in typical and practical dataset's sustainable performance prediction challenges. As indicated by the recognized model, FS displayed how to minimize redundancy while effectively enhancing CA in prediction.

In their research[28], attempted to measure the results of RNN + ESN regarding how well they could predict test and fake cardiovascular applications. A summary of findings of the detections shows that the RNN + ESN could predict with precision behaviors that were tested 15 to 20 beats previously. However, the ESN revealed a significant decrease in the total time required to complete computation when the system allocated applications that were running in real-time. In taking advantage of Particle Swarm Optimization (PSO)[29], could improve their ESN to reach higher prediction ranges. As a result of this method, the authors' design functioned substantially superior to previous methods by predicting up to 10,000 steps in an advance period.

Furthermore, it used < 1% of neurons. Effective Hybrid Enhanced Echo State Network (HEESN) was shown by[30] as a method for solving the complicated and unpredicted nature of time series with multiple variables

cardiovascular signals. The authors validated the model they suggested by implementing multiple optimization techniques, such as initial FS, unknown data restoration, and results value in the normalization process. Tests frequently demonstrated how the put-forward hypothesis performed better than the original data set regarding CA in predictions. In their research, a distinct reservoir architecture theory called Simple-ESN (S-ESN) was presented by[31]. This knowledge was motivated by the α-helix design of biomolecules. Also, there is a probability that more thorough categories of ESNs will increase advantages from the improved reliability and precision in predicting that this bio-inspired model enables. They also exploited a fresh Moth-Flame Optimization Method (MFOM) involving more fine-tuning, causing a boost in CA when employed on updated time series samples—the hybrid Particle Swarm Optimization and Gradient Descent to optimize ESN[32] for time series prediction. The hybrid optimization model resulted in enhanced prediction CA and convergence speed.

The following works emphasized the requirement of FS in the diagnosis of CKD detection:

The technique[33] adopted an ensemble DL that fused several FS with an SVM that functioned as the meta-learner. This helped the technique to recognize CKD. This ensemble model that was developed using Mutual Information (MI)-FS is higher than all the additional models that underwent training with FS, and because of that, it has a prediction of nearly 90%. Implementing a Deep Belief Network (DBN) that was recently adapted[34], reported the results of a study on the diagnosis of CKD. Considering a level of sensitivity of 87.5% and a prediction CA of 98.5%, researchers could predict CKD accurately. The focus of this study was to prove how to use DL, such as DBN, to structure higher CDSS and improve the CA of the diagnosis of CKD. In order to deal with the drawbacks of the Local Search (LS) that is employed for the training process of Neural Networks (NN) for the recognition of CKD[35,36], have assisted in the findings by presenting definite methods. Also, the NN-GA + NN-CS was invented to predict CKD diagnosis problems and improve the NN's data value array by applying the Genetic Algorithm (GA) and Cuckoo Search (CS) correspondingly.

For the objective of further enhancing the collection of data for CKD prediction[37], developed a model that applied the use of an improved FS. During the study, they employed the Enhanced Decision Tree (EDT) to predict the potential risk of the CKD version to the restricted amount dataset. To select the most dynamic features from the assumed data, the researchers additionally employed the Recursive Feature Elimination (RFE). The conclusions of the research demonstrate that the methods that have been suggested improving the CA of the medical diagnosis of CKD. The DBN, which includes the SoftMax activation function and this Categorical Cross-entropy loss function, is used to comply with the demand for effective preclinical CKD prediction. This was performed to achieve the objectives of the research development. In line with the results of the research studies, the study's hypothesis was improved as it possessed a level of sensitivity of 87.5% and a CA of 98.5% when it involved the early detection of CKD.

Authors[38] introduced a method for identifying CKD cases subject to a successful hybrid approach. This model originated using SVM, ReliefF, and Principal Component Analysis (PCA). On a pair of distinct sets of test results, the prediction model attained a CA of 92.5% and 98.5% when using the decreased feature set; the ReliefF + PCA was applied for computing the feature values.

An ensemble FS combined with Improved Teacher Learner Based Optimization (ITLBO) was invented[39] to identify optimal features for CKD prediction higher CA.

Methods

Recurrent neural network (RNN)

An example of an Artificial Neural Network (ANN) is an RNN defined by developing a directed graph over a time-series. The RNN executes a step-by-step iteration of the specified input data series to analyze the sets of data that it collects into the system[40]. The RNN successfully refreshes the state of the hidden layer via data from the currently selected input and previous state data[41]. The Hidden State (HS) stores the network's memory to store the provided data, Eq. (1), which is the state updating factor.

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

where:

- h_t is the current HS,
- x_t is the current input,
- h_{t-1} is the previous HS,
- W_{xh} and W_{hh} are weights,
- b_h is a biased term,
- f is a non-linear activation function (e.g., tanh).

The output at each time step is computed based on the current state using another transformation, Eq. (2).

$$y_t = g\left(W_{hy}h_t + b_y\right) \tag{2}$$

where y_t is the output, W_{hy} is the weight matrix, b_y is a bias term, and g is an activation function. The RNN supports vanishing or exploding gradients during backpropagation, making learning long-term dependencies challenging.

Reservoir computing (RC)

RC is a type of RNN that addresses the challenges, particularly the vanishing and exploding gradient problems. The RC integrates a reservoir of an extensive, fixed, randomly initialized network of neurons. This neuron

reservoir acts as a memory system that captures and preserves the subtleties of the temporal sequences. Further, the internal weights of the network model remain unchanged during the training method, which avoids the complexities related to weight adjustment and eliminates the vanishing and exploding gradients issues.

Echo state networks

ESN, introduced by[42], is a subset of RC that leverages the reservoir for processing temporal information. ESN (Fig. 1) is organized into three essential layers: input, hidden, known as the reservoir, and output. Inputs are represented by a matrix with dimensions (V, T), where 'V' denotes the number of features and T the number of time steps.

Several essential variables categorize the network:

- *Reservoir Size* (N): Specifies the number of units within the hidden layer.
- *Sparsity* Indicates the fraction of non-zero entries in the reservoir's weight matrix (W), typically below 10%, to ensure sparse connectivity.
- *Input Weights* (W_{in}): Initialized from a binomial distribution, these weights connect the input layer to the reservoir and have dimensions $(N \times V)$.
- *Reservoir Weights* (W): Set over sampling from a uniform distribution, with dimensions $(N \times N)$ and adjusted density for desired sparsity.

(i) *Reservoir Dynamics*: Central to the ESN's ability to process temporal data is the reservoir's dynamic state update, governed by Eq. (3).

$$x\,(t) = f\,(W_{\text{in}} u\,(t) + W x(t-1)) \tag{3}$$

This equation governs the evolution of the reservoir's internal state $x\,(t)$ at time t.

It describes how the current state depends on:

(a) The input signal $(u(\,t\,))$, mapped into the reservoir using the input weight matrix (W_{in}).
(b) The previous state of the reservoir $(x(t-1))$, impacted by the internal networks of the reservoir, is represented by the weight matrix (W).

The nonlinear activation function $f\ (tanh)$ ensures that the state captures complex temporal relationships. This mechanism allows the reservoir to encode the history of inputs in its dynamic state, enabling the ESN to process temporal data efficiently without requiring iterative weight updates within the reservoir.

This transformation, facilitated by the input weights W_{in} and the internal reservoir weights 'W', enable the ESN to maintain a memory of input history, which is vital for its predictive capabilities.

(ii) *Training the Output Weights*: Unlike the input and reservoir weights, the output weights (W_{out}) are trainable. The training process includes adjusting. 'W_{out}' to map the reservoir states to the desired result. The output at time 't' is computed as Eq. (4).

$$\mathbf{y}\,(t) = \mathbf{W}_{\text{out}} \mathbf{x}\,(t) \tag{4}$$

where '$\mathbf{y}\,(t)$' is the result of the network at time 't' and $\mathbf{W}_{\text{out}}$ is the result weight matrix. Training W_{out} typically involves collecting the states of the reservoir '$\mathbf{x}\,(t)$' over the training set and using LR to find W_{out} that minimizes the difference between the network result and the actual selected result[43].

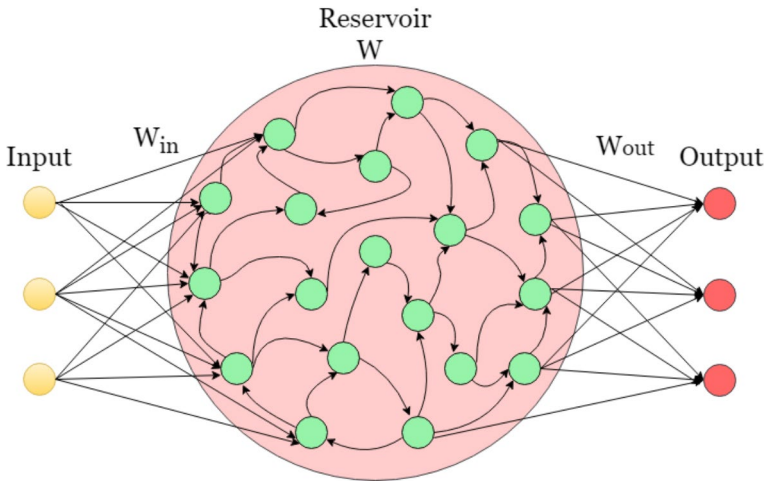

Fig. 1. ESN Model. The ESN consists of an input layer, a reservoir with sparsely connected neurons, and an output layer. Inputs are mapped to the reservoir via input weights (W_{in}), where temporal patterns are processed. The output layer uses trained weights (W_{out}) to generate predictions based on reservoir states.

Training Process for ESN involves using a sequence of training input-output pairs $(u(1), d(1)), \ldots, (u(T), d(T))$ to train an ESN considered by weights $[W_{\text{in}}, W, W_{\text{back}}, W_{\text{out}}]$. The aim is for the ESN's output $y(n)$ to closely match the target or teacher output $d(n)$ when provided with the corresponding input $u(n)$. In the initial phase, W_0, the matrix representing internal connections within the reservoir is caused randomly. The generated W_0 is then normalized to W_1, adjusting its spectral radius to 1. Then the W_1 is scaled to $W = \alpha W_1$, where 'α' is less than 1 to ensure stability. Then, the input weights W_{in} and the analysis weights W_{back} are also initialized randomly.

In the network preparation phase, the network's initial state, $x(0)$, is set to '0', and the training data $u(n)$ and the corresponding target results $d(n-1)$ are presented to the network. For the initial step, $d(0)$ is set to '0'. The network's state is updated iteratively for each time step from $n = 0$ to T using the function f, which includes the current input, the previous state, and the input from the previous target result. After an initial *"washout"* period T_0, the states $x(n)$ and targets $d(n)$ are collected. The resultant weights W_{out} are then computed by solving a linear system that maps the collected reservoir states to the target results. This is achieved using the pseudoinverse of the state matrix M multiplied by the target matrix T, resulting in W_{out}^T, which is then transposed to obtain W_{out}. The trained network[44] then processes the new input sequences $u(n)$ for prediction. The network updates its state with these new inputs and the previously predicted outputs, using the learned weights to generate new result predictions $y(n+1)$. The entire flow of the training process is presented in the following Algorithm 1:

Inputs:

- A sequence of training input-output pairs $(u(1), d(1)), \ldots, (u(T), d(T))$.
- **Network variables:** reservoir size N, input as V, output as L, sparsity s, and feedback scale α.

Outputs:

- A trained ESN with weights $W_{\text{in}}, W, W_{\text{back}}, W_{\text{out}}$.

Algorithm:

1 Initialization:

- Generate W_0 for internal connections randomly.
- Normalize W_0 to get W_1 with a unit spectral radius.
- Scale W_1 to $W = \alpha W_1$, ensuring a spectral radius of α.
- Randomly initialize W_{in} and W_{back}.

2 Network Preparation:

- Set initial network state $x(0) = 0$.
- Introduce training $u(n)$ and target result $d(n-1)$, with $d(0) = 0$, to update network states.

3 State Update and Collection:

- For $n = 0$ to T, update state, EQU (5)

$$x(n+1) = f\left(W_{\text{in}}\, u(n+1) + Wx(n) + W_{\text{back}}\, d(n)\right) \tag{5}$$

- Collect states $x(n)$ and target $d(n)$ post-washout period T_0 in matrices M and T.

4 Compute Output Weights (W_{out}):

- Solve for W_{out}^T using EQU (6)

$$\left(W_{\text{out}}^T\right) = M^{-1}T \tag{6}$$

- Transpose to get W_{out}.

5 Exploitation:

- With the network now trained, apply new input sequences $u(n)$ for prediction.
- Update state with new inputs and previously predicted outputs EQU (7)

$$x(n+1) = f\left(W_{\text{in}}\, u(n+1) + Wx(n) + W_{\text{back}}\, y(n)\right) \tag{7}$$

- Predict new outputs EQU (8)

$$y(n+1) = \sigma\left(W_{\text{out}}\,(u(n+1), x(n+1))\right) \tag{8}$$

Algorithm 1. For Training Process of ESN.

PSO algorithm

PSO[45] symbolizing the social behaviour experiential in natural swarms[46]. It aims to find the optimal solutions in a multidimensional space by adjusting the positions and velocities of a group of particles.

Considering a problem within a D-dimensional space, a swarm of size N is defined as $Swarm = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N\}$, where each particle $\mathbf{X}_i$ (for $i = 1$ to N) is a mass-less entity operating within this space. Let $T_{\max}$ denote the total iterations required for the optimization process. The state of the i^{th} particle at the t^{th} iteration in this D-dimensional space is represented by the position vector $\mathbf{X}_i^t = \left(x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t\right)$, and its velocity vector is denoted by $\mathbf{V}_i^t = \left(v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t\right)$. In each iteration, the dynamics of the particles are ruled by their velocities and positions, which are updated based on the personal best positions and the global best position observed so far.

(i) *Velocity Update*: At all stages, the velocity of every single object is updated depending on its data and the collective information of the swarm.

$$\mathbf{V}_i^{t+1} = \mathbf{V}_i^t + c_1 \cdot \text{rand}\,() \cdot \left(\mathbf{P}_i^t - \mathbf{X}_i^t\right) + c_2 \cdot \text{rand}\,() \cdot \left(\mathbf{X}_{\text{gbest}}^t - \mathbf{X}_i^t\right) \tag{9}$$

where $V_i^{(t)}$ represents the current velocity of the particle and P_i and X_{gbest} are the personal best and global best positions, respectively. This Eq. (9) combines three components: the particle's current velocity (promoting inertia), the attraction to its best-known position (encouraging personal exploration), and the attraction to the global best position (leveraging collective knowledge). The randomness introduced by *rand()* ensures diversity in the swarm's search, preventing premature convergence to local optima.

(ii) *Position Update*: Directly succeeding in the change of the velocity, the current location of each object changes by applying its fresh velocity to the location it is presently in. This step simulates the movement of particles through the search space Eq. (10).

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + \mathbf{V}_i^{t+1} \tag{10}$$

where:

- c_1 and c_2 are learning factors controlling the impact of the personal and global bests.
- rand () is a function generating a random number between 0 and 1.
- $\mathbf{P}_i^t$ is the best position of the i^{th} particle until iteration 't'.
- $\mathbf{X}_{\text{gbest}}^t$ Shows the best position that each object in the swarm identified during cycle 'X'.

The performance of PSO is susceptible to its parameters, including the number of particles 'N', learning factors $\{c_1; c_2\}$, and the inertia weight. Experimentation knowledge may guide the tuning of these parameters for optimal performance.

Proposed methodology

Adaptive particle swarm optimization (APSO)

This proposed APSO research aims to improve the standard PSO's capability to maintain a balanced exploration and exploitation by the optimization method. For the problem within a D-dimensional space considered in PSO[47], the swarm of particles is initialized with positions $\mathbf{X}_i$ and velocities $\mathbf{V}_i$ randomly distributed within the search space limits.

Each particle 'i' has a position vector $\mathbf{X}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and a velocity vector $\mathbf{V}_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, where $\boldsymbol{D}$ is the dimensionality of the search space. The Dynamic Inertia Weight (w) is adapted at each iteration t to balance exploration and exploitation. It is done by decreasing w linearly from $w_{\max}$ to $w_{\min}$ over iterations, Eq. (11).

$$w\,(t) = w_{\text{Max}} - \left(\frac{w_{\text{Max}} - w_{\text{Min}}}{T_{\text{Max}}}\right) t \tag{11}$$

where $T_{\max}$ is the *Max()* iterations, the personal learning coefficient (c_1) and the social learning coefficient (c_2), are dynamically used in response to the swarm's performance. If there is detected stagnation in the improvement of the global best solution, 'c_1' is incremented to enhance the exploration capabilities of the swarm[48]. Conversely, if there is a need to intensify the exploitation of the swarm's collective data, then 'c_2' is modified. For particles not showing improvement over 'Δt' iterations, reset velocities to boost the study of new areas, Eq. (12).

$$\mathbf{V}_i = \text{rand}\,() \cdot (\mathbf{X}_{\text{gbest}} - \mathbf{X}_i) \tag{12}$$

This directs stagnant particles toward the current global best position $\mathbf{X}_{\text{gbest}}$. At each iteration '$t$', update the velocity of each particle using the adaptive inertia weight, personal best position $\mathbf{P}_i$, and global best position $\mathbf{X}_{\text{gbest}}$, Eq. (13).

$$\mathbf{V}_i(t+1) = w\,(t) \cdot \mathbf{V}_i\,(t) + c_1 \cdot \text{rand}\,() \cdot (\mathbf{P}_i - \mathbf{X}_i\,(t)) + c_2 \cdot \text{rand}\,() \cdot (\mathbf{X}_{\text{gbest}} - \mathbf{X}_i\,(t)) \tag{13}$$

Update the position of each particle by adding the new velocity, Eq. (14)

$$\mathbf{X}_i(t+1) = \mathbf{X}_i\,(t) + \mathbf{V}_i(t+1) \tag{14}$$

Next, each particle's fitness is computed based on its updated position. Then, the personal best $\mathbf{P}_i$ is updated if the current position generates a better fitness value. Update the global best $\mathbf{X}_{\text{gbest}}$ if any particle achieves a better fitness than the current global best. The optimization method continues until the termination criterion $T_{\max}$is referred. Algorithm 2 presents the complete flow of the proposed APSO[49].

Input: Objective function $f(\mathbf{X})$, Dimension 'D', Swarm size 'N', *Max()* iterations

T_{Max}, w_{Max}, w_{Min}, Initial $\{c_1; c_2\}$

Output: The best solution $\mathbf{X}_{\text{gbest}}$ and its fitness

1 **Initialization:**

- **For Each** particle $i = 1$ to N :
- **Initialize** position $\mathbf{X}_i$ and velocity $\mathbf{V}_i$ randomly within the search space.
- **Evaluate** the fitness of each particle: $f(\mathbf{X}_i)$.
- **Set** personal best $\mathbf{P}_i = \mathbf{X}_i$.
- **Determine** the global best position $\mathbf{X}_{\text{gbest}}$ from the initial population.

2 **For** $t = 1$ to T_{max} **do:**

- **Adapt Parameters:**
  - Fine-tune 'w' dynamically using EQU (11).
  - Adapt $\{c_1; c_2\}$ based on the optimization state or performance metrics.
- **For Each** particle $i = 1$ to N **Do:**
  - **Velocity Update** using adaptive $\{w, c_1, c_2\}$, EQU (13)
  - **Position Update, EQU (14)**
  - **Evaluate** the fitness of $\mathbf{X}_i(t+1)$.
  - **Update** Personal Best if necessary.
- Update Global Best $\mathbf{X}_{\text{gbest}}$ if any particle finds a better position.
- Velocity Reset Mechanism: For particles not improving over 'Δt', reset $\mathbf{V}_i$ to encourage exploration.

3 **End For**

4 **Return** $\mathbf{X}_{\text{gbest}}$ and its fitness.

Algorithm 2. For APSO

Balancing spectral radius using random matrix theory for enhancing ESN

Balancing spectral radius is crucial for better ESN performance because ESN with a lower spectral radius will fail the long-term memory capability; conversely, the higher spectral radius results in performance unpredictability and overfitting. So, it is necessary to balance the suitable setup of spectral radius. To ensure this rightful balance, the Random Matrix Theory (RMT) is employed; the RMT employs the model to analyze the eigenvalue distribution of large random matrices to enhance the reservoir with optimal spectral features.

(i) *Spectral Radius and Stability*: The spectral radius 'ρ' of a matrix is defined as the *Max()* absolute value of its eigenvalues. In ESN, the spectral radius of the reservoir weight matrix 'W' determines the network's dynamical behaviour, Eq. (15)

$$\rho(W) = \text{Max}\,|\lambda_i|\,, \tag{15}$$

where 'λ_i' are the eigenvalues of 'W'. For a matrix W of size $N \times N$, where the entries are tired from a random distribution with mean '*0*' and variance 'σ^2', the Marchenko-Pastur Law provides a way to understand the distribution of eigenvalues.

(ii) *Marchenko-Pastur Law*: The MPL[50] applies to matrices with independent, identically distributed (i.i.d.) entries, where the sum of rows and columns tends to infinity, but their ratio remains constant.

Consider a large $N \times M$ matrix X, where N represents the number of rows and M the number of columns, and the entries of X are i.i.d. random variables with mean '0' and variance 'σ^2'. Define the feature ratio of the matrix as $q = N/M$, with $N, M \to \infty$ but q remaining constant. The empirical distribution of the eigenvalues of the matrix XX^T (or = X^TX) converges almost surely to a deterministic measure known as the Marchenko-Pastur distribution.

The density of this distribution is given by the MPL, Eqs. (16), and (17).

$$\rho(\lambda) = \frac{1}{2\pi\sigma^2}\frac{\sqrt{(\lambda_{\max} - \lambda)(\lambda - \lambda_{\min})}}{q\lambda for}\lambda \in [\lambda_{\min}, \lambda_{\max}] \tag{16}$$

where

$$\lambda_{\min} = \sigma^2(1 - \sqrt{q})^2, \lambda_{\max} = \sigma^2(1 + \sqrt{q})^2 \tag{17}$$

where q is the aspect ratio of the matrix dimensions. By scaling W to ensure its eigenvalues fall within these limits, the spectral radius is adjusted to achieve an optimal balance between memory size and stability. This optimization is essential for processing temporal data effectively while maintaining computational efficiency.

APSO-based echo state network (APSO-ESN)

The proposed APSO-ESN uses APSO to optimize each sub-reservoir in ESN before adding it to the network. The APSO was selected over other optimization due to its adaptive capabilities, which balance exploration and exploitation during optimization. This generates it mainly effective for addressing the challenges in ESN parameter tuning, such as avoiding local optima, optimizing the spectral radius for stability, and reducing the computational cost. APSO's dynamic adjustment of parameters like inertia weight and learning coefficients ensures efficient convergence while maintaining a diverse search space, making it suitable for the complex, high-dimensional optimization required in ESN training.

After the integration of the required sub-reservoirs, the weights of the result of the network are determined in one instance, resulting in a significant decrease in the computation cost. The enhancement process starts with constructing the sub-reservoir using the MPL[51].

(i) Constructing the Sub-Reservoir Using MPL.

Using MPL[52], this step constructs the sub-reservoir. This step ensures that the spectral radius and density of the reservoir weight matrix are selected to balance memory capacity, stability, and sensitivity to inputs.

The steps involving the matrix structure u*sing the MPL are presented below*.

1 *Initial Weight Matrix Setup*: The matrix structure starts with setting the sub-reservoir's weight matrix W, Eq. (18).

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix} \tag{18}$$

where N is the number of neurons in the sub-reservoir, making W an $N \times N$ matrix. Each element w_{ij} Within W is initialized from a distribution with mean 0 and variance σ^2, following the principles of Random Matrix Theory (RMT).

2 *Determination of Feature Ratio*: The feature ratio $q = N/M$ is calculated, where N is the sum of rows (units in the sub-reservoir), and M is the sum of columns (connections per unit).
3 *Application of Marchenko-Pastur Distribution*: Using the MPL, the eigenvalue distribution of WW^T (or $= W^TW$) is determined as $N, M \to \infty$ with a constant ratio 'q'. The limits characterize the distribution $\lambda_{\min} = \sigma^2(1 - \sqrt{q})^2$ and $\lambda_{\max} = \sigma^2(1+ \sqrt{q})^2$, where '$\lambda$' ranges between these limits.
4 *Spectral Radius Optimization*: The spectral radius $\rho(W)$ is adjusted by scaling 'W' to fit within the desired bounds determined by the Marchenko-Pastur Distribution[53]. The goal is to set $\rho(W)$ to a value that optimizes the balance between memory capacity and stability, frequently requiring $\rho(W)$ to be <1 for stability in ESN contexts.

(ii) Fitness Function:

Given P training instances, the fitness of the i^{th} sub-reservoir, which comprises m_iNeurons employ a sigmoid activation function determined by its capacity to minimize the prediction error[54]. The state matrix for this sub-reservoir is denoted as $S_{(P\times m_i)} = [s_1, s_2, \ldots, s_{m_i}]$, where each s_j denotes the state vector in response to the input sequence. Define the state space of the i^{th} sub-reservoir as $U_i = \text{span}(s_1, s_2, \ldots, s_{m_i})$. The discrepancy between the preceding sub- reservoir's error vector $f_{(i-1)}$ (with f_0 being the target vector) and U_i is given by Eq. (19).

$$d\left(f_{(i-1)}, U_i\right) = \inf_{\theta \in U_i} \left\| f_{(i-1)} - \theta \right\| \tag{19}$$

Apply a modified Gram-Schmidt process to the vectors $s_1, s_2, \ldots, s_{m_i}$ to attain orthonormal vectors $\eta_1, \eta_2, \ldots, \eta_{m_i}$, these vectors η_j form an orthonormal basis for U_i. This means that any vector $\theta \in U_i$ can be expressed as a linear integration of these orthonormal vectors, Eq. (20).

$$\theta = \sum_{j=1}^{m_i} c_j \eta_j \tag{20}$$

where c_j are the coefficients for the linear combination.

The advantage of having an orthonormal basis is that these coefficients can be easily calculated as Eq. (21).

$$c_j = \langle \theta, \eta_j \rangle \tag{21}$$

for $j = 1,2,\ldots,m_i$, where $\langle\cdot,\cdot\rangle$ denotes the inner product of two vectors. This is due to the orthonormality of the η_j's, where $\langle\eta_j,\eta_k\rangle = \delta_{jk}$, and δ_{jk} is the Kronecker delta, which is 1 if $j = k$ and 0 otherwise. The discrepancy $d\left(f_{(i-1)},U_i\right)$ can now be minimized by projecting $f_{(i-1)}$ onto the subspace U_i.

The prediction of $f_{(i-1)}$ onto U_i is given by Eq. (22)

$$P_{U_i}\left(f_{(i-1)}\right) = \sum_{j=1}^{m_i}\left\langle f_{(i-1)},\eta_j\right\rangle\eta_j \tag{22}$$

This prediction minimizes the distance between $f_{(i-1)}$ and any vector in U_i, effectively solving for θ that minimizes $\left\|f_{(i-1)}-\theta\right\|$. The minimized discrepancy $d\left(f_{(i-1)},U_i\right)$ is then the norm of the difference between $f_{(i-1)}$ and its prediction onto U_i, Eq. (23)

$$d\left(f_{(i-1)},U_i\right) = \left\|f_{(i-1)} - P_{U_i}\left(f_{(i-1)}\right)\right\| \tag{23}$$

After computing the discrepancy for the i^{th} sub-reservoir, the process can be constant for subsequent sub-reservoirs by updating the error vector $f_{(i)}$ to be the discrepancy vector $f_{(i-1)} - P_{U_i}\left(f_{(i-1)}\right)$ and it is then moving to the next sub-reservoir. This iterative process continues until all sub-reservoirs have been processed, which allows for a hierarchical decomposition of the target vector T across the network of sub-reservoirs.

The fitness value [56] is designed to minimize the prediction of the error vector $f_{(i-1)}$ onto the subspace spanned by η_j, is quantified by the norm of the difference between the original error vector and its prediction onto the subspace. This is related to the minimized discrepancy $d\left(f_{(i-1)},U_i\right)$.

Mathematically, the fitness Φ_i for the i^{th} sub-reservoir can be defined as follows: Eq. (24).

$$\Phi_i = -\left\|f_{(i-1)} - P_{U_i}\left(f_{(i-1)}\right)\right\|^2 \tag{24}$$

where $f_{(i-1)}$ is the error vector passed from the previous sub-reservoir, and $P_{U_i}\left(f_{(i-1)}\right)$ is its projection onto the subspace U_i. The fitness function evaluates how well the i^{th} sub-reservoir represents the target vector. By projecting the error vector onto the subspace spanned by the state vectors of the sub-reservoir, the discrepancy between the original error vector and its projection is minimized. This ensures that each sub-reservoir contributes meaningfully to reducing the overall prediction error. To compute this, a modified Gram-Schmidt process is applied to orthogonalize the state vectors, creating an orthonormal basis for the subspace. This orthonormal basis simplifies the prediction method and ensures that the sub-reservoir effectively captures the residual information, optimizing the overall network's performance.

(iii) APSO Optimized ESN Training.

Using APSO to optimize ESN training starts with pretraining for Spectral Radius Optimization using APSO (Fig. 2). In this stage, a swarm of particles is deployed, each representing a possible configuration of the ESN parameters. These particles explore the parameter space, guided by a fitness function that evaluates the ESN's performance by its prediction CA based on the current parameter settings. Each particle's position reflects a specific spectral radius value among other reservoir parameters, and its fitness is computed by how well the ESN, configured with these parameters, performs on a set of training data. The APSO dynamically adjusts the particles' positions (i.e., parameter configurations) across iterations, focusing on those adjustments that improve the ESN's performance (Algorithm 3).

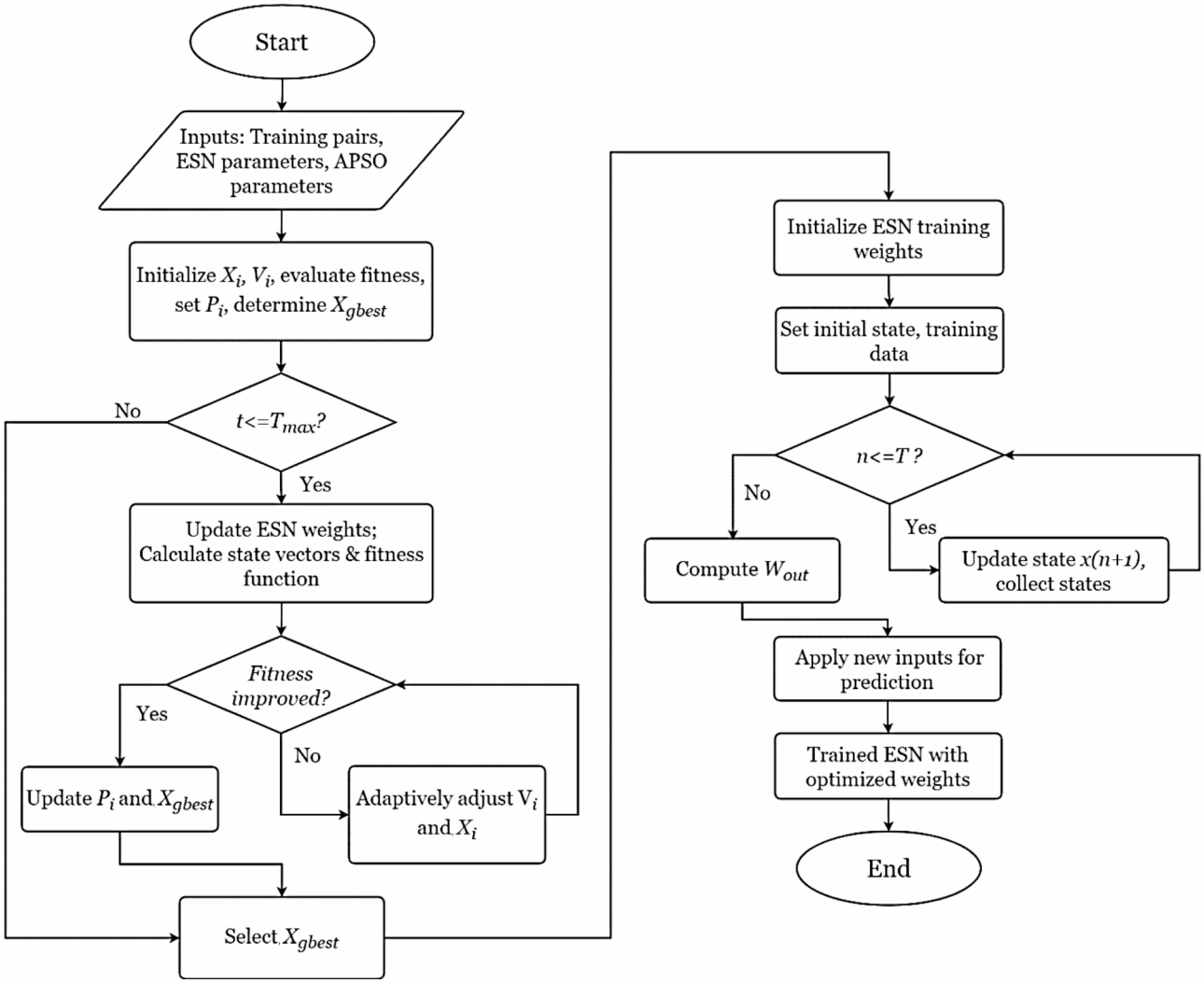

Fig. 2. APSO-based ESN Flowchart. The flowchart illustrates the training process of ESN optimized using APSO. The left section represents the APSO optimization of ESN parameters, while the true section depicts the ESN training and prediction method using the optimized weights. This approach ensures enhanced CA and efficiency in CKD prediction.

Input:
- A sequence of training input-output pairs $(u(1), d(1)), \ldots, (u(T), d(T))$.
- **ESN network parameters:** reservoir size N, input dimension V, output dimension L, sparsity s, and feedback scale α.
- APSO parameters: swarm size N, maximum iterations $T_{\max}$, inertia weight bounds $w_{\max}, w_{\min}$, and learning coefficients c_1, c_2.

Output:
- A trained ESN with optimized weights $W_{\text{in}}, W, W_{\text{back}}, W_{\text{out}}$.

Algorithm Steps:

1 APSO Pretraining for Spectral Radius Optimization:
- **Initialization:**
  - **For Each** particle $i = 1$ to N, initialize X_i and V_i randomly within the search space.
  - **Evaluate** the initial fitness of each particle using the ESN's performance criterion $f(X_i)$.
  - Set $P_i = X_i$ as the personal best and determine X_{gbest} from the initial population.
- **Fitness Calculation:**
  - **For** $t = 1$ to $T_{\max}$, repeat:
    - **For Each** particle, update the ESN sub-reservoir weight matrix using the MPL based on X_i.
    - Compute state vectors $s_1, s_2, \ldots, s_{m_i}$ for the updated sub-reservoir.
    - Orthonormalize state vectors to $\eta_1, \eta_2, \ldots, \eta_{m_i}$.
    - Apply the fitness function $\Phi_i = -\left\| f_{(i-1)} - P_{U_i}\left(f_{(i-1)}\right)\right\|^2$.
    - Update P_i and X_{gbest} based on fitness improvements.
  - Finalize by selecting X_{gbest} as the optimized spectral radius configuration.

2 ESN Training with Optimized Parameters:
- **Initialization:**
  - Generate W_0 for internal networks randomly and normalize to get W_1 with a unit spectral radius.
  - Scale W_1 to $W = \alpha W_1$ using the optimized spectral radius from APSO.
  - Randomly initialize W_{in} and W_{back}.
- **Network Preparation:**
  - Set initial network state $x(0) = 0$.
  - Introduce training data $u(n)$ and target output $d(n-1)$ to update network states.
- **State Update and Collection:**
  - For $n = 0$ to T, update state using $x(n+1) = f\left(W_{\text{in}}\, u(n+1) + Wx(n) + W_{\text{back}}\, d(n)\right)$.
  - Collect states $x(n)$ and targets $d(n)$ in matrices M and T.
- **Compute Output Weights W_{out} :**
  - Solve $W_{\text{out}}^T = M^{-1}T$ and transpose to obtain W_{out}.
- **Exploitation:**
  - Apply new input sequences for prediction, updating the state with $x(n+1)$ and predicting new outputs $y(n+1)$.

3 Output
- A trained ESN with optimized weights $W_{\text{in}}, W, W_{\text{back}}, W_{\text{out}}$

4 End If

Algorithm 3. For APSO-based ESN Training

Experiment analysis

Dataset

The MIMIC-III (Medical Information Mart for Intensive Care III) dataset is a comprehensive, publicly available database that has become an invaluable resource for researchers in healthcare analytics, medical informatics, and critical care. Developed by the MIT Lab for Computational Physiology and made available through PhysioNet, the dataset encompasses a wide array of de-identified health-related information collected from patients admitted to critical care units at the Beth Israel Deaconess Medical Center in Boston, Massachusetts, between 2001 and 2012.

The MIMIC-III dataset features the following key features:

(a) *Patient Demographics*: In order to promote the study of public health and imbalances in healthcare results, it is vital to have data on gender, ethnicity, age, and other demographic factors.
(b) *Clinical Data*: It is feasible to examine the development of the disease, the results of treatment, and the Clinical Decision-Making System (CDMS) processes over comprehensive data involving tests achieved in laboratories, indicators of health, drugs, clinical scripts, and additional data.
(c) *Notes and Reports*: Significant data from medical notes, doctor prescriptions, patient discharge summaries, and reports from radiology are required for tasks requiring natural language processing (NLP) and qualitative evaluations.
(d) *Temporal Data*: Records of the administration of drugs, test results, and body sign data are recorded in time, which allows for studies into temporal pattern recognition, lively health state simulation, and time-series analysis.

Class labeling using ICD code for CKD prediction

In this study on CKD prediction, we implemented a binary classification system based on the presence of specific ICD-10 codes inpatient data. Individuals with records containing any of the following ICD-10 codes were classified as having CKD (assigned a binary value of 1), indicating various stages of the disease:

(a) *N18.1*: CKD, Stage I, considered by kidney failure with normal or increased GFR (≥ 90 *mL/Min*).
(b) *N18.2*: CKD, Stage II (mild), denoted by kidney failure with a mildly decreased GFR (60–89 *mL/Min*).
(c) *N18.3*: CKD, Stage III (moderate), involving kidney failure with a moderately decreased GFR (30–59 *mL/Min*).
(d) *N18.4*: CKD, Stage IV (severe), marked by kidney failure with a severely decreased GFR (15–29 *mL/Min*).
(e) *N18.5*: CKD, Stage V, indicating kidney failure (GFR < 15 *mL/Min/1.73* m^2) or patients on dialysis.

Patients without these codes were categorized as non-CKD (assigned a binary value of 0). The dataset exhibits a significant imbalance between CKD and non-CKD cases. The final cohort contained 11,752 patient records that met all the specified requirements for the study on CKD prediction, of which 1369 patients were identified with CKD, and the rest were non-CKD.

The following criteria guided the selection of patient records from the MIMIC-III dataset:

(a) *ICU Stay Duration*: Only patients with an ICU stay of at least 2 days were included.
(b) *Data Completeness*: Inclusion required complete records of lab results and vital signs.
(c) *Ethnicity Information*: Patients with documented ethnicity were selected.

This study's window was set to include data from the first 48 h after ICU admission. Within the defined window, we collected data points, including lab results, vital signs, and other relevant clinical information.

Feature selection

Out of 61 features, the proposed FS identified 23 key features as the most relevant for CKD prediction. These FS are classification based on their clinical importance and data type (Table 1).

Handling missing values and outliers

(i) *Missing Value*: Missing data impacts the CA of predictive models by reducing statistical impact and hypothetically introducing bias.

The missing data in the dataset was addressed using the following approaches:

(a) *Imputation*: For features with a lower percentage of missing values, imputation methods were used to fill in the gaps. Continuous variables were imputed with the median value of the feature, while definite variables used the mode.
(b) *Exclusion*: In cases where a feature had a maximum percentage of missing data, which could not be reliably assigned without introducing bias, the researcher selected to exclude the feature from the model. This decision was based on a threshold set by analyzing the impact of missing data on model performance and interpretability.
(c) *Advanced Techniques*: For life-threatening features with missing values that were too important to exclude but challenging to impute using rudimentary methods, we applied techniques such as k-NN imputation.

(ii) *Managing Outliers*: Outliers can change predictions and affect the model's ability to generalize from the training data to hidden data.

This proposed model outlier management approach involved:

- *Detection*: This study used statistical methods such as the Interquartile Range (IQR) and Z-scores to identify outliers in the dataset.
- *Treatment*: For outliers deemed errors or noise, we applied capping (using the 1st and 99th percentiles) and transformation techniques (e.g., Log Transformation) to minimize their impact.

Data and class imbalance

A class imbalance develops when the sum of instances in a single group significantly exceeds that of another group. This is reflected in models biased towards the class that forms most cases. The MIMIC-III had a higher imbalance with fewer non-CKD records than CKD records.

Initial analysis revealed that only a small fraction of the patient records was classified as CKD based on the specified ICD-10 codes. This imbalance poses a risk of model bias, where the predictive algorithm might

Category	Features
KF and Metabolic Indicators (MI)	Specific Gravity, Albumin, Creatinine, Blood Urea Nitrogen (BUN), eGFR (Estimated Glomerular Filtration Rate), Urine Protein Qualitative, Sodium, Potassium, Calcium, Phosphate, Sugar, Blood Urea, Protein
Blood Composition (BC)	Blood Glucose, Hemoglobin, Packed Cell Volume (Hematocrit), White Blood Cell (WBC), Red Blood Cell (RBC), C-Reactive Protein
Chronic Conditions (CC)	Hypertension, Diabetes Mellitus, Anemia

Table 1. Classification of FS.

overly favor the majority class (non-CKD), resulting in poor sensitivity for detecting CKD cases. We utilized oversampling of the minority class (CKD patients) and undersampling of the majority class (non-CKD patients) to balance the dataset. Oversampling techniques, such as the Synthetic Minority Over-sampling Technique (SMOTE), were particularly useful in generating synthetic instances of the minority class, enriching the dataset without losing valuable information.

While the MIMIC-III and preprocessing steps provided a robust foundation for CKD prediction, certain features security attention. The dataset, sourced from a single medical center, may limit the generalizability of results across assorted populations. While addressing class imbalance, techniques like SMOTE may introduce synthetic data objects. Additionally, excluding records with significant missing values might have presented selection bias. The 48-hour ICU window was a deliberate design optimal for motivation on early prediction; however, it limits insights into long-term CKD progression. Lastly, the dataset's absence of genetic, environmental, and lifestyle variables restricts a more comprehensive analysis of CKD risk factors. Future studies should address these aspects to enhance model applicability and reliability.

Model evaluation

The dataset was segregated into training and testing sets in an 80:20 split (nearly 9,402 and 2,350 records). The proposed APSO Optimized ESN is trained using a set of hyperparameters summarized in (Table 2).

The ESN is defined by a reservoir of 500 neurons processing 23 input features for binary classification. The reservoir's connectivity is maintained sparse with a sparsity value of 0.1, while a feedback scale of 0.3 regulates the impact of previous states. This study deploys a failure period of 50 steps to ensure stable reservoir dynamics, permitting the network to reach a steady state before training. The hyperbolic '*tan*' function is the activation function, introducing necessary non-linearity to the system.

The weight initialization ranges are carefully chosen, with input weights controlled between − 0.1 and 0.1 to prevent input overload and reservoir weights between − 1 and 1 to maintain the echo state feature. A small noise level of 0.0001 is introduced during training to enhance the model's robustness and generalization capability.

This study applies a swarm of 50 particles evolving over 100 iterations for the APSO component. The inertia weights decrease linearly from 0.9 to 0.4, balancing the exploration and exploitation phases and learning coefficients of 2.0 guide particles based on personal and social experience, while velocity bounds of [−4.0, 4.0] prevent redundant particle movement. This parameter configuration ensures effective optimization while maintaining computational efficiency.

In this study, five-fold cross-validation is applied for the metrics such as CA, precision, recall, Specificity, and F1-score.

- *True Positive (TP)*: The positive class is precisely predicted using archetypal methods.
- *True Negative (TN)*: The model precisely predictions the negative class.
- *False Positive (FP)*: The model causes an incorrect prediction about the positive class.
- *False Negative (FN)*: The algorithm incorrectly predicts the negative class.

(a) *Accuracy*: Measures the proportion of correctly predicted observations to the total observations Eq. (25).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{25}$$

(b) *Precision*: Evaluate the percentage of TP predictions among all optimistic predictions made by the model Eq. (26).

Hyperparameter	Value/range
Reservoir Size (N)	500
Input Dimension (V)	23 (based on selected features)
Output Dimension (L)	1 (binary classification)
Sparsity (s)	0.1
Feedback Scale (α)	0.3
Washout Period (T_0)	50-time steps
Activation Function (f)	tanh
Input Weight Range (W_in)	[−0.1, 0.1]
Reservoir Weight Range (W)	[−1, 1]
Noise Level	0.0001
Swarm Size	50
Maximum Iterations (T_max)	100
Inertia Weight Bounds (w_max, w_min)	[0.9, 0.4]
Learning Coefficients (c_1, c_2)	[2.0, 2.0]
Velocity Bounds (v_max, v_min)	[−4.0, 4.0]

Table 2. Hyperparameters.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{26}$$

(c) *Recall (Sensitivity)*: Measures the proportion of TP correctly identified by the model, Eq. (27).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{27}$$

(d) *F1-score*: In the context of precision and recall, the F1-Score is the harmonic mean of both Eq. (28).

$$F1_{\text{score}} = 2 * \frac{\text{Precision} - \text{Recall}}{\text{Precision} + \text{Recall}} \tag{28}$$

(e) *Specificity*: Specificity measures the model's ability to find TN correctly and is calculated using Eq. (29).

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{29}$$

Using the above metrics, The following ML is used to compare with the proposed APSO + ESN model: FFNN, RNN, LSTM, GRU, ESN, PSO, APSO (Proposed).

Results and analysis

The comparison results in Fig. 3 applied the complete set of FS for its relevance to CKD and non-CKD classification, focusing on precision, recall, and F1-score metrics. The analysis demonstrated that the APSO + ESN improved precision by 0.61%, recall by 1.33%, and F1-score by 0.91% over the standalone ESN for CKD classification. These results indicate the model's improved capability in accurately predicting CKD when APSO is used to optimize the ESN parameters. When comparing APSO against standard PSO, APSO presented a precision increase of 2.07%, recall by 2.75%, and F1-score by 7.21%. This comparison highlights APSO's effectiveness in optimizing model parameters, leading to better performance in CKD prediction.

Analyzing the impact of ignoring the KF and MI classes, as detailed in Fig. 4, shows findings regarding the predictive models' reliance on renal-specific parameters for CKD classification. The APSO + ESN illustrations a precision of 0.873, recall of 0.801, and F1-score of 0.812 for CKD classification. The results show a decrease in performance compared with the complete set of FS. For non-CKD classification, the proposed model achieved a precision of 0.836, recall of 0.811, and F1-score of 0.838, which, while still leading among the models evaluated, suggests a diminished capacity to differentiate non-CKD instances accurately without these indicators.

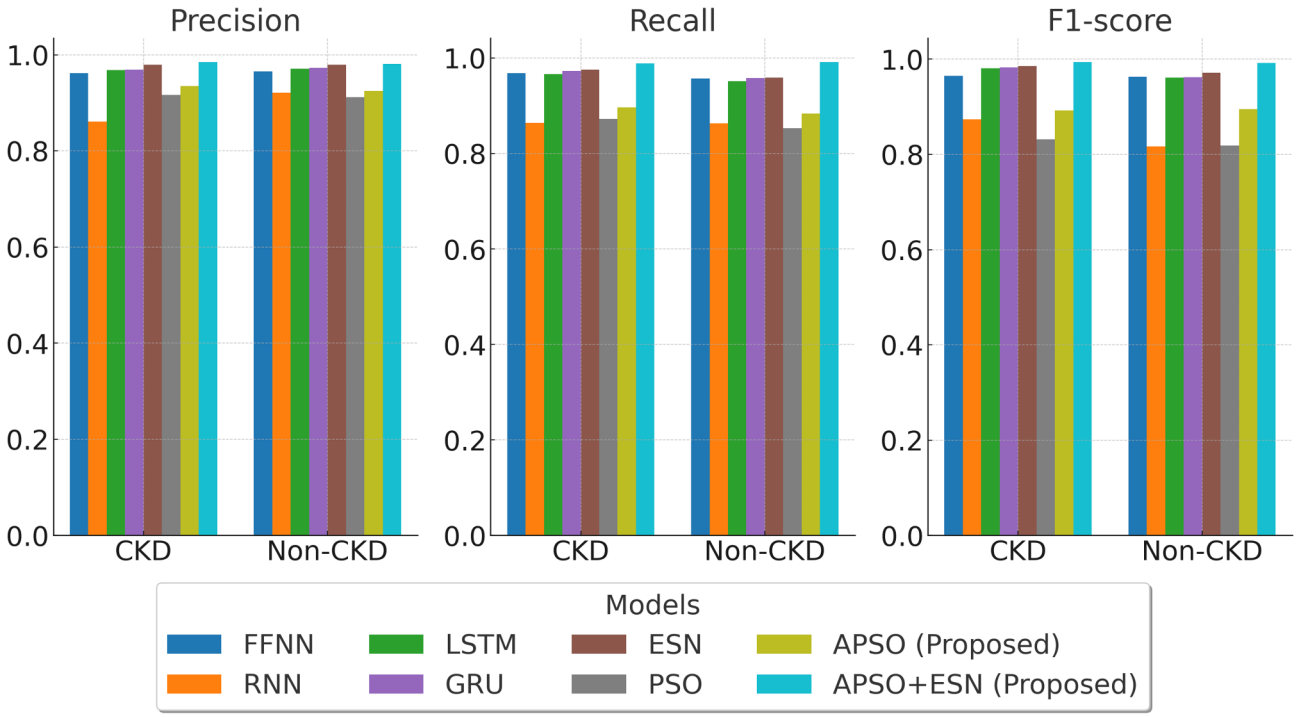

Fig. 3. Comparison of ML models' performance in CKD classification using all FS (Kidney Function, Blood Composition, and Chronic Conditions). The proposed APSO + ESN outperforms other models (e.g., FFNN, RNN, LSTM, GRU, PSO) in precision, recall, and F1-score metrics, demonstrating its superior capability to record temporal dependencies and optimize CKD prediction CA effectively.

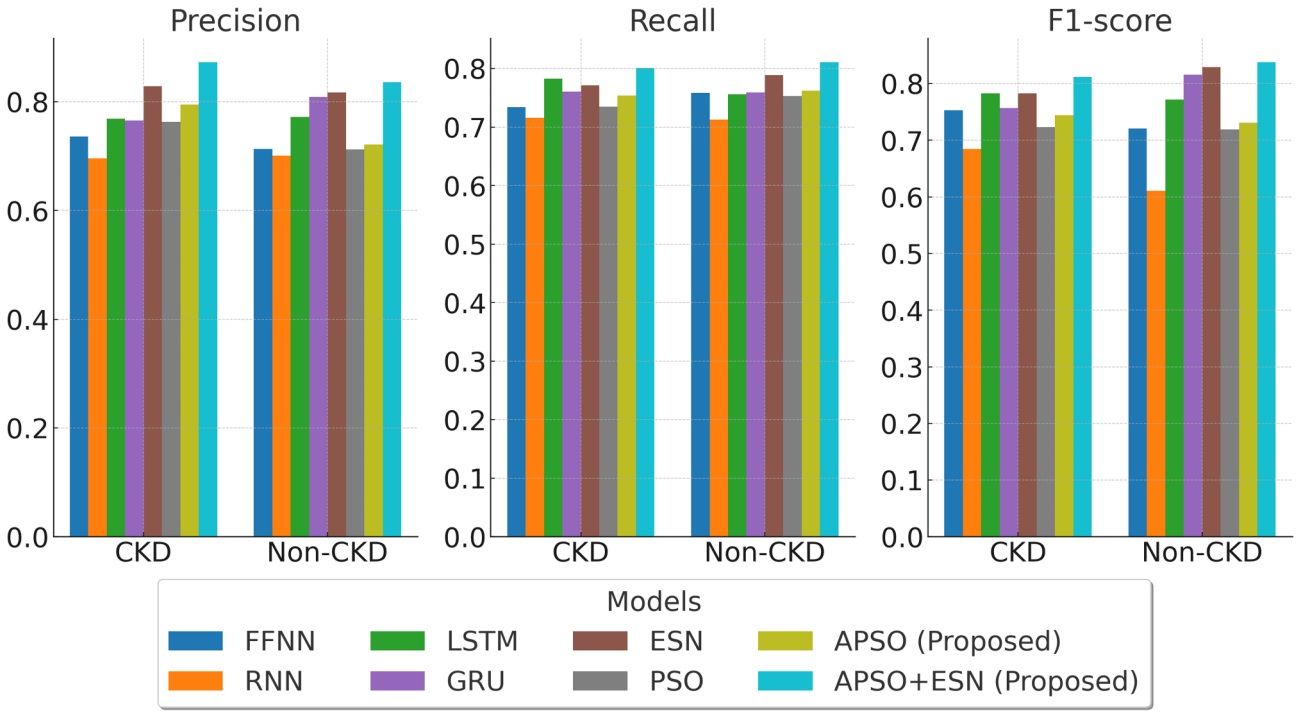

Fig. 4. Impact of excluding KF and MI features on model performance for CKD classification. The proposed APSO + ESN proves significant performance degradation, with an 11.37% decrease in precision and a 19.01% drop in recall, underscoring the critical role of KF and MI features in accurate CKD prediction.

Specifically, removing these critical indicators results in an 11.37% decrease in precision, a 19.01% decline in the recall, and an 18.31% reduction in the F1-score for the APSO + ESN. Across all models, the absence of KF and MI uniformly impacts performance metrics. The APSO + ESN, however, continues to outperform other models, including FFNN, RNN, LSTM, GRU, ESN, and PSO-optimized models, in this constrain of FS setup.

Figure 5 compares models without the BC class. Without the BC class, the APSO + ESN demonstrates a precision of 0.888, a recall of 0.879, and an F1-score of 0.887 for CKD classification. Compared to its performance with the complete set of FS (precision: 0.985, recall: 0.989, F1-score: 0.994), the model parades a noticeable decrease in all metrics. This reduced performance displays the significance of the BC features in enhancing the model's predictive CA. For non-CKD classification, the APSO + ESN maintains high performance with a precision of 0.886, a recall of 0.896, and an F1-score of 0.891. Although slightly reduced compared to the complete set of FS, this performance still confirms the model's ability to adapt and utilize the remaining features effectively for accurate prediction.

As shown in Fig. 6, in the absence of CC data, the APSO + ESN records a precision of 0.918 and a recall of 0.921 for CKD classification, culminating in an F1-score of 0.937. The model achieves a precision and recall of 0.926 for non-CKD classification, with an F1-score of 0.918. These results indicate a decrease in performance metrics compared to the model's results when utilizing a complete set of FS. Across all analyses, the APSO + ESN consistently outperforms other models under numerous feature set conditions, demonstrating its superior design and optimization approach.

Figure 7 shows the accuracy of the compared models for different feature set configurations. The APSO + ESN had outperformed all other models with a CA of 0.996. The closest competitor is ESN, which has a CA of 0.986. On comparing the proposed model without KF metrics, the proposed model had a reduced CA of 0.857, whereas the other models had a more significant decrease; notably, FFNN reduced to 0.742. The decrease across models ranges from 11 to 22%, with APSO + ESN displaying one of the most minor percentage decreases. For other feature classes, such as those without BC and CC, the proposed APSO + ESN has reduced CA to 0.902 and 0.932. The other models have decreased performance, proving both feature classes' significance. It is to be noted that while CC pays to CKD prediction, their absence is less detrimental to the model's CA than the omission of KF or BC data.

Figure 8 shows the results of all models under five-fold cross-validation. FFNN performance scores range from around 0.712 to 0.836, whereas RNN's performance is more variable, with scores from approximately 0.608 to 0.867. The LSTM and GRU illustrate robust performances compared to RNN, with LSTM scores variable from around 0.773 to 0.854 and GRU scores slightly higher, ranging from approximately 0.802 to 0.917. Comparatively, the ESN outperforms other algorithms except those proposed in later folds, with scores increasing to 0.966. Compared to PSO, the proposed APSO has shown increased performance, ranging from 0.734 to 0.858. Out of all models, the proposed APSO + ESN significantly performed, scoring around 0.911 to 0.969.

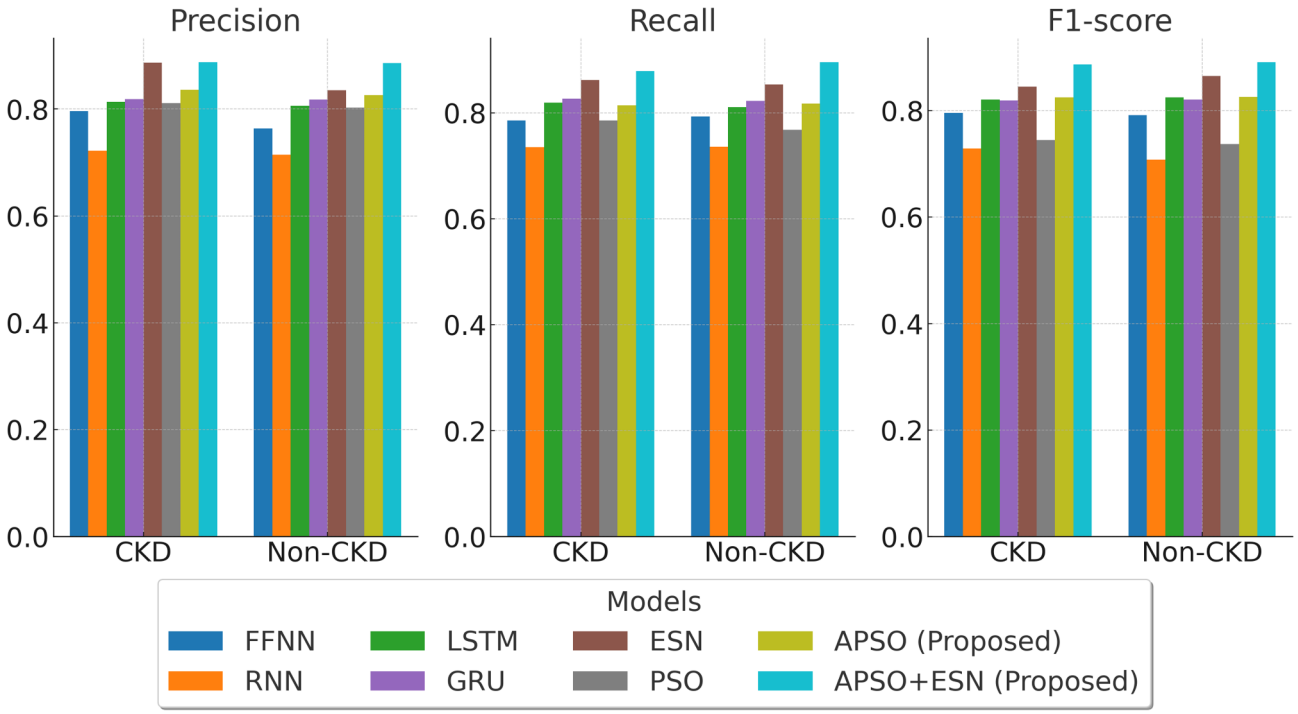

Fig. 5. Performance comparison of CKD prediction models without BC features. The APSO + ESN remains robust but shows a noticeable decline in accuracy and recall, indicating the importance of BC features in enhancing predictive accuracy and reliability, particularly in CKD classification.

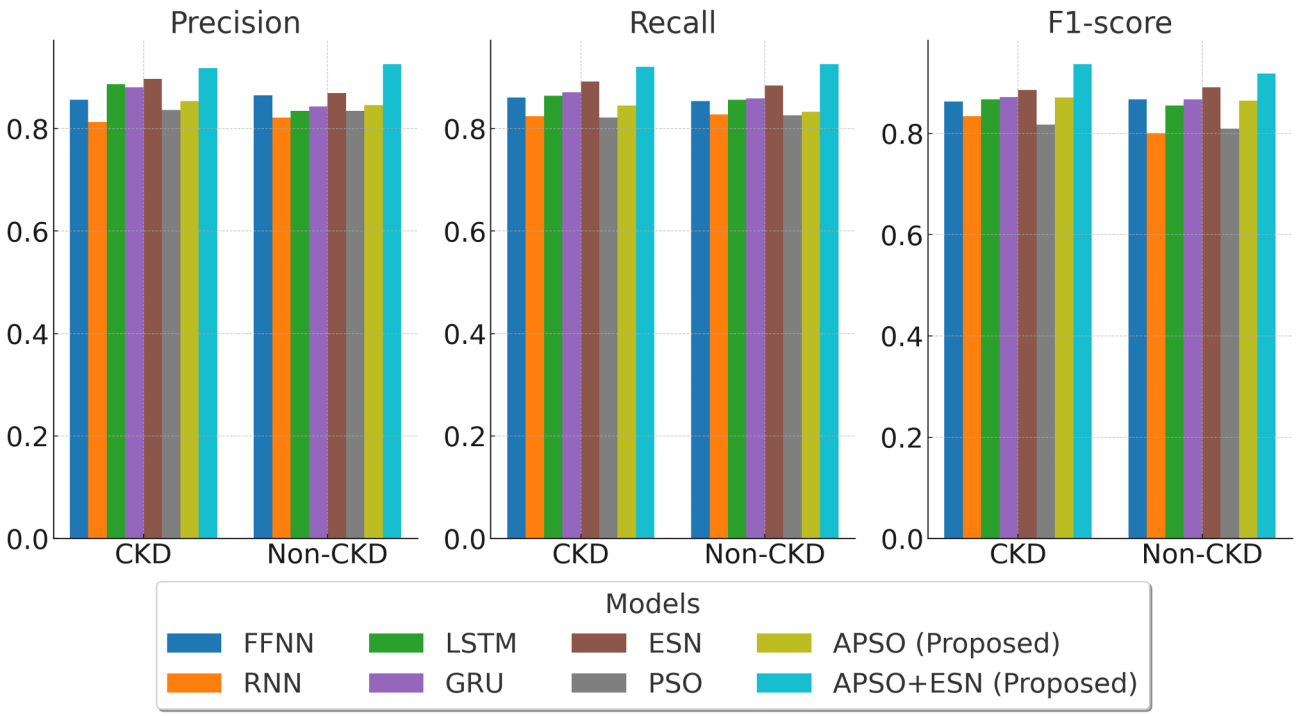

Fig. 6. Effect of excluding CC features on CKD prediction model performance. While the APSO + ESN retains higher CA and precision than other models, the results highlight that CC features play a helpful role in refining classification metrics, albeit with less impact than KF or BC classes.

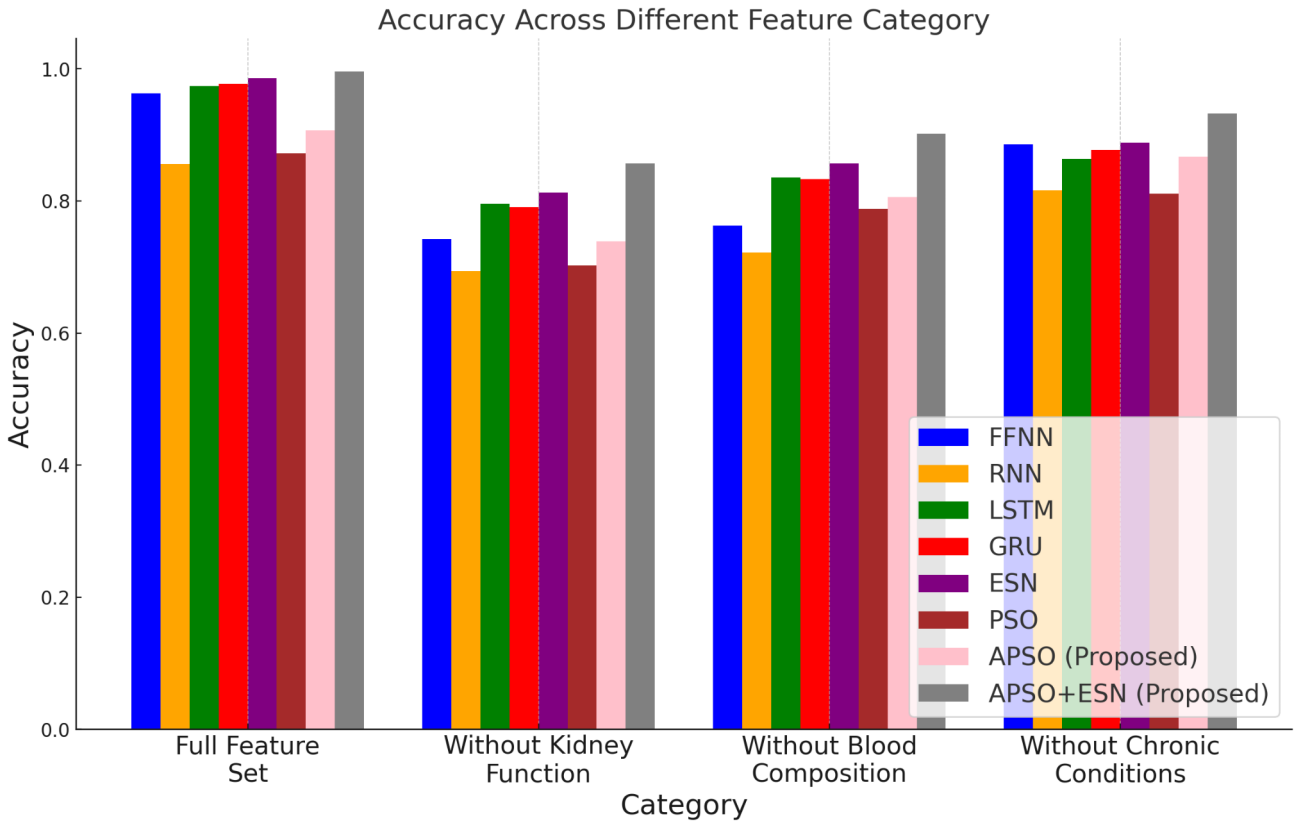

Fig. 7. Accuracy comparison of CKD prediction models across different feature set configurations. The APSO + ESN consistently outperforms other models under all configurations, demonstrating its resilience and adaptability. Notably, accuracy drops significantly when key feature classes (e.g., KF and MI) are excluded, reinforcing their critical role in CKD diagnosis.

As shown in Fig. 9, the chart visualizes the feature importance scores for various features considered under KF and MI, BC, and CC. The importance scores, ranging from 0 to 1, reflect the relative significance of each feature in assessing KF and related conditions based on their impact on model performance. Table 1 displays higher importance scores for features within the "KF and MI" class, reflecting their critical role in assessing KF as indicated by the drop in model CA when these features are omitted. Features related to "BC" and "CC" are also significant but have slightly lower scores, consistent with the lesser impact on CA observed when these classes are excluded from the model training.

The selection of 23 features from the original 61 was guided by their clinical relevance to CKD diagnosis and their predictive significance in improving model performance. The proposed FS prioritized features that demonstrated strong correlations with KF, MI, BC, and CC, commonly predicted in clinical works as key symbols for CKD. Statistical methods, such as Recursive Feature Elimination (RFE), were applied to identify features with the highest importance scores, while domain expertise in nephrology was used to validate the selection. This process retained only the most relevant and non-redundant features, balancing predictive impact with interpretability. The effectiveness of these FS is further proved by the high CA and model performance achieved during evaluation.

In the subsequent analysis, the proposed model's CA, loss and execution time were compared against other *state-of-the-art* models such as Neural Network with Cuckoo Search (NN-CS), Neural Network with Genetic Algorithm (NN-GA), Hierarchical Modular Artificial Neural Network (HMANN), Ensemble Decision Trees (EDT), Modular Deep Belief Networks (MDBN), Hybrid Parallel Computing Model (HPCM). Figure 10 shows the comparative performance among the compared models.

The NN-CS showed a computation time of 52.92 min and a CA of 95.86% with a loss of 0.141. The NN-GA required less time at 32.83 min, with a slightly lower CA of 95.51% and a marginally higher loss of 0.153. The HMANN outperformed these two in CA and loss, achieving a CA of 98.73% and the lowest loss among the conventional models at 0.0468, within a relatively short computation time of 26.33 min. The EDT showcased its efficiency with less computation time of 6.47 min, a high CA of 97.52%, and a loss of 0.0746. The MDBN, with a computation time of 11.72 min, recorded a CA of 95.70% and the highest loss of 0.186 among the models evaluated. The HPCM developed a high computation time at 67.33 min, achieving a CA of 95.83% with a loss of 0.1335. The proposed model demonstrated superior performance, with a CA nearing perfection at 99.60% and a less loss of 0.014, requiring a computation time of 56.28 min.

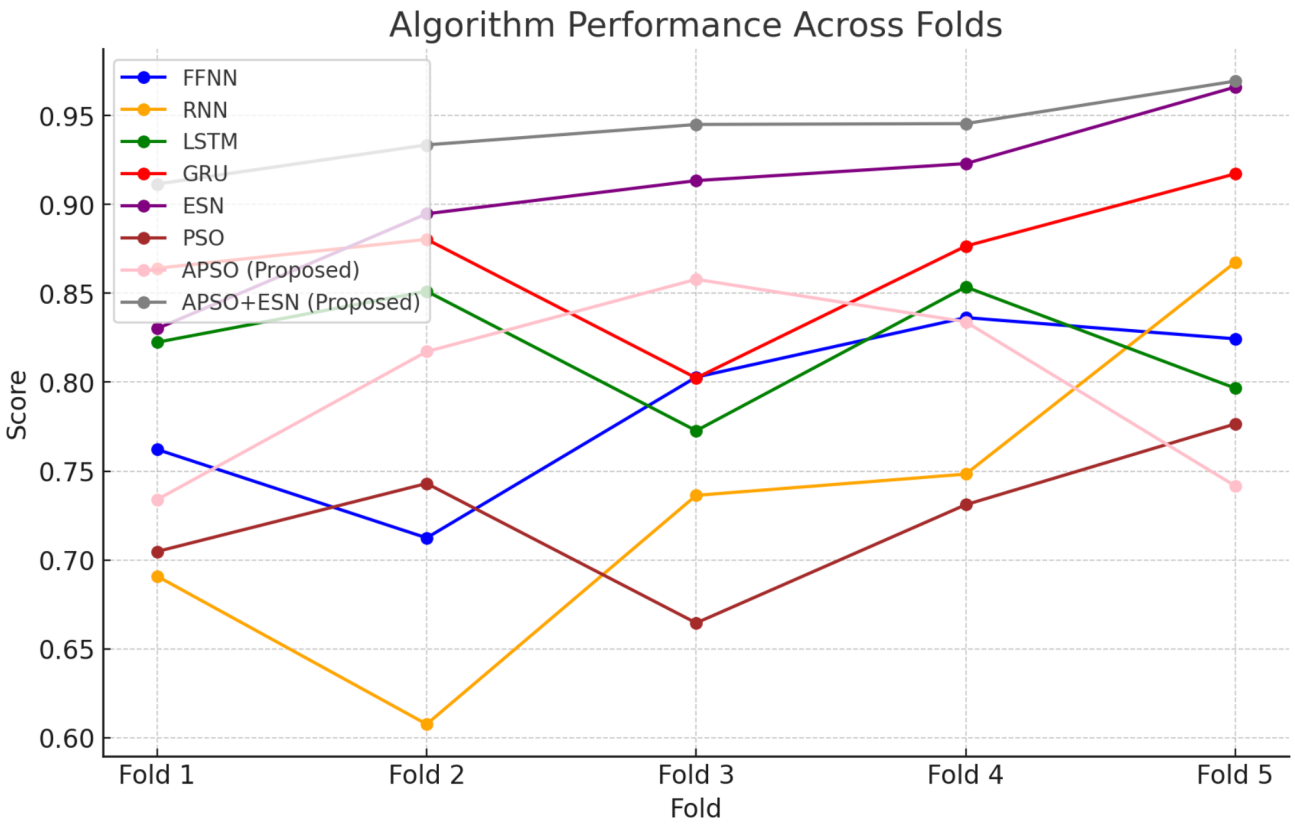

Fig. 8. Accuracy measures for five-fold cross-validation. This figure compares the accuracy of numerous models, including FFNN, RNN, LSTM, GRU, ESN, PSO, and the proposed APSO + ESN, across five cross-validation folds. Compared to other models, the APSO + ESN consistently achieves the highest CA, demonstrating its robustness and generalizability in CKD prediction.

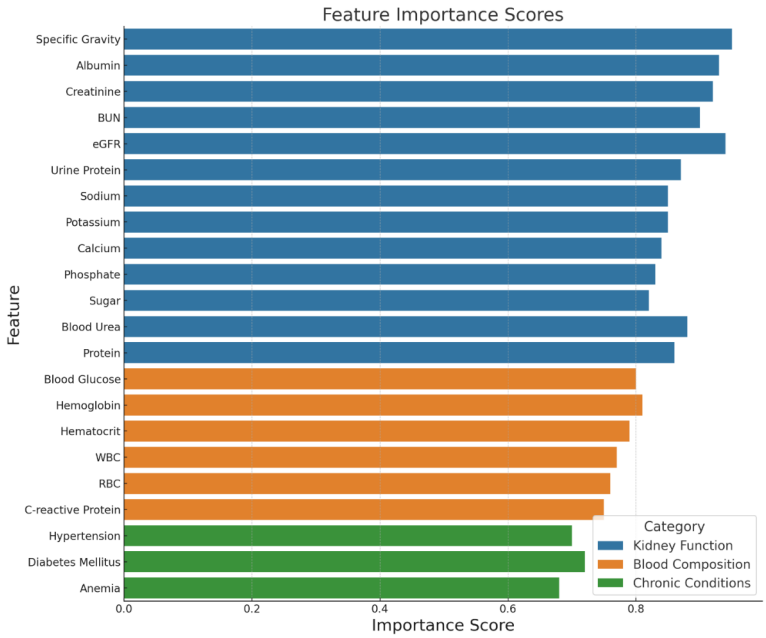

Fig. 9. Feature importance scores for categories in CKD prediction (KF, MI, BC, and CC). The analysis reveals that KF and MI hold the highest importance, aligning with their substantial impact on model CA and clinical relevance in CKD diagnosis.

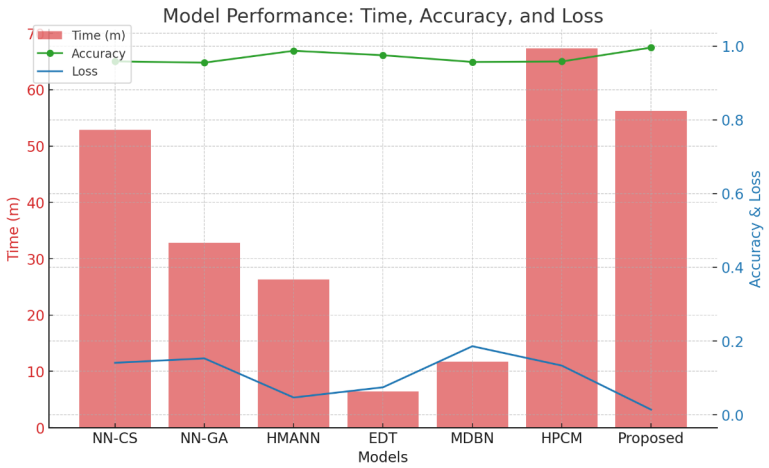

Fig. 10. Comparative analysis of the proposed APSO + ESN against state-of-the-art models (e.g., NN-CS, NN-GA, HMANN). The APSO + ESN achieves the highest CA (99.6%) and the lowest loss (0.014) while maintaining competitive execution time, starting it as a prime model for CKD prediction.

Model	Precision (Mean ± SD)	Recall (Mean ± SD)	F1-Score (Mean ± SD)	*p*-value (Precision)	*p*-value (Recall)	*p*-value (F1-Score)
APSO + ESN	0.985 ± 0.003	0.989 ± 0.004	0.994 ± 0.002	-	-	-
ESN	0.976 ± 0.004	0.978 ± 0.005	0.987 ± 0.003	< 0.01	< 0.01	< 0.01
PSO	0.964 ± 0.006	0.962 ± 0.007	0.978 ± 0.005	< 0.001	< 0.001	< 0.001
GRU	0.941 ± 0.008	0.947 ± 0.009	0.955 ± 0.007	< 0.001	< 0.001	< 0.001
LSTM	0.951 ± 0.007	0.950 ± 0.008	0.960 ± 0.006	< 0.001	< 0.001	< 0.001

Table 3. Statistical analysis.

Statistical significance analysis

The statistical significance analysis (Table 3) confirms the superior performance of the APSO + ESN compared to other models, including standalone ESN, PSO, GRU, and LSTM. The APSO + ESN achieved a precision of 0.985 ± 0.003, recall of 0.989 ± 0.004, and F1-score of 0.994 ± 0.002, demonstrating minimal variability across multiple runs. In comparison, the ESN achieved slightly lower metrics (precision: 0.976 ± 0.004, recall: 0.978 ± 0.005, and F1-score: 0.987 ± 0.003), and the differences were statistically significant with p-values less than 0.01 for all three metrics.

The APSO + ESN also outperformed PSO, which showed more significant variability in its results (precision: 0.964 ± 0.006, recall: 0.962 ± 0.007, and F1-score: 0.978 ± 0.005), with p-values of less than 0.001. Similarly, significant improvements were experiential when compared to GRU (precision: 0.941 ± 0.008, recall: 0.947 ± 0.009, F1-score: 0.955 ± 0.007) and LSTM (precision: 0.951 ± 0.007, recall: 0.950 ± 0.008, F1-score: 0.960 ± 0.006), with p-values consistently below 0.001.

These findings demonstrate that the APSO + ESN significantly enhances precision, recall, and F1-score metrics while maintaining low variability, highlighting its reliability and robustness. The p-value results validate the statistical significance of the performance improvements, particularly in CKD prediction tasks, highlighting the success of the proposed optimization strategy in enhancing ESN performance.

Convergence analysis

The convergence analysis (Fig. 11) across the models highlights distinct differences in their optimization performance. Traditional models like FFNN, RNN, LSTM, and GRU show slower convergence rates, with RNN and GRU displaying variations during optimization due to gradient instability issues inherent in recurrent models. The standalone ESN achieves faster convergence than these traditional models, featuring its fixed reservoir structure and reduced computational complexity. However, its convergence plateaus earlier, indicating the limitations of manual parameter selection in achieving optimal performance.

As a standalone optimizer, the APSO converges faster than traditional models and exhibits a more consistent reduction in error, demonstrating the strength of adaptive mechanisms in balancing exploration and exploitation during optimization. The proposed APSO + ESN shows the fastest and smoothest convergence curve, stabilizing at a significantly lower loss than all other models. This outcome highlights the synergistic effect of combining APSO's dynamic optimization with the ESN's efficient handling of temporal dependencies, further enhanced by Random Matrix Theory (RMT) for spectral radius tuning.

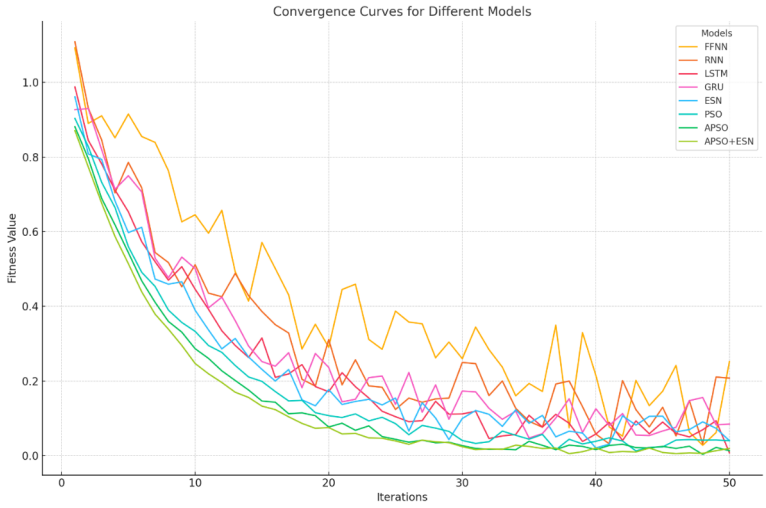

Fig. 11. Convergence analysis.

Discussion

The results presented in this study demonstrate the superior performance of the APSO + ESN in CKD prediction across multiple metrics, including precision, recall, F1-score, and accuracy. These findings have significant implications for practical applications, particularly in developing Clinical Decision Support Systems (CDSS). By achieving high CA (99.6%) and robust performance even under constrained feature sets, the APSO + ESN can support early CKD diagnosis, enabling timely interventions and personalized treatment plans.

In practical settings, this enhanced predictive capability could be integrated into CDSS to assist healthcare providers in identifying high-risk patients, prioritizing resources, and tailoring treatment procedures based on discrete patient profiles. The high precision and recall metrics indicate that the model can reduce FN and FP, minimizing the risk of misdiagnosis and redundant treatments.

Furthermore, the model's adaptability to numerous feature set configurations underscores its potential utility in assorted clinical setups, where complete data may not continuously be available. The demonstrated effectiveness of the APSO + ESN in leveraging temporal requirements and optimizing parameters also proposes its applicability to other time-series-based healthcare predictions, such as monitoring disease progression or predicting readmission risks. By integrating this model into clinical workflows, healthcare systems could enhance patient outcomes and operational efficiency while addressing critical challenges in CKD management.

Despite these strengths, the APSO + ESN also has some limitations. The computational overhead introduced by APSO optimization is higher than that of traditional methods, which may pose challenges for deployment in resource-constrained environments. Additionally, while the model demonstrates resilience to missing features, its performance still declines when critical features such as KF and MI are excluded, as shown by a significant drop in precision and recall. This dependency highlights the requirement for complete and high-quality data for optimal performance. Another limitation is the interpretability of the model's internal mechanisms, a common challenge in reservoir computing, which may hinder its adoption in clinical systems where explainability is vital.

In conclusion, the APSO + ESN significantly improves CKD prediction but requires careful attention to its computational demands and data quality requirements. Future work should address these limitations, such as optimizing computational efficiency and improving interpretability, to enhance its practicality in diverse healthcare backgrounds.

Conclusion and future work

This work recommended an improved ESN for the early prediction of CKD using a longitudinal time-series dataset. Optimizing the CA of the ESN is as easy as optimizing the variables, which significantly impact how well the model performs. The current research advanced an APSO to improve the ESN variables with RMT. Researchers examined the proposed method's performance on the MIMIC-III dataset. When compared to other models, this one has a higher CA of 99.6%, precision of 100%, and recall of 100% scores. Several differences in feature sets were employed to evaluate the model's adaptability. Several feature types were rejected by training, including KF, MI, BC, and CC. It has been trained with one data set, and the high computational needs of the APSO are identified as the research problems proposed in the case study.

While the results are promising, this study is subject to certain limitations. The reliance on a single dataset (MIMIC-III) with inherent class imbalance and demographic constraints may limit the generalizability of the findings. The computational demands of the APSO-optimized ESN pose challenges for real-time deployment in resource-constrained environments. Additionally, the temporal space of 48 h used for analysis, though effective for early-stage prediction, might not fully capture CKD's long-term progression.

Future research will address these limitations by exploring external datasets for validation across diverse peoples, optimizing computational efficiency for real-time deployment, and incorporating longitudinal and

heterogeneous data to capture CKD progression more comprehensively. Expanding the feature set to include genetic and lifestyle factors will enhance the model's predictive capabilities and clinical applicability.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 25 June 2024; Accepted: 18 February 2025
Published online: 26 February 2025

References

1. Akter, S. Comprehensive performance assessment of deep learning models in early prediction and risk identification of chronic kidney disease. *IEEE Access.* **9**, 165184–165206. https://doi.org/10.1109/ACCESS.2021.3129491 (2021).
2. Singh, V., Asari, A. K. & Rajasekaran, R. A deep neural network for early detection and prediction of chronic kidney disease. *Diagnostics* **12**(1), 116. https://doi.org/10.3390/diagnostics12010116 (2022).
3. Kumar, S. S. & Baskaran, T. S. Classification of chronic kidney disease in adults using enhanced recurrent neural networks. *Int. J. Intell. Syst. Appl. Eng.* **12**(7s), 191–200 (2024).
4. Bhaskar, N. & Suchetha, M. A computationally efficient correlational neural network for automated prediction of chronic kidney disease. *IRBM* **42**(4), 268–276 (2021).
5. Zhu, Y., Bi, D., Saunders, M. & Ji, Y. Prediction of chronic kidney disease progression using recurrent neural network and electronic health records. *Sci. Rep.* **13**(1), 22091 (2023).
6. Akter, S. et al. CKD. Net: A novel deep learning hybrid model for effective, real-time, automated screening tool towards prediction of multi-stages of CKD along with eGFR and creatinine. *Expert Syst. Appl.* **223**, 119851 (2023).
7. Saif, D., Sarhan, A. M. & Elshennawy, N. M. Deep-kidney: An effective deep learning framework for chronic kidney disease prediction. *Health Inform. Sci. Syst.* **12**(1), 3 (2023).
8. Bhaskar, N., Suchetha, M. & Philip, N. Y. Time series classification-based correlational neural network with bidirectional LSTM for automated detection of kidney disease. *IEEE Sens. J.* **21**(4), 4811–4818 (2020).
9. Yildiz, E. N., Cengil, E., Yildirim, M. & Bingol, H. Diagnosis of chronic kidney disease based on CNN and LSTM. *Acadlore Trans. Mach. Learn.* **2**(2), 66–74 (2023).
10. Wang, Y., Guan, Z., Hou, W. & Wang, F. TRACE: Early detection of chronic kidney disease onset with Transformer-Enhanced feature embedding. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare: VLDB Workshops, Poly 2021 and DMAH 2021, Virtual Event, August 20, 2021, Revised Selected Papers 7* 166–182 (Springer, 2021).
11. Thanmayi, K. & Suseela, K. Detection of chronic kidney diagnosis using RNN algorithm. In *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)* 1–5 (IEEE).
12. Elkholy, S. M. M., Rezk, A. & Saleh, A. A. E. F. Early prediction of chronic kidney disease using deep belief network. *IEEE Access* **9**, 135542–135549 (2021).
13. Pradeepa, P. & Jayakumar, M. K. Modelling of IDBN with LSNN-based optimal feature selection for the prediction of CKD using real-time data. *Multimed. Tools Appl.* **82**(4), 6309–6344 (2023).
14. Chen, G. et al. Prediction of chronic kidney disease using adaptive hybridized deep convolutional neural network on the internet of medical things platform. *IEEE Access* **8**, 100497–100508 (2020).
15. Alsekait, D. M. et al. Toward comprehensive chronic kidney disease prediction based on ensemble deep learning models. *Appl. Sci.* **13**(6), 3937 (2023).
16. Hore, S., Chatterjee, S., Shaw, R. K., Dey, N. & Virmani, J. Detection of chronic kidney disease: A NN-GA-based approach. In *Nature Inspired Computing. Advances in Intelligent Systems and Computing* Vol. 652 (eds Panigrahi, B. et al.) (Springer, Singapore, 2018).
17. Chatterjee, S., Banerjee, S., Basu, P., Debnath, M. & Sen, S. Cuckoo search coupled artificial neural network in detection of chronic kidney disease. In *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech), Kolkata, India* 1–4 (2017). https://doi.org/10.1109/IEMENTECH.2017.8077016.
18. 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).
19. Chaudhuri, A. K., Sinha, D., Banerjee, D. K. & Das, A. A novel enhanced decision tree model for detecting chronic kidney disease. *Netw. Model. Anal. Health Inf. Bioinf.* **10**, 1–22 (2021).
20. Singh, V. & Jain, D. A hybrid parallel classification model for the diagnosis of chronic kidney disease (2023).
21. Bouazizi, S. & Ltifi, H. Enhancing accuracy and interpretability in EEG-based medical decision making using an explainable ensemble learning framework application for stroke prediction. *Decis. Support Syst.* **178**, 114126 (2024).
22. Wang, S., Ding, C., Wang, Z., Shen, L. & Wang, J. Using normalized echo state network to detect abnormal ECG patterns. *Int. J. Imaging Syst. Technol.* **34**(1), e22940 (2024).
23. Liu, J. et al. Echo state network optimization using binary grey Wolf algorithm. *Neurocomputing* **385**, 310–318 (2020).
24. Trierweiler Ribeiro, G., Guilherme Sauer, J., Fraccanabbia, N., Mariani, C. & dos Santos Coelho, L. Bayesian optimized echo state network applied to short-term load forecasting. *Energies* **13**(9), 2390 (2020).
25. Bala, A., Ismail, I., Ibrahim, R., Sait, S. M. & Oliva, D. An improved grasshopper optimization algorithm-based echo state network for predicting faults in airplane engines. *IEEE Access* **8**, 159773–159789 (2020).
26. Zhang, Q., Qian, H., Chen, Y. & Lei, D. A short-term traffic forecasting model based on echo state network optimized by improved fruit fly optimization algorithm. *Neurocomputing* **416**, 117–124 (2020).
27. Ahmad, Z., Mahmood, T., Rehman, A., Saba, T. & Alamri, F. S. Enhancing time series forecasting with an optimized binary gravitational search algorithm for echo state networks. *IEEE Access* (2023).
28. Shahi, S. et al. Long-time prediction of arrhythmic cardiac action potentials using recurrent neural networks and reservoir computing. *Front. Physiol.* **12**, 734178 (2021).
29. González-Zapata, A. M., Tlelo-Cuautle, E., Ovilla-Martinez, B., Cruz-Vega, I. & De la Fraga, L. G. Optimizing echo state networks for enhancing large prediction horizons of chaotic time series. *Mathematics* **10**(20), 3886. https://doi.org/10.3390/math10203886 (2022).
30. Fu, S., Fang, X. & Chen, X. Hybrid enhanced echo state network for nonlinear prediction of multivariate chaotic time series. In *2022 IEEE International Conference on Networking, Sensing and Control (ICNSC)* 1–6 (IEEE, 2022).
31. Gong, Y., Lun, S., Li, M. & Lu, X. An echo state network model with the protein structure for time series prediction. *Appl. Soft Comput.* **153**, 111257 (2024).
32. Lukoševičius, M. A practical guide to applying echo state networks. *Lecture Notes Comput. Sci.* **7700**, 659–686 (2012).
33. Emara, H. M. et al. Cervical cancer detection: A comprehensive evaluation of CNN models, vision transformer approaches, and fusion strategies. *IEEE Access* https://doi.org/10.1109/ACCESS.2024.3473741 (2024).

34. Saif, D., Sarhan, A. M. & Elshennawy, N. M. Early prediction of chronic kidney disease based on ensemble of deep learning models and optimizers. *J. Electr. Syst. Inf. Technol.* **11**, 17. https://doi.org/10.1186/s43067-024-00142-4 (2024).
35. Tusar, M. T. H. K., Islam, M. T. & Raju, F. I. Detecting chronic kidney disease (CKD) at the initial stage: A novel hybrid feature-selection method and robust data preparation pipeline for different ML techniques. In *5th International Conference on Computing and Informatics (ICCI)* 400–407 (IEEE, 2022).
36. Li, Y. & Padman, R. Enhancing end stage renal disease outcome prediction: A multi-sourced data-driven approach. arXiv preprint arXiv:2410.01859 (2024).
37. Manonmani, M. & Balakrishnan, S. An ensemble feature selection method for prediction of CKD. In *2020 International Conference on Computer Communication and Informatics, Coimbatore, India* 1–6. https://doi.org/10.1109/ICCCI48352.2020.9104137.
38. Arora, A., Sehgal, C. & Agarwal, N. An analysis of machine learning algorithms for chronic kidney disease prediction. In *2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* 581–586 (IEEE, 2024).
39. Soltani, R., Benmohamed, E. & Ltifi, H. Optimized echo state network based on PSO and gradient descent for choatic time series prediction. In *IEEE 34th International Conference on Tools with Artificial Intelligence, Macao, China, 2022* 747–754 (2022). https://doi.org/10.1109/ICTAI56018.2022.00115.
40. Ramu, K. et al. Hybrid CNN-SVM model for enhanced early detection of chronic kidney disease. *Biomed. Signal Process. Control* **100**, 107084 (2025).
41. Yang, W., Ahmed, N. & Barczak, A. Comparative analysis of machine learning algorithms for CKD risk prediction. *IEEE Access* (2024).
42. An, X. et al. A few-shot identification method for stochastic dynamical systems based on residual multi peaks adaptive sampling. *Chaos Interdiscip. J. Nonlinear Sci.* **34**(7), 073118 (2024).
43. Yang, Y. & Li, H. Neural ordinary differential equations for robust parameter estimation in dynamic systems with physical priors. *Appl. Soft Comput.* **169**, 112649. https://doi.org/10.1016/j.asoc.2024.112649 (2024).
44. Lin, W. et al. Input and output matter: Malicious traffic detection with explainability. *IEEE Netw.* https://doi.org/10.1109/MNET.2024.3481045 (2024).
45. Yue, S., Zeng, S., Liu, L., Eldar, Y. C. & Di, B. Hybrid near-far field channel estimation for holographic MIMO communications. *IEEE Trans. Wirel. Commun.* **23**(11), 15798–15813. https://doi.org/10.1109/TWC.2024.3433491 (2024).
46. Shi, J., Zhao, B., He, J. & Lu, X. The optimization design for the journal-thrust couple-bearing surface texture based on particle swarm algorithm. *Tribol. Int.* **198**, 109874. https://doi.org/10.1016/j.triboint.2024.109874 (2024).
47. Yang, M. et al. Two-stage day-ahead multi-step prediction of wind power considering time-series information interaction. *Energy* **312**, https://doi.org/10.1016/j.energy.2024.133580 (2024).
48. Li, N., Dong, J., Liu, L., Li, H. & Yan, J. A novel EMD and causal convolutional network integrated with transformer for ultra short-term wind power forecasting. *Int. J. Electr. Power Energy Syst.* **154**, 109470. https://doi.org/10.1016/j.ijepes.2023.109470 (2023).
49. Xu, X., Lin, Z., Li, X., Shang, C. & Shen, Q. Multi-objective robust optimisation model for MDVRPLS in refined oil distribution. *Int. J. Prod. Res.* **60**(22), 6772–6792. https://doi.org/10.1080/00207543.2021.1887534 (2022).
50. Zhou, Z., Zhou, X., Qi, H., Li, N. & Mi, C. Near miss prediction in commercial aviation through a combined model of grey neural network. *Expert Syst. Appl.* **255**, 124690. https://doi.org/10.1016/j.eswa.2024.124690 (2024).
51. Guo, C., Hu, J., Hao, J., Čelikovský, S. & Hu, X. Fixed-time safe tracking control of uncertain high-order nonlinear pure-feedback systems via unified transformation functions. *Kybernetika* **59**(3), 342–364. https://doi.org/10.14736/kyb-2023-3-0342 (2023).
52. Panneerselvam, S. et al. Federated learning-based fire detection method using local MobileNet. *Sci. Rep.* **14**, 30388. https://doi.org/10.1038/s41598-024-82001-w (2024).
53. Vincent, A. C. S. R. & Sengan, S. Edge computing-based ensemble learning model for health care decision systems. *Sci. Rep.* **14**, 26997. https://doi.org/10.1038/s41598-024-78225-5 (2024).
54. Robert Vincent, A. C. S. & Sengan, S. Effective clinical decision support implementation using a multi filter and wrapper optimisation model for Internet of Things based healthcare data. *Sci. Rep.* **14**, 21820. https://doi.org/10.1038/s41598-024-71726-3 (2024).

Acknowledgements
Not applicable.

Author contributions
Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft, writing - review & editing, visualization: A.T. and B.R.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to T.A. or B.R.

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.