Applied Soft Computing Journal 89 (2020) 106117

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Non linear system identification using kernel based exponentially extended random vector functional link network

Tatiana Chakravorti *, Penke Satyanarayana

Electronics and Communication Engineering, KLEF (Deemed to be University), Vijayawada, Andhra Pradesh 522502, India

ARTICLE INFO

Article history:
Received 29 June 2019
Received in revised form 5 January 2020
Accepted 20 January 2020
Available online 24 January 2020

Keywords:
Kernel based exponentially extended RVFLN
System identification
Water cycle algorithm
SISO system
Monte Carlo simulations

ABSTRACT

Identification of nonlinear systems finds extensive applications in control design and stability analysis. To identify complex nonlinear systems, the neural network has drawn the attention of many researchers due to its broad application area. In this paper, an improved identification method based on Kernel Exponentially Extended Random Vector Functional Link Network (KERVFLN) has been proposed for nonlinear system identification. Good generalization capability, fast learning speed, simple architecture and the direct connection between input and output nodes along with non linear enhancement nodes with random weights of traditional Random Vector Functional Link Network (RVFLN) are very essential to industrial applications. To avoid the selection of the number of hidden nodes and hidden mapping function, kernel function has been used in this paper to increase the stability. The input is extended using trigonometric expansion which increases the accuracy of the algorithm when ever there is a sudden random change. In case of KERVFLN the number of enhancement nodes and its corresponding activation function need not to be known if its corresponding kernel function is given. To verify the accuracy of the proposed model, some benchmark Monte Carlo simulations and one SISO system are carried out through simulation study and the obtained results are compared with some established techniques such as original RVFLN, Extreme Learning Machine (ELM), and Least Mean Square (LMS). The efficiency of the proposed technique has been tested with the real time data set as well. The prediction accuracy of the proposed method KERVFLN is higher than the normal RVFLN for different nonlinear systems which is clear from the performance evaluation section.

1. Introduction

System identification is a method of identifying any mathematical model of any nonlinear system by measuring the system inputs and outputs. Identification of complex non linear plants is a major concern in control system [1–4]. Most of the practical systems are nonlinear in nature. Time series models require less information to design the model but the performances of these models are very slow. Previously the statistical models include massively used techniques like Auto-regressive Moving Average (ARMA) and Auto-regressive Integrated Moving Average (ARIMA) in different areas. The two models, ARMA and ARIMA are having a lot implementation in prediction but they are having certain drawbacks like the ineffectiveness in tracking the random change in time series data [5–8]. Recent research shows that neural network seems to be promising to identify any nonlinear complex systems when the whole model information is not present [9,10]. Several identification algorithms for nonlinear system identification have been proposed in the literature [11–15]. Nonlinear plants identification is a huge challenge because of the difficulty to decide a general model structure which can represent data from a nonlinear system. Most of the practical or industrial systems possess inherent nonlinear nature due to intermediation, harmonic generation, gain expansion, desensitization etc. Identification of those complex nonlinear plants possesses an important role in design and analysis of control system.

Artificial neural networks (ANNs) have proved to be a strong learning technique to perform tasks in nonlinear environments much better than ARMA and ARIMA. Some of the main advantages of using ANN are their excellent performance for an approximation of nonlinear systems. In literature, different types of paradigms of ANNs are present by the researchers for the task of nonlinear system identification [16–19]. These neural networks are capable of identification of nonlinear complex systems, but its computational complexity is very high. But ANN based methods suffer from certain disadvantages likes generalization, overfitting local minima and lower convergence speed whenever the data is very random. Therefore it exhibit unstable characteristics for smaller variations in the time series data. The hybrid models

* Corresponding author.
E-mail addresses: tatiana@kluniversity.in (T. Chakravorti), satece@kluniversity.in (P. Satyanarayana).

https://doi.org/10.1016/j.asoc.2020.106117
1568-4946/

combining both statistical and AI based approaches [20–23] gives better results than the previous model like ARMA-ANN [24] and ARIMA. ARMA model combined with non linear auto regression neural network (NARNN) provide somewhat good prediction accuracy but suffer from higher computational complexity due to their iterative learning process.

The tradition neural networks use gradient based learning algorithms resulting in slow convergence rate and the possibility of being trapped in local minima. To overcome these limitations single layer feedforward neural networks (SLFN) with random hidden layer weights has been proposed. Single layer feed forward neural networks have been extensively used to solve different problems such as regression, classification, and nonlinear system identification because of their universal approximation proficiency [25–27]. Conventional techniques such as back-propagation are used to train SLFN. These methods deteriorate because of slow confluence, getting captured in a local minimum. This method is very sensitive to learning rate setting. Another massively used SLFN technique is Extreme Learning Machine (ELM). However, the learning performance of ELM is highly influenced by the choice of the hidden layer neurons and their activation functions. Science with a chosen number of hidden layer neurons, ELM produces a large variation in the prediction accuracy for different trial runs and therefore kernel functions are introduced. In order to overcome the above mentioned problems, a non iterative model has been introduced known as Random Vector Functional Link Network (RVFLN). A randomized interpretation of the functional link is called Random Vector Functional Link Networks (RVFLN), is proposed in [28], which shows that certain values of the weights can be randomly generated from the input layer to hidden layer. The conventional RVFLN is known for its simple architecture, good generalization capabilities and fast learning speed. The architecture of RVFLN is similar to Extreme Learning Machine (ELM) only there is a direct connection present between the input and output nodes with random weights which increases the prediction accuracy. The proficiency and the capability of the RVFLN were discussed in [29] and the generalization accuracy of RVFLN was shown in [30]. In [31], the authors proved that the technique RVFLN network does a universal approximation for any continuous operation on a bounded fixed dimensional set. From that invention, RVFLN has been used as a problem solver in multi-dimensional area. Conditional feasibilities are modeled using RVFLN in literature. There have been a lot of studies present in the literature which shows that the generalization capacity of RVFLN is better and the learning speed is also higher than the traditional SLFN [32]. But the selection of the hidden layer nodes and the mapping function is still a challenging problem in case of RVFLN. To overcome the deficiencies of the conventional RVFLN, kernel functions are used for both the direct links and the hidden layer nodes to provide better stability, generalization and regression accuracy. The random selection of the hidden nodes can be solved by introducing the kernel function. Different types of kernel function can be used such as sigmoid kernel, Gaussian kernel, and polynomial kernel. Further to optimize the kernel parameters the efficient watercycle algorithm [33–35] has been used in this paper whose main advantages are swift convergence rate and lower computational complexity as compared to various other optimization techniques such as Genetic Algorithm [36,37], particle swam optimization, cuckoo search algorithm, artificial bee colony and so on [38–40]. Many researches with RVFLN have been done to solve regression and identification problems [41–44]. Different advance techniques have been presented in the literature such as Kernel Extreme Learning machine (KELM) and Kernel Ridge Regression (KRR) [45]. The authors have been done a comparative study as well and it has been found that KELM and KRR are not exactly same as KRVFLN. In case of KRVFLN a direct link is present between the input and output nodes which improves the accuracy.

In this paper, the authors have proposed a new kernel based exponentially extended RVFLN (KERVFLN) to improve the performance of the traditional RVFLN. To avoid the selection of the number of hidden nodes and hidden mapping function, Kernel function has been introduced. The simple trigonometric expansion involves sinusoids which attempt to model the nonlinearity. But this expansion cannot efficiently track the sudden amplitude change, especially when the change is very fast. In order to track the sudden amplitude change the expansion is modified which use exponentially varies sinusoids. The proposed KERVFLN has better accuracy and generalization capability than many other existing methods in different types of application.

In this paper, the authors have taken different nonlinear systems to show the performance of the proposed method. The proposed method has been found to be capable of accurate identification of different nonlinear plants. Thus the significant contributions of this paper can be categorized as follows:

1. The basic RVFLN method is chosen for fast learning and accurate identification purpose
2. The randomness of the RVFLN technique is mitigated by implementing the kernel matrix
3. Different kernel functions are investigated to achieve better accuracy
4. In order to track the random amplitude changes in the signal, the exponentially varies sinusoids has been used which has been tested for different random systems as well.
5. The randomly selected kernel parameters are optically tuned by an efficient optimization technique, there by imparting more accurate results.

The remaining part of this paper is arranged as: In Section 2, the proposed identification method of nonlinear systems is introduced. Section 3 describes the performance evaluation of the proposed method and the structure of the nonlinear systems. Finally, Section 4 presents the conclusion and the remarks of the proposed method in the paper.

2. Proposed identification method

2.1. Original RVFLN

Single hidden layer neural network (SLFN) is one of the hugely used feed-forward neural network. In this technique, the input nodes are connected to the hidden layer nodes and then the hidden layer nodes are connected to the output nodes. There are no interconnections between the same layers or between the input and output layers. Random vector functional link network (RVFLN) is a condition where the direct connections are present between the input–output nodes. The data is mapped from the input layer to the hidden layer, i.e. $h\left(w_j^T x + b_j\right)$ where h(.) is the activation function, the input weights are represented as w_j, b_j is the hidden layer node bias and x is the input and can be represented as $x = [x_1, x_2, \ldots, x_Z]$. In this case, the input nodes actually work as a direct input–output connection to connect the inputs with the outputs. Finally, the RVFLN network is modeled as:

$$f(x) = \sum_{j=1}^{J} \beta_j h\left(w_j^T x + b_j\right) + \sum_{i=J+1}^{J+Z} \beta_i x \tag{1}$$

According to Eq. (1) the total input data is Z and there are J hidden layer nodes. β_j is the output weights connected between the output nodes and jth hidden layer nodes. β_i is the output weight which is connected between the direct input–output connections.

2.2. Proposed extended RVFLN

For Extended RVFLN, a nonlinear trigonometric series expansion has been introduced in this paper which can be represented as:

$$f(x) = \begin{cases} x \\ \sin(p\pi x) \\ \cos(p\pi x) \end{cases} \tag{2}$$

where p = (1, … , G) is the expansion index which is taken 1 here and the expansion order is G.

If the inputs are delayed by M number of samples, then the trigonometric expansion can be represented as:

$$\begin{aligned} &f(n_1) = \{1, x(n_1), x(n_1-1), \ldots, x(n_1-M+1) \\ &\sin[\pi x(n_1)], \cos[\pi x(n_1)], \sin[2\pi x(n_1)], \cos[2\pi x(n_1)], \ldots, \\ &\sin[G\pi x(n_1)], \cos[G\pi x(n_1)] \\ &\sin[\pi x(n_1-1)], \cos[\pi x(n_1-1)], \sin[2\pi x(n_1-1)], \\ &\cos[2\pi x(n_1-1)], \ldots, \sin[G\pi x(n_1-1)], \cos[G\pi x(n_1-1)] \\ &\ldots \\ &\sin[\pi x(n_1-M+1)], \cos[\pi x(n_1-M+1)], \ldots, \\ &\sin[G\pi x(n_1-M+1)], \cos[G\pi x(n_1-M+1)]\} \end{aligned} \tag{3}$$

2.3. Proposed exponentially extended RVFLN

This expansion involves sinusoids which attempt to model the nonlinearity in a nonlinear system. But this expansion cannot efficiently track the sudden amplitude change, especially when the change is very fast. In order to track the sudden amplitude change the expansion is modified which use exponentially varies sinusoids. The modified extended signal is given as:

$$\begin{aligned} &f(n_1) = \{1, x(n_1), x(n_1-1), \ldots, x(n_1-M+1) \\ &e^{-a|x(n_1)|}\sin[\pi x(n_1)], e^{-a|x(n_1)|}\cos[\pi x(n_1)], \ldots, \\ &e^{-a|x(n_1)|}\sin[G\pi x(n_1)], e^{-a|x(n_1)|}\cos[G\pi x(n_1)] \\ &e^{-a|x(n_1-1)|}\sin[\pi x(n_1-1)], e^{-a|x(n_1-1)|}\cos[\pi x(n_1-1)], \ldots, \\ &e^{-a|x(n_1-1)|}\sin[G\pi x(n_1-1)], e^{-a|x(n_1-1)|}\cos[G\pi x(n_1-1)] \\ &\ldots \\ &e^{-a|x(n_1-M+1)|}\sin[\pi x(n_1-M+1)], e^{-a|x(n_1-M+1)|}\cos[\pi x \\ &(n_1-M+1)], \ldots, e^{-a|x(n_1-M+1)|}\cos[G\pi x(n_1-M+1)]\}^T \end{aligned} \tag{4}$$

where a can be represented as an exponential parameter which is a constant. Different types of activation function can be used to model this Extended RVFLN model. In this case, we assume G = 1 and the Eq. (4) is reduced as follows:

$$\begin{aligned} &X = f(n_1) = \{1, x(n_1), x(n_1-1), \ldots, x(n_1-M+1) \\ &e^{-a|x(n_1)|}\sin[\pi x(n_1)], e^{-a|x(n_1)|}\cos[\pi x(n_1)] \\ &e^{-a|x(n_1-1)|}\sin[\pi x(n_1-1)], e^{-a|x(n_1-1)|}\cos[\pi x(n_1-1)] \\ &\ldots \\ &e^{-a|x(n_1-M+1)|}\sin[\pi x(n_1-M+1)], \\ &e^{-a|x(n_1-M+1)|}\cos[\pi x(n_1-M+1)]\}^T \end{aligned} \tag{5}$$

Generally, there are different types of activation functions have been used by the researchers. Here tanh(.) function has been taken as the activation function which is given in Eq. (6).

$$f(X) = \sum_{j=1}^{J} \beta_j \tanh\left(w_j^T X + b_j\right) \tag{6}$$

where X is the expanded input signal. The cost function is the sum of the squared errors and this can be represented as:

$$C = \sum_{i=1}^{L} \left\| \sum_{j=1}^{J} \beta_j h\left(w_j^T X_i + b_j\right) - y_i \right\|^2 \tag{7}$$

where the total number of extended inputs to the hidden layer is L. Further Eq. (7) can be represented as:

$$\arg\min \beta \, \|H\beta - y\|^2 \tag{8}$$

where

$$H = \begin{bmatrix} X_1 & h(w_1, b_1, X_1) & \ldots & \ldots & h\left(w_J, b_J, X_1\right) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ X_L & h(w_1, b_1, X_L) & \ldots & \ldots & h\left(w_J, b_J, X_L\right) \end{bmatrix}_{L\times J} \tag{9}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \ldots \\ \ldots \\ \beta_J \end{bmatrix}_{J\times 1} \quad \text{and} \quad y = \begin{bmatrix} y_1 \\ \ldots \\ \ldots \\ y_L \end{bmatrix}_{L\times 1} \tag{10}$$

Here the hidden layer output is H, the sampled output is y and the output weight is β.

$$\beta = \begin{cases} \left(H^T H + \dfrac{I}{C}\right)^{-1} H^T y\_L \geq J \\ H^T \left(H^T H + \dfrac{I}{C}\right)^{-1} y\_L < J \end{cases} \tag{11}$$

The Eq. (11) represents offline learning of the ERVFLN but this research work L is always less than J. So in this case the β value is represented as:

$$\beta = \left\{ H^T \left(H^T H + \frac{I}{C}\right)^{-1} y\_L < J \right. \tag{12}$$

$$h_1 = X(t) \tag{13}$$

$$h_2 = h_2(X(t)) = \left[h(\omega_1 X(t) + b_t) \ldots h\left(\omega_j X(t) + b_j\right)\right] \tag{14}$$

$$H_k = [H_1/H_2] \tag{15}$$

$$H_1 = [X(1) \ldots X(L)]^T \tag{16}$$

$$H_2 = \begin{bmatrix} h(\omega_1 X(1) + b_1) & \ldots & \ldots & \ldots & h\left(\omega_j X(1) + b_j\right) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ h(\omega_1 X(L) + b_1) & \ldots & \ldots & \ldots & h\left(\omega_j X(L) + b_j\right) \end{bmatrix}_{(L\times J)} \tag{17}$$

$$\begin{aligned} f(X) &= [h_1/h_2]\left[\frac{H_1^T}{H_2^T}\right]\left(\frac{I}{C} + [H_1/H_2]\left[\frac{H_1^T}{H_2^T}\right]\right)^{-1} Y \\ &= \left(h_1 H_1^T + h_2 H_2^T\right)\left(\frac{I}{C} + H_1 H_1^T + H_2 H_2^T\right)^{-1} Y \end{aligned} \tag{18}$$

The kernel matrix has been defined as follows:

$$\lambda = H_1 H_1^T \tag{19}$$

$$\widetilde{\lambda} = H_2 H_2^T \tag{20}$$

$$\lambda_{t,\tau} = (X(t), X(\tau)) = K(X(t), X(\tau)) \tag{21}$$

$$\widetilde{\lambda}_{t,\tau} = h_2(X(t)).h_2(X(\tau)) \tag{22}$$

$$\widetilde{\lambda}_{t,\tau} = \widetilde{K}\left(X\left(t\right).X\left(\tau\right)\right) \tag{23}$$

Two kernel functions have been used here one is polynomial kernel and the other one is Morlet wavelet kernel function. Both the kernel function is represented as follows:

Polynomial Kernel function:

$$K\left(X, X_i\right) = \left(\gamma\left(X.X_i\right) + r\right)^d \qquad \gamma > 0 \tag{24}$$

Morlet Wavelet Kernel function:

$$K\left(X, X_i\right) = \cos\left(\omega\frac{\|X - X_i\|}{b}\right) e^{\left(-\left(\frac{\|X-X_i\|^2}{f}\right)\right)} \tag{25}$$

Here, γ, b, e and f are real constant parameters.

The final output of KERVFLN is represented as:

$$f\left(X\left(t\right)\right) = \left(\begin{bmatrix} K\left(X\left(t\right), X\left(1\right)\right) \\ \cdots \\ \cdots \\ K\left(X\left(t\right), X\left(L\right)\right) \end{bmatrix} + \begin{bmatrix} \widetilde{K}\left(X\left(t\right), X\left(1\right)\right) \\ \cdots \\ \cdots \\ \widetilde{K}\left(X\left(t\right), X\left(L\right)\right) \end{bmatrix}\right) \times \left(\frac{I}{C} + \lambda + \widetilde{\lambda}\right)^{-1} Y \tag{26}$$

The value of γ, b, e and f are real constant parameters and are optimized using water cycle algorithm and the values are 1, 0.8, 1.5 and 1.2.

Water Cycle Algorithm (WCA):

The water cycle algorithm is based on the rain drops. It is basically based on the movement of water. The water cycle algorithm mimics the flow of rivers and streams toward the sea and was derived by observing the water cycle process. The initial population is selected randomly within a range known as the upper bound (UP) and lower bound (LB). The population of the streams is the initial population which is randomly generated after the raining process. The best stream which is having the minimum cost function is chosen as the sea. After that a random number of good streams those which are having cost function values close to the best value are chosen as rivers. All other remaining streams flow into the rivers and the sea. At first, a matrix of streams represents the initial population of the water cycle and the size is. This matrix is randomly generated where the rows represent the population size Npop, and column represent the number of design variables, D, respectively. At first, Npop streams are created. Those which are having minimum objective function is considered as sea.

The WCA algorithm is used to optimize different parameters of the kernel function in order to enhance the ability of the KRVFLN. Before initiating the optimization technique, an initial population matrix is created. Initially, different kernel parameters are randomly selected, which can be considered as the initial raindrops.

$$totalpopulation = \begin{bmatrix} Sea \\ River_1 \\ River_2 \\ River_3 \\ \cdots \\ \cdots \\ Stream_{Nsr} + 1 \\ Stream_{Nsr} + 2 \\ Stream_{Nsr} + 3 \\ \cdots \\ \cdots \\ Stream_{Npop} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \ldots & x_D^1 \\ x_1^2 & x_2^2 & x_3^2 & \ldots & x_D^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_1^{Npop} & x_2^{Npop} & x_3^{Npop} & \ldots & x_D^{Npop} \end{bmatrix} \tag{27}$$

Nsr is the total number of rivers plus one sea. Nstream is the rest of the population of the streams which are flowing into the river or directly flowing into the sea. Each stream carries different amount of water. The mathematical formulation of streams for each river and the sea is as follows

$$NS_n = round\left\{\left|\frac{Cost_n - Cost_{Nsr+1}}{\sum_{n-1}^{N_{sr}} C_n}\right| \times N_{streams}\right\} \tag{28}$$

where n = 1, 2, 3 …, Nsr

NSn is the number of streams which flow into the specific rivers and the sea. For the exploitation phase of the water cycle, new position of streams and rivers are generated as:

$$\vec{X}_{stream}\left(t+1\right) = \vec{X}_{stream}\left(t\right) + rand \times C \times \left(\vec{X}_{sea}\left(t\right) - \vec{X}_{stream}\left(t\right)\right) \tag{29}$$

$$\vec{X}_{stream}\left(t+1\right) = \vec{X}_{stream}\left(t\right) + rand \times C \times \left(\vec{X}_{River}\left(t\right) - \vec{X}_{stream}\left(t\right)\right) \tag{30}$$

$$\vec{X}_{River}\left(t+1\right) = \vec{X}_{River}\left(t\right) + rand \times C \times \left(\vec{X}_{sea}\left(t\right) - \vec{X}_{River}\left(t\right)\right) \tag{31}$$

Here t is the iteration index, and the best value of C is 2 where C lies between 1 and 2. If the solution given by a stream is more optimal than that of its connecting river, the positions of the river and the stream are exchanged and this is similar in case of river and the sea.

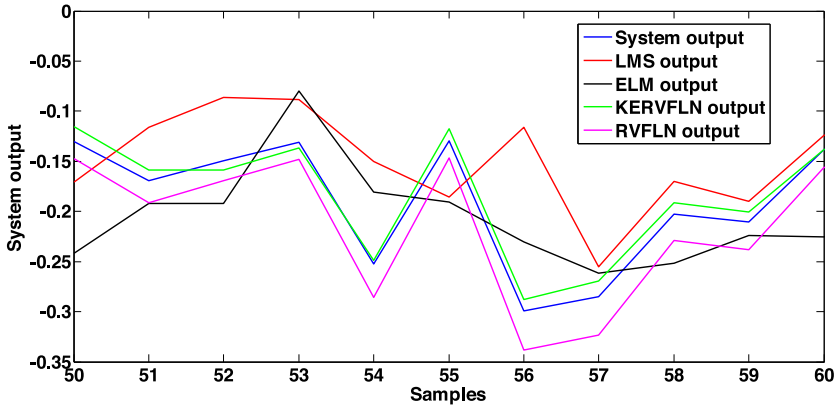
The revised position of the stream and river denotes the cost function of each raindrop. If the revised cost function of the river performs better than the connecting sea, then the initial river becomes the new sea and therefore the sea becomes river. The updated streams initially associated with the river are now incorporated with the sea; similar processes are followed between the sea, river, sea, and streams based on the discharge intensity. The second step of the WCA is the mutation, the increase in rain and evaporation process forms this step. This occurs as the streams and rivers connecting to the sea which increases the performance and capability of the method. A minute value is chosen that symbolizes the convergence of the stream or river along with the sea.

An evaporation operator has been used between a river and the sea which is represented as follows

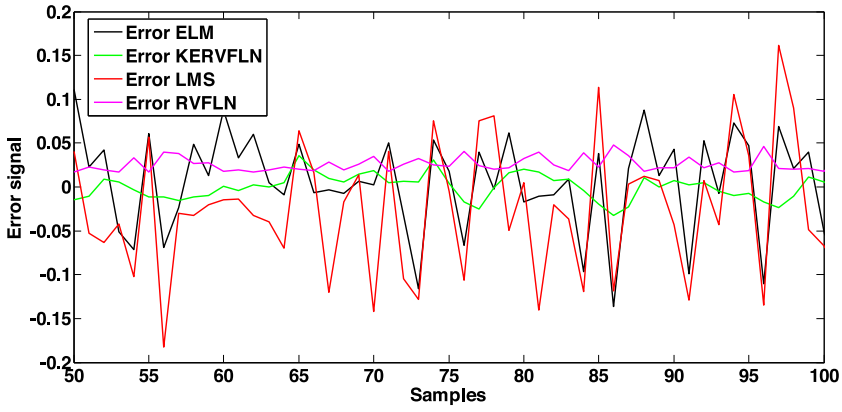
$$if\ \left\|\vec{X}_{Sea}^{t} - \vec{X}_{River}^{t}\right\| < d_{\max} \tag{32}$$

where rand < 0.1 and j = 1, 2, 3 …, Nsr-1. This operator is introduced to avoid premature convergence to local optima. After evaporation the raining process is applied and new streams are formed in different locations. Uniform random search is used to specify the new location of the newly formed streams. The search intensity near the sea is controlled by dmax. The value of dmax adaptively decreases as follows:

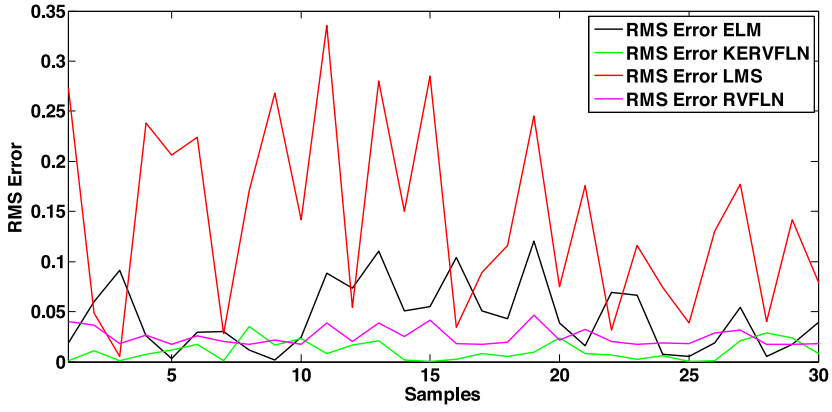
$$d_{\max}\left(t+1\right) = d_{\max}\left(t\right) - \frac{d_{\max}\left(t\right)}{Max.Iteration}$$
$$t = 1, 2, 3, \ldots, Max\_Iteration \tag{33}$$



(a)



(b)



(c)

Fig. 1. Outputs of the nonlinear system1 (a) comparison of the predicted output (b) comparison of the error signal (c) comparison of the RMS error plot of the proposed method with other established methods.

3. Performance evaluation of kernel based exponentially extended RVFLN in different non linear system identification

Monte Carlo Simulations:

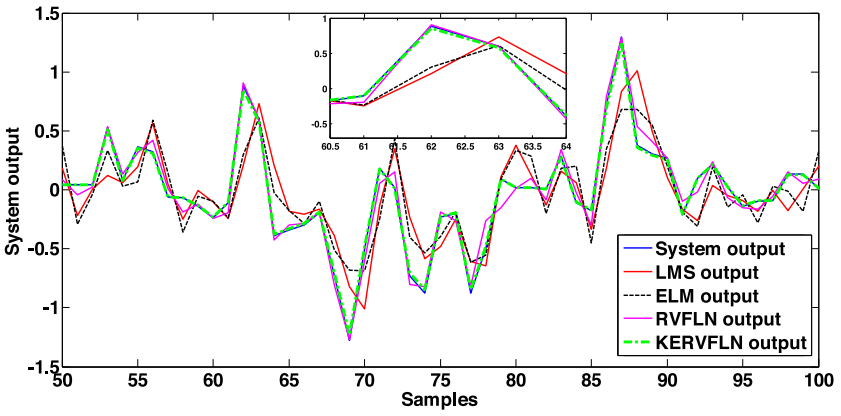
In this paper, different nonlinear model identification has been used to investigate the robustness and convergence of the proposed KERVFLN technique. In performance evaluation section, Monte Carlo simulations have been used. Four different types of nonlinear systems are considered in this section. All the systems are described as follows:

3.1. Nonlinear system 1

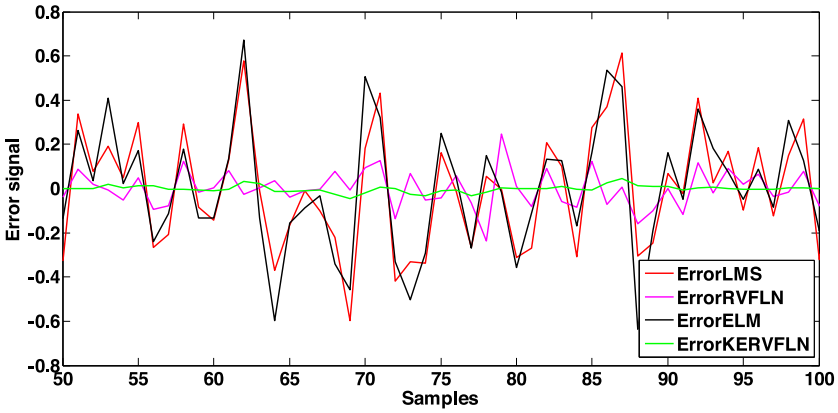
The mathematical formulation of the first nonlinear system to be given as follows:

$$Y(m) = \frac{4.0X^2(m) - 1.2X^2(m) - 3.0X(m) + 1.2}{0.4X^5(m) + 0.8X^4(m) - 1.2X^3(m) + 0.2X^3(m) - 3.0} \tag{34}$$

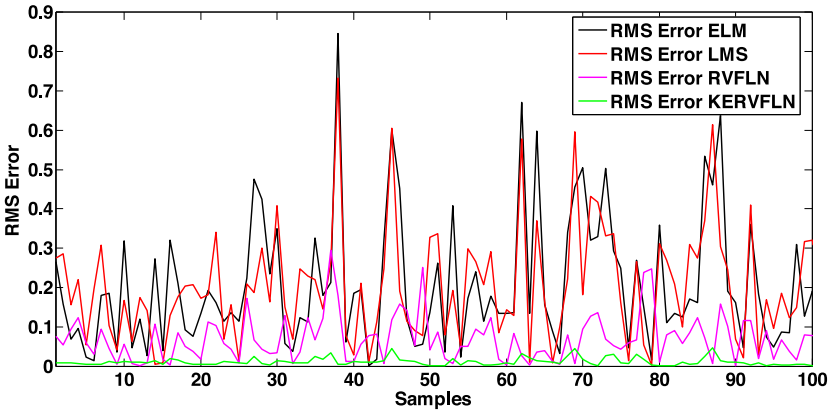
where X(m) is the random input signal and the range is [0, 1].



(a)



(b)



(c)

Fig. 2. Outputs of the nonlinear system 2 (a) comparison of the predicted output (b) comparison of the error signal (c) comparison of the RMS error plot of the proposed method with other established methods.

3.2. Nonlinear system 2

In this paper, the second nonlinear system is chosen as [46] which can be represented as:

$$Y(m) = \frac{Y(m-1)}{1+Y^2(m-1)} + X^3(m) \tag{35}$$

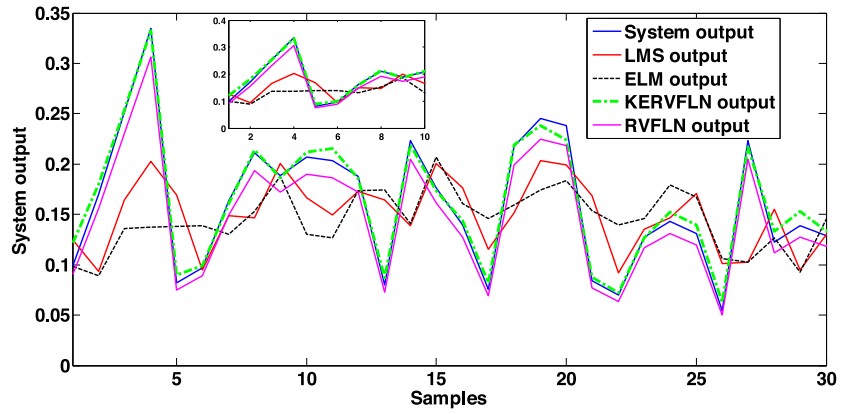
X(m) and Y(m) are the input signal and the output signal of the unknown nonlinear system, respectively. The input X(m) is a random signal which is uniformly distributed in the range of $[-1, 1]$.

3.3. Nonlinear system 3

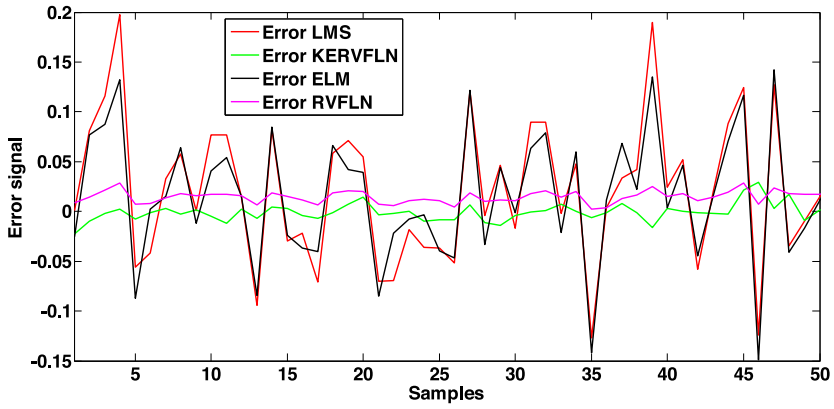
The relationship between the input and output of the third nonlinear system is as follows [47]:

$$Y(m) = 0.6\sin^3(X(m)) - \frac{2}{X^3(m)+2} - 0.1\cos(4\pi X(m-4)) + 1.125 \tag{36}$$

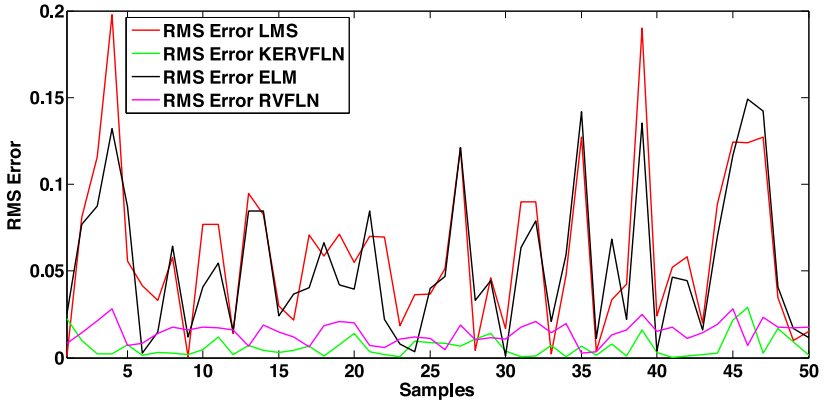
The input X(m) is randomly distributed signal in the range $[-0.5, 0.5]$.



(a)



(b)



(c)

Fig. 3. Outputs of the nonlinear system 3 (a) comparison of the predicted output (b) comparison of the error signal (c) comparison of the RMS error plot of the proposed method with other established methods.

3.4. Nonlinear system 4

The fourth nonlinear system is a memory less sigmoid function and the mathematical relationship of the system 4 is given as follows [48]:

$$Y(m) = \gamma\left(\frac{1}{1+\exp\left(-AB(m)\right)} - \frac{1}{2}\right) \tag{37}$$

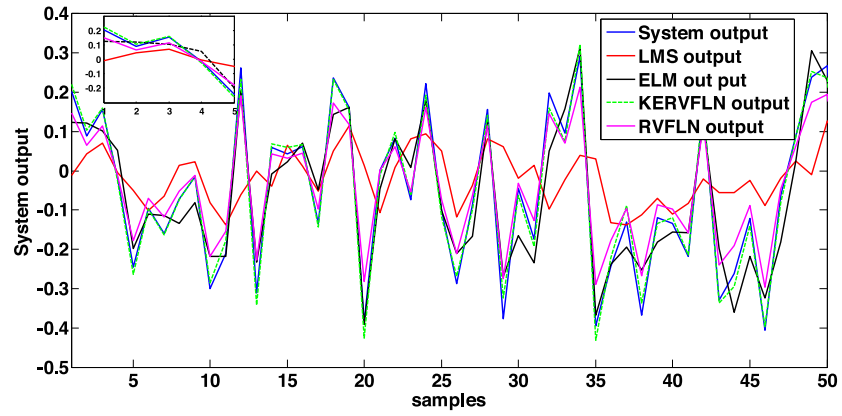
where γ is taken as 2 which is known as the gain parameter.

$$A = \begin{cases} 4 & if\ B(m) > 0 \\ 1 & \\ \frac{1}{2} & if\ B(m) \le 0 \end{cases} \tag{38}$$

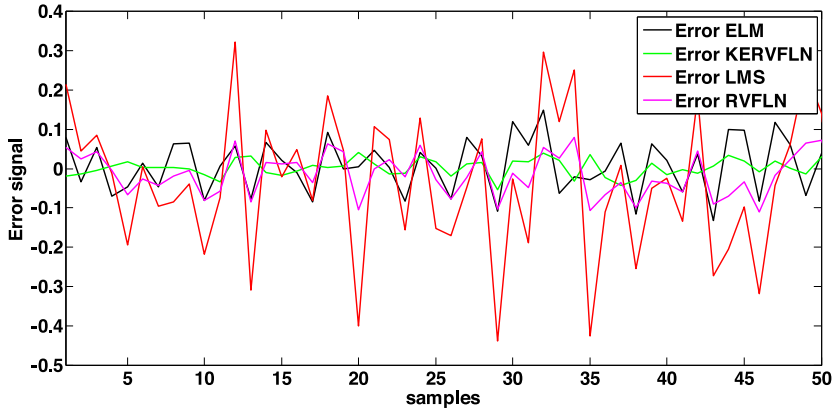
B(n) is taken as:

$$B(m) = \frac{3}{2}X(m) - \frac{3}{10}X^2(m) \tag{39}$$

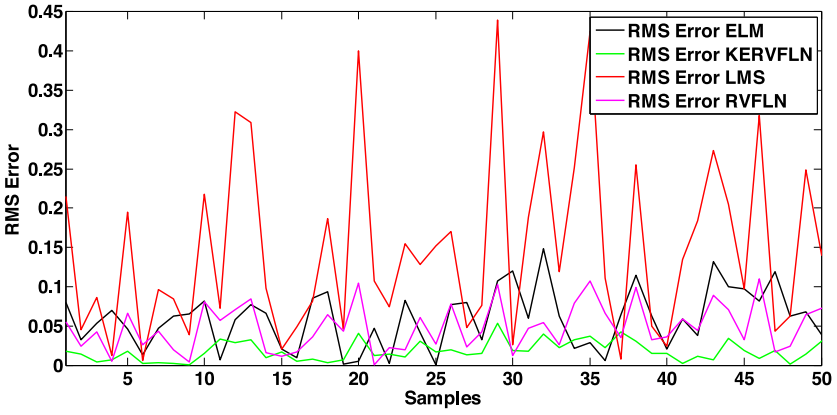
In this example X(m) is the same as system 2.



(a)



(b)



(c)

Fig. 4. Outputs of the nonlinear system 4 (a) comparison of the predicted output (b) comparison of the error signal (c) comparison of the RMS error plot of the proposed method with other established methods.

3.5. Non linear system 5

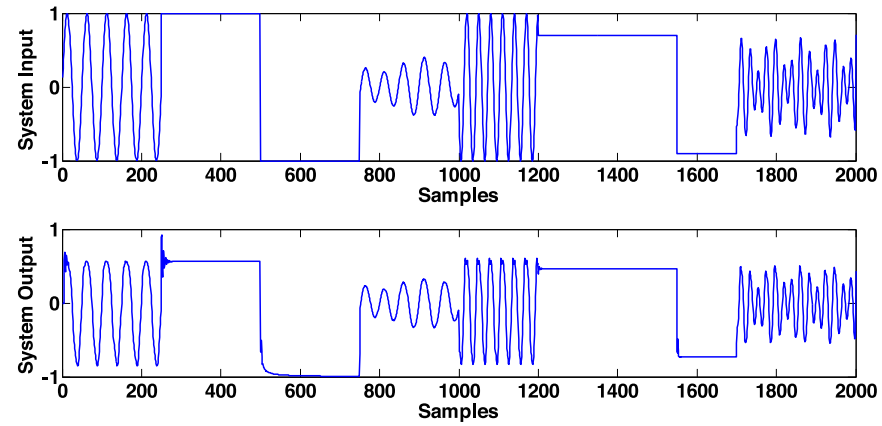
A single input single output (SISO) nonlinear dynamic system has been taken in this example and the mathematical representation is as follows [49]:

$$Y(m) = f[y(m-1), y(m-2), y(m-3), u(m-1), u(m-2)] \tag{40}$$

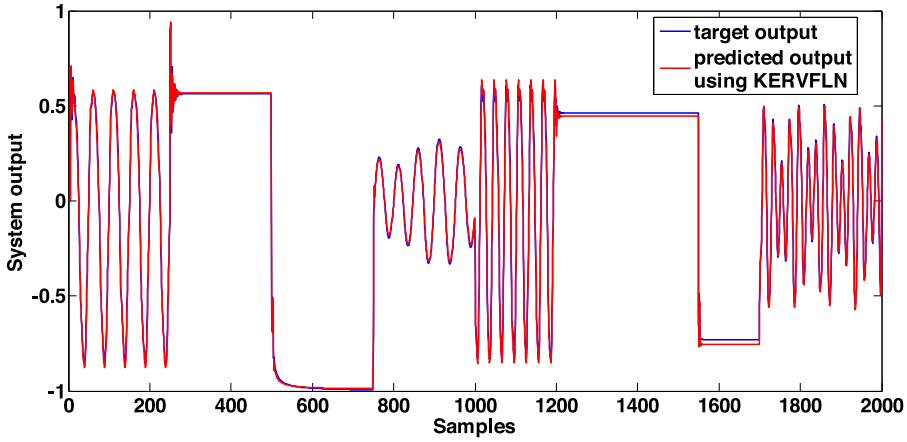
where,

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2} \tag{41}$$

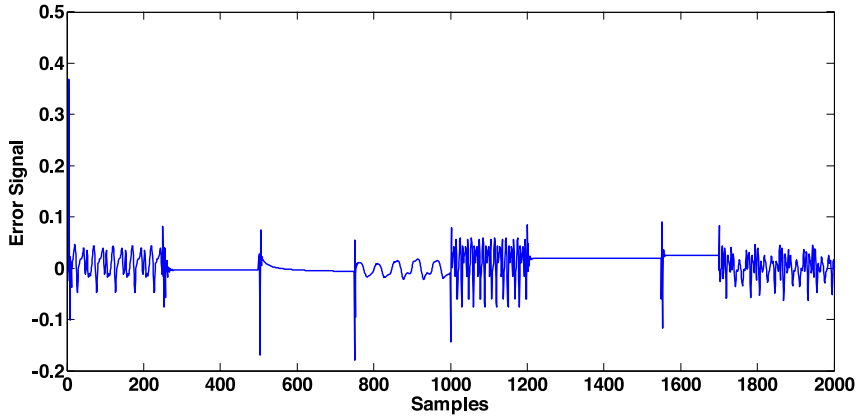
$$x_1 = u(m), x_2 = y(m-1), x_3 = u(m-1), x_4 = y(m-2) \tag{42}$$



(a)



(b)



(c)

Fig. 5. Outputs of the nonlinear SISO system 5 (a) System input system output (b) Comparison of the target output and the predicted output using KERVFLN (c) Error signal.

$$u(m) = \begin{cases} \sin\left(\frac{\pi m}{25}\right) & 0 \le m < 250 \\ 1.0 & 250 \le m < 500 \\ -1.0 & 500 \le m < 700 \\ 0.3\sin\left(\frac{\pi m}{25}\right) + 0.1\sin\left(\frac{\pi m}{32}\right) + 0.6\sin\left(\frac{\pi m}{10}\right) & 750 \le m < 1000 \end{cases} \tag{43}$$

Fig. 1(a) shows the comparison between the predicted output and the actual signal of the proposed technique KERVFLN with the other established techniques; they are original RVFLN, ELM, and LMS for nonlinear system 1. From Fig. 1(a) it can be said that KERVFLN is giving much better accuracy than the other methods. Figs. 2(a), 3(a) and 4(a) also describe the same as above for the nonlinear system 2, 3 and 4, respectively. Fig. 1(b) is the plot of the error signal of the proposed method RERVFLN with the other methods as described above. The difference between the predicted output and the actual signal is the error signal. The error is very less in case of KERVFLN and it is highest for LMS.

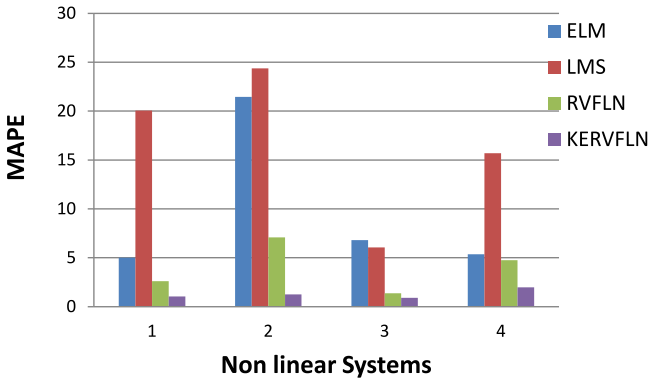


Fig. 6. MAPE value of different non linear systems with different techniques.

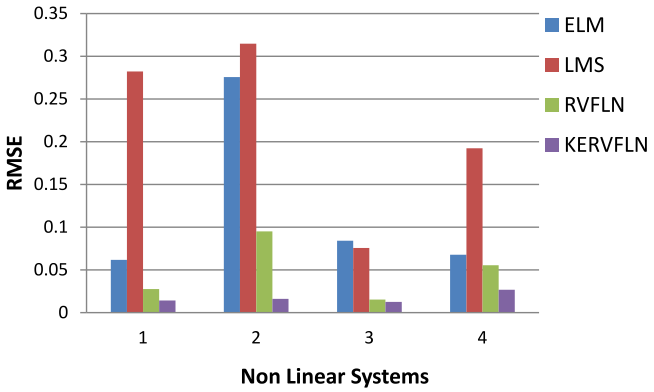


Fig. 7. RMSE values of different non linear systems with different techniques.

Figs. 2(b), 3(b) and 4(b) also describe the same as Fig. 1(b) for the nonlinear system 2, 3 and 4, respectively. Fig. 1(c) shows the RMSE plot of the KERVFLN, Original RVFLN, ELM, and LMS where the accuracy of KERVFLN is higher than others. Only in case of Fig. 3 the performance of the LMS shows better than the ELM but both are having much less accuracy than RVFLN and KERVFLN. Fig. 5(a) describes the input and output of the non linear SISO system. Fig. 5(b) describes the comparison of the target output and the predicted output using KERVFLN where it seems that the prediction accuracy of KERVFLN is very high. Fig. 5(c) represents the error signal which is the prediction error between the target output and the predicted output.

The performance of the proposed method kernel exponentially extended RVFLN (KERVFLN) has been tested through the performance matrices such as mean absolute percentage error (MAPE) and root mean square error (RMSE). All these performance matrices have been calculated between the absolute and the predicted values. The mathematical formulation of MAPE and RMSE are described as follows:

$$MAPE = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{|A_c\,(i) - A_P\,(i)|}{A_c\,(i)}\right) * 100 \tag{44}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|A_c\,(i) - A_P\,(i)|^2} \tag{45}$$

where N is the sample numbers, A_c is the actual signal from any system and A_p is the predicted signal from the proposed algorithm KERVFLN. Table 1 describes the comparative study of MAPE and RMSE values of the proposed method with other established methods; they are original RVFLN, ELM, and LMS for nonlinear system 1. The MAPE and RMSE of the proposed method KERVFLN is the lowest which is clear from Table 1. The MAPE and RMSE are highest in case of LMS technique which is 20.058 and 0.2822, respectively. With other nonlinear systems, it is clear from Tables 2–4 that the prediction efficiency of the proposed method is much higher than the other established methods. In case of nonlinear system 4, the MAPE of original RVFLN is 4.75 which is much higher than the MAPE value of KERVFLN is 1.9828. Figs. 6 and 7 describe the MAPE and RMSE values of different non linear systems with different algorithms. Basically it represents the clear view of Tables 1 to 4.

At the end, the proposed algorithm has been tested for real time non linear electronic circuit and the description of the data has been given in [50]. Fig. 8 describes the predicted out put of the proposed algorithm KERVFLN and RVFLN. It has been found that the proposed algorithm gives very good prediction accuracy even better than normal RVFLN. Fig. 8 (a) and (b) describe the predicted output with the system output. Fig. 8(c) describes the error signal and Fig (d) describes the RMS Error of the real time system output. The whole procedure has been described in a flowchart in Fig. 9 to have a clear view for the readers.

Table 1
Performance evaluation of nonlinear system 1.

	Techniques	MAPE	RMSE
System 1	ELM	4.9978	0.0618
	LMS	20.058	0.2822
	RVFLN	2.5992	0.0275
	KERVFLN	1.0499	0.0141

Table 2
Performance evaluation of nonlinear system 2.

	Techniques	MAPE	RMSE
System 2	ELM	21.4544	0.2757
	LMS	24.3780	0.3147
	RVFLN	7.0922	0.0951
	KERVFLN	1.2459	0.0160

Table 3
Performance evaluation of system 3.

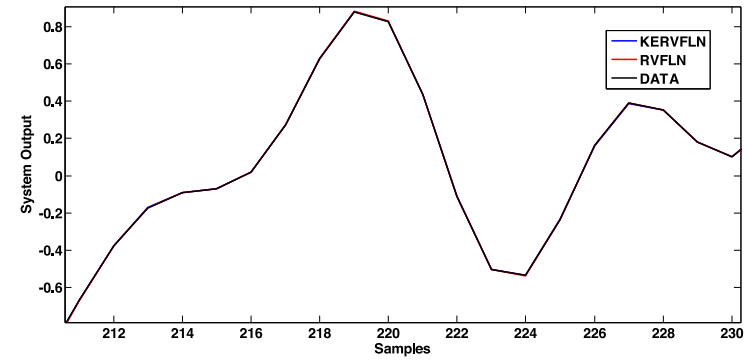
	Techniques	MAPE	RMSE
System 3	ELM	6.8003	0.0842
	LMS	6.0662	0.0756
	RVFLN	1.3594	0.0152
	KERVFLN	0.9071	0.0125

Table 4
Performance evaluation of system 4.

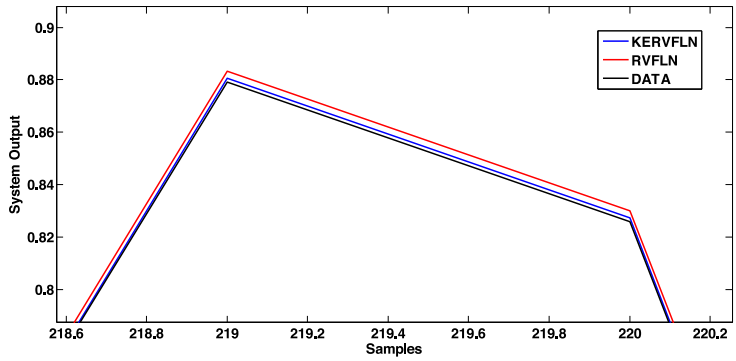
	Techniques	MAPE	RMSE
System 4	ELM	5.3472	0.0676
	LMS	15.694	0.1922
	RVFLN	4.750	0.0554
	KERVFLN	1.9828	0.0267

4. Conclusion

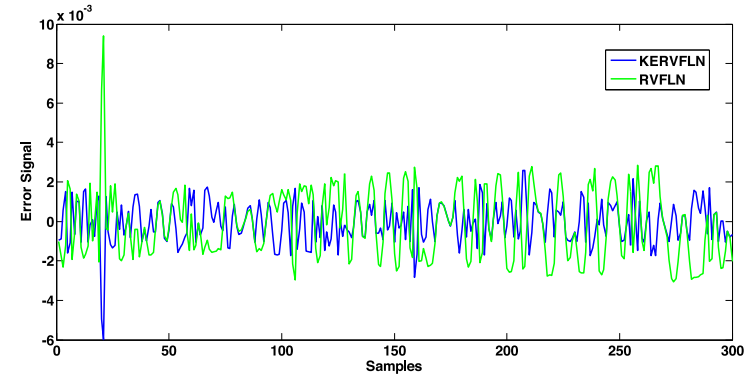
Most of the practical and industrial engineering plants exhibit nonlinearity due to its nonlinear boundary conditions, structural joints, and nonlinear material properties. Meanwhile, the presence of nonlinearity in the practical systems can head to a wide bound of structural behavior. Identification of complex nonlinear plants is a major concern in control system nowadays. The aim of this paper is the identification of nonlinear plans or nonlinear systems. A new kernel exponentially extended RVFLN has been developed in this paper for identification purpose. Different types of nonlinear systems are examined using the proposed



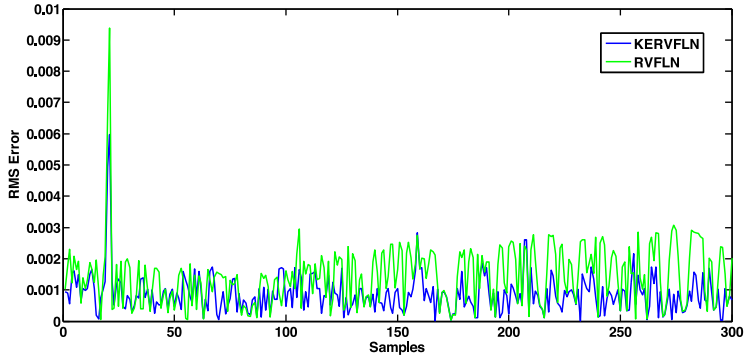
(a)



(b)



(c)



(d)

Fig. 8. Output of the real time non linear system (a) Comparison of system output with predicted output (b) Zoomed section of the predicted output (c) Error Signal (d) RMS Error using KERVFLN and RVFLN.

method KERVFLN to check the robustness, prediction accuracy and generalization efficiency. Real time system data also has been taken to explore the efficiency of the proposed model. The modeling efficiency of the proposed method has been compared with other established methods such as original RVFLN, ELM, and LMS algorithm in the performance evaluation section. The proposed methods givers much better prediction accuracy than all the other techniques as described above.

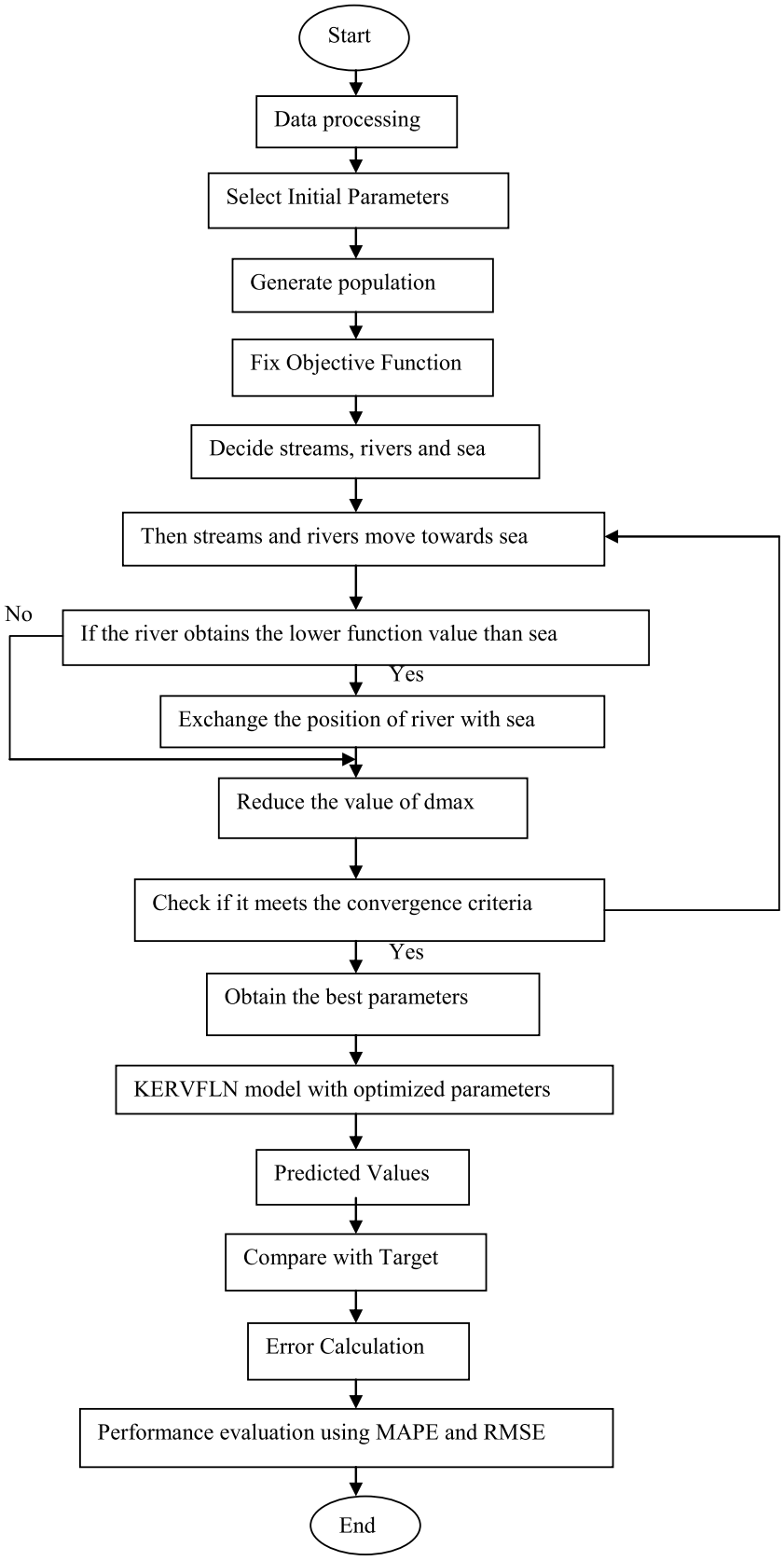


Fig. 9. Flowchart of the proposed model.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2020.106117.

CRediT authorship contribution statement

Tatiana Chakravorti: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation. **Penke Satyanarayana:** Conceptualization, Visualization.

References

[1] Yanjiao Wang, Feng Ding, Filtering-based iterative identification for multivariable systems, IET Control Theory Appl. 10 (8) (2016) 894–902.
[2] Liqian Dou, Ran Ji, Jingqi Gao, Identification of nonlinear aeroelastic system using fuzzy wavelet neural network, Neurocomputing 214 (2016) 935–943.
[3] Ilyes Boulkaibet, Khaled Belarbi, Sofiane Bououden, Tshilidzi Marwala, Mohammed Chadli, A new TS fuzzy model predictive control for nonlinear processes, Expert Syst. Appl. 88 (2017) 132–151.
[4] Tara Baldacchino, Keith Worden, Jennifer Rowson, Robust nonlinear system identification: Bayesian mixture of experts using the t-distribution, Mech. Syst. Signal Process. 85 (2017) 977–992.
[5] Feng Ding, Dandan Meng, Jiyang Dai, Qishen Li, Ahmed Alsaedi, Tasawar Hayat, Least squares based iterative parameter estimation algorithm for stochastic dynamical systems with ARMA noise using the model equivalence, Int. J. Control Autom. Syst. 16 (2) (2018) 630–639.
[6] Chengpu Yu, Keyou You, Lihua Xie, Quantized identification of ARMA systems with colored measurement noise, Automatica 66 (2016) 101–108.
[7] Wen-chuan Wang, Kwok-wing Chau, Dong-mei Xu, Xiao-Yun Chen, Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition, Water Resour. Manage. 29 (8) (2015) 2655–2675.
[8] Jui-Sheng Chou, Ngoc-Tri Ngo, Time series analytics using sliding window metaheuristic optimization-based machine learning system for identifying building energy consumption patterns, Appl. Energy 177 (2016) 751–770.
[9] Shuen Wang, Zhenzhen Han, Fucai Liu, Yinggan Tang, Nonlinear system identification using least squares support vector machine tuned by an adaptive particle swarm optimization, Int. J. Mach. Learn. Cybern. 6 (6) (2015) 981–992.
[10] Michael Feldman, Simon Braun, Nonlinear vibrating system identification via Hilbert decomposition, Mech. Syst. Signal Process. 84 (2017) 65–96.
[11] Navid Vafamand, Mohammad Mehdi Arefi, Alireza Khayatian, Nonlinear system identification based on takagi-sugeno fuzzy modeling and unscented kalman filter, ISA Trans. 74 (2018) 134–143.
[12] Vincent Laurain, Roland Tóth, Dario Piga, Wei Xing Zheng, An instrumental least squares support vector machine for nonlinear system identification, Automatica 54 (2015) 340–347.
[13] Wei-Chiang Hong, Ming-Wei Li, Jing Geng, Yang Zhang, Novel chaotic bat algorithm for forecasting complex motion of floating platforms, Appl. Math. Model. 72 (2019) 425–443.
[14] Guo-Feng Fan, Li-Ling Peng, Wei-Chiang Hong, Short term load forecasting based on phase space reconstruction algorithm and bi-square kernel regression model, Appl. Energy 224 (2018) 13–33.
[15] Yongquan Dong, Zichen Zhang, Wei-Chiang Hong, A hybrid seasonal mechanism with a chaotic cuckoo search algorithm with a support vector regression model for electric load forecasting, Energies 11 (4) (2018) 1009.
[16] Dalia Yousri, Dalia Allam, M.B. Eteiba, Ponnuthurai Nagaratnam Suganthan, Static and dynamic photovoltaic models' parameters identification using chaotic heterogeneous comprehensive learning particle swarm optimizer variants, Energy Convers. Manage. 182 (2019) 546–563.
[17] Swagatam Das, Sankha Subhra Mullick, Ponnuthurai N. Suganthan, Recent advances in differential evolution–an updated survey, Swarm Evol. Comput. 27 (2016) 1–30.
[18] W. Chine, A. Mellit, V. Lughi, A. Malek, G. Sulligoi, A. Massi Pavan, A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks, Renew. Energy 90 (2016) 501–512.
[19] Rasit Ata, Artificial neural networks applications in wind energy systems: a review, Renew. Sustain. Energy Rev. 49 (2015) 534–562.
[20] Abinet Tesfaye Eseye, Jianhua Zhang, Dehua Zheng, Short-term photovoltaic solar power forecasting using a hybrid wavelet-PSO-SVM model based on SCADA and meteorological information, Renew. Energy 118 (2018) 357–367.
[21] Lei Yang, Miao He, Junshan Zhang, Vijay Vittal, Support-vector-machine-enhanced markov model for short-term wind power forecast, IEEE Trans. Sustain. Energy 6 (3) (2015) 791–799.
[22] Ye Ren, P.N. Suganthan, Narasimalu Srikanth, A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods, IEEE Trans. Sustain. Energy 6 (1) (2014) 236–244.
[23] Ye Ren, Ponnuthurai N. Suganthan, Narasimalu Srikanth, Gehan Amaratunga, Random vector functional link network for short-term electricity load demand forecasting, Inform. Sci. 367 (2016) 1078–1093.
[24] Cyril Voyant, Marc Muselli, Christophe Paoli, Marie-Laure Nivet, Numerical weather prediction (NWP) and hybrid ARMA/ANN model to predict global radiation, Energy 39 (1) (2012) 341–355.
[25] Muhammad Naveed Akhter, Saad Mekhilef, Hazlie Mokhlis, Noraisyah Mohamed Shah, Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques, IET Renew. Power Gener. 13 (7) (2019) 1009–1023.
[26] Seyed Amin Bagherzadeh, Nonlinear aircraft system identification using artificial neural networks enhanced by empirical mode decomposition, Aerosp. Sci. Technol. 75 (2018) 155–171.
[27] Mostafa Mehdipour Ghazi, Berrin Yanikoglu, Erchan Aptoula, Plant identification using deep neural networks via optimization of transfer learning parameters, Neurocomputing 235 (2017) 228–235.
[28] Satchidananda Dehuri, Sung-Bae Cho, A comprehensive survey on functional link neural networks and an adaptive PSO–BP learning for CFLNN, Neural Comput. Appl. 19 (2) (2010) 187–205.
[29] Mrutyunjaya Sahani, Pradipta Kishore Dash, FPGA-based online power quality disturbances monitoring using reduced-sample HHT and class-specific weighted RVFLN, IEEE Trans. Ind. Inf. 15 (8) (2019) 4614–4623.
[30] Yoh-Han Pao, Gwang-Hoon Park, Dejan J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (2) (1994) 163–180.
[31] Boris Igelnik, Yoh-Han Pao, Stochastic choice of basis functions in adaptive function approximation and the functional-link net, IEEE Trans. Neural Netw. 6 (6) (1995) 1320–1329.
[32] Cheng Lian, Lingzi Zhu, Zhigang Zeng, Yixin Su, Wei Yao, Huiming Tang, Constructing prediction intervals for landslide displacement using bootstrapping random vector functional link networks selective ensemble with neural networks switched, Neurocomputing 291 (2018) 1–10.
[33] Ali Sadollah, Hadi Eskandar, Ardeshir Bahreininejad, Joong Hoon Kim, Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems, Appl. Soft Comput. 30 (2015) 58–71.
[34] Ali Sadollah, Hadi Eskandar, Joong Hoon Kim, Water cycle algorithm for solving constrained multi-objective optimization problems, Appl. Soft Comput. 27 (2015) 279–298.
[35] Ali Sadollah, Hadi Eskandar, Ardeshir Bahreininejad, Joong Hoon Kim, Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems, Appl. Soft Comput. 30 (2015) 58–71.
[36] Arvind Yadav, Snehamoy Chatterjee, Sk Md. Equeenuddin, Prediction of suspended sediment yield by artificial neural network and traditional mathematical model in Mahanadi river basin, India, Sustain. Water Resour. Manage. 4 (4) (2018) 745–759.
[37] Arvind Yadav, Snehamoy Chatterjee, Sk Md. Equeenuddin, Suspended sediment yield estimation using genetic algorithm-based artificial intelligence models: case study of Mahanadi river, India, Hydrol. Sci. J. 63 (8) (2018) 1162–1182.
[38] Huaiguang Jiang, Yingchen Zhang, Eduard Muljadi, Jun Jason Zhang, David Wenzhong Gao, A short-term and high-resolution distribution system load forecasting approach using support vector regression with hybrid parameters optimization, IEEE Trans. Smart Grid 9 (4) (2016) 3341–3350.
[39] Jianzhou Wang, He Jiang, Yujie Wu, Yao Dong, Forecasting solar radiation using an optimized hybrid model by cuckoo search algorithm, Energy 81 (2015) 627–644.
[40] Anula Khare, Saroj Rangnekar, A review of particle swarm optimization and its applications in solar photovoltaic system, Appl. Soft Comput. 13 (5) (2013) 2997–3006.
[41] Kang-Kang Xu, Han-Xiong Li, Hai-Dong Yang, Kernel-based random vector functional-link network for fast learning of spatiotemporal dynamic processes, IEEE Trans. Syst. Man Cybern.: Syst. 49 (5) (2017) 1016–1026.
[42] F. Cao, Y. Tan, M. Cai, Sparse algorithms of random weight networks and applications, Expert Syst. Appl. 41 (5) (2014) 2457–2462.
[43] Y. He, X. Wang, J. Huang, Fuzzy nonlinear regression analysis using a random weight network, Inform. Sci. 364 (2016) 222–240.
[44] W. Dai, Q. Liu, T. Chai, Particle size estimate of grinding processes using random vector functional link networks with improved robustness, Neurocomputing 169 (2) (2015) 361–372.
[45] Ponnuthurai Nagaratnam Suganthan, On non-iterative learning algorithms with closed-form solution, Appl. Soft Comput. 70 (2018) 1078–1082.
[46] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Trans. Neural Netw. 1 (1) (1990) 4–27.

[47] V. Patel, V. Gandhi, S. Heda, N.V. George, Design of adaptive exponential functional link network-based nonlinear filters, IEEE Trans. Circuits Syst. I 63 (9) (2016) 1434–1442.
[48] D. Comminiello, M. Scarpiniti, L.A. Azpicueta-Ruiz, J. Arenas-García, A. Uncini, Functional link adaptive filters for nonlinear acoustic echo cancellation, IEEE Trans. Audio Speech Lang. Process. 21 (7) (2013) 1502–1512.
[49] Chia-Feng Juang, Chin-Teng Lin, A recurrent self-organizing neural fuzzy inference network, IEEE Trans. Neural Netw. 10 (4) (1999) 828–845.
[50] Torbjörn Wigren, Johan Schoukens, Three free data sets for development and benchmarking in nonlinear system identification, in: 2013 European Control Conference (ECC), IEEE, 2013, pp. 2933–2938.