

A Tensor Based Stacked Fuzzy Networks for Efficient Data Regression

Jie Li

Inner Mongolia University

Jiale Hu

Inner Mongolia University

Guoliang Zhao (✉ guoliangzhao@imu.edu.cn)

Inner Mongolia University <https://orcid.org/0000-0002-0323-1299>

Sharina Huang

Inner Mongolia Agriculture University: Inner Mongolia Agricultural University

Yang Liu

Inner Mongolia University

Research Article

Keywords: Extreme learning machine (ELM), Random vector functional link network (RVFL), Tensor-based type-2 random vector functional link network (T2-RVFL), Tensor-based type-2 Extreme learning machine (TT2-ELM), Tensor stacked fuzzy neural network (TSFNN)

Posted Date: November 11th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1035440/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A Tensor Based Stacked Fuzzy Networks for Efficient Data Regression

Jie Li¹, Jiale Hu², Guoliang Zhao^{3*†}, Sharina Huang⁴ and Yang Liu⁵

¹College of Electronic Information Engineering, Inner Mongolia University, Hohhot, 010021, China.

²College of Electronic Information Engineering, Inner Mongolia University, Hohhot, 010021, China.

³College of Electronic Information Engineering, Inner Mongolia University, Hohhot, 010021, China.

⁴College of Science, Inner Mongolia Agricultural University, Hohhot, 010018, China.

⁵College of Electronic Information Engineering, Inner Mongolia University, Hohhot, 010021, China.

*Corresponding author(s). E-mail(s): guoliangzhao@imu.edu.cn;

Contributing authors: lijieeffort@163.com; hjl18747978781@163.com; hsrn1982@163.com; yangliuimu@163.com;

†These authors contributed equally to this work.

Abstract

Random vector functional link and extreme learning machine have been extended by the type-2 fuzzy sets with vector stacked methods, this extension leads to a new way to use tensor to construct learning structure for the type-2 fuzzy sets-based learning framework. In this paper, type-2 fuzzy sets-based random vector functional link, type-2 fuzzy sets-based extreme learning machine and Tikhonov-regularized extreme learning machine are fused into one network, a tensor way of stacking data is used to incorporate the nonlinear mappings when using type-2 fuzzy sets. In this way, the network could learning the sub-structure by three sub-structures' algorithms, which are merged into one tensor structure via the type-2 fuzzy mapping results. To the stacked single fuzzy neural network, the consequent part parameters learning are implemented by unfolding tensor-based matrix regression. The newly proposed stacked single fuzzy neural network shows a new way to design the hybrid fuzzy neural network with the higher order fuzzy sets and higher order data structure. The effective of the proposed stacked single fuzzy neural network are verified by the classical testing benchmarks and several statistical testing methods.

Keywords: Extreme learning machine (ELM), Random vector functional link network (RVFL), Tensor-based type-2 random vector functional link network (T2-RVFL), Tensor-based type-2 Extreme learning machine (TT2-ELM), Tensor stacked fuzzy neural network (TSFNN)

1 Introduction

The random vector functional link (RVFL) and extreme learning machine (ELM) are two popular randomized single layer forward learning networks, which provide us a unified framework for both regression and multi-class classification with single layer. Then the semi-supervised RVFL and ELM networks can be merged into a joint optimization framework, it shows that the algorithm is efficient in moderate scale data classification (Peng et al, 2020). The parameters could be regularized when ridge regression is used (Yildirim and Revan Özkale, 2019). When singular value decomposition (SVD) is used for algorithm iterative solution searching, the SVD update algorithm scales better and works faster than SVD computed from scratch (Grigorievskiy et al, 2016). Multi-label learning method could also use the multi-label radial basis function neural network and Laplacian ELM (Xu et al, 2019), in this algorithm, the clustering algorithm determines the number of hidden nodes, and the center of the activation function could be determined by the data itself, then the output is solved by a Laplacian ELM. Inspired by biological intelligent systems, bio-inspired learning model blooms a lot recently (Huang and Chen, 2016; Alencar et al, 2016; Christou et al, 2019), its can be applied to many areas, such as, anomalous trajectory classification (Sekh et al, 2020), long-term time series prediction (Grigorievskiy et al, 2014), T-S fuzzy model identification (Wei et al, 2020), dictionary learning-based image classification (Zeng et al, 2020a), anomaly detection (Hashmi and Ahmad, 2019), HRV recognition (Bugnon et al, 2020), energy system (Yaw et al, 2020), mislabeled samples detection (Akusok et al, 2015), concept drift detection (Yang et al, 2020c), etc.

Generalization performance is the main concern for the learning algorithms, balancing computational complexity and generalization ability have been extended via ELM (Ragusa et al, 2020); with the designed data and modeled parallel ELMs, large-scale learning tasks could be tackled by ELM (Ming et al, 2018), moreover, a tradeoff should be made among efficiency and scalability, the algorithm should have complementary advantages. With the aid of graph learning and adaptive unsupervised/semi-supervised clustering method, flexible and discriminative data

embedding could be achieved (Zeng et al, 2020b; Zheng et al, 2020). By using the regularized corentropy criterion and half-quadratic optimization technique, convergence speed and performance are both showed superiorities than the original (Yang et al, 2020a), and the robust type algorithm has been studied in (Yang et al, 2020b). When inverse-free recursive algorithm is used to update the inverse of the networks' Hermitian matrix, efficient inverse-free algorithm is designed to update the regularized pseudo-inverse has been proposed for ELMs (Zhu and Wu, 2020).

When big data environment is encountered, a fast parameter selection scheme for modeling the large amount of data is needed, alternating direction method and maximally splitting method could be applied to the algorithms to minus the number of the sub-model and coefficients training (Lai et al, 2020). Concerning the credit probability for network output for each prediction, probabilistic output from the original architecture of ELM is proposed, iterative way of learning is eliminated, and the merits of ELM is preserved (Wong et al, 2020). Using Bayesian inferences, multiple-instance learning-based ELM has proven to be efficient in classification problems (Wang et al, 2020). Optimally pruned ELM (Miche et al, 2010) is presented to both regression and classification problems, the proposed algorithm could counter the effect of noise. To move forward, a L_2 regularization penalty applied to the optimally pruned ELM, and a double-regularized ELM using LARS and Tikhonov regularization is proposed (Miche et al, 2011b). Missing data case for the regression problem is studied in (Yu et al, 2013). When training sample selection method is designed based on the fuzzy C-means clustering algorithm, and the proposed small training samples selection-based hierarchical ELM could reduce the computational time (Xu et al, 2020).

Random vector functional link networks (RVFL) (Zhang and P.N.Suganthan, 2016) could also use the techniques mentioned above, such as ridge regression (Zhang and Suganthan, 2017), and its extended version, the new learning paradigm is named as RVFL plus (RVFL+) (Zhang, 2020), it has been used in neuro-imaging-based parkinson's disease diagnosis (Xue et al, 2018; Shi et al, 2019). Motivated by the ELMs and RVFLs, generalized Moore-Penrose inverse and triangular type-2 fuzzy sets are used to extend the

ELM, and tensor-based ELM has been proposed (Huang et al, 2019). RVFL network has been also expanded to tensor case (Zhao and Wu, 2019), the type-reduction method for general type-2 fuzzy sets are removed in this type of network.

Motivated by the above-mentioned material, we have noticed that the type-2 fuzzy sets and tensor structure provide a new way to model the complex data, whether the ELMs, RVFLs or the neuro-fuzzy systems could be used under the tensor structure. Our target is to unveil the links or laws behind the data, and a new tensor-based stacked networks for efficient data regression is studied. To get the merits of the algorithms, a good way is to fuse the algorithms into one frame, the balance of performance and structure simplicity would be achieved.

To fuse different concept and techniques into the algorithm, it is inevitable to extract the different aspect of the data, then different view results of the data can be obtained, then the original proposed algorithms could be used to minimize the testing error. Go back to the type-2 fuzzy sets, it could map the data with different parameter-specified fuzzy membership functions with at least three type-1 fuzzy membership functions, then the multi-view on the data could be obtained. A question follows this is how to fuse the results into one data structure, tensor is the suitable choice for these types of learning methods, this is the motivation of the work.

The structure of the rest of this article is as follows: Section 2 introduces the three algorithms that is used by stacking system, Section 2.1 introduces the tensor-based type-2 RVFL, Section 2.2 presents the tensor-based type-2 ELM, and Section 2.3 is the introduction to the TROP-ELM, it is used to compare algorithms' performance. In Section 3, the structure of tensor based hybrid single fuzzy neural networks, that is, a stacked single fuzzy neural network is presented. Simulation results and discussions are given in Section 4. Finally, conclusions are inferred in Section 5.

2 Preliminary

In this station, the tensor-based type-2 RVFL, tensor based type-2 ELM and Tikhonov regularized OP-ELM is introduced.

2.1 Tensor-based type-2 RVFL

The RVFL usually adopts the activation functions to construct the network, for example the Radbas ($y = e^{-s^2}$) functions. In the hidden layers of the network, where y and s are defined as the outputs and inputs, respectively. The enhancement nodes of tensor-based RVFL are replaced with IT2 fuzzy sets. The structure of tensor-based type-2 RVFL (TT2-RVFL) is represented in Fig. 1. Activation function Radbas of RVFL is extended to interval type-2 fuzzy set IT2Radbas, and the extended RVFL is constructed using IT2Radbas. The Radbas activation function in RVFL is extended to interval type-2 fuzzy sets IT2Radbas, and IT2Radbas is used to construct IT2RVFL.

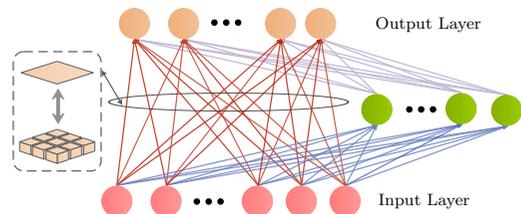


Fig. 1 Structure of the TT2-RVFL network.

Fig. 2 shows the standard formulation of the IT2 fuzzy set. The incentive function is named IT2Radbas, and the interval type-2 fuzzy set could be constructed with this type of activation functions.

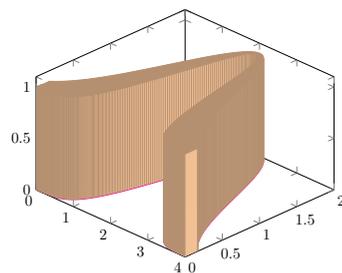


Fig. 2 Structure of the it2 fuzzy set.

The membership function (MF) of type-2 fuzzy sets in TT2-RVFL is defined as follows

$$\bar{g}_i(x_i) = \exp(-k_1 s^2), \quad (1)$$

$$\underline{g}_i(x_i) = \exp(-k_2 s^2), \quad (2)$$

where $s = x_i - m_i, k_1 = \frac{1}{\sigma_i^2}, k_2 = \frac{1}{\sigma_i^2}$ ($i = 1, 2, \dots, L$).

Given testing data set $\{\mathbf{D}_t\}_{t=1}^N$, where $\mathbf{D}_t = (\mathbf{x}_t, \mathbf{y}_t)$, $\mathbf{x}_t \in \mathbb{R}^N$ and $\mathbf{x}_t = (x_{t1}, x_{t2}, \dots, x_{tN})$, $\mathbf{y}_t \in \mathbb{R}$ and $\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]^T$. For a lower MF matrix $\underline{\Phi} \in \mathbb{R}^{N \times L \times 1}$ can be structured with the following $\mathbb{R}^{N \times L}$ matrices as follows

$$\underline{\Phi}_{:,1,1} = \begin{bmatrix} \underline{g}_1(\mathbf{a}_{11}\mathbf{x}_1 + b_{11}) & \underline{g}_1(\mathbf{a}_{12}\mathbf{x}_1 + b_{12}) \\ \vdots & \vdots \\ \underline{g}_1(\mathbf{a}_{11}\mathbf{x}_N + b_{11}) & \underline{g}_1(\mathbf{a}_{12}\mathbf{x}_N + b_{12}) \\ \vdots & \vdots \\ \underline{g}_L(\mathbf{a}_{L1}\mathbf{x}_1 + b_{L1}) & \underline{g}_L(\mathbf{a}_{L2}\mathbf{x}_1 + b_{L2}) \\ \vdots & \vdots \\ \underline{g}_L(\mathbf{a}_{L1}\mathbf{x}_N + b_{L1}) & \underline{g}_L(\mathbf{a}_{L2}\mathbf{x}_N + b_{L2}) \end{bmatrix},$$

where b_{il} is bias and $\mathbf{a}_{il} = [w_{i1}, w_{i2}, \dots, w_{iK}]$ ($i = 1, 2, \dots, L; l = 1, 2$) is input weights, respectively; they are randomly generated. By the definition of the IT2 fuzzy sets' lower MF, the relationship between input \mathbf{x}_t and expected output \mathbf{y}_t could be approximated by the lower MF matrix. The principal MF matrix $\hat{\Phi} \in \mathbb{R}^{N \times L \times 1}$ and the upper MF matrix $\bar{\Phi} \in \mathbb{R}^{N \times L \times 1}$ can also be structured similarly, with the following formulas

$$\hat{\Phi}_{:,1,2} = \begin{bmatrix} \hat{g}_1(\mathbf{a}_{11}\mathbf{x}_1 + b_{11}) & \hat{g}_1(\mathbf{a}_{12}\mathbf{x}_1 + b_{12}) \\ \vdots & \vdots \\ \hat{g}_1(\mathbf{a}_{11}\mathbf{x}_N + b_{11}) & \hat{g}_1(\mathbf{a}_{12}\mathbf{x}_N + b_{12}) \\ \vdots & \vdots \\ \hat{g}_L(\mathbf{a}_{L1}\mathbf{x}_1 + b_{L1}) & \hat{g}_L(\mathbf{a}_{L2}\mathbf{x}_1 + b_{L2}) \\ \vdots & \vdots \\ \hat{g}_L(\mathbf{a}_{L1}\mathbf{x}_N + b_{L1}) & \hat{g}_L(\mathbf{a}_{L2}\mathbf{x}_N + b_{L2}) \end{bmatrix},$$

It also forms the upper membership function value-filled tensor, the slices of tensor are

$$\bar{\Phi}_{:,1,3} = \begin{bmatrix} \bar{g}_1(\mathbf{a}_{11}\mathbf{x}_1 + b_{11}) & \bar{g}_1(\mathbf{a}_{12}\mathbf{x}_1 + b_{12}) \\ \vdots & \vdots \\ \bar{g}_1(\mathbf{a}_{11}\mathbf{x}_N + b_{11}) & \bar{g}_1(\mathbf{a}_{12}\mathbf{x}_N + b_{12}) \\ \vdots & \vdots \\ \bar{g}_L(\mathbf{a}_{L1}\mathbf{x}_1 + b_{L1}) & \bar{g}_L(\mathbf{a}_{L2}\mathbf{x}_1 + b_{L2}) \\ \vdots & \vdots \\ \bar{g}_L(\mathbf{a}_{L1}\mathbf{x}_N + b_{L1}) & \bar{g}_L(\mathbf{a}_{L2}\mathbf{x}_N + b_{L2}) \end{bmatrix}.$$

where \mathbf{a}_{ij} ($i = 1, 2, \dots, L; j = 1, 2$) is a random generated weighted vector constructed by tensor.

Remark 1 The uncertain weight method (Runkler et al, 2018) is applied to compute the principal MF matrices $\hat{\Phi}$, which reflects the impact of MF value on the whole defuzzification result of the set. The mapping results are obtained as follows:

$$u_{UW}(x) = \frac{1}{2}(u(x) + \bar{u}(x)) \cdot (1 + \underline{u}(x) - \bar{u}(x))^\zeta, \quad (3)$$

where $\zeta > 0$ measures the uncertainty of lower membership value on type reduction results through the formula $(1 + \underline{u}(x) - \bar{u}(x))^\zeta$. The uncertainty weight method expands the simple method to obtain the mean value of upper and lower MF value. The legible output can be given as follows:

$$(1 + \underline{u}(x) - \bar{u}(x))^\zeta = \begin{cases} 0, & \underline{u}(x) = 0, \bar{u}(x) = 1, \\ 1, & \underline{u}(x) = \bar{u}(x). \end{cases} \quad (4)$$

Formula (4) shows that for $\zeta = 1$, the defuzzification results increase linearly with uncertainty; the defuzzification results are less than linear when $\zeta < 1$; the defuzzification results would be greater than linear when $\zeta > 1$.

Remark 2 For the expansion of fuzzy sets from type-1 fuzzy MFs to IT2 fuzzy set, when (4) is used for defuzzification, the extension of IT2 fuzzy set to the enhanced part is called IT2-RVFL (interval type-2 random vector function link).

Finally, a 3-tensor $\Phi \in \mathbb{R}^{N \times L \times 3}$ is established by the three foregoing membership function $\underline{\Phi}_{:,1,1}$, $\hat{\Phi}_{:,1,2}$ and $\bar{\Phi}_{:,1,3}$. Thereinafter, because of the relevant usage of \mathcal{A} in tensor equations, Φ will be changed to another capital letter \mathcal{A} in the next section. It can be known from the relevant content of the tensor equation that in the enhanced node of TT2-RVFL, the weighting matrix is a matrix of $L \times 3$, so the weighting matrix of TT2-RVFL can be defined as

$$\mathcal{X}_1 = \begin{bmatrix} \mathbf{w}_{L+11} & \mathbf{w}_{L+12} & \mathbf{w}_{L+13} \\ \mathbf{w}_{L+21} & \mathbf{w}_{L+22} & \mathbf{w}_{L+23} \\ \vdots & \vdots & \vdots \\ \mathbf{w}_{2L1} & \mathbf{w}_{2L2} & \mathbf{w}_{2L3} \end{bmatrix}. \quad (5)$$

To TT2-RVFL, the output model can be fused into one matrix by the following equation

$$\mathbf{Y}_1 = \alpha \mathcal{A}_1 \mathcal{X}_1 + (1 - \alpha) \mathbf{X}_1 \Omega_1. \quad (6)$$

where $\alpha \in [0, 1]$ is called the equilibrium coefficient of TT2-RVFL, \mathcal{A}_1 is denoted as mapping consequences of non-linear interval type-2 activation function, \mathcal{X}_1 is the weighted matrices for the enhanced part, matrix $\mathbf{Y}_1 = [\mathbf{y} \mathbf{y}]$; define \mathbf{a}_i ($i = 1, 2, \dots, L$) as a weight vector from input layer to the intensification nodes are stochastically generated, in such a way, the activation functions in $\underline{\Phi}$, $\bar{\Phi}$ are not fully saturated; \mathbf{X}_1 is the input matrix that is structured by input samples, and $\Omega_1 = [\omega \omega]$ is used to denote the unresolved input weight matrix.

Remark 3 In the light of equation (6), when $\alpha = 0$, TT2-RVFL will be degenerated to RVFL; when $0 < \alpha < 1$, TT2-RVFL is a mixed model of tensor-based extreme learning machine and RVFL; when $\alpha = 1$, TT2-RVFL will be transformed to tensor-based extreme learning machine.

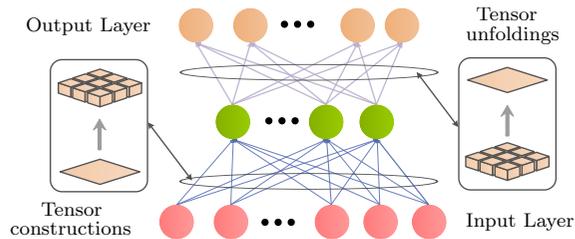


Fig. 3 Structure of the TT2-ELM network.

2.2 Tensor based type-2 ELM

The TT-2ELM (tensor based type-2 extreme learning machine) was first proposed in (Huang et al, 2019). The advantage of tensor structure is that the information of quadratic MF can be contained and modelled directly into one high dimensional array. This characteristic can avoid the type reduction operation in the process of type-2 fuzzy reasoning. Therefore, the tensor structure can seamlessly embed type-2 fuzzy sets into ELM.

Fig. 3 intuitively shows the structure of TT2-ELM. Obviously, TT2-ELM is a single hidden layer feed forward neural network. For the specified test dataset, which is expressed as (\mathbf{x}_t, y_t) , where $\mathbf{x}_t = (x_{t1}, x_{t2}, \dots, x_{tK}) \in \mathbb{R}^K$ represents inputs, and $y_t \in \mathbb{R}$ represents outputs. The

mathematical model of TT2-ELM is formed as

$$\mathcal{A}_2 *_N \mathcal{X}_2 = \mathcal{Y}_2, \quad (7)$$

where $\mathcal{A}_2 \in \mathbb{R}^{I_1 \dots I_2 \times J_1 \dots J_2}$ can be reshaped by Φ with a specified size $N \times L \times 3$, the $N \times L \times 3$ is regression tensor's dimension, N is the training patterns, $\mathcal{X}_2 \in \mathbb{R}^{J_1 \dots J_2}$ is the value of output weight, $\mathcal{Y}_2 \in \mathbb{R}^{I_1 \dots I_2}$ is the output matrix, Obviously, when $N = 2$, equation (7) degenerates into matrices case.

According to (Huang et al, 2019) (Theorem 1), for multi-linear system (7), if there exist any $\mathcal{X}_2 \in \mathbb{R}^{J_1 \dots J_2}$, then the multi-linear system (7) is solvable, and the solution of the equation (7) is

$$\mathcal{X}_2 = \mathcal{A}_2^{(1)} *_N \mathcal{Y}_2, \quad (8)$$

where $\mathcal{A}_2^{(1)}$ is a solution of $\mathcal{A}_2 *_N \mathcal{X}_2 *_N \mathcal{A}_2 = \mathcal{A}_2$. The resultant \mathcal{X}_2 in formula (8) is the solution of TT2-ELM.

For the equation (7), if no $\mathcal{X}_2 \in \mathbb{R}^{J_1 \dots J_2}$ can be obtained via the analysis, then the multi-linear system (7) is unsolvable, and the equation (7) has a minimum norm solution alternatively.

The following tensor equation can be obtained.

$$(\mathcal{A}_2^T *_N \mathcal{A}_2) *_N \mathcal{X}_2 = \mathcal{A}_2^T *_N \mathcal{Y}_2. \quad (9)$$

The gain tensor $\mathcal{A}_2^T *_N \mathcal{A}_2$ in equation (9) can be considered as 'square tensor', and the solution of tensor equation can still be obtained. Based of theory of tensor and kernel, the following equation can be determined.

$$\mathcal{A}_2 *_N (\mathcal{A}_2^{(1)} *_N \mathcal{A}_2) *_N \mathcal{Z} = \mathcal{A}_2 *_N \mathcal{Z}, \quad (10)$$

where \mathcal{Z} is a random tensor that can satisfy formula (10). As long as this tensor has a suitable order, it can be established.

A generally accepted condition is $(\mathcal{I} - \mathcal{A}_2^{(1)} *_N \mathcal{A}_2) *_N \mathcal{Z}$ gratifies $\mathcal{A}_2 *_N (\mathcal{I} - \mathcal{A}_2^{(1)} *_N \mathcal{A}_2) *_N \mathcal{Z} = 0$. This condition indicates that $(\mathcal{I} - \mathcal{A}_2^{(1)} *_N \mathcal{A}_2) *_N \mathcal{Z}$ is included in the null space of $\mathcal{A}_2 *_N \mathcal{X}_2 = 0$. Therefore, A general solution can be obtained as following

$$\mathcal{X}_2 = \mathcal{A}_2^{(1)} *_N \mathcal{Y}_2 + (\mathcal{I} - \mathcal{A}_2^{(1)} *_N \mathcal{A}_2) *_N \mathcal{Z}, \quad (11)$$

where $\mathcal{A}_2^{(1)}$ represents the 1-inverse of \mathcal{A}_2 . The minimum solution for the tensor equation consists of two parts, one of which comes from the null space of equation $\mathcal{A}_2 *_{\mathcal{N}} \mathcal{X}_2 = 0$. Note that the tensor equation that is formulated by (7) and (11), we have

$$\begin{aligned} \mathcal{X}_2 = & (\mathcal{A}_2^T *_{\mathcal{N}} \mathcal{A}_2)^{(1)} *_{\mathcal{N}} \mathcal{A}_2^T *_{\mathcal{N}} \mathcal{Y}_2 + \\ & (\mathcal{I} - (\mathcal{A}_2^T *_{\mathcal{N}} \mathcal{A}_2)^{(1)} *_{\mathcal{N}} (\mathcal{A}_2^T *_{\mathcal{N}} \mathcal{A}_2)) *_{\mathcal{N}} \mathcal{Z}. \end{aligned} \quad (12)$$

For equation (12), if $\mathcal{Z} = 0$, \mathcal{X}_2 is the minimizer. In the light of Corollary 2.14(1) (Behera and Mishra, 2017), the existence of $(\mathcal{A}_2^T *_{\mathcal{N}} \mathcal{A}_2)^{(1)}$ makes

$$\mathcal{A}_2^+ = (\mathcal{A}_2^T *_{\mathcal{N}} \mathcal{A}_2)^{(1)} *_{\mathcal{N}} \mathcal{A}_2^T. \quad (13)$$

Subsequently, the least squares solution of equation (13) is shown below

$$\mathcal{X}_2 = \mathcal{A}_2^+ *_{\mathcal{N}} \mathcal{Y}_2. \quad (14)$$

Equation (14) is the minimum norm solution of equation (7).

2.3 Tikhonov Regularized OP-ELM

In (Miche et al, 2010), Yoan Miche et al. first proposed OP-ELM, it is an improvement of ELM. Fig. 4 shows the Tikhonov regularized ELM (TROP-ELM) (Miche et al, 2011a). First, the OP-ELM network uses three different types of activation functions to form the kernel, and the assembled kernel is able to improve robustness and generality. For the initial condition of the ELM algorithm, Sigmoid kernel is used in its structure, while OP-ELM could use linear kernel, Sigmoid kernel and Gaussian kernel. Second, compared with the original proposed ELM, the multi-response sparse regression algorithm (MRSR) and the verification method leave one out (LOO) are also introduces in OP-ELM. The main role is to prune irrelevant variables by pruning the related neurons of SLFN constructed by ELM. The MRSR algorithm can rank neurons according to the usefulness of neurons, and the actual pruning technique is performed by leave one out validation results.

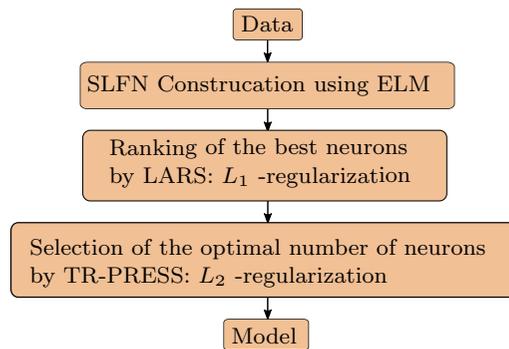


Fig. 4 Implementation steps of TROP-ELM

3 Tensor based hybrid single fuzzy neural networks design

In the previous section, three single fuzzy neural networks, that are, the TT2-ELM, TT2-RVFL and TROP-ELM are briefed. In this section, the three networks are stacked into one network, and a regression method that is based on the three regression results are introduced. The architecture is shown in Fig. 5.

The framework of tensor-based stacked single fuzzy neural network that is design based on three single fuzzy neural networks (TT2-ELM, TT2-RVFL and TROP-ELM), while the three algorithms have been proposed by researchers. The layer 2 is the hidden layer that is constituted by the three algorithms to construct a tensor structure. The tensor structure could be constructed in layer 3, then the tensor is unfolded into three different matrices for each single network. The final regression uses a simple normalized scalar weighting, which is the part to be optimized in the future.

The unfolding of tensor could use the definitions from (Yu et al, 2019), which are appended as definition 1.

Definition 1 ($\mathbf{m}, \boldsymbol{\nu}$)-unfolding. Consider an tensor with N -dimensional, and the tensor follows the dimension that $\mathcal{A} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$, $\chi(\mathbf{m}, \boldsymbol{\nu})$ is a k -partition of $m \in \mathbb{A}_N^n$, where $\boldsymbol{\nu} \in \mathbb{A}_{n+1}^{k+1}$ and $1 = \nu_1 < \nu_2 < \dots < \nu_{k+1} = n + 1$. The $(\mathbf{m}, \boldsymbol{\nu})$ -unfolding of \mathcal{A} , denoted by $\mathcal{A}_{\|\mu, nu\|}$, is a tensor with the size of $\left(\prod_{\nu=\nu_1}^{\nu_2-1} J_{m_\nu}\right) \times \dots \times \left(\prod_{\nu=\nu_k}^{\nu_{k+1}-1} J_{m_\nu}\right) \times J_{\bar{m}_1} \times \dots \times J_{\bar{m}_{N-n}}$ such that the (j_1, j_2, \dots, j_N) -th entry of \mathcal{A} is the entry of $\mathcal{A}_{\|\mathbf{m}, \boldsymbol{\nu}\|}$ at the position

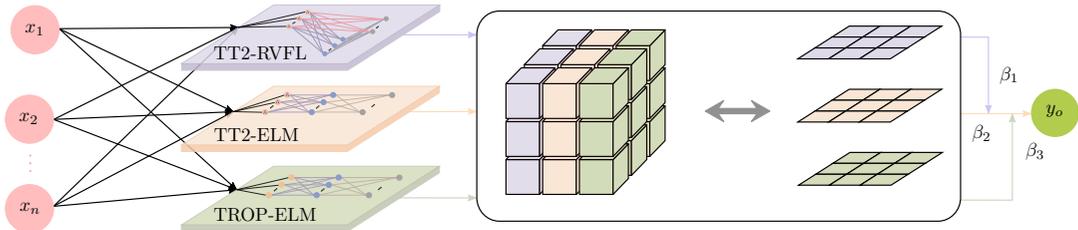


Fig. 5 The structure of TensorSFNN.

$(\varrho(jm_1; J_{m_1}), \dots, \varrho(jm_k; J_{m_k}), j\bar{m}_1, \dots, j\bar{m}_{N-n})$, where $\mathbf{J} = [J_1 \ J_2 \ \dots \ J_N]^T$, $\bar{\mathbf{m}} \in \mathbb{A}_N(\mathbf{m})$ and $\varrho(\cdot; \cdot)$ is the linear index of a multi-dimensional array, see, e.g., (Baranyi, 2016) and (Baranyi et al, 2014).

For example, $\mathbf{m} = [m_1 \ m_2 \ m_3]^T = [1 \ 2 \ 3]^T \in \mathbb{R}^3$, $\boldsymbol{\nu} = [1 \ 3 \ 4]^T \in \mathbb{A}_{3+1}^{2+1}$, then $\chi(\mathbf{m}, \boldsymbol{\nu}) = \{\mathbf{m}_1 \ \mathbf{m}_2\}$ is a 2-partition of \mathbf{m} , where $\mathbf{m}_1 = [m_1 \ m_2]^T = [1 \ 2]^T$, $\mathbf{m}_2 = m_3 = 3$.

Remark 4 By comparing the structures of TT2-RVFL, TT2-ELM and TROP-ELM, we have made a few modifications to TT2-RVFL and TT2-ELM. The original activation function of TT2-RVFL and TT2-ELM is Sigmoid function, while TROP-ELM uses linear activation function and Gaussian activation function in addition to Sigmoid function. In order to help ensure that their structures correspondingly and facilitate the composition of tensor structures, linear activation function and Gaussian activation function are also added to TT2-RVFL and TT2-ELM. After adding two distinct activation functions, TT2-ELM and TT2-RVFL obtained better performance than the original proposed ones based on our test.

The 3-tensor $\mathcal{A} \in \mathbb{R}^{N \times L \times 3}$ generates three matrices, and the row number of matrices is just equal to sample number. Three matrices are denoted by N_1, N_2 and N_3 , respectively, and $N_k \in \mathbb{R}^{N \times L}$, $k = 1, 2, 3$. The three matrices, N_1, N_2 and N_3 could reconstruct the tensor $\mathcal{A} \in \mathbb{R}^{N \times L \times 3}$ easily, that is, N_1 is the first aspect of $A(:, :, 1)$, which is the mapping results from the TT2-RVFL. N_2 is the second aspect of $A(:, :, 2)$, which is the mapping results from the TT2-ELM. and N_3 is the three aspect of $A(:, :, 3)$, which is the mapping results from the TROP-ELM.

To the type-2 fuzzy networks, the LMF, UMF and defuzzification of the secondary membership function are used to solve the consequent parameters' learning problem. To the regression

layer, N_1, N_2 and N_3 are three matrices that is unfolded from the tensor, regression equation can be denoted as $N_i W_i = \mathbf{y}_{ti}$, where $N_i \in \mathbb{R}^{N \times L \times 3}$, $W_i \in \mathbb{R}^{L \times 3}$ and $\mathbf{y}_t \in \mathbb{R}^{N \times 1}$.

To make the network perform best, first, the error of each network is calculated, in addition, root mean square error is used as the measurement standard. Second, the best-performing network are found out, which is used to train the other two network errors. After processing, the network is recorded as y'_t . Finality, The output are averaged based on y'_t , that is

$$\hat{y}_{kt} = \sum_{k=1}^3 \alpha_k y'_t, \quad \sum_{k=1}^3 \alpha_k = 1 \quad (15)$$

which is the average results of the three type-2 fuzzy networks, the MF values are obtained from lower MF and upper MF, since the matrices are the unfolded results from the tensor.

And the defuzzification result could be calculated from secondary MFs, it could also be used for the stacked SFNN. The unconstraint regression result of $\mathbf{A}_k \boldsymbol{\beta}_k = \mathbf{y}_t$ could be denoted as $\hat{\boldsymbol{\beta}}_k = (\mathbf{A}_k^T \mathbf{A}_k)^{-1} \mathbf{A}_k^T \mathbf{y}_t$. In statistics, this method is known as ridge regression, it is associated to the Levenberg–Marquardt algorithm and Andrey Tikhonov method to solve the regularization of ill-posed non-linear least-squares problems. Suppose that for a known matrix \mathbf{N}_k and vector \mathbf{y}_t , a vector \mathbf{x} is expected to be found, such that

$$\mathbf{A}_k \boldsymbol{\beta}_k = \mathbf{y}_t. \quad (16)$$

In most of the cases, ordinary least squares estimation leads to an overdetermined (overfitted), or more often an underdetermined (underfitted) system of equations. Therefore, in solving the ill-posed inverse-problem, the inverse mapping operators that has the undesirable tendency of

amplifying noise (The eigenvalue value is maximum in the reverse mapping and the singular value is minimum in the forward mapping). Moreover, every element of the reconstructed version of \mathbf{x} is implicitly nullified by ordinary least squares, instead of taking a model to as a prior of \mathbf{x} . For the purpose of the minimize residuals sum of squares, and the particular solution also satisfies some suitable qualities, a regularization term which can be added to this primary minimization problem, it can be succinctly scripted as follows

$$\|\mathbf{A}_k \boldsymbol{\beta}_k - \mathbf{y}_t\|^2 + \|\Gamma_k \boldsymbol{\beta}_k\|^2, \quad (17)$$

where $\|\cdot\|$ is the Euclidean norm, Γ_k is a appropriately selected Tikhonov weighting matrix.

Under many circumstances, the matrix Γ_k is selected as a multiple of the character matrix $\alpha_k I$, By L_2 regularization, solutions with smaller norms can be found (Ng, 2004). At other times, in the event that the fundamental vectors are considered primarily uninterrupted, a low-pass operator can be accustomed to strengthen flatness. This canonicalization enhances the conditioning of the problem, which leads to a straightforward numerical solution. An approximated solution is signified through $\hat{\mathbf{x}}$, which is presented by:

$$\hat{\boldsymbol{\beta}}_k = (\mathbf{A}_k^T \mathbf{A}_k + \Gamma^T \Gamma)^{-1} \mathbf{A}_k^T \mathbf{y}_t. \quad (18)$$

the individual algorithm of the stacked single fuzzy neural network could use the regularized results for the tensor unfolded structure's learning method.

4 Simulation results for the datasets

In this section, the UCI benchmark dataset and the other four actual datasets are tested to evaluate the performance of this method. Among all the simulations, root mean square error (RMSE) was utilized to assess the performance of the TSFNN in this paper and the four comparison methods, which is given by

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^N (\hat{y}_t - y_t)^2} \quad (19)$$

where \hat{y}_t is the predicted signal, y_t is the target signal, and N is the length of the testing sequence.

All experiments are performed on a computer with AMD Ryzen 7 4800U with Radeon Graphics 1.80 GHz and 16 GB RAM. The result of total 5000 times were performed on each data set.

Table 1 Attributes of the testing datasets.

Dataset	#Attributes	#Train set	#Test set
Abalone	9	2923	1524
Airfoil self noise	6	1052	451
Auto-Mpg	7	274	118
Bank	9	5734	2458
Concrete slump	11	91	12
Diabetes	2	538	230
Delta aileron	6	1052	451
Delta elevators	7	6661	2856
Energy efficiency	9	537	231
Wine quality white	12	3429	1469
Electrical.class	10	5502	2359
Electrical.detect	7	8400	3601
Electrical.No fault	6	1656	709
Electrical.LG fault	6	790	339
Electrical.LLG fault	6	794	340
Electrical.LLL fault	6	767	329
Electrical.LLLG fault	6	793	340
Asteroid	7	1750	750
Covid19.Beijing	3	345	149
Covid19.Shanghai	3	345	149
Covid19.Tianjin	3	345	149
Covid19.Chongqing	3	345	149
Covid19.Arizona	3	316	136
Covid19.Washington	3	325	140
Covid19.California	3	316	136
Covid19.Illinois	3	317	136
Hertz50	96	277	120
Amprion	96	277	120
TenneTTSO	96	277	120
TransnetBW	96	277	120

the Abalone, Airfoil self noise, Auto-Mpg, Bank, Concrete slump, Diabetes, Delta aileron, Delta elevators, Energy efficiency and Wine quality white datasets could be download via the following url:

<https://archive.ics.uci.edu/ml/datasets.php>

4.1 Regression problems

In this section, ten realistic world regression problems are used for testing. Abalone is a dataset that is used to predict abalone's age by physical measurements, which includes the whole weight, shucked weight and viscera weight of abalone in Tasmania, and 4447 samples with 9 attributes is included in the dataset. Airfoil Self-Noise dataset is obtained from a series of aerodynamic and acoustic tests on two-dimensional, and three-dimensional airfoil blade profiles in an anechoic wind tunnel, it contains 1503 samples and 6 attributes. The data set of Auto-MPG assembles miles each gallon data with dissimilar car brands, it contains 392 samples with 8 attributes.

Table 2 Comparison results with TSFNN, TT2-RVFL, TT2-ELM, OP-ELM and TROP-ELM (A brief introduction is listed in Table 1).

Dataset	Method	Training (RMSE)		Testing (RMSE)	
		Mean	Std	Mean	Std
Abalone	TSFNN	2.11e+00	2.62e-02	2.13e+00	7.73e-02
	TT2-RVFL	2.13e+00	3.20e-02	2.27e+00	1.89e-01
	TT2-ELM	2.21e+00	2.63e-02	2.24e+00	7.03e-02
	OP-ELM	2.16e+00	6.48e-02	3.72e+01	6.07e+02
	TROP-ELM	2.11e+00	2.67e-02	5.58e+01	1.13e+03
Airfoil self noise	TSFNN	4.01e+00	1.18e-01	3.99e+00	1.75e-01
	TT2-RVFL	4.44e+00	2.21e-01	4.59e+00	2.73e-01
	TT2-ELM	4.80e+00	6.60e-02	4.82e+00	1.55e-01
	OP-ELM	4.35e+00	1.23e-01	4.47e+00	1.94e-01
	TROP-ELM	4.35e+00	1.19e-01	4.47e+00	2.49e-01
Auto MPG	TSFNN	3.73e-01	2.05e-02	3.60e-01	3.67e-02
	TT2-RVFL	4.62e-01	4.25e-02	5.20e-01	6.26e-02
	TT2-ELM	5.15e-01	1.26e-02	5.35e-01	3.02e-02
	OP-ELM	3.93e-01	2.35e-02	4.48e-01	7.37e-02
	TROP-ELM	3.86e-01	1.72e-02	4.45e-01	8.36e-02
Bank	TSFNN	2.85e-02	7.82e-04	2.85e-02	1.02e-03
	TT2-RVFL	5.23e-02	3.61e-03	5.27e-02	3.75e-03
	TT2-ELM	3.14e-02	3.69e-04	3.15e-02	8.61e-04
	OP-ELM	2.92e-02	5.95e-04	2.94e-02	9.07e-04
	TROP-ELM	2.92e-02	5.96e-04	2.94e-02	9.77e-04
Concrete slump	TSFNN	1.52e+00	1.81e-01	1.29e+00	2.16e-01
	TT2-RVFL	3.51e+00	5.71e-01	5.50e+00	1.15e+00
	TT2-ELM	2.27e+00	1.27e-01	2.71e+00	3.24e-01
	OP-ELM	1.91e+00	3.58e-01	2.89e+00	6.98e-01
	TROP-ELM	1.54e+00	1.88e-01	3.02e+00	2.40e+00
Delta aileron	TSFNN	3.63e-03	4.23e-05	3.65e-03	6.54e-05
	TT2-RVFL	3.67e-03	4.79e-05	3.70e-03	6.45e-05
	TT2-ELM	3.64e-03	4.56e-05	3.67e-03	6.47e-05
	TROP-ELM	3.93e-03	2.59e-04	3.95e-03	3.13e-04
	OP-ELM	3.92e-03	2.56e-04	3.94e-03	2.78e-04
Delta elevators	TSFNN	3.47e+00	2.65e-02	3.46e+00	4.56e-02
	TT2-RVFL	3.52e+00	3.59e-02	3.55e+00	6.96e-02
	TT2-ELM	3.58e+00	1.79e-02	3.59e+00	4.18e-02
	OP-ELM	3.50e+00	2.42e-02	3.52e+00	4.43e-02
	TROP-ELM	3.50e+00	2.40e-02	3.52e+00	4.79e-02
Diabetes	TSFNN	1.44e-01	4.21e-03	2.29e-01	5.32e+00
	TT2-RVFL	1.49e-01	4.79e-03	1.58e-01	9.59e-03
	TT2-ELM	1.54e-01	4.20e-03	1.58e-01	9.72e-03
	OP-ELM	—	—	—	—
	TROP-ELM	1.45e-01	4.27e-03	2.93e-01	2.54e+00
EnergyEfficiency ¹	TSFNN	2.47e+00	9.55e-02	2.42e+00	1.45e-01
	TT2-RVFL	3.62e+00	3.73e-01	3.82e+00	4.28e-01
	TT2-ELM	2.91e+00	6.23e-02	2.95e+00	1.46e-01
	OP-ELM	2.54e+00	9.72e-02	2.67e+00	1.57e-01
	TROP-ELM	2.48e+00	9.29e-02	2.62e+00	1.58e-01
wine quality white	TSFNN	6.23e-01	9.39e-03	6.15e-01	2.62e-02
	TT2-RVFL	6.43e-01	1.18e-02	6.64e-01	2.30e-02
	TT2-ELM	6.45e-01	8.69e-03	6.53e-01	2.04e-02
	OP-ELM	6.38e-01	1.14e-02	3.25e+00	9.09e+01
	TROP-ELM	6.27e-01	9.59e-03	1.09e+01	3.05e+02

The bank dataset simulates the customer's patience who select their favored services in the bank according to 8 factors, for example residential area, distance, virtual temperature regulating bank option and so on, it contains 8192 samples with 8 attributes. Concrete Slump dataset contains information about the factors that affect slump flow of concrete, it includes 103 samples with 11 attributes. Diabetes is a dataset that

investigates the reliance of the grade of serum C-peptide on various factors, it can be used to measure residual insulin secretion patterns, 768 samples with 4 attributes are included in the dataset. Delta ailerons and Delta elevators are recorded ailerons' data and elevators' data for delta, and they have 7129 samples with 6 attributes, and 9247 with 7 attributes, respectively. Energy efficiency is a dataset that is obtained by energy analysis of 12 various architectural shapes, which is simulated in Ecotect, and there are 768 samples and 8 features in it, the regression problem is for forecast 2 authentic valued responses that are cooling load and heating load. Wine quality white is a dataset associated with red and white wine samples, it contains 4898 samples with 12 attributes.

The information of the dataset is presented in Table 1, these datasets involve four small-scale datasets and six moderate-scale datasets. The mean and standard of 5000 experimental results of Abalone, Airfoil self-noise, Auto-Mpg, Bank, Concrete slump, Diabetes, Delta ailerons, Delta elevators, Energy efficiency and Wine quality white are showed in Table 2. TT2-RVFL, TT2-ELM, OP-ELM and TROP-ELM are used for algorithm comparison.

Results of Friedman test on these ten datasets for five method (TT2-RVFL, TT2-ELM, OP-ELM, TROP-ELM and TSFNN) are listed in Table 3. It can be inferred that the training error and testing error of the proposed method is the smallest. The stacked tensor-based hybrid single fuzzy neural networks of proposed can share advantages of TT2-RVFL, TT2-ELM and TROP-ELM, and the tensor quantization method of fuzzy system may be a method of extending type-2 fuzzy modeling.

Table 3 Results of Friedman test on ten datasets (The testing results are listed in bracket).

Algorithm	Mean rank	χ^2	p value
TSFNN	1.55 (1.34)		
TT2-RVFL	3.93 (4.07)	103393.205	< 0.05 (< 0.05)
TT2-ELM	4.48 (3.93)	(86753.311)	
OP-ELM	2.76 (2.85)		
TROP-ELM	2.28 (2.81)		

4.2 Simulations for other datasets

In this section, four datasets which are Electrical Fault detection and classification datasets, Asteroid Dataset, Novel Corona Virus 2019 Dataset and Wind Power Generation Data are used for testing model performance.

4.2.1 Electrical Fault Detection and Classification Dataset

Power systems consist of many complex dynamic and interactive elements that are always vulnerable to interference or electrical failures. Transmission lines are the most critical part of the power system, and the prominent role of transmission lines is to transmit electricity from the source area to the distribution destination in the network. The faults of power system transmission lines should be first correctly detected and classified, and should be eliminated in the shortest possible time. Electrical Fault detection and classification dataset contains the current and voltage of the line under different fault conditions (Jamil et al, 2015). The dataset contains the detection of power system faults and classifying fault types for the power system faults. The dataset of power system fault detection contains 12001 sampling data with six inputs (I_a , I_b , I_c , V_a , V_b , V_c) and two outputs (0 and 1), no fault is denoted by 0, and fault is denoted by 1.

The dataset of classifying fault types contains 7861 group data with six inputs (I_a , I_b , I_c , V_a , V_b , V_c) and four outputs (G, C, B and A), they represents 4 generators of 11×10^3 V, respectively. No fault occurs is denoted by 0, and fault occurs is denoted by 1. The combination of G, C, B and A represents various failures, and the failures faults are shown in Table 5.

The faults of the system are judged according to the current and voltage of the power system. The dataset consists of two outputs that represents whether the system faults. Fig. 6 shows that no faults in Power Electrical detection dataset. Similarly, Fig. 7 shows that faults occurs in Power Electrical detection dataset. The transverse axis delegates samples, and the longitudinal axis delegates the values of V_a , V_b , V_c , I_a , I_b and I_c . Compared with Fig. 6 and Fig. 7, if there is no fault in the electric power system, the values of current and voltage are generally stable, and its change trend is similar to the sin function, which

is also consistent with the characteristics of AC in the electric power system. However, once a fault occurs in the power system, the values of current and voltage are abnormal, which represents different fault locations. This anomaly can be clearly seen from Fig. 7. The comparison result of detection of power system faults dataset that is observed to determine whether the power system is faulty is shown in Table 4.

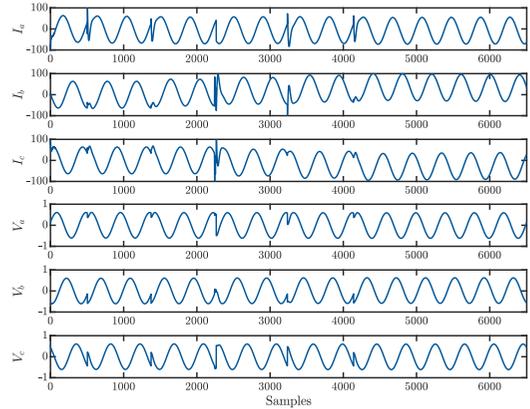


Fig. 6 The data of no faults in Electrical detect dataset.

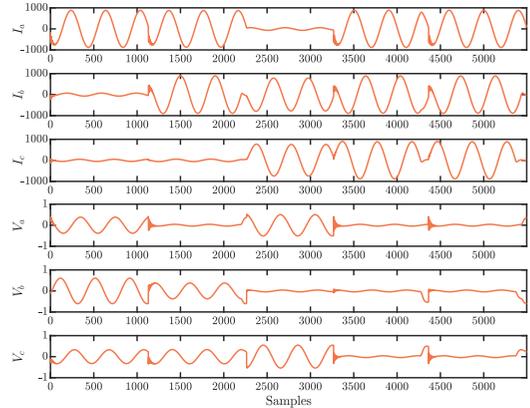


Fig. 7 The data has faults in Electrical detect dataset.

Comparison results of TT2-RVFL, TT2-ELM, OP-ELM, TROP-ELM and TSFNN on the Electrical Fault detection dataset are used to test the performance of the five algorithms. Results of TSFNN in the Table 4 show that the generalization ability of TSFNN is better than the other four algorithms. Moreover, data indicates power system failure can be considered as no power system failure added noise by comparing with Fig. 6 and Fig.

7. Therefore, Table 4 also show that the disturbance reject ability of TSFNN is better than the other four algorithms.

Table 4 Comparison results with TSFNN, TT2-RVFL, TT2-ELM, OP-ELM and TROP-ELM for Electrical-class, Electrical-detect (A brief introduction is listed in Table 1).

Dataset	Method	Training (RMSE)		Testing (RMSE)	
		Mean	Std	Mean	Std
Electrical _detect	TSFNN	4.78e-04	2.34e-03	7.67e-04	2.76e-03
	TT2-RVFL	3.39e-02	1.27e-02	3.07e-02	1.15e-02
	TT2-ELM	2.59e-02	9.87e-03	2.61e-02	9.94e-03
	OP-ELM	3.10e-03	1.73e-02	3.10e-03	1.73e-02
	TROP-ELM	3.60e-03	1.85e-02	3.61e-03	1.86e-02

For Electrical Fault classification case, we decompose Electrical Fault classification dataset into five parts according to the different locations of faults. It can be seen from Table 5 that the dataset provides six kinds of faults, but we do not find LL fault ($[G, C, B, A] = [0, 0, 1, 1]$) in the dataset, which represents the fault between Phase A and Phase B.

Through the above analysis, we know that in the power system, the faults data can be regarded as the no faults data with added noise. Thus, in the five extracted datasets, the dataset of LG fault, LLG fault, LLL fault, and LLLG fault can be treated as imposing different noise on the dataset with no fault status. Moreover, the extracted dataset represents only one power system fault, so its data is more pure and has more obvious characteristics and trends. Through the above-mentioned analysis, the anti-interference performance and generalization performance of the algorithm can be further verified. Results of TSFNN in the Table 6 fully proof that the excellent disturbance rejection and generalization performance of TSFNN.

Table 5 Faults represented by G, C, B and A.

$[G, C, B, A]$	Faults
[0, 0, 0, 0]	No fault
[1, 0, 0, 1]	LG fault (Between phase A and phase G)
[0, 0, 1, 1]	LL fault (Between phase A and phase B)
[1, 0, 1, 1]	LLG fault (Between Phases A, B and ground)
[0, 1, 1, 1]	LLL fault (Between all three phases)
[1, 1, 1, 1]	LLL fault (Three phase symmetrical fault)

4.2.2 Asteroid Dataset

The Asteroid Dataset is officially maintained by Jet Propulsion Laboratory of California Institute of Technology, which is an organization under

Table 6 Comparison results with TSFNN, TT2-RVFL, TT2-ELM, OP-ELM and TROP-ELM for various faults in Electrical dataset. (A brief introduction is listed in Table 1).

Dataset	Method	Training (RMSE)		Testing (RMSE)	
		Mean	Std	Mean	Std
Electrical _No fault	TSFNN	2.32e-03	2.02e-03	2.98e-03	2.34e-03
	TT2-RVFL	8.55e-03	3.84e-03	8.33e-03	4.09e-03
	TT2-ELM	7.16e-03	3.07e-03	7.79e-03	3.72e-03
	OPELM	3.75e-02	9.75e-03	3.80e-02	1.42e-02
	TROP-ELM	3.72e-02	9.86e-03	3.78e-02	1.37e-02
Electrical _LG fault	TSFNN	1.17e-03	1.12e-03	1.99e-03	1.94e-03
	TT2-RVFL	4.12e-03	1.85e-03	5.52e-03	3.76e-03
	TT2-ELM	3.70e-03	1.56e-03	5.69e-03	3.98e-03
	OPELM	1.85e-02	5.94e-03	1.89e-02	7.76e-03
	TROP-ELM	1.85e-02	5.85e-03	1.90e-02	7.35e-03
Electrical _LLG fault	TSFNN	2.02e-03	1.60e-03	2.84e-03	1.80e-03
	TT2-RVFL	5.91e-03	2.46e-03	7.99e-03	5.08e-03
	TT2-ELM	5.15e-03	2.03e-03	8.09e-03	5.34e-03
	OPELM	2.60e-02	7.28e-03	2.65e-02	8.87e-03
	TROP-ELM	2.60e-02	7.40e-03	2.64e-02	9.32e-03
Electrical _LLL fault	TSFNN	1.62e-03	7.97e-04	2.44e-03	2.59e-03
	TT2-RVFL	3.54e-03	1.47e-03	4.72e-03	3.30e-03
	TT2-ELM	3.13e-03	1.19e-03	4.80e-03	3.49e-03
	OPELM	2.40e-02	5.44e-03	2.70e-02	3.19e-02
	TROP-ELM	2.40e-02	5.52e-03	2.62e-02	2.54e-02
Electrical _LLL fault	TSFNN	2.28e-03	1.10e-03	3.16e-03	1.70e-03
	TT2-RVFL	3.92e-03	1.62e-03	4.77e-03	2.86e-03
	TT2-ELM	3.39e-03	1.29e-03	4.76e-03	2.98e-03
	OPELM	3.10e-02	8.00e-03	3.19e-02	1.39e-02
	TROP-ELM	3.10e-02	8.12e-03	3.18e-02	1.05e-02

NASA. The data set is publicly available in JPL Small-Body Database Search Engine. This dataset could also be obtained from [kaggle](#). Table 7 shows basic column definition for Asteroid dataset. A portion of the data is extract by us as a comparison test dataset when using Asteroid dataset. The 2500 samples in the dataset presents 7 attributes, these samples are applied to validate the proposed algorithm. These properties are Geometric albedo, Eccentricity, Semi-major axis, inclination angle about the x-y elliptic plane, Earth Minimum Orbit Intersection Distance and RMS for the Asteroid, respectively.

The comparison results with five methods are demonstrated in Table 8. The results demonstrate that the TSFNN performs best with respect to training error, while TT2-RVFL and TT2-ELM performs best in testing error. Meanwhile, the performance of OP-ELM and TROP-ELM is bad. Because the approach of proposed in this paper is superposition of TT2-RVFL, TT2-ELM and TROP-ELM. The main reason why TSFNN, OP-ELM and TROP-ELM perform well in training and poor in testing is that these three methods all use multi-response sparse regression (MRSR), it is a variable sorting technology that is extended from

Table 7 Basic column definition for Asteroid dataset.

Attributes	Description
SPK-ID	Object primary SPK-ID
Object ID	Object internal database ID
Object fullname	Object full name/designation
pdes	Object primary designation
name	Object IAU name
NEO	Near-Earth Object (NEO) flag
PHA	Potentially Hazardous Asteroid (PHA) flag
H	Absolute magnitude parameter
Diameter	object diameter (from equivalent sphere) km Unit
Albedo	Geometric albedo
Diameter_sigma	1-sigma uncertainty in object diameter km Unit
Orbit_id	Orbit solution ID
Epoch	Epoch of osculation in modified Julian day form
Equinox	Equinox of reference frame
e	Eccentricity
a	Semi-major axis au Unit
q	perihelion distance au Unit
i	inclination; angle with respect to x-y ecliptic plane
tp	Time of perihelion passage TDB Unit
moid_ld	Earth Minimum Orbit Intersection Distance au Unit

the least angle regression algorithm (Similä and Tikka, 2005) and (Efron et al, 2004). According to the usefulness of neurons, the MRSR algorithm can obtain a ranking of the neurons in OP-ELM (Miche et al, 2010). TROP-ELM is an improvement of OP-ELM, the MRSR method is also used for the input data of TROP-ELM. Due to the proposed TSFNN includes TROP-ELM, thus the TSFNN is affected by MSRS method.

To the MRSR, an important feature is that the obtained ordering is exact in the case of linear problems. The Asteroid dataset collects the attributes of asteroid. A part of its data is used, it is also nonlinear. And the OP-ELM and the TROP-ELM have one detail that the neural networks they constructed are linear between the hidden layer and the output layer, the role of MRSR algorithm is that will get an exact ranking of the neurons. The sequence obtained by sorting can be used to sort the kernels of model. When the whole dataset is nonlinear, so the exact ranking of neurons cannot be obtained by OP-ELM. Similarly, TROP-ELM and TSFNN are also affected by this flaw. Therefore, TSFNN performs well in the training part, and in the test part, due to MRSR method, the extracted data set features cannot be well applied to the testing set, resulting in poor performance of TSFNN in the testing phase.

According to the data in Table 8, the performance of TT2-RVFL and TT2-ELM is the best. The TT2-RVFL and the TT2-ELM are constructed by tensor structure and interval type-2 fuzzy sets. The membership degree of type-2 fuzzy set is characterized by type-1 fuzzy set. Since the

Table 8 Comparison results with TSFNN, TT2-RVFL, TT2-ELM, OP-ELM and TROP-ELM for Asteroid (A brief introduction is listed in Table 1).

Dataset	Method	Training (RMSE)		Testing (RMSE)	
		Mean	Std	Mean	Std
Asteroid	TSFNN	3.23e-02	5.16e-04	2.45e+01	5.62e+02
	TT2-RVFL	3.24e-02	5.34e-04	3.34e-02	1.35e-03
	TT2-ELM	3.23e-02	5.29e-04	3.34e-02	1.37e-03
	OP-ELM	3.33e-02	8.53e-04	5.08e+02	1.96e+04
	TROP-ELM	3.33e-02	8.41e-04	5.77e+02	1.34e+04

type-1 fuzzy set has a strong ability to deal with uncertainty in the system, so type-2 fuzzy set greatly strengthens the processing ability of fuzzy system for uncertainty and nonlinearity, and it has good performance in nonlinear systems with high uncertainty. Therefore, type-2 fuzzy systems have strong generalization ability. And the tensor structure is also good at dealing with uncertain systems, which can improve the generalization performance of the system.

The merits of type-2 fuzzy sets, and the tensor structure are inherited by the tensor-based type-2 fuzzy system. On the basis of the above analysis, TT2-RVFL and TT2-ELM have good performance on Asteroid dataset, their training error and testing error are minimal in Table 8.

From the test performance of TSFNN, since TSFNN contains TT2-RVFL and TT2-ELM, it makes up for the defect of insufficient generalization ability of TROP-ELM in nonlinear system. This also proves the excellent generalization ability of type-2 fuzzy systems, the advantages of the stacked tensor-based hybrid single fuzzy neural networks indicate that the stacked way of networks designing can inherit the merits of the used algorithms, and the stacked structure of the three algorithms are complementary with each other.

4.2.3 Novel Corona Virus 2019 Dataset

COVID-19 affected cases in the Corona Virus 2019 dataset contains date information label. This dataset contains daily level information about the number of affected cases, the number of deaths and the rehabilitation of the new coronavirus in 2019. It is worth noting that this is a time series data, so any number of cases given a fixed date is cumulative. the data from national centers for disease control and prevention are collected in [github](#). The data is updated daily. Eight cities in Beijing, Shanghai, Chongqing Tianjin, Arizona, Washington, California and Illinois are used for testing,

and the time stamp ranges from 22 Jan, 2020 to 29 May, 2021.

There is no doubt that the extracted data of eight cities is composed of time series dataset, and is a small-scale dataset with three attributes. Four of the eight cities that are selected were from China, four were from the United States, the outbreaks in both regions were predicted. Although the data set itself is a small-scale one, because the attribute of the data set is only three, obviously, attribute is not sufficient. The performance of the proposed network is tested in the case that the feature attributes of the dataset are insufficient.

Table 9 Comparison results with TSFNN, TT2-RVFL, TT2-ELM, OP-ELM and TROP-ELM for Covid19_Beijing, Covid19_Shanghai, Covid19_Tianjin, Covid19_Chongqing, Covid19_Arizona, Covid19_Washington, Covid19_California, Covid19_Illinois (A brief introduction is listed in Table 1).

Dataset	Method	Training (RMSE)		Testing (RMSE)	
		Mean	Std	Mean	Std
Beijing	TSFNN	5.92e+01	3.74e+00	6.01e+01	7.07e+00
	TT2-RVFL	6.25e+01	2.82e+00	6.30e+01	6.67e+00
	TT2-ELM	6.25e+01	2.82e+00	6.28e+01	6.63e+00
	OP-ELM	7.59e+01	1.15e+01	7.64e+01	1.33e+01
	TROP-ELM	7.60e+01	1.14e+01	7.65e+01	1.33e+01
Shanghai	TSFNN	2.53e+01	1.88e+00	2.78e+01	3.42e+00
	TT2-RVFL	2.97e+01	2.00e+00	3.15e+01	2.97e+00
	TT2-ELM	2.96e+01	1.94e+00	3.08e+01	3.08e+00
	OP-ELM	6.01e+01	2.96e+01	6.10e+01	3.03e+01
	TROP-ELM	6.11e+01	3.00e+01	6.19e+01	3.05e+01
Tianjin	TSFNN	9.82e+00	5.86e-01	1.02e+01	1.26e+00
	TT2-RVFL	1.04e+01	5.15e-01	1.09e+01	1.30e+00
	TT2-ELM	1.04e+01	5.10e-01	1.07e+01	1.30e+00
	OP-ELM	1.52e+01	4.26e+00	1.57e+01	8.46e+00
	TROP-ELM	1.51e+01	4.26e+00	1.57e+01	1.38e+01
Chongqing	TSFNN	2.84e+01	2.74e+00	2.86e+01	6.40e+00
	TT2-RVFL	2.99e+01	2.67e+00	3.10e+01	6.20e+00
	TT2-ELM	2.98e+01	2.67e+00	3.06e+01	6.16e+00
	OP-ELM	3.85e+01	5.48e+00	4.15e+01	1.48e+01
	TROP-ELM	3.85e+01	5.54e+00	4.18e+01	1.95e+01
Arizona	TSFNN	1.22e-01	1.28e-02	1.16e-01	3.52e-02
	TT2-RVFL	1.24e-01	1.34e-02	1.22e-01	3.20e-02
	TT2-ELM	1.24e-01	1.34e-02	1.22e-01	3.19e-02
	OP-ELM	1.26e-01	1.40e-02	1.23e-01	3.20e-02
	TROP-ELM	1.26e-01	1.39e-02	1.23e-01	3.19e-02
Washington	TSFNN	1.16e-01	1.21e-02	1.12e-01	3.11e-02
	TT2-RVFL	1.17e-01	1.24e-02	1.19e-01	2.78e-02
	TT2-ELM	1.17e-01	1.24e-02	1.19e-01	2.77e-02
	OP-ELM	1.18e-01	1.29e-02	1.20e-01	2.80e-02
	TROP-ELM	1.18e-01	1.28e-02	1.20e-01	2.80e-02
California	TSFNN	6.41e-01	7.67e-02	6.14e-01	2.05e-01
	TT2-RVFL	6.50e-01	7.92e-02	6.44e-01	1.87e-01
	TT2-ELM	6.50e-01	7.92e-02	6.43e-01	1.86e-01
	OP-ELM	6.61e-01	8.31e-02	6.52e-01	1.88e-01
	TROP-ELM	6.61e-01	8.31e-02	6.52e-01	1.88e-01
Illinois	TSFNN	2.34e-01	2.36e-02	2.26e-01	6.20e-02
	TT2-RVFL	2.38e-01	2.44e-02	2.40e-01	5.62e-02
	TT2-ELM	2.38e-01	2.45e-02	2.40e-01	5.60e-02
	OP-ELM	2.40e-01	2.57e-02	2.42e-01	5.60e-02
	TROP-ELM	2.40e-01	2.56e-02	2.42e-01	5.60e-02

The comparison results are shown in Table 9. Fig. 8 is the results that is carried out based on samples and attributes from four datasets, Beijing, Shanghai, Tianjin and Chongqing. From the results, TSFNN has the best performance compared with the other four algorithms. By comparing the data results in Table 9, on the whole, the data results of the five comparison methods are similar. Moreover, from the data results, the results of Beijing, Shanghai, Tianjin and Chongqing are significantly worse than those of Arizona, Washington, California and Illinois.

Similarly, the same operation is performed on the four datasets of Arizona, Washington, California and Illinois, and the results are shown in Fig. 9.

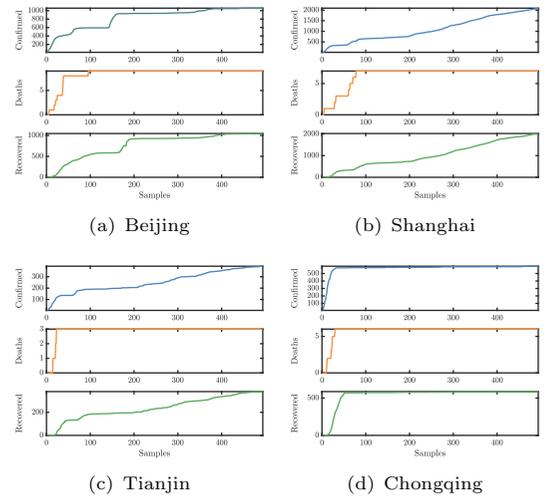


Fig. 8 The data for Novel Corona Virus 2019 Dataset in Beijing, Shanghai, Tianjin and Chongqing.

It can be regarded from Fig. 8 and Fig. 9 that, due to the characteristics of the COVID-19 dataset itself, the overall trend of the data is rising, reflecting the characteristics of its time series, and the growth rate of the curve reflects the situation of the new coronavirus. From the figures, these eight datasets are more suitable for regression problems, and the data is relatively stable. Therefore, there is little difference between TSFNN and four comparison methods in Table 9.

Compared with Fig. 8 and Fig. 9, the curve in Fig. 9 is smoother and the overall trend is more obvious. Although the curve in Fig. 8 shows an overall upward trend, the data soon go back to

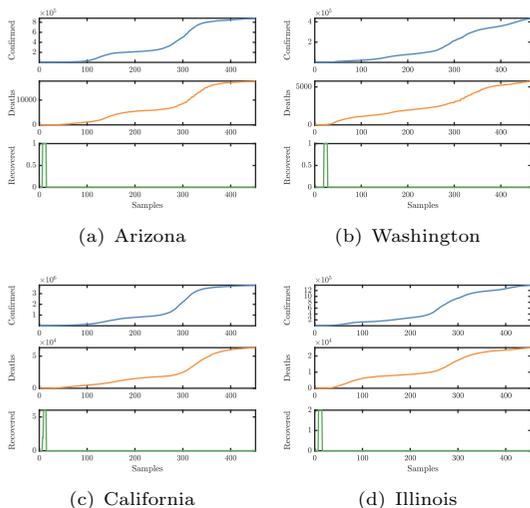


Fig. 9 The data for Novel Corona Virus 2019 Dataset in Arizona, Washington, California and Illinois.

the stable state. Therefore, the data in Fig. 9 is more suitable for forecasting regression problems than the data in Fig. 8, which is also the reason that in Table 9 the overall performance of the data sets corresponding to four cities in China on five algorithms is not as good as that of the data sets corresponding to four cities in the United States.

Through the above analysis, the data characteristics of the data set itself lead to data differences in table 9. Because of this difference, four data sets in China are comparable to four data sets in the United States. On the premise that the eight data sets are small-scale time series, the data characteristics of the four data sets in China are quite different, and the performance in the regression problem is poor, while the data in the United States are more obvious and more suitable for regression analysis.

In addition, four data sets such as Beijing can be regarded as data sets with more complex data structures than four data sets such as Arizona, and the features are more diverse, not limited to reflect the upward trend of data. Obviously, combined with the actual current situation. The four data sets related to the United States can better predict the future epidemic situation in the United States. As far as the data in the data set itself is concerned, the four data sets such as Arizona are more "pure" and suitable for regression problems. Therefore, the performance of the five methods on

these four data sets is better, while the performance of Beijing and other four and data sets is relatively poor.

According to the overall data of Table 9, TSFNN still performs best compared with the other four comparison methods. In four datasets of Beijing, Shanghai, Tianjin and Chongqing, although the overall data is relatively poor due to the problems with the data itself, the performance of TSFNN is still the best in these four datasets. TSFNN also performs best in Arizona, Washington, California and Illinois datasets which has the better data than 4 datasets for China. This shows in the excellent feature extraction and generalization ability of TSFNN. It is demonstrated that the scheme of tensor stacked neural network can integrate the advantages of its members and enhance the ability of data feature extraction.

4.2.4 Wind Power Generation Dataset

The wind power generation or wind energy is the use of wind power to provide mechanical power through wind turbines and generate electricity by rotating generators. Wind power is a popular and sustainable renewable energy, and its impact on the environment is much smaller than burning fossil fuels. Wind power generation dataset is collected from four German transmission system operators which are 50Hertz, Amprion, TenneT TSO and TransnetBW, the date ranges from 23 August 2019 to 22 September 2020. It contains non-normalized power generation data with 15 minutes per interval, a total of 96 sets of data with 96 points a day. The measurement unit of these data is terawatts hours. Obviously, there are 397 samples with 96 attributes in Wind power generation dataset.

The Wind Power Generation dataset is a small and medium-sized dataset. Through the data set, the prediction ability of the model can be verified in the case of sufficient data set attributes. Meanwhile, 50Hertz, Amprion, TransnetBW and TenneT TSO manage Germany's east, south, west and north regions, respectively; and the same is true of their geographical location. For wind power systems, power generation in different geographical locations has different characteristics, and this data difference caused by different geographical locations is also affected by the seasons. In Germany, the Officially spring is during March, April

and May, the Summer is taken from June through to August. The autumn is during the months of September, October and November and winter is from December to February. Obviously, geographical and seasonal factors are important for wind power systems. Therefore, among the 96 attributes of this data set, by virtue of the characteristics of the wind power generation system, the entire data set will have different characteristics in different months. Thus, these 96 attributes contain rich feature information, which can better assess the performance of the model.

Table 10 Comparison results with TSFNN, TT2-RVFL, TT2-ELM, OP-ELM and TROP-ELM for Hertz50, Amprion, TenneT TSO, TransnetBW (A brief introduction is listed in Table 1).

Dataset	Method	Training (RMSE)		Testing (RMSE)	
		Mean	Std	Mean	Std
Hertz50	TSFNN	2.95e+00	1.73e-01	4.15e+00	1.13e+00
	TT2-RVFL	8.16e+01	1.39e+01	9.39e+01	1.70e+01
	TT2-ELM	7.62e+01	1.36e+01	1.05e+02	2.04e+01
	OP-ELM	3.19e+00	1.83e-01	7.53e+00	7.79e-01
	TROP-ELM	3.19e+00	1.85e-01	7.51e+00	7.56e-01
Amprion	TSFNN	6.92e+00	3.24e+00	1.20e+01	3.10e+00
	TT2-RVFL	5.72e+01	1.02e+01	6.79e+01	1.21e+01
	TT2-ELM	5.34e+01	9.95e+00	7.34e+01	1.41e+01
	OP-ELM	8.49e+00	4.12e+00	2.94e+01	7.86e+00
	TROP-ELM	8.50e+00	4.13e+00	2.93e+01	7.96e+00
TenneT TSO	TSFNN	2.12e+00	1.12e-01	2.99e+00	2.56e-01
	TT2-RVFL	8.33e+01	1.54e+01	9.65e+01	1.92e+01
	TT2-ELM	7.69e+01	1.50e+01	1.07e+02	2.34e+01
	OP-ELM	2.23e+00	1.23e-01	5.61e+00	1.41e+00
	TROP-ELM	2.23e+00	1.24e-01	5.59e+00	1.38e+00
Transnet BW	TSFNN	1.58e+00	9.14e-02	4.47e+05	2.23e+07
	TT2-RVFL	1.88e+01	4.16e+00	2.17e+01	4.56e+00
	TT2-ELM	1.77e+01	3.90e+00	2.36e+01	5.10e+00
	OP-ELM	1.71e+00	9.61e-02	4.30e+04	1.63e+06
	TROP-ELM	1.71e+00	9.69e-02	8.52e+06	5.71e+08

The comparison results of TSFNN and TT2-RVFL, TT2-ELM, OP-ELM and TROP-ELM on 50Hertz, Amprion, TenneT TSO and TransnetBW datasets are shown in Table 10. Overall, results in the Table 10 show that the training error of TSFNN on four datasets is the best, while the test error is the best on the data sets of Hertz50, Amprion and TenneT TSO, and the worse on TransnetBW dataset. For Hertz50, Amprion and TenneT TSO datasets, TSFNN, OPELM and TROP-ELM perform better than TT2-RVFL and TT2-ELM, especially for training error and testing error, TT2-RVFL and TT2-ELM are much worse than the other three algorithms. Among the three methods of TSFNN, OP-ELM and TROP-ELM, TSFNN has better performance.

This result shows the excellent generalization ability and feature extraction ability of TSFNN. In the superposition of TSFNN, its members TT2-RVFL and TT2-ELM are also improved, as shown in Remark 4. The comparison results of TSFNN with TT2-RVFL and TT2-ELM prove that this method of composing kernel space with multiple activation functions can enhance the performance of the network. For TransnetBW dataset, the results in Table 10 show that TSFNN, OP-ELM and TROP-ELM perform better on the training set than TT2-RVFL and TT2-ELM.

On the testing set, TT2-RVFL and TT2-ELM perform better than the other three methods. Moreover, the performance of TSFNN, OP-ELM and TROP-ELM on testing set is far from that on training set. And the performance of TT2-RVFL and TT2-ELM is normal, and there is no difference significantly between the performance on the testing set and the training set. This shows that the generalization ability of type-2 fuzzy system is better than the ability of type-1 system, due to the TROP-ELM is a member of TSFNN network, the overall test performance of TSFNN is poor. Because the performance of TSFNN on the training set shows that it has good feature extraction ability on the data set, better results are obtained, and the results of OP-ELM and TROP-ELM are also this reason. At the same time, OP-ELM and TROP-ELM also benefit from the MRSR function. The MRSR function pairs are sorted, and then through the pruning links in OP-ELM and TROP-ELM, better results will be obtained for the data set with more attributes. This sort and pruning result on the training set may not be suitable for test set data, especially when the Wind Power Generation dataset has 96 attributes. This shows that the sorting and pruning of the network has a certain limit, and the effect on some data sets needs to be tested, which may produce large singular values, thus affecting the final results.

In the previous analysis, the stacked tensor-based hybrid single fuzzy neural networks system proposed in this paper has the advantages of concentrating its member networks. At the same time, it inherits the defects of its member network. The performance of TSFNN and other four comparison methods on TransnetBW dataset in Table 10 proves this phenomenon again. At the same time, it also shows that this ability to concentrate the advantages of its member networks also has the

upper limit. When a member network performs poorly, the advantages of other member networks are difficult to fully supplement this defect.

TSFNN can obtain the advantages of these member networks in different problems by combining different member networks, and then these advantages will be concentrated by tensor superposition. Meanwhile, the defects of some member networks are complemented by tensor overlay networks. However, when the performance of some member functions is poor to a certain extent, this complementary mechanism is difficult to completely eliminate the defects of the member function. When it exceeds the "threshold" of this complementary mechanism, the poor performance member network will dominate the performance of the whole network, resulting in poor performance of the whole network.

5 Conclusions

In this paper, a stacked tensor-based hybrid single fuzzy neural networks (TSFNN) was proposed, which is a neural network combination model. The TT2-RVFL, TT2-ELM and TROP-ELM are used to form the TSFNN network, and TT2-RVFL and TT2-ELM are optimized by using the kernel space method to enhance their performance. TSFNN overlays the hidden layer output network of its member network by tensor. In this process, the tensor-based superposition system inherits the advantages of the type-2 fuzzy sets and the fuzzy reasoning ability of the fuzzy system. Simultaneously, the good performance of the linear system that is generated by MRSR method and pruning method in TROP-ELM is also extracted by tensor structure.

Because tensor results can concentrate the advantages of member networks, TSFNN has good generalization ability and anti-noise ability. TSFNN will also inherit the defects of its member networks. In general, the ability of TSFNN to concentrate the advantages of member networks will make up for the shortcomings of some member networks, but this complementarity is limited, and when the data is too complex, there will still be underfitting, for example, TransnetBW dataset in Table 10.

By and large, the proposed TSFNN algorithm has excellent generalization ability, anti-noise ability and feature extraction ability. These capabilities are demonstrated and validated on 10 UCI standard datasets and 4 real world datasets. The TSFNN algorithm supplements the tensor-based model optimization and model combination methods, indicating that the tensor structure superposition neural network is a feasible neural network combination method. For the purpose of obtain a fast model of the data set, unfolding of tensor is used, and then the regression results are obtained by matrix regression. Tensor regression and tensor equation can also be used in this direction, which is a future optimization direction.

Acknowledgment

This document is the results of the research project funded by the National Natural Science Foundation of China under number 12161065, 61603126 and 61966026, Natural Science Foundation of Inner Mongolia (2020MS06016, 2019MS01005), and the work is also supported by the Research Foundation for Advanced Talents Research Foundation of Inner Mongolia University (21700-5185130).

Declaration of conflicting interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

- Akusok A, Veganzones D, Miche Y, et al (2015) MD-ELM: Originally mislabeled samples detection using OP-ELM model. *Neurocomputing* 159:242–250. <https://doi.org/10.1016/j.neucom.2015.01.055>
- Alencar AS, Rocha Neto AR, Gomes JPP (2016) A new pruning method for extreme learning machines via genetic algorithms. *Applied Soft Computing* 44:101–107. <https://doi.org/10.1016/j.asoc.2016.03.019>
- Baranyi P (2016) TP-Model Transformation-Based-Control Design Frameworks. Springer

- International Publishing, Switzerland, <https://doi.org/10.1007/978-3-319-19605-3>
- Baranyi P, Yam Y, Várlaki P (2014) Tensor Product Model Transformation in Polytopic Model-Based Control. CRC Press-Taylor and Frances Group, <https://doi.org/10.1201/9781315218045>
- Behera R, Mishra D (2017) Further results on generalized inverses of tensors via the einstein product. *Linear and Multilinear Algebra* 65(8):1662–1682. <https://doi.org/10.1080/03081087.2016.1253662>
- Bugnon L, Calvo RA, Milone D (2020) Dimensional affect recognition from HRV: An approach based on supervised SOM and ELM. *IEEE Transactions on Affective Computing* 11(1):32–44. <https://doi.org/10.1109/TAFFC.2017.2763943>
- Christou V, Tsiouras MG, Giannakeas N, et al (2019) Hybrid extreme learning machine approach for heterogeneous neural networks. *Neurocomputing* 361:137–150. <https://doi.org/10.1016/j.neucom.2019.04.092>
- Efron B, Hastie T, Johnstone I, et al (2004) Least angle regression. *The Annals of Statistics* 32:407–499. <https://doi.org/10.1214/009053604000000067>
- Grigorievskiy A, Miche Y, Ventelä A, et al (2014) Long-term time series prediction using OP-ELM. *Neural Networks* 51:50–56. <https://doi.org/10.1016/j.neunet.2013.12.002>
- Grigorievskiy A, Miche Y, Käpylä M, et al (2016) Singular value decomposition update and its application to (Inc)-OP-ELM. *Neurocomputing* 174:99–108. <https://doi.org/10.1016/j.neucom.2015.03.107>
- Hashmi AS, Ahmad T (2019) GP-ELM-RNN: Garson-pruned extreme learning machine based replicator neural network for anomaly detection. *Journal of King Saud University - Computer and Information Sciences* <https://doi.org/10.1016/j.jksuci.2019.09.007>
- Huang SRN, Chen MH (2016) Constructing optimized interval type-2 TSK neuro-fuzzy systems with noise reduction property by quantum inspired BFA. *Neurocomputing* 173:1839–1850. <https://doi.org/10.1016/j.neucom.2015.09.060>
- Huang SRN, Zhao GL, Chen MH (2019) Tensor extreme learning design via generalized Moore–Penrose inverse and triangular type-2 fuzzy sets. *Neural Computing and Applications* 31(9):5641–5651. <https://doi.org/10.1007/s00521-018-3385-5>
- Jamil M, Sharma SK, Singh R (2015) Fault detection and classification in electrical power transmission system using artificial neural network. *SpringerPlus* 4(1):334. <https://doi.org/10.1186/s40064-015-1080-x>
- Lai XP, Cao JW, Huang XF, et al (2020) A maximally split and relaxed ADMM for regularized extreme learning machines. *IEEE Transactions on Neural Networks and Learning Systems* 31(6):1899–1913. <https://doi.org/10.1109/TNNLS.2019.2927385>
- Miche Y, Sorjamaa A, Bas P, et al (2010) OP-ELM: Optimally pruned extreme learning machine. *IEEE Transactions on Neural Networks* 21(1):158–162. <https://doi.org/10.1109/TNN.2009.2036259>
- Miche Y, van Heeswijk M, Bas P, et al (2011a) TROP-ELM: A double-regularized ELM using LARS and Tikhonov regularization. *Neurocomputing* 74(16):2413–2421. <https://doi.org/10.1016/j.neucom.2010.12.042>
- Miche Y, van Heeswijk M, Bas P, et al (2011b) TROP-ELM: A double-regularized ELM using LARS and Tikhonov regularization. *Neurocomputing* 74(16):2413–2421. <https://doi.org/10.1016/j.neucom.2010.12.042>
- Ming YW, Zhu E, Wang M, et al (2018) DMP-ELMs: Data and model parallel extreme learning machines for large-scale learning tasks. *Neurocomputing* 320:85–97. <https://doi.org/10.1016/j.neucom.2018.08.062>
- Ng AY (2004) Feature selection, L_1 vs. L_2 regularization, and rotational invariance. <https://doi.org/10.1145/1015330.1015435>

- Peng Y, Li QX, Kong WZ, et al (2020) A joint optimization framework to semi-supervised RVFL and ELM networks for efficient data classification. *Applied Soft Computing* 97:106,756. <https://doi.org/10.1016/j.asoc.2020.106756>
- Ragusa E, Gastaldo P, Zunino R, et al (2020) Balancing computational complexity and generalization ability: A novel design for ELM. *Neurocomputing* 401:405–417. <https://doi.org/10.1016/j.neucom.2020.03.046>
- Runkler TA, Coupland S, John R, et al (2018) Interval type-2 defuzzification using uncertainty weights. In: *Frontiers in Computational Intelligence*. Springer, p 47–59, https://doi.org/10.1007/978-3-319-67789-7_5_4
- Sekh AA, Dogra DP, Kar S, et al (2020) ELM-HTM guided bio-inspired unsupervised learning for anomalous trajectory classification. *Cognitive Systems Research* 63:30–41. <https://doi.org/10.1016/j.cogsys.2020.04.003>
- Shi J, Xue ZY, Dai YK, et al (2019) Cascaded multi-column RVFL+ classifier for single-modal neuroimaging-based diagnosis of parkinson's disease. *IEEE Transactions on Biomedical Engineering* 66(8):2362–2371. <https://doi.org/10.1109/TBME.2018.2889398>
- Similä T, Tikka J (2005) Multiresponse sparse regression with application to multidimensional scaling. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 97–102, https://doi.org/10.1007/11550907_163
- Wang PP, Zheng XQ, Ku JH, et al (2020) Multiple-Instance learning approach via bayesian extreme learning machine. *IEEE Access* 8:62,458–62,470. <https://doi.org/10.1109/ACCESS.2020.2984271>
- Wei CY, Li CS, Feng C, et al (2020) A T-S fuzzy model identification approach based on evolving MIT2-FCRM and WOS-ELM algorithm. *Engineering Applications of Artificial Intelligence* 92:103–653. <https://doi.org/10.1016/j.engappai.2020.103653>
- Wong SY, Yap KS, Li X (2020) A new probabilistic output constrained optimization extreme learning machine. *IEEE Access* 8:28,934–28,946. <https://doi.org/10.1109/ACCESS.2020.2971012>
- Xu XZ, Shan D, Li S, et al (2019) Multi-label learning method based on ML-RBF and laplacian ELM. *Neurocomputing* 331:213–219. <https://doi.org/10.1016/j.neucom.2018.11.018>
- Xu XZ, Li S, Liang TM, et al (2020) Sample selection-based hierarchical extreme learning machine. *Neurocomputing* 377:95–102. <https://doi.org/10.1016/j.neucom.2019.10.013>
- Xue ZY, Shi J, Dai YK, et al (2018) Transcranial sonography based diagnosis of parkinson's disease via cascaded kernel RVFL+. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp 574–577, <https://doi.org/10.1109/EMBC.2018.8512384>
- Yang J, Cao JW, Wang TL, et al (2020a) Regularized correntropy criterion based semi-supervised ELM. *Neural Networks* 122:117–129. <https://doi.org/10.1016/j.neunet.2019.09.030>
- Yang J, Cao JW, Xue AK (2020b) Robust maximum mixture correntropy criterion-based semi-supervised ELM with variable center. *IEEE Transactions on Circuits and Systems II: Express Briefs* 67(12):3572–3576. <https://doi.org/10.1109/TCSII.2020.2995419>
- Yang Z, Al-Dahidi S, Baraldi P, et al (2020c) A novel concept drift detection method for incremental learning in nonstationary environments. *IEEE Transactions on Neural Networks and Learning Systems* 31(1):309–320. <https://doi.org/10.1109/TNNLS.2019.2900956>
- Yaw O, Yap KS, Wong SY, et al (2020) Enhancement of neural network based multi agent system for classification and regression in energy system. *IEEE Access* 8:163,026–163,043. <https://doi.org/10.1109/ACCESS.2020.3012983>
- Yildirim H, Revan Özkale M (2019) The performance of ELM based ridge regression via the regularization parameters. *Expert Systems with Applications* 134:225–233. <https://doi.org/10.1016/j.eswa.2019.05.039>

- Yu Q, Miche Y, Eirola E, et al (2013) Regularized extreme learning machine for regression with missing data. *Neurocomputing* 102:45–51. <https://doi.org/10.1016/j.neucom.2012.02.040>
- Yu Y, Li Z, Liu XD, et al (2019) A nested tensor product model transformation. *IEEE Transactions on Fuzzy Systems* 27(1):1–15. <https://doi.org/10.1109/TFUZZ.2018.2851575>
- Zeng Y, Li Y, Chen J, et al (2020a) ELM embedded discriminative dictionary learning for image classification. *Neural Networks* 123:331–342. <https://doi.org/10.1016/j.neunet.2019.11.015>
- Zeng YJ, Chen JC, Li Y, et al (2020b) Clustering via adaptive and locality-constrained graph learning and unsupervised ELM. *Neurocomputing* 401:224–235. <https://doi.org/10.1016/j.neucom.2020.03.045>
- Zhang L, P.N.Suganthan (2016) A comprehensive evaluation of random vector functional link networks. *Information Sciences* 367–368:1094–1105. <https://doi.org/10.1016/j.ins.2015.09.025>
- Zhang L, Suganthan PN (2017) Benchmarking ensemble classifiers with novel co-trained kernel ridge regression and random vector functional link ensembles [research frontier]. *IEEE Computational Intelligence Magazine* 12(4):61–72. <https://doi.org/10.1109/MCI.2017.2742867>
- Zhang PB (2020) A new learning paradigm for random vector functional-link network: RVFL+. *Neural Networks* 122:94–105. <https://doi.org/10.1016/j.neunet.2019.09.039>
- Zhao GL, Wu W (2019) Tensor-based type-2 random vector functional link network. In: 2019 Chinese Control And Decision Conference (CCDC), pp 3278–3285, <https://doi.org/10.1109/CCDC.2019.8833161>
- Zheng F, Liu ZY, Chen YJ, et al (2020) A novel adaptive multi-view non-negative graph semi-supervised ELM. *IEEE Access* 8:116,350–116,362. <https://doi.org/10.1109/ACCESS.2020.2998428>
- Zhu HF, Wu YP (2020) Inverse-free incremental learning algorithms with reduced complexity for regularized extreme learning machine. *IEEE Access* 8:177,318–177,328. <https://doi.org/10.1109/ACCESS.2020.3022174>