ARTICLE IN PRESS
FSS:7338
Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems ••• (••••) •••–•••
FUZZY sets and systems
www.elsevier.com/locate/fss

Fuzzy wavelet extreme learning machine

Pegah Golestaneh, Maryam Zekri *, Farid Sheikholeslam

Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 8415683111, Iran

Received 30 May 2016; received in revised form 23 August 2017; accepted 5 December 2017

Abstract

Incorporating the time-frequency localization properties of wavelets and the learning abilities of neural network (NN), the approximate reasoning characteristics of fuzzy inference system and the advantages of ELM (one-pass learning and good generalization performance at extremely fast learning speed) can exhibit their characteristics to reveal an effective solution in many applications. Following that, this paper presents a novel structure called fuzzy wavelet extreme learning machine (FW-ELM). The main objectives of FW-ELM are to significantly reduce the network complexity by reducing the number of linear learning parameters, and to decrease the sensitivity to random initialization procedure while the acceptable accuracy and generalization performances are preserved. In the proposed structure, each fuzzy rule corresponds to a sub-wavelet neural network and consists of wavelets with different dilations and translations. In this model, in order to achieve a balance between network complexity and performance accuracy, in the THEN-part of each fuzzy rule, one coefficient is considered for each two inputs. In this work, first, the equivalence of an FW model and an SLFN is proved and then ELM can be directly applied to the model. All free parameters of membership functions and wavelet coefficients are generated randomly, and only the output weights are determined analytically using a one-pass learning method. To evaluate FW-ELM, it is compared with popular fuzzy models like OS-Fuzzy-ELM, Simpl_eTS, ANFIS and several other relevant algorithms such as ELM, BP and SVR on various benchmark datasets. Simulation results demonstrate the remarkable efficiency resulting from the proposed approach. Performance accuracy of FW-ELM is shown to be comparable with OS-Fuzzy-ELM and better than the rest of the well-known methods.

Keywords: Fuzzy wavelet (FW) model; Extreme learning machine (ELM); Wavelet neural network (WNN)

1. Introduction

Soft computing techniques such as artificial neural networks (ANN) [1], Fuzzy Models [2] and wavelet networks [3] have been applied in large variety of areas such as artificial intelligence, pattern recognition, robotics, system identification and so on [4–8]. Among ANNs, feed forward neural network (FFNN) is one of the mostly used learning mechanisms [9,10]. Although FFNNs have learning and generalization abilities, they suffer from network complexity because they may require a large number of neurons [11]. By combining the benefits of neural network (learning

* Corresponding author.
E-mail addresses: p.golestaneh@ec.iut.ac.ir (P. Golestaneh), mzekri@cc.iut.ac.ir (M. Zekri), sheikh@cc.iut.ac.ir (F. Sheikholeslam).

https://doi.org/10.1016/j.fss.2017.12.006
0165-0114/© 2017 Elsevier B.V. All rights reserved.

ability) and fuzzy inference system (FIS), fuzzy neural network (FNN) has been developed [12–15]. FNN employs the learning ability of neural network and the approximate reasoning characteristics of FIS to present a technique to deal with complex problems, which include uncertain or ill-defined factors [11,15].

Multiresolution and time frequency localization properties of wavelets make them powerful approaches to solve difficult problems in the fields of mathematics, physics and engineering [16–19]. By inspiring neural network, fuzzy theory and wavelet transform, fuzzy wavelet neural network (FWNN) has been constructed [20]. FWNNs can improve model accuracy, generalization ability and computational power of neuro-fuzzy systems [21–23]. In FWNNs, the consequent part of each fuzzy rule includes a linear finite set of wavelet functions. A fuzzy wavelet network consists of three types of parameters: wavelet coefficients, parameters of membership functions, and the output layer weights. These parameters should be initialized using an appropriate method, and then they should be updated during training steps of the network [24–29]. In [25], to find unknown parameters of the network, a fast gradient-based-training algorithm, i.e., the Broyden–Fletcher–Gold Farb–Shanno method is used. In [27], an adaptive FWNN was proposed for the control of affine nonlinear systems, in which TSK (Takagi Sugeno Kang) fuzzy model and WNN are combined. The network size, the number of fuzzy rules, the number of wavelets in each sub-WNN and initial weights are determined in a reasonable number of iterations by OLS (Ordinary Least Squares) algorithm, and the algorithm provides good initializations for the learning procedure.

In [28], the algorithm gives the initial parameters by the clustering algorithm and then updates them with a combination of back propagation and recursive least-squares methods. In [29], initially, the antecedent parameters are selected as small random values, the initial consequent layer parameters are determined by least square methods, and training is done using one of the most popular quasi-Newton algorithms known as DFP algorithm.

Generally, in FWNNs, finding efficient methods for parameter initialization and iterative learning step is one of the main issues. Not only these factors are important for determining the speed of training algorithms, but also affect the accuracy of the model.

It is to be noted that all the above-mentioned neural networks are trained by iterative learning methods, and most of them use gradient-based learning algorithms such as back propagation (BP). In spite of good accuracy provided by these algorithms, they suffer from several drawbacks such as finding an appropriate method for adjusting and updating parameters, long computation time, and local optima [30]. To overcome these problems, in [30], Huang–Bin has been presented a method for training a single hidden layer feed forward neural network (SLFN) called extreme learning machine (ELM). In contrast to traditional learning algorithms such as BP, in ELM, parameters of the hidden layer and weights between the hidden layer and the input layer need not be tuned and are assigned randomly for once, and remain fixed throughout learning procedure. The output weights, which link the hidden layer and the output layer, are determined analytically.

According to conventional neural network theories, SLFNs have universal approximation properties when all the parameters of the network are allowed to be adjustable. Ref. [31] in an incremental constructive method proves that input weights and hidden layer biases of SLFNs need not be tuned. Consequently, SLFNs can also work as universal approximators with randomly selected hidden nodes and adjustable output weights.

A number of extensions of ELM have been deployed in various learning fields [32–35]. Huang et al. proposed an online sequential learning machine (OS-ELM) for function approximation and classification of real-world problems [34]. In [35], Huang et al. extended OS-ELM to OS-Fuzzy-ELM, which is based on TSK fuzzy model. Training procedure of OS-Fuzzy-ELM can be done by loading the data chunk by chunk or one by one. In [35], it is shown that efficiency and computation time of OS-Fuzzy-ELM are similar to or better than other well-known learning algorithms such as (ANFIS-Full, ANFIS-Reduced and simplified eTS algorithms (Simpl_eTS) [35]).

Various combination of WNNs and ELM have been proposed in prognostic, health monitoring, forecasting, identification problems, etc. [36–39]. In [36], the authors have proposed summation wavelet-ELM (SW-ELM), which uses summation of an inverse hyperbolic Sine and a Morlet wavelet function as activation function of the hidden layer nodes and Nguyen Widrow method to reduce the impact of a random initialization procedure. In [37], to perform estimation/prediction tasks, features are used to build a model with SW-ELM algorithm. In [38], an automatic system was proposed for target recognition based on wavelet extreme learning machine. In [39], to improve forecasting accuracy, a hybrid wavelet-ELM model was developed. It is noteworthy that in the most extensions of ELM, network complexity and impact of parameter randomness are important issues and should be considered.

To overcome the aforementioned problems, this paper presents a novel structure of fuzzy wavelet model based on theory of multiresolution analysis (MRA) of wavelet transforms, fuzzy concepts and ELM. This method is called fuzzy

Please cite this article in press as: P. Golestaneh et al., Fuzzy wavelet extreme learning machine, Fuzzy Sets Syst. (2017), https://doi.org/10.1016/j.fss.2017.12.006

wavelet extreme learning machine (FW-ELM). In the proposed approach, incorporating the time–frequency localization properties of wavelets, the learning abilities of neural network (NN), the approximate reasoning characteristics of fuzzy inference system and also the advantages of ELM (one-pass learning and good generalization performance at extremely fast learning speed) can exhibit their characteristics to reveal an effective solution in many applications. The goal of the combination of fuzzy wavelet network and ELM is to reduce the complexity of the network and to decrease the impact of a random initialization procedure while preserving the performance accuracy. In order to apply ELM algorithm to the proposed fuzzy wavelet model, first equivalence of an SLFN with an FW model is proved. Furthermore, in the presented model, first, all parameters of membership functions and wavelet coefficients are randomly assigned and then the output weights in the consequent part of each rule are analytically determined through a least square solution using a one-pass learning method. In FW-ELM, each fuzzy rule includes sigmoid membership function in IF-part and Mexican hat wavelet function in THEN-part.

To evaluate the performance of FW-ELM, it is compared with other well-known methods using real machine learning benchmark problems in function approximation, regression, classification, identification of nonlinear systems and time series prediction.

These methods can categories as follows:

- Methods based on ELM (ELM [30], OS-F-ELM [35], CFWNN-ELM [40] and WNN-ELM [40])
- Fuzzy Wavelet Models based on iterative methods such as BP (Ref. [5], Ref. [11], Ref. [20], Ref. [23], Ref. [25] and Refs. [27–29])
- Fuzzy models based on iterative methods.

The simulation results demonstrate that the performance accuracy of FW-ELM method is similar to OS-Fuzzy-ELM and better than other mentioned algorithms. The main features of this paper are as follows:

1- In this study, to establish an appropriate tradeoff between network complexity and performance accuracy, a new structure of FW model was proposed, in which, in the THEN-part of each fuzzy rule, one coefficient is considered for each two inputs.
2- By combining the advantages of NN, FIS, wavelet theory and ELM, FW-ELM is able to have good performance accuracy, while the number of the consequent weight parameters are considerably reduced and sensitivity to parameter randomness is decreased.
3- In this algorithm, each fuzzy rule corresponds to sub-wavelets with different translation and dilation values rather than using constants or linear equations as in TSK model. Thus, the sub-WNNs at different dilation values (i.e., different resolution levels) are used to capture different behaviors (global or local) of the learning functions. In fact, the mentioned explanations emphasize the difference between our approach and OS-Fuzzy-ELM.
4- Unlike FWNNs with iterative learning algorithms (the parameters of the network should be tuned iteratively), our structure uses only one-pass learning algorithm (at extremely fast learning speed) to analytically determine the linear learning parameters of consequent parts of fuzzy rules.
5- In FWNNs, finding an appropriate method for initialization procedure is an important issue, and an effective method should be selected to obtain acceptable performance accuracy, while in our approach, all free parameters of membership functions and wavelet coefficients are generated randomly and remain fixed.

The rest of this paper is organized as follows. A brief background of FWNN and ELM is explained in Section 2. Section 3 demonstrates the equivalence of an SLFN with a fuzzy wavelet model. Then, fuzzy wavelet structures are introduced and in the following, FW-ELM method based on proposed fuzzy wavelet model is described. In Section 4, to verify the performance and efficiency of FW-ELM, simulations are carried out on real-world benchmark datasets. Finally, conclusions are given in Section 5.

2. Background: FWNN and ELM

The structure of FWNN and ELM algorithm are introduced in this section.

Please cite this article in press as: P. Golestaneh et al., Fuzzy wavelet extreme learning machine, Fuzzy Sets Syst. (2017), https://doi.org/10.1016/j.fss.2017.12.006

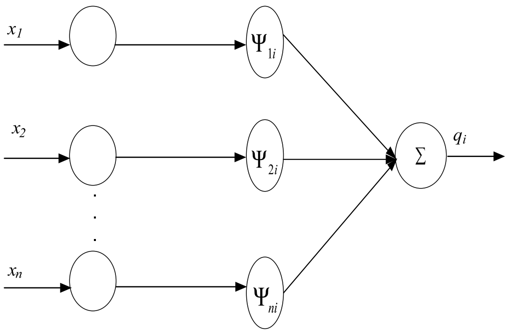

Fig. 1. Structure of WNN [28].

2.1. Structure of FWNNs

By integrating fuzzy inference system (TSK model) with WNN, the structure of FWNN is constructed. This network can be expressed by a set of fuzzy rules as follows [11,28]:

$$R^i : \text{IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and } \dots \text{ and } x_n \text{ is } A_n^i \qquad \text{THEN } y_i = q_i \sum_{j=1}^{n} \Psi_{ij}(x_j) \quad (i = 1 : L) \tag{1}$$

where L and n is the number of fuzzy rules and the dimension of input vector, respectively. $\boldsymbol{x} = [x_1, x_2, \dots, x_n]^T$ is the input vector and y_i $(1 \le i \le L)$ is the output of the ith rule, which is equal to a linear combination of finite set of wavelets.

A_j^i is a membership function for jth input of the ith rule. It can include Gaussian functions, Sigmoid functions, B-Spline functions of different order, etc.

Wavelets are waveforms with limited duration and zero mean value with local properties in both time space and frequency space. The structure of each wavelet neural network (WNN) with n inputs and one output is described in Fig. 1.

An WNN is a neural network in which wavelet functions act as activation functions. The output of the wavelet neural network is given as:

$$y_i = q_i \sum_{j=1}^{n} \Psi_{ij}(x_j) \tag{2}$$

where n is the dimension of the input vector and q_i is the weight coefficient between the inputs and the ith output. By dilation and translation of a mother wavelet function $\psi(.)$, Ψ_{ij} is obtained as follows:

$$\Psi_{ij}(x_j) = \psi\left(\frac{x_j - b_{ij}}{d_{ij}}\right), \quad d_{ij} \neq 0. \tag{3}$$

In (3), d_{ij} and b_{ij} are free wavelet parameters that should be initialized and updated.

It can be noticed that in each WNN, only one weight coefficient is used for each rule; therefore the number of linear parameters is equal to that of rules. According to (1), the overall output of the FWNN structure is given in Fig. 2 and is defined as (4)

$$\hat{y} = \frac{\sum_{i=1}^{L} [\prod_{j=1}^{n} A_j^i(x_j)] y_i}{\sum_{i=1}^{L} [\prod_{j=1}^{n} A_j^i(x_j)]}. \tag{4}$$

It should be noted that this FWNN structure is an n input and one output model. This model can be extended to n inputs and m outputs ($\boldsymbol{y_i} = [y_{i1}, \dots, y_{ik}]^T$, $i = 1, \dots, L$, $k = 1, \dots, m$).

In FWNNs, parameter initialization procedure and finding algorithms for updating parameters are very important factors, which affect the speed of training algorithms and accuracy of the model.

Please cite this article in press as: P. Golestaneh et al., Fuzzy wavelet extreme learning machine, Fuzzy Sets Syst. (2017), https://doi.org/10.1016/j.fss.2017.12.006

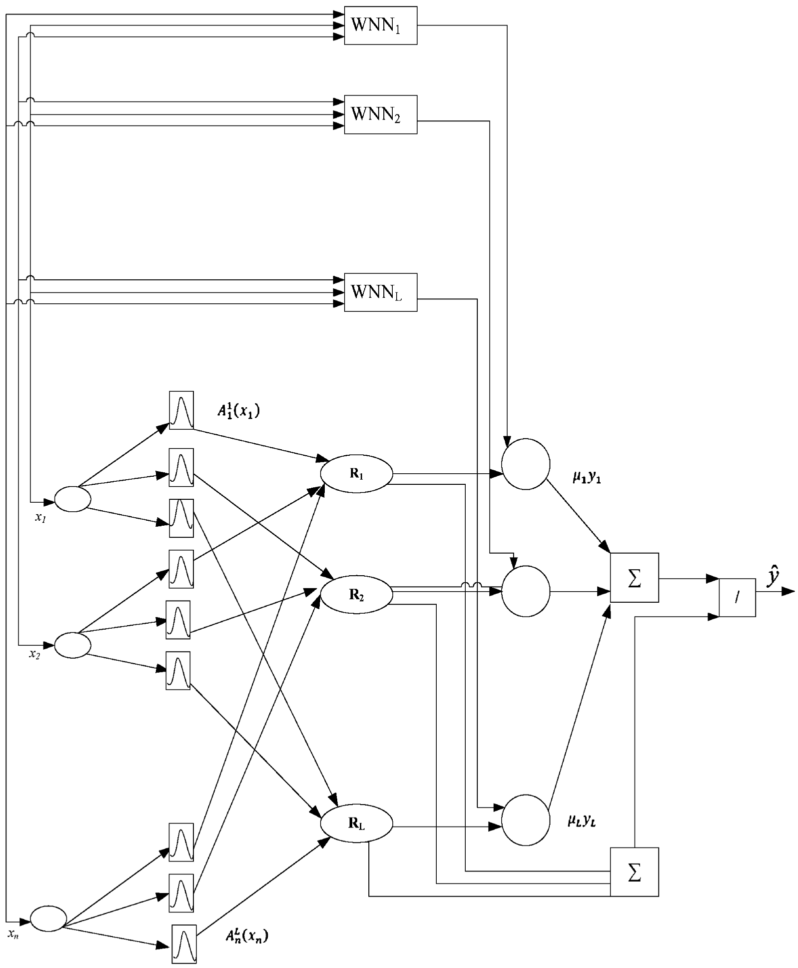

Fig. 2. Structure of FWNN [28].

2.2. Single hidden layer feed forward networks (SLFNs)

For N arbitrary distinct training data $(\boldsymbol{x}_i, \boldsymbol{t}_i) \in R^n \times R^m$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T$ and $\boldsymbol{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T$, the standard SLFN with $\tilde{N}$ hidden nodes is mathematically modeled as [30]:

$$\sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i G_i(\boldsymbol{x}_j) = \sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i G(\boldsymbol{w}_i.\boldsymbol{x}_j + b_i) = \boldsymbol{O}_j, \quad j = 1, \ldots, N, \tag{5}$$

where $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the weight vector linking the ith hidden node to the output nodes, $\boldsymbol{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ is the weight vector connecting the ith hidden node to the input nodes, $b_i \in R$ is the bias of ith hidden node and $\boldsymbol{O}_j = [O_{j1}, O_{j2}, \ldots, O_{jm}]^T$ is the output of the SLFN.

$G(\boldsymbol{w}_i.\boldsymbol{x}_j + b_i)$ is the output corresponding to the ith hidden node and jth input vector and $(\boldsymbol{w}_i.\boldsymbol{x}_j)$ denotes the inner product of $\boldsymbol{w}_i$ and $\boldsymbol{x}_j$. Structure of an SLFN is depicted in Fig. 3.

That standard SLFNs with $\tilde{N}$ hidden nodes with activation function $G(x)$ can be approximated by these N samples with zero error means that $\sum_{j=1}^{\tilde{N}} \|\boldsymbol{o}_j - \boldsymbol{t}_j\| = 0$, i.e., there exist $\boldsymbol{\beta}_i$, $\boldsymbol{w}_i$ and b_i such that

$$\sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i G(\boldsymbol{w}_i.\boldsymbol{x}_j + b_i) = \boldsymbol{t}_j, \quad j = 1, \ldots, N. \tag{6}$$

Please cite this article in press as: P. Golestaneh et al., Fuzzy wavelet extreme learning machine, Fuzzy Sets Syst. (2017), https://doi.org/10.1016/j.fss.2017.12.006

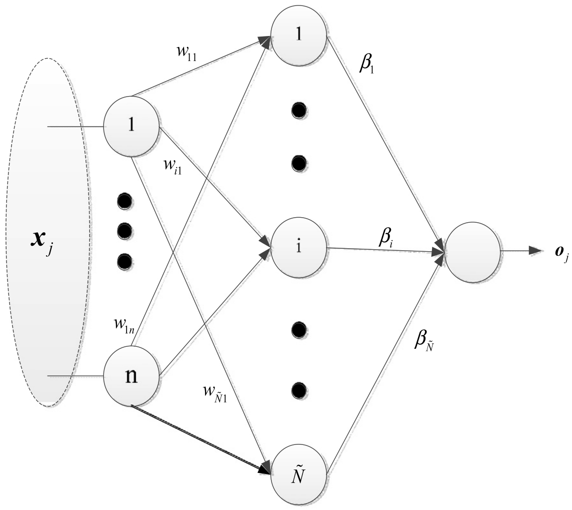

Fig. 3. Structure of an SLFN.

Then (6) can be compactly written as:

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{T}, \tag{7}$$

where

$$\boldsymbol{H}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \begin{bmatrix} G(\boldsymbol{w}_1.\boldsymbol{x}_1 + b_1) & \ldots & G(\boldsymbol{w}_{\tilde{N}}.\boldsymbol{x}_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ G(\boldsymbol{w}_1.\boldsymbol{x}_N + b_1) & \ldots & G(\boldsymbol{w}_{\tilde{N}}.\boldsymbol{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \tag{8}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \boldsymbol{t}_1^T \\ \vdots \\ \boldsymbol{t}_N^T \end{bmatrix}_{N \times m}. \tag{9}$$

H is the hidden layer output matrix [10,11]; the ith column of this matrix is the ith hidden node output with respect to inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$.

According to [30], if the activation function $G(x)$ is infinitely differentiable, the required hidden nodes are $\tilde{N} \leq N$.

2.3. Extreme learning machine algorithm (ELM)

Extreme Learning Machine is an efficient and simple learning method for SLFNs. ELM learning method not only provides good generalization performance, but also makes learning procedure very fast [30]. ELM algorithm is a one-pass learning method in which parameters of the hidden units and input weights (weights between the input layer and the hidden layer) can be randomly generated for once, and later SLFN can act as a linear system, so that the output weights can be analytically determined simply through a generalized inverse method. ELM not only minimizes the training error, but also tends to minimize the norm of the output weights [30].

Learning scheme of ELM can be summarized in the following steps:

ELM Algorithm: For N as the training data $(\boldsymbol{x}_i, \boldsymbol{t}_i) \in R^n \times R^m$, for $i = 1, \ldots, N$, $G(.)$ as activation function and with $\tilde{N}$ as the hidden nodes [30]

1- Randomly assigned input weight $\boldsymbol{w}_i$ and bias b_i, $i = 1, \ldots, \tilde{N}$.
2- Calculate the hidden layer output matrix ($H_{N \times \tilde{N}}$).
3- According to step 2 and (10), calculate the output weight matrix ($\boldsymbol{\beta}$):

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{T}. \tag{10}$$

Please cite this article in press as: P. Golestaneh et al., Fuzzy wavelet extreme learning machine, Fuzzy Sets Syst. (2017), https://doi.org/10.1016/j.fss.2017.12.006

In (10), $\boldsymbol{H}^\dagger$ is a Moore–Penrose generalized inverse of matrix $\boldsymbol{H}$ [41,42]. Various methods can be used to calculate the Moore–Penrose generalized inverse of a matrix such as orthogonal projection, orthogonalization, iterative methods and singular value decomposition (SVD). The orthogonal projection can be used in two different cases [33, 41, and 42]:

1- When $\boldsymbol{H}^T\boldsymbol{H}$ is nonsingular, then $\boldsymbol{H}^\dagger = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T$.
2- When $\boldsymbol{H}\boldsymbol{H}^T$ is nonsingular, then $\boldsymbol{H}^\dagger = \boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T)^{-1}$.

3. Fuzzy wavelet extreme learning machine

In this section, the structure of the proposed fuzzy wavelet extreme learning machine (FW-ELM) is described. First, equivalence of an SLFN with a fuzzy wavelet (FW) model is proved. Then, to construct FW-ELM structure, ELM is applied to proposed FW model.

3.1. Equivalence of an SLFN with an FW model

FW model is an FIS (Fuzzy Inference System) model (TSK model) in which THEN-part of each fuzzy rule contains a linear combination of finite set of wavelet functions. In this part, according to the structure of a generalized SLFN, the equivalence of an SLFN with an FW model is proved.

Theorem. *An FW model is equivalent to an SLFN.*

Proof. The TSK fuzzy model in an n inputs and m outputs system can be expressed as [12]:

$$R^i : \text{IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and } \dots \text{ and } x_n \text{ is } A_n^i \qquad \text{THEN}\big(y_1 \text{ is } \gamma_1^i\big)\dots\big(y_m \text{ is } \gamma_m^i\big), \tag{11}$$

where the input and output vector are $\boldsymbol{x} = [x_1, \dots, x_n]^T$ and $\boldsymbol{y} = [y_1, \dots, y_m]^T$, respectively. A_j^i ($j = 1, \dots, n$, $i = 1, \dots, L$) are fuzzy membership functions corresponding to the ith rule, L is the number of rules and $\boldsymbol{\gamma}_i = [\gamma_1^i, \gamma_2^i, \dots, \gamma_m^i]^T$. γ_k^i is a linear combination of input variables and is defined as:

$$\gamma_k^i = q_{ik,0} + q_{ik,1}x_1 + q_{ik,2}x_2 + \dots + q_{ik,n}x_n \tag{12}$$

As mentioned above, since an FW model is an FIS model, it can be expressed as (13):

$$R^i : \text{IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and } \dots x_n \text{ is } A_n^i \quad \text{THEN } \big(y_1 \text{ is } {}_S\boldsymbol{\beta}_1^i\big)\dots\big(y_m \text{ is } {}_S\boldsymbol{\beta}_m^i\big). \tag{13}$$

$${}_S\boldsymbol{\beta}_k^i = q_{ik,0} + q_{ik,1}\Psi_{i1}(x_1) + q_{ik,2}\Psi_{i2}(x_2) + \dots + q_{ik,n}\Psi_{in}(x_n) \tag{14}$$

${}_S\boldsymbol{\beta}_k^i$ is a set of linear combination of wavelet functions. S in ${}_S\boldsymbol{\beta}_k^i$, denotes standard FW structure. $\Psi_{ij}(x_j)$ ($j = 1, \dots, n$, $i = 1, \dots, L$) represents wavelet function of the jth input of the ith rule.

A_j^i ($j = 1, \dots, n$, $i = 1, \dots, L$) are fuzzy membership functions corresponding to the ith rule. Both T-NORM and S-NORM fuzzy logic operation can be used in (13); therefore, the firing strength of the ith rule can be expressed as:

T-NORM operation:

$$\mu_i(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{a}_i) = A_1^i(x_1, c_{1i}, a_{1i}) \otimes \dots \otimes A_n^i(x_n, c_{ni}, a_{ni}).$$

S-NORM operation: (15)

$$\mu_i(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{a}_i) = A_1^i(x_1, c_{1i}, a_{1i}) \oplus \dots \oplus A_n^i(x_n, c_{ni}, a_{ni}). \tag{15}$$

The system output of FW model for the given input $\boldsymbol{x}$ is weighted sum of the output of each normalized rule and is calculated as:

$$\hat{y} = \frac{\sum_{i=1}^{L} {}_S\boldsymbol{\beta}_i \mu_i(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{a}_i)}{\sum_{i=1}^{L} \mu_i(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{a}_i)} = \sum_{i=1}^{L} {}_S\boldsymbol{\beta}_i G_{FW}(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{a}_i). \tag{16}$$

Table 1
Equivalence of an SLFN with an FW model.

FW model	SLFN
$\sum_{i=1}^{L} {}_S\boldsymbol{\beta}_i G_{FW}(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{a}_i)$	$\sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i G(\boldsymbol{w}_i . \boldsymbol{x}_j + b_i)$
Total number of rules (L)	The number of hidden nodes ($\tilde{N}$)
$G_{FW}(.)$ normalized firing strength for each rule	$G(.)$ the output function of hidden nodes

In which $G_{FW}(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{a}_i) = \mu_i(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{a}_i) / \sum_{i=1}^{L} \mu_i(\boldsymbol{x}; \boldsymbol{c}_i, \boldsymbol{a}_i)$ is normalized firing strength for rule R^i.

By comparing (16) with (6), FW model is equivalent to a generalized SLFN. The equivalence of an SLFN with an FW model with more details is shown in Table 1. Consequently, ELM can be applied to FW model, where the output function of hidden nodes is $G_{FW}(.)$ and ${}_S\boldsymbol{\beta}_i$ is the output weight vector that should be determined analytically. As given in (15) and (16), the output functions for hidden nodes in an SLFN are based on the membership functions of FW model. □

3.2. Fuzzy wavelet structures

In the previous section, equivalence of an SLFN with an FW model was proved. Before we present our proposed structure and construct an FW model based on ELM algorithm, three well-known FW structures (standard FW structure [20], FW structure proposed by Yilmaz and et al. [43] and FW model used in [28]), are selected to apply ELM algorithm to them. These structures are respectively named, structures A, B and C that are different in number of linear learning parameters and type of membership functions.

In the following part, these three structures are explained.

Structure A:

This structure is based on standard FW (13) and (14) where in the THEN-part of each fuzzy rule, for each input, one coefficient is considered so that for all L rules, the number of linear coefficients are $(n + 1) \times L$.

Structure B:

This structure is based on standard FW structure (13) and (14), while in the IF-part of each fuzzy rule, wavelet membership function is used. It is noted that the number of linear learning parameters in this structure is equal to structure A.

According to [43], normalized Mexican hat wavelet function can be used as a membership function (A_j^i) and is defined as:

$$A_j^i = \frac{(1 - (\frac{x_j - b_{ji}}{d_{ji}})^2) \exp(0.5(\frac{x_j - b_{ji}}{d_{ji}})^2) + \varepsilon}{1 + \varepsilon}, \tag{17}$$

where $\varepsilon = 0.4461$.

Structure C:

This structure is based on fuzzy wavelet network used in [28]. This structure is defined as (1) and THEN-part of ith rule defined as $y_k^i = q_{ik,0} + q_{ik,1} \sum_{j=1}^{n} \Psi_{ij}(x_j)$. Therefore, in the THEN-part of each fuzzy rule, one coefficient is considered for all inputs. Hence for all L rules, the number of linear learning parameters is $2 \times L$ (one coefficient for all inputs and a constant coefficient).

The strengths and weaknesses of the above structures are discussed in the simulation results. As mentioned before, three structures have different numbers of linear learning parameters. Among them, structure C comes with the least number of linear parameters. As shown in Section 4, these three structures are not suitable for all examples in simulation results section. Therefore, in order to achieve a balance between network complexity and performance accuracy, we propose FW-ELM structure based on a new FW model.

Accordingly, in the proposed FW structure, in the THEN-part of each fuzzy rule, for each two inputs, one coefficient is considered. More details of the proposed structure are given in the following part.

Table 2
Details of four FW structures.

Structures	The number of linear learning parameters
Structure A	$(n+1) \times L$
Structure B	$(n+1) \times L$
Structure C	$2 \times L$
The proposed structure	$((\lceil \frac{n}{2} \rceil + 1) \times L)$

3.3. FW-ELM structure

In this section, the proposed FW structure is defined as:

$$R^i : \text{IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and } \dots x_n \text{ is } A_n^i \quad \text{THEN } \left(y_1 \text{ is } \beta_1^i\right) \dots \left(y_m \text{ is } \beta_m^i\right). \tag{18}$$

Structure of β_k^i is dependent on whether n is odd or even.
If n is even:

$$\beta_k^i = q_{ik,0} + q_{ik,1} \sum_{j=1}^{2} \Psi_{ij}(x_j) + q_{ik,2} \sum_{j=3}^{4} \Psi_{ij}(x_j) + \cdots + q_{ik,\lceil \frac{n}{2} \rceil} \sum_{j=n-1}^{n} \Psi_{ij}(x_j). \tag{19}$$

If n is odd:

$$\beta_k^i = q_{ik,0} + q_{ik,1} \sum_{j=1}^{2} \Psi_{ij}(x_j) + q_{ik,2} \sum_{j=3}^{4} \Psi_{ij}(x_j) + \cdots + q_{ik,\lceil \frac{n}{2} \rceil - 1} \sum_{j=n-2}^{n-1} \Psi_{ij}(x_j) + q_{ik,\lceil \frac{n}{2} \rceil} \Psi_{in}(x_n). \tag{20}$$

As seen in (19) and (20), the number of linear coefficients in the THEN-part of each fuzzy rule (linear learning parameters) is $(\lceil \frac{n}{2} \rceil + 1)$ where $\lceil n \rceil$ represents ceiling of n that is the least integer greater than or equal to n. In Table 2, the proposed structure and three other structures are compared.

In (18), Sigmoid membership function is used in the IF-part of the rules as follows [7]:

$$A(x) = sig(x; a, c) = \frac{1}{1 + \exp(-a(x-c))}, \tag{21}$$

where c is the central position and a controls the slope at the crossover point $x = c$.

In the consequent part of the proposed FW model, Mexican hat wavelet function is used as follows [28]:

$$\psi(x) = \left(1 - x^2\right) \exp\left(-\frac{x^2}{2}\right).$$

Mathematical model of FW-ELM

For N arbitrary distinct training data $(\boldsymbol{x}_i, \boldsymbol{t}_i)$ where $\boldsymbol{x}_i = [x_{i1}, \dots, x_{in}]^T$, $\boldsymbol{t}_i = [t_{i1}, \dots, t_{im}]^T$ and L fuzzy rules, the mathematical model of FW model containing three parameters $(\boldsymbol{\beta}_i, \boldsymbol{c}_i, \boldsymbol{a}_i)$ can be defined as

$$f_L(\boldsymbol{x}_j) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G_{FW}(\boldsymbol{x}_j, \boldsymbol{c}_i, \boldsymbol{a}_i) = \boldsymbol{t}_j, \quad j = 1, \dots, N. \tag{22}$$

$$\boldsymbol{\beta}_i = \Psi_{ie}^T(\boldsymbol{x}_j) \boldsymbol{q}_{i\text{FW}}. \tag{23}$$

Here $\boldsymbol{\beta}_i$, is based on the linear combination of finite set of wavelets. According to (19) and (20), depending on n (odd or even), $\Psi_{ie}(\boldsymbol{x})$ vector can be defined as follows:

If n is even:

$$\Psi_{ie}(\boldsymbol{x}) = \left[1, \sum_{j=1}^{2} \Psi_{ij}(x_j), \dots, \sum_{j=n-1}^{n} \Psi_{ij}(x_j)\right]^T.$$

If n is odd:

$$\Psi_{ie}(\boldsymbol{x})=\left[1,\sum_{j=1}^{2}\Psi_{ij}(x_j),\ldots,\sum_{j=n-2}^{n-1}\Psi_{ij}(x_j),\Psi_{in}(x_n)\right]^T. \tag{24}$$

In (23), $\boldsymbol{q}_{i_{FW}}$ $(i=1,\ldots,L)$ for the ith rule is expressed by

$$\boldsymbol{q}_{i_{FW}}=\begin{bmatrix} q_{i1,0} & \cdots & q_{ik,0} & \cdots & q_{im,0} \\ \vdots & \vdots & & \vdots & \vdots \\ q_{i1,\lceil\frac{n}{2}\rceil} & \cdots & q_{ik,\lceil\frac{n}{2}\rceil} & \cdots & q_{im,\lceil\frac{n}{2}\rceil} \end{bmatrix}_{(\lceil\frac{n}{2}\rceil+1)\times m}. \tag{25}$$

In fact, the kth column of $\boldsymbol{q}_{i_{FW}}$ includes linear parameters in (19) or (20). Consequently, the dimension of the above matrix in the proposed FW structure is $((\lceil\frac{n}{2}\rceil+1)\times m)$.

According to (16), (22) and (23), the total output for proposed FW model can be written as

$$f_L(\boldsymbol{x}_j)=\sum_{i=1}^{L}\Psi_{ie}^T(\boldsymbol{x}_j)\boldsymbol{q}_{i_{FW}}G_{FW}(\boldsymbol{x}_j,\boldsymbol{c}_i,\boldsymbol{a}_i)=\boldsymbol{t}_j,\quad j=1,\ldots,N. \tag{26}$$

The above N equations can be compactly written as:

$$\boldsymbol{H}\boldsymbol{Q}=\boldsymbol{T}. \tag{27}$$

$$\begin{aligned}&\boldsymbol{H}(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_L,\boldsymbol{a}_1,\ldots,\boldsymbol{a}_L;\boldsymbol{b}_1,\ldots,\boldsymbol{b}_L,\boldsymbol{d}_1,\ldots,\boldsymbol{d}_L;\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N)\\ &\quad=\left[\Psi_{ie}^T(\boldsymbol{x}_j;\boldsymbol{b}_1,\boldsymbol{d}_1)G_{FW}(\boldsymbol{x}_j;\boldsymbol{c}_1,\boldsymbol{a}_1)\ldots\Psi_{ie}^T(\boldsymbol{x}_j;\boldsymbol{b}_L,\boldsymbol{d}_L)G_{FW}(\boldsymbol{x}_j;\boldsymbol{c}_L,\boldsymbol{a}_L)\right]\end{aligned} \tag{28}$$

$$\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{q}_{1\mathrm{FW}}\\ \vdots\\ \boldsymbol{q}_{\mathrm{LFW}}\end{bmatrix}_{L(\lceil\frac{n}{2}\rceil+1)\times m}\quad\text{and}\quad\mathbf{T}=\begin{bmatrix}\boldsymbol{t}_1^T\\ \vdots\\ \boldsymbol{t}_N^T\end{bmatrix}_{N\times m}, \tag{29}$$

where $\boldsymbol{H}$ is the hidden unit matrix and is the parameter matrix for the fuzzy wavelet model. The elements of $\boldsymbol{Q}$ are linear parameters.

In (28), $(\boldsymbol{c}_i=[c_{i1},\ldots,c_{in}]^T,\boldsymbol{a}_i=[a_{i1},\ldots,a_{in}]^T, i=1:L)$ are parameters of Sigmoid membership function and $(\boldsymbol{b}_i=[b_{i1},\ldots,b_{in}]^T,\boldsymbol{d}_i=[d_{i1},\ldots,d_{in}]^T, i=1:L)$ are wavelet coefficients (translations and dilations).

According to the proof of the equivalence between an SLFN and an FW model, ELM can be applied to the proposed FW model. The resulting algorithm is called FW-ELM. In (28), all nonlinear parameters including parameters of membership functions and wavelet functions $(\boldsymbol{c}_i,\boldsymbol{a}_i$ and $\boldsymbol{b}_i,\boldsymbol{d}_i$ $i=1,\ldots,L)$ are randomly generated and will remain fixed. Linear parameters (elements of Q) are calculated according to ELM with one pass FW-ELM algorithm. The procedure can be summarized as follows:

FW-ELM algorithm:

For N arbitrary distinct training data $(\boldsymbol{x}_i,\boldsymbol{t}_i)$ where $\boldsymbol{x}_i=[x_{i1},\ldots,x_{in}]^T$, $\boldsymbol{t}_i=[t_{i1},\ldots,t_{im}]^T$ and L fuzzy rules.

1- Randomly assigned parameters of membership function and wavelet coefficients $(\boldsymbol{c}_i=[c_{i1},\ldots,c_{in}]^T$, $\boldsymbol{a}_i=[a_{i1},\ldots,a_{in}]^T$, $\boldsymbol{b}_i=[b_{i1},\ldots,b_{in}]^T$, $\boldsymbol{d}_i=[d_{i1},\ldots,d_{in}]^T$, $i=1:L)$.
2- According to (28), calculate $\boldsymbol{H}$ (matrix of the hidden layer).
3- By using $\boldsymbol{H}\boldsymbol{Q}=\boldsymbol{T}$, determine $\boldsymbol{Q}$:

$$\boldsymbol{Q}=\boldsymbol{H}^{\dagger}\boldsymbol{T}. \tag{30}$$

$\boldsymbol{H}^{\dagger}$, has been described in explanations of (10).

To clarify the advantages of FW-ELM over OS-Fuzzy-ELM [35] as well as FWNN [28], the explanation is provided below:

To compare FW-ELM with OS-Fuzzy-ELM, dimensions of matrix $\boldsymbol{Q}$ and $\boldsymbol{H}$ in (27) are considered as indices to compare computation complexity of the models (L, n, m are the number of rules, the dimension of input and output vector, respectively).

Table 3
Comparison results between FW-ELM and OS-FUZZY-ELM.

Method	Dimension of $\boldsymbol{Q}$	Dimension of $\boldsymbol{H}$	# Linear learning parameters for each rule
FW-ELM	$L(\lceil \frac{n}{2} \rceil + 1) \times m$	$N \times (L(\lceil \frac{n}{2} \rceil + 1))$	$(\lceil \frac{n}{2} \rceil + 1) \times m$
OS-Fuzzy-ELM [35]	$L(n + 1) \times m$	$N \times (L(n + 1))$	$(n + 1) \times m$

Table 4
Regression problem datasets [35].

Datasets	Datasets	# Observations		# Attributes
		Training	Testing	
A-1	Census (House8L)	10,000	12,784	8
A-2	Bank	4,500	3,692	8
A-3	Kinematics robot arm	4,000	4,192	8

According to Table 3, the number of linear coefficients of consequent part in OS-Fuzzy-ELM is $(n + 1)$ and in FW-ELM, it is almost half-time $(\lceil n/2 \rceil + 1)$. Therefore, in comparison to OS-Fuzzy-ELM, FW-ELM is able to significantly reduce the complexity of the network by decreasing the number of linear learning parameters.

In FWNN [28], one linear coefficient is used in the consequent part of each rule and RLS is used to update weight parameters of the model. Initial values of the wavelet parameters are found by K-means clustering. All nonlinear parameters of FWNN are updated by applying the BP method with adaptive learning rate. Briefly, the method proposed in [28] needs initialization procedure and iterative updating algorithms while in FW-ELM, all nonlinear parameters are randomly generated and linear parameters are determined using only a one-pass learning method.

4. Simulation results

In this section, the performance accuracy of the proposed fuzzy wavelet ELM and three other structures are compared with various fuzzy algorithms such as (OS-Fuzzy-ELM, ANFIS-Full, ANFIS-Reduced, Simpl_eTS) [35], FWNN [28] as well as ELM, BP, CFWNN-ELM [40], WNN-ELM [40], SVR [30] as other relevant approaches, on a number of real benchmark datasets.

For each data set, 50 independent runs are performed and the mean values and standard deviations (SD) of the accuracy metrics calculated for both training and testing data are reported. Regression, classification, time series prediction, identification of nonlinear systems and function approximation are the machine learning problems taken into account in this study.

All simulations have been conducted in Matlab 2012 running on CPU 2.66 GHZ with RAM 4 GB. For time series prediction, identification of nonlinear systems, function approximation and regression problems, all the free parameters are chosen from [0, 1] and for the classification problem, they are chosen from [−1, 1].

For the sake of a fair comparison between FW-ELM and OS-Fuzzy-ELM in regression and classification examples, the number of hidden nodes in both algorithms is chosen equal. The number of fuzzy rules in the structures A, B and C are selected such that the number of linear learning parameters is almost equal to the proposed FW structure.

Remark. Since the number of linear learning parameters in structure A and B is the same, only the results for structure B are reported in the following tables. For the ease of read, the best SDs, the lowest number of learning parameters and the best training and testing accuracy are shown in boldface.

4.1. *Example 1 – regression problems*

FW-ELM structure and three other mentioned structures are applied to three benchmark real-world regression datasets [44]. The details of datasets are given in Table 4. All input and output data are normalized to [0, 1].

In Table 5, SDs and average values of root-mean-square error (RMSE) for training and testing data of the proposed algorithm and three other structures are compared with those of ANFIS, OS-Fuzzy-ELM and Simpl_eTS.

Table 5
Performance comparison for benchmark regression problems.

Datasets	Algorithms	Training (RMSE)		Testing (RMSE)		# Rules	# Linear learning parameters
		Mean	SD	Mean	SD		
A-1	FW-ELM	0.0677	0.0012	0.0690	**0.0015**	8	**40**
	Structure B	0.0670	0.0006	0.0701	0.0267	**6**	54
	Structure C	0.0705	**0.0001**	0.0728	0.0146	20	**40**
	OS-fuzzy-ELM [35]	0.0649	0.0023	**0.0661**	0.0019	8	72
	ANFIS-reduced [35]	0.0648	0.0027	0.0667	0.0027	8	
	Simpl_eTS [35]	0.0818	0.0036	0.0814	0.0030	20	
	ANFIS-full [35]	0.0576	0.0001	0.1058	0.0237	256	
A-2	FW-ELM	0.0431	**0.00003**	0.0446	**0.00002**	15	**75**
	Structure B	0.0428	0.0001	0.00445	0.0040	**12**	108
	Structure C	0.0442	0.0002	0.0454	0.0050	38	76
	OS-fuzzy-ELM [35]	0.0380	0.0014	**0.0390**	0.0016	15	135
	ANFIS-reduced [35]	0.0376	0.0015	0.0394	0.0019	16	
	Simpl_eTS [35]	0.0506	0.0029	0.0512	0.0033	42	
	ANFIS-full [35]	0.0228	0.0014	0.1327	0.0054	256	
A-3	FW-ELM	0.0701	0.0003	0.0743	**0.0006**	50	250
	Structure B	0.0698	**0.0001**	0.0745	0.0072	**30**	270
	Structure C	0.0712	0.0006	**0.0732**	0.0086	50	**100**
	OS-fuzzy-ELM [35]	0.0761	0.0024	0.0853	0.0023	50	450
	ANFIS-reduced [35]	0.0721	0.0031	0.0823	0.0031	64	
	Simpl_eTS [35]	0.1451	0.0050	0.1460	0.0055	74	
	ANFIS-full [35]	0.0251	0.0010	0.1229	0.0053	256	

As shown in Table 5, while the number of consequent weight parameters in the proposed method is almost half of the number of parameters of OS-Fuzzy-ELM, the performance accuracy of FW-ELM is comparable with OS-Fuzzy-ELM and is better than that of other algorithms.

For instance, for dataset A-3, the number of consequent weight parameters in OS-Fuzzy-ELM is 450 while it is 250 for the proposed method (reduced almost by half). And also both SD and Mean for the proposed method is better than OS-Fuzzy-ELM.

Comparing with (ANFIS and Simpl_eTS), the proposed method provides the best testing accuracy in a very compact network with fewer fuzzy rules.

Table 5 also shows that FW-ELM yields smaller SDs, which means that FW-ELM is more stable and it is able to decrease the impact of parameter randomness. As can be seen in Table 5, SD values in FW-ELM are fewer than those in other structures.

For dataset (A-3), performance accuracy (in testing step) of FW-ELM and structure C are better than those in all other approaches.

4.2. Example 2 – classification problems

The performance of FW-ELM, structure B, OS-Fuzzy-ELM, CFWNN-ELM [40], WNN-ELM [40], Simpl_eTS and evolving Takagi–Sugeno (eTS) models are compared on three real-world benchmark datasets [44]. Details of the datasets are listed in Table 6. All input data are normalized between $[-1, 1]$.

The comparative results of FW-ELM and the well-known methods are shown in Table 7. As shown in this table, while the number of consequent weight parameters in FW-ELM is almost half of that of parameters in OS-Fuzzy-ELM, the performance of FW-ELM is comparable with OS-Fuzzy-ELM and better than most of the other mentioned methods. Structure C is not appropriate for classification problems; therefore, its results are not reported here.

For instance, as observed in Table 8, for dataset (C-3), the testing accuracy of the proposed method is as well OS-Fuzzy-ELM while the proposed method provides the smallest SD value (0.0042) and also the smallest number of linear learning parameters (this number is 3330 for OS-Fuzzy-ELM, while it is 1710 for the proposed method

Table 6
Classification problem datasets [35].

Datasets	Datasets	# Observations		# Classes	# Attributes
		Training	Testing		
C-1	Image segmentation	1,100	1,210	7	19
C-2	Page blocks	2,700	2,773	5	10
C-3	Satellite image	4,400	2,035	6	36

Table 7
Performance comparison for benchmark classification problems.

Datasets	Algorithms	Training (%)		Testing (%)		# Rules/nodes	# Linear learning parameters
		Accuracy	SD	Accuracy	SD		
C-1	FW-ELM	96.27	**0.0047**	93.64	**0.0136**	10	**770**
	Structure B	**96.38**	0.0432	88.64	0.0137	**8**	1120
	OS-fuzzy-ELM [35]	95.836	0.6575	**94.410**	0.9596	10	1400
	Simpl_eTS [35]	93.047	0.9087	91.673	1.3275	21	–
	eTS [35]	94.320	0.8503	92.667	1.1681	21	–
	CFWNN-ELM [40]	98.46	0.13	95.40	0.80	300	–
	WNN-ELM [40]	98.43	0.18	95.16	0.72	100	–
C-2	FW-ELM	96.19	0.0014	95.82	**0.0040**	10	**300**
	Structure B	96.53	**0.0003**	89.73	0.0365	**7**	385
	OS-fuzzy-ELM [35]	**96.72**	0.2087	**95.99**	0.2200	10	550
	Simpl_eTS [35]	95.167	0.6184	94.546	0.4557	14	–
	eTS [35]	95.746	0.4189	95.130	0.3145	14	–
C-3	FW-ELM	91.11	**0.0037**	88.50	**0.0042**	15	**1710**
	Structure B	92.07	0.0257	87.95	0.0059	**10**	2220
	OS-fuzzy-ELM [35]	**92.841**	0.6591	**89.40**	0.4474	15	3330
	Simpl_eTS [35]	91.496	0.7998	88.157	0.8005	35	–
	eTS [35]	92.144	0.5460	88.430	1.0078	37	–
	CFWNN-ELM [40]	93.29	0.37	88.82	0.48	500	–
	WNN-ELM [40]	93.17	0.32	88.53	0.52	500	–

(reduced almost by half)). This again shows that FW-ELM has acceptable performance and can decrease the impact of parameter randomness.

4.3. Example 3 – predicting chaotic time series

In this section, the Mackey Glass time series is used to evaluate FW-ELM in comparison to FWNN [28] and the other fuzzy algorithms discussed in [28]. This time series is generated by the following differential equation [28]:

$$\dot{x} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t). \tag{31}$$

One thousand input-output data pairs of the following format (where $\tau = 17$ and $x(0) = 1.2$) are extracted [20]:

$$\left\{ \begin{matrix} X^d = [x(t-18) \quad x(t-12) \quad x(t-6) \quad x(t)] \\ y^d = x(t+6) \end{matrix} \right\}. \tag{32}$$

To validate the proposed approach, the first 500 pairs are used as the training dataset, while the remaining 500 pairs are used as the testing dataset [28]. Similar to [28], as a performance index, the root mean square error is considered as (33)

$$RMSE = \frac{1}{\sigma}\sqrt{\frac{\sum_{i=1}^{N}(y(k+1)-y_d(k+1))^2}{N}}. \tag{33}$$

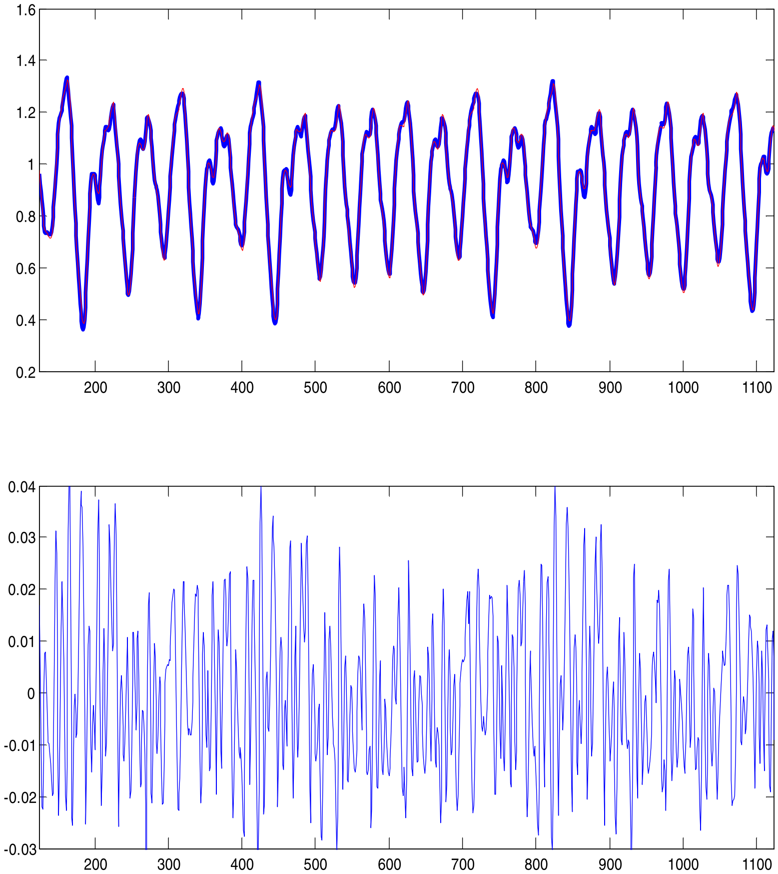

Fig. 4. The comparison between predicted output using FW-ELM, original function (top part) and prediction errors (bottom part).

Table 8
Comparison of FW-ELM with other works.

Method	# Rules	Epoch	# Parameters	RMSE (final)
FW-ELM	8	–	**24**	**0.000083**
Structure B	5	–	25	0.0001
Structure C	12	–	**24**	0.0001
FWNN [28]	**3**	70	51	0.0030
FWNN [28]	6	–	–	0.0066
FWNN with GD And fixed learning rate [28]	3	51	150	0.01
ANFIS [28]	–	–	–	0.007

In (33), σ is the standard deviation and N is the number of the training data, y is output of the network and y_d is the desired output.

In Fig. 4, the comparison between predicted output using FW-ELM, original function and prediction errors for 1000 time steps are illustrated.

As can be seen in Table 8, the RMSE for FW-ELM is much better than that for the other well-known methods (RMSE is 0.000083). It should be noted that the number of linear learning parameters in all structures is much less than (half of) those of the other rival approaches.

It is to be noted that the proposed scheme, structures B and C are non-iterative learning algorithm, but the other methods are trained using iterative approaches.

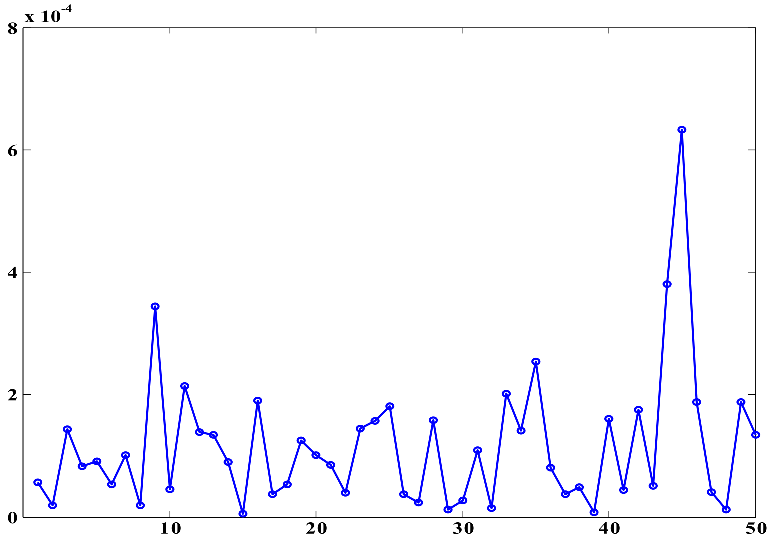

Fig. 5. RMSE values for 50 trials.

Table 9
Comparison results for identification of nonlinear plant.

Method	# Rules	Epoch	# Learning parameters	RMSE (final)
FW-ELM	4		16	**0.0000013**
Structure B	3	–	18	0.000386
FWNN [28]	**2**	80	18	0.0000049
FWNN [28]	3	27	200	0.028
RFNN [28]	16	112	100	0.0003 (MSE)
TRFN-S [28]	3	33	**10**	0.0084
ANFIS [28]	36	144	400	0.03

Fig. 5 shows the variation of RMSE values for 50 trails. As can be seen in Fig. 5, FW-ELM is not sensitive to parameter randomness.

4.4. Example 4 – identification of nonlinear dynamic plant

This example discusses the performance of rival approaches in identification of a second-order nonlinear plant [11].

$$y(k) = f\big(y(k-1), y(k-2), y(k-3), u(k), u(k-1)\big)$$
$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2}, \tag{34}$$

where $u(k)$, $u(k-1)$ are the current and one-step delayed inputs of the plant and $y(k-1)$, $y(k-2)$, $y(k-3)$ are one, two and three steps delayed outputs. According to (34), the current output of the plant is a nonlinear function of the previous inputs and outputs.

In this paper similar to [11], for FW-ELM the current state of the system and the control signal are considered as inputs. The RMSE is considered as a performance criteria, where N is the number of training data which are distributed uniformly in $[-1, 1]$ (here $N = 1000$)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y(k+1) - y_d(k+1))^2}{N}}. \tag{35}$$

In (35), N is the number of training data, $y(n)$ and $y_d(n)$ are the desired and estimated outputs, respectively.

The performance of FW-ELM is compared with the other rival approaches in Table 9. Structure C does not suit this problem; therefore the result for this structure is not reported.

Please cite this article in press as: P. Golestaneh et al., Fuzzy wavelet extreme learning machine, Fuzzy Sets Syst. (2017), https://doi.org/10.1016/j.fss.2017.12.006

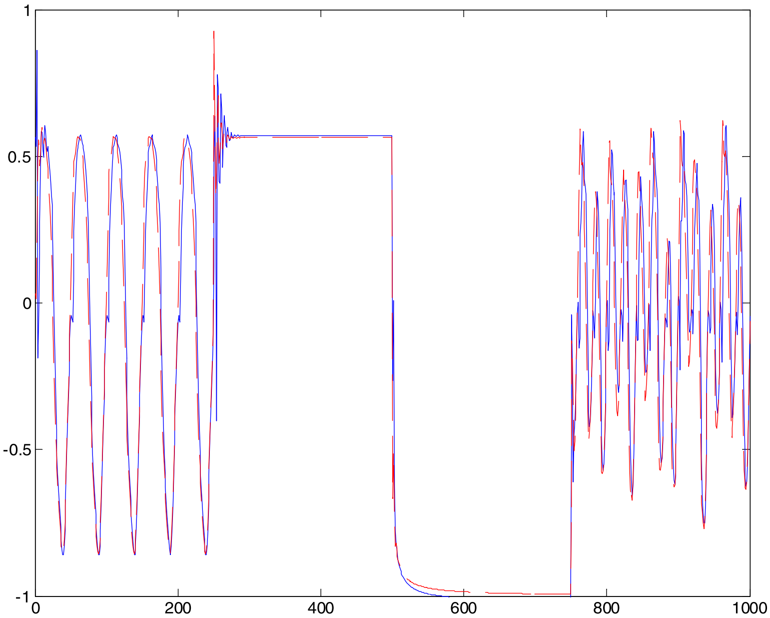

Fig. 6. Comparison between the original function (solid line) and the output of FW-ELM (dashed line).

Briefly, it is observed that the performance accuracy of FW-ELM is better than the best of the other reported algorithms, while its number of learning parameters is fewer than the others. It should be noted that FW-ELM is a non-iterative method; however, the other approaches are based on iterative methods and obtain their best accuracy after several epochs.

For testing the ability of FW-ELM algorithm in identification of the plant, the excitation signal is considered as (36)

$$u(k) = \begin{cases} \sin(\pi k/25) & k < 250 \\ 1.0 & 250 \leq k < 500 \\ 0.3\sin(\pi k/25) + 0.1\sin(\pi k/32) + 0.6\sin(10), & 750 \leq k < 1000. \end{cases} \tag{36}$$

The ability of FW-ELM in identification of nonlinear dynamic plant is shown in Fig. 6.

4.5. Example 5 – approximation of 'SinC' function

In this example, to approximate the 'SinC' function, FW-ELM, three other mentioned structures, BP, ELM, CFWNN-ELM [40], WNN-ELM [40] and SVR algorithms are used.

$$y(x) = \begin{cases} \frac{\sin(x)}{x}, & x \neq 0 \\ 1, & x = 0. \end{cases} \tag{37}$$

In this example for both training and testing data $(\boldsymbol{x}_i, \boldsymbol{t}_i)$, 5000 data points are sampled. Input data are uniformly randomly distributed over $[-10, 10]$. It is to be noted that large uniform noise distributed in $[-0.2, 0.2]$ is added to the training data while testing data is kept noise-free [30]. The average and SDs of RMSE values and the number of rules are shown in Table 10.

For training FW-ELM, the number of hidden nodes is set to 15, while for both BP and ELM, 20 hidden nodes are required. According to the results, for testing data, the performance accuracy of the proposed approach is as well as CFWNN-ELM and is better than other methods. And in comparison to the other methods, the proposed method obtains the best SD value while using the smallest number of nodes/SVs. In this example, since $n = 1$ so it is expected that all three structures and also the proposed structure can obtain similar results. As can be seen in Table 10, the results are almost the same.

The true and the approximated function of FW-ELM structure is shown in Fig. 7.

Table 10
Comparison results for approximation Of SINC function.

Algorithms	Training		Testing		# Nodes/no of SVs
	Mean	SD	Mean	SD	
FW-ELM	0.1159	**4.97e−05**	0.0082	**6.27e−04**	**15**
Structure B	0.1160	0.0001	0.0092	0.0015	**15**
Structure C	0.1159	0.0001	0.0094	0.0018	**15**
ELM [30]	**0.1148**	0.0037	0.0097	0.0028	20
BP [30]	0.1196	0.0042	0.0159	0.0041	20
SVR [30]	0.1149	0.0007	0.0130	0.0012	2499.9
CFWNN-ELM [40]	–	–	**0.0062**	0.0012	30
WNN-ELM [40]	–	–	0.0651	0.0309	30

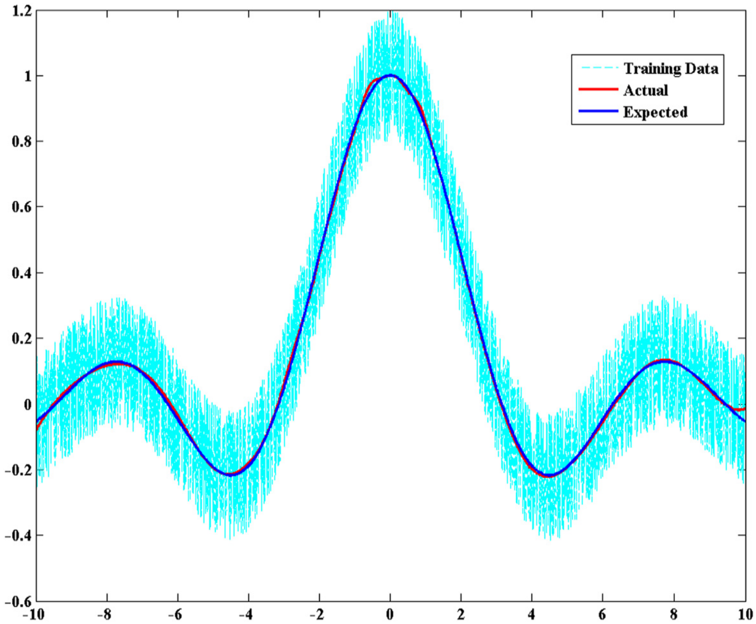

Fig. 7. Comparison between the original function and the output of FW ELM structure.

5. Conclusions

This work presents fuzzy wavelet extreme learning machine (FW-ELM) structure based on a combination of fuzzy rules, the theory of multiresolution analysis of wavelet transform and ELM. Specifically, as one of the advantages of this scheme is that it is able to efficiently reduce the complexity of the network and handle the impact of parameter randomness in initialization phase while the model has only one pass learning step and can provide acceptable accuracy and generalization performance. The capability of the algorithm is verified by applying FW-ELM model on benchmark real datasets. The results of comparison indicate that the number of linear learning parameters is efficiently reduced in comparison to the other reported works. For identification of plants and prediction of time series, in comparison to FWNNs, FW-ELM achieves the best performance with a smaller number of learning parameters and using a one-pass learning method. Furthermore, for classification and regression problems, the performance of FW-ELM is comparable with that of the OS-Fuzzy-ELM and better than other reported works while the number of linear learning parameters is decreased and SDs is smaller.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.fss.2017.12.006.

References

[1] S. Haykin, Neural Networks: A Comprehensive Foundation, McMillan, New York, 1994.
[2] Q. Zhang, Using wavelet networks in nonparametric estimation, IEEE Trans. Neural Netw. 8 (Mar. 1997) 227–236.
[3] L. Wang, Adaptive Fuzzy Systems and Control, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[4] J. Wang, C. Wang, C. Chen, The bounded capacity of fuzzy neural networks (FNNs) via a new fully connected neural fuzzy inference system (F-CONFIS) with its applications, IEEE Trans. Fuzzy Syst. 22 (6) (Dec. 2014) 1373–1386.
[5] C.-K. Lin, H_∞ reinforcement learning control of robot manipulators using fuzzy wavelet networks, Fuzzy Sets Syst. 160 (12) (2009) 1765–1786.
[6] P. Ong, Z. Zainuddin, Calibrating wavelet neural networks by distance orientation similarity fuzzy C-means for approximation problems, Appl. Soft Comput. 42 (2016) 156–166.
[7] J. Jang, C. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, Prentice Hall, Upper Saddle River, NJ, 1997.
[8] L. Reyneri, Unification of neural and wavelet networks and fuzzy systems, IEEE Trans. Neural Netw. 10 (4) (1999) 801–814.
[9] G. Daqi, Y. Genxing, Influences of variable scales and activation functions on the performances of multilayer feed forward neural networks, Pattern Recognit. 36 (4) (2003) 869–878.
[10] D. Svozil, V. Kvasnicka, J. Pospichal, Introduction to multi-layer feed-forward neural networks, Chemom. Intell. Lab. Syst. 39 (1) (1997) 43–62.
[11] R.H. Abiyev, O. Kaynak, Fuzzy wavelet neural networks for identification and control of dynamic plants—A novel structure and a comparative study, IEEE Trans. Ind. Electron. 55 (8) (Aug. 2008) 3133–3140.
[12] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications for modeling and control, IEEE Trans. Syst. Man Cybern. SMC-15 (1) (Feb. 1985) 116–132.
[13] W. Cheng, C. Juang, A fuzzy model with online incremental SVM and margin-selective gradient descent learning for classification problems, IEEE Trans. Fuzzy Syst. 22 (2) (Apr. 2014) 324–337.
[14] S. Horikawa, T. Furuhashi, Y. Uchikawa, On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm, IEEE Trans. Neural Netw. 3 (5) (Sep. 1992) 801–806.
[15] J. Mendel, Fuzzy logic systems for engineering: a tutorial, Proc. IEEE 83 (3) (1995) 345–377.
[16] P. Flandrin, Time-Frequency/Time Scale Analysis, Academic Press, San Diego, 1999.
[17] J. Zhang, G.G. Walter, Y. Miao, W. Wayne Lee, Wavelet neural networks for function learning, IEEE Trans. Signal Process. 43 (6) (June 1995).
[18] P. D'Urso, E.A. Maharaj, Wavelets-based clustering of multivariate time series, Fuzzy Sets Syst. 193 (2012) 33–61.
[19] H. Pan, L. Xia, Efficient object recognition using boundary representation and wavelet neural network, IEEE Trans. Neural Netw. 19 (12) (Dec. 2008) 2132–2149.
[20] D. Ho, Ping-Au Zhang, Jinhua Xu, Fuzzy wavelet networks for function learning, IEEE Trans. Fuzzy Syst. 9 (1) (Feb. 2001) 200–211.
[21] Y. Huang, F. Lin, J. Hwang, J. Chang, K. Ruan, Wavelet fuzzy neural network with asymmetric membership function controller for electric power steering system via improved differential evolution, IEEE Trans. Power Electron. 30 (4) (Apr. 2015) 2350–2362.
[22] C. Lu, Wavelet fuzzy neural networks for identification and predictive control of dynamic systems, IEEE Trans. Ind. Electron. 58 (7) (Jul. 2011) 3046–3058.
[23] R. Cheng, Y. Bai, A novel approach to fuzzy wavelet neural network modeling and optimization, Int. J. Electr. Power Energy Syst. 64 (2015) 671–678.
[24] F. Lin, Y. Huang, K. Ruan, An intelligent second-order sliding-mode control for an electric power steering system using a wavelet fuzzy neural network, IEEE Trans. Fuzzy Syst. 22 (6) (2014) 1598–1611.
[25] S. Yilmaz, Y. Oysal, Fuzzy wavelet neural network models for prediction and identification of dynamical systems, IEEE Trans. Neural Netw. 21 (10) (Oct. 2010) 1599–1609.
[26] C. Lu, Wavelet fuzzy neural networks for identification and predictive control of dynamic systems, IEEE Trans. Ind. Electron. 58 (7) (Jul. 2011) 3046–3058.
[27] M. Zekri, S. Sadri, F. Sheikholeslam, Adaptive fuzzy wavelet network control design for nonlinear systems, Fuzzy Sets Syst. 159 (20) (Jun. 2008) 2668–2695.
[28] M. Davanipoor, M. Zekri, F. Sheikholeslam, Fuzzy wavelet neural network with an accelerated hybrid learning algorithm, IEEE Trans. Fuzzy Syst. 20 (3) (2012) 463–470.
[29] Y. Oysal, S. Yilmaz, An adaptive fuzzy wavelet network with gradient learning for nonlinear function approximation, J. Intell. Syst. 23 (2) (2014).
[30] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[31] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feed forward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (Jul. 2006) 879–892.
[32] G. Huang, G.-B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Netw. 61 (2015) 32–48.
[33] G.-Bin Huang, Hongming Zhou, Xiaojian Ding, Rui Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 42 (2) (Apr. 2012) 513–529.
[34] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (Nov. 2006) 1411–1423.
[35] H.-J. Rong, G.-B. Huang, N. Sundararajan, P. Saratchandran, Online sequential fuzzy extreme learning machine for function approximation and classification problems, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 39 (4) (Aug. 2009) 1067–1072.

Please cite this article in press as: P. Golestaneh et al., Fuzzy wavelet extreme learning machine, Fuzzy Sets Syst. (2017), https://doi.org/10.1016/j.fss.2017.12.006

[36] K. Javed, R. Gouriveau, N. Zerhouni, SW-ELM: a summation wavelet extreme learning machine algorithm with a priori parameter initialization, Neurocomputing 123 (2014) 299–307.

[37] K. Javed, R. Gouriveau, N. Zerhouni, P. Nectoux, Enabling health monitoring approach based on vibration data for accurate prognostics, IEEE Trans. Ind. Electron. 62 (1) (Jan. 2015) 647–656.

[38] E. Avci, R. Coteli, A new automatic target recognition system based on wavelet extreme learning machine, Expert Syst. Appl. 39 (16) (2012) 12340–12348.

[39] N. Shrivastava, B. Panigrahi, A hybrid wavelet-ELM based short term price forecasting for electricity markets, Int. J. Electr. Power Energy Syst. 55 (2014) 41–50.

[40] J. Cao, Z. Lin, G.-B. Huang, Composite function wavelet neural networks with extreme learning machine, Neurocomputing (Dec. 2009) 1405–1416.

[41] C.R. Rao, S.K. Mitra, Generalized Inverse of Matrices and Its Applications, Wiley, New York, 1971.

[42] D. Serre, Matrices: Theory and Applications, Springer, New York, 2002.

[43] S. Yilmaz, Y. Oysal, Nonlinear system modeling and control with dynamic fuzzy wavelet neural network 2015, in: International Symposium on Innovations in Intelligent Systems and Applications, INISTA, 2015.

[44] C. Blake, C. Merz, UCI Repository of Machine Learning Databases, Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, 1998 [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html.

Please cite this article in press as: P. Golestaneh et al., Fuzzy wavelet extreme learning machine, Fuzzy Sets Syst. (2017), https://doi.org/10.1016/j.fss.2017.12.006