IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 31, 2018, accepted February 28, 2018, date of publication March 12, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2815503

Sparse Recursive Least Mean p-Power Extreme Learning Machine for Regression

JING YANG[1,2], (Member, IEEE), YI XU[1], (Student Member, IEEE), HAI-JUN RONG[3], (Member, IEEE), SHAOYI DU[4], AND BADONG CHEN[4], (Senior Member, IEEE)

[1]Institute of Control Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[2]Computational NeuroEngineering Laboratory, Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
[3]State Key Laboratory for Strength and Vibration of Mechanical Structures, School of Aerospace, Xi'an Jiaotong University, Xi'an 710049, China
[4]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Hai-jun Rong (hjrong@xjtu.edu.cn)

This work was supported by the National Natural Science Foundation-Shenzhen Joint Research Program under Grant U1613219, in part by the National Science Council of ShaanXi Province under Grant 2014JM8337, and in part by the Fundamental Research Funds for the Central Universities.

ABSTRACT Real industrial processes usually are equipped with onboard control or diagnostic systems and limit to store a complicated model. Also, measurement samples from real processes are contaminated with noises of different statistical characteristics and are produced by one-by-one way. In this case, learning algorithms with better learning performance and compact model for systems with noises of various statistics are necessary. This paper proposes a new online extreme learning machine (ELM) algorithm, namely, sparse recursive least mean p-power ELM (SRLMP-ELM). In SRLMP-ELM, a novel cost function, i.e., the sparse least mean p-power (SLMP) error criterion, provides a mechanism to update the output weights sequentially and automatically tune some parameters of the output weights to zeros. The SLMP error criterion aims to minimize the combination of the mean p-power of the errors and a sparsity penalty constraint of the output weights. For real industrial system requirements, the proposed on-line learning algorithm is able to provide more higher accuracy, compact model, and better generalization ability than ELM and online sequential ELM, whereas the non-Gaussian noises impact the processes, especially impulsive noises. Simulations are reported to demonstrate the performance and effectiveness of the proposed methods.

INDEX TERMS Sparse recursive least mean p-power, extreme learning machine, online sequential learning, non-gaussian noises, alpha-stable noises.

I. INTRODUCTION

Online system identification is a significant problem that people often need to face in the fields of engineering technologies, natural sciences or social sciences [1]–[5]. In many practical applications, e.g., forecasting of renewable energy generation [1], stock forecast [2], and weather forecast [3], the datum samples are often stained with the large stochastic noises of different statistic characteristics, such as Gaussian, impulsive, or mixed distribution. Furthermore, the amount of expensive memory is always less and limit to store a complicated model in many practical systems. The onboard control or diagnostic systems in industries are the typical cases. Therefore, on-line sequential learning algorithms which are highly efficient, better learning performance and compact structure for systems with various noise statistics are keenly sought for both researchers and enterprise groups.

Neural networks have been intensively studied as the basis for solving this problem [6]–[8]. Extreme Learning Machine (ELM), a new fast neural learning algorithm, is proposed to train a single layer feedforward network (SLFN) with hidden neuron weights randomly initialized and fixed. It's obviously different from other traditional training algorithms, hidden neuron weights need to be tuned, such us back-propagation (BP) algorithm and its various improved algorithms [9], [10]. In contrast to the full parameter determination algorithms, ELM has fast learning speed [11], universal approximation capability [11], [12] and provides a unified learning paradigm for regression and classification [13]. For online identification problem, the datum samples are often arriving in the order of time, Liang *et al.* [14] propose the online sequential ELM (OS-ELM), which can learn the data one-by-one or chunk-by-chunk with fixed or varying chunk sizes. And many different improvements have been proposed and successfully applied in some applications [15]–[22].

In ELM, OS-ELM and many variants of them, mean squared error (MSE) criterion is exclusively adopted to construct their cost functions. Since the MSE criterion only takes into account the second-order statistics, it makes sense in the signal processing with Gaussian assumption. Consequently, ELM suffers from two drawbacks: 1) MSE minimization learning can easily suffer from overfitting. The problem will be serious while the characteristic of the learned dataset can't be represented by the training data [23], [24]. 2) ELM may perform poorly in the data under nonlinear and non-Gaussian situations, as it only captures the second-order statistics in the samples [25].

To overcome the overfitting drawback, Deng *et al.* [17] proposed a l_2-type regularized ELM based on structural risk minimization principle and weighted least square. The generalization performance of the proposed algorithm was improved significantly in most cases without increasing training time. A kernel ELM with higher generalization was proposed in [13] where a unified framework is provided to simplify and unify different learning methods, including LS-SVM, PSVM, feedforward neural networks and etc. However, a more complicated networks and longer testing time are required while the sparsity of the network is lost.

The second drawback is obvious in some practical applications. In many real-world circumstances, e.g., the energy spectrums of brain magnetic resonance images [26], multiple access interference in communication systems (broadband power-line communications [27], [28], wireless sensor networks [29], [30]), and other scenarios [31]–[35], the datum encountered have more impulsive characteristic than that predicted by a Gaussian distribution, even the combination of the impulsive and Gaussian distribution. These impulsive distribution problems, i.e., the non-Gaussian heavy-tailed distribution problems, cannot be satisfactorily solved by the MSE criterion. On the other hand, in many real industrial production processes, the measurement noises of the instrument have another kind of statistical characteristics, named non-Gaussian light-tailed distribution, of which bounded uniform distribution is a particular case. At this time, the best performance is also difficult to be achieved by the MSE criterion. To solve this problem, a new online ELM algorithm, namely recursive least mean p-power ELM (RLMP-ELM) [25] is proposed in our previous work. The least mean p-power (LMP) error criterion for cost function provides a mechanism to tune the output weights sequentially. The aim of the LMP error criterion is to minimize the mean p-power of the error. Generally, the mean square error criterion is used in the ELM. Under the non-Gaussian noises situation, the novel learning algorithm is able to provide on-line predictions of variables with different statistics and obtain better performance than ELM and OS-ELM with the same number of hidden neurons. However, the accuracy of the proposed model is obviously influenced by the hidden units' number, just like ELM model [24]. The classic ELM usually requires more hidden neurons than that of conventional neural networks to achieve matched performance, since ELM generates hidden layers randomly. A long running time is resulted in the testing phase of ELM for its large network size. This is a hinder for ELM to efficiently develop in some test time sensitive scenarios. Thus, the topic on improving the compactness of ELM while maintaining high model accuracy has attracted great interest [36].

To find the optimal number of hidden neurons, the ELM model is trained in a dynamic way that the number of hidden neurons will be changed during the training process [36]. In the way of neurons growing, only appropriate neurons are added into the network, such as incremental ELM (I-ELM) [12], [37]–[39] and bidirectional ELM (B-ELM) [40]. Thus, the more compact networks can be obtained. In pruning ELM (P-ELM) [41]–[43], the traditional ELM is used to construct an initial network, then some hidden neurons will be removed since they contribute less to the training performance. A least angle squares regression (l_1-type regularization) to minimize training error is used to rank neurons [41] and later improved with a cascade of l_1- and l_2-types regularization [42] by the same authors. Deferent from I-ELMs with frozen existing hidden nodes, the adaptive growth ELM (AG-ELM) [44] can automatically increase, decrease or stay the same hidden neurons at any step of the training process. A sparse Bayesian approach [24] is presented to learn a compact ELM model through automatically tuning most of the output weights to zeros with an assumed prior distribution. Bai *et al.* [45] proposed a sparse ELM (S-ELM) by involving in the quadratic programming (QP) problem and analytically solves the problems, which greatly reduces the storage space and testing time. These two sparse models are both proposed for classification problems.

In the fields of the systems identification and the adaptive filtering, a sparsity constraint approximating l_0-norm is applied as a penalty term to the widely used algorithms, such as least mean square (LMS) or recursive least square (RLS) algorithms to achieve sparsity models. For systems identification, Chen *et al.* [46] combines a l_1-norm penalty on the coefficients into the quadratic LMS cost function, which generates a zero attractor in the LMS iteration and takes advantage of the sparsity of the underlying signal to improve the MSE performance of the LMS algorithm. A weighted l_1-norm sparsity constraint is used in the RLS algorithm to estimate a sparse tap-weight vector in the adaptive filtering setting. The proposed algorithm improves the MSE performance of the conventional RLS algorithm and decreases the computational requirements of the RLS [47]. For the system identification setting, Eksioglu develops a new sparse RLS algorithm using a general convex function of the system estimate as a regularizing term [48]. The sparsity penalty constraint (SPC) used as regularization term can improve the generation ability of the learning system by removing redundant data and keeping a minimal set of centers that covers the area where inputs will likely appear, i.e., to prevent overfitting [49]. On the other hand, a sparse model reduces the complexity in terms of computation and memory [47]–[49].

Inspired by these literatures and sparse ELM algorithms above, we present a novel sparse ELM algorithm by incorporating a sparsity penalty term into the least mean p-power error criterion as the cost function, while the more initial hidden neurons are selected first and the parameters of hidden layer are randomly generated as in the conventional ELM. It is deferent from our previous work RLMP-ELM algorithm, which constructs its cost function only with the least mean p-power error criterion. For simplicity, the new method is named as the sparse recursive least mean p-power ELM (SRLMP-ELM) algorithm.

The SRLMP-ELM finds sparse representatives for the output weights by recursively learning to minimize the cost function and automatically tunes some parameters of the output weights to zeros during learning phase for the effect of the sparsity penalty term. The proposed algorithm gains sparsity by pruning the corresponding hidden neurons which parameters of the output weight are tuned to zeros. Hence, the SRLMP-ELM is proposed to improve the robustness and accuracy of ELM algorithm that produces a poor and unreliable solution for on-line identification problems when the output data are stained with various noise disturbances. Simultaneously, the novel algorithm improves the generation ability of the classic ELM and reduces the model complexity and storage space of the system. Compared with our previous work, the RLMP-ELM algorithm, our new method can achieve more compact models and shorter testing time without sacrificing the accuracy of the systems for the same processes. Simulation results show that this proposed method with different p and ρ values has more accurate solution and more compact network structure compared with the existing ELM and OS-ELM algorithms, while similar accuracy and more compact network compared with the RLMP-ELM algorithm.

The remainder of this paper is as follows. We provide a brief review of the ELM and sparse LMP error criterion in Section2. In Section3, the proposed SRLMP-ELM algorithm is described. The performance of this proposed algorithm is subsequently verified on different artificial dataset and real-world datasets in Section4. Section5 summarizes the conclusions from this study.

II. PRELIMINARY

A. EXTREME LEARNING MACHINE

In the ELM, hidden lay is generated randomly and need not be adjusted, and only the output weight vector is tuned based on application dependent training data. The training speed is much faster than that of the traditional SLFNs because much fewer parameters need to be adjusted here [11]. Consider N arbitrary distinct samples $(\mathbf{x}_k, t_k)$, where $\mathbf{x}_k \in \mathbb{R}^n$ is the kth input vector and $t_k \in \mathbb{R}$ is the associated desired value. ELM could have single or multiple output nodes. For simplicity, we consider the case with single output node and the output of an ELM with $\tilde{N}$ hidden nodes equals as,

$$\begin{aligned} f(\mathbf{x}_k) &= \sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{x}_k; a_i, b_i) \\ &= \boldsymbol{\beta}^T g_k, \quad k = 1, \cdots, N. \end{aligned} \tag{1}$$

where $g(\cdot)$ is the activation function and could be additive nodes and RBF nodes. a_i and b_i are the learning parameters of hidden nodes, $\boldsymbol{\beta} \in \mathbb{R}^{\tilde{N}}$ and $g_k \in \mathbb{R}^{\tilde{N}}$ are the output weight vector and the hidden nodes' output vector with respect to the input $\mathbf{x}_k$. Just mentioned above, the parameters of hidden nodes a_i and b_i in ELM are randomly set and are not subject to any optimization.

The output weight vector $\boldsymbol{\beta}$ is trained using the least mean square (LMS) algorithm based on the minimization of the following mean square error (MSE) cost function,

$$\begin{aligned} J_{MSE} &= \frac{1}{N}\sum_{k=1}^{N} e_k^2 = \frac{1}{N}\|\mathbf{H}\boldsymbol{\beta} - T\| \\ &= E(e_k^2) \end{aligned} \tag{2}$$

where E denotes the expectation operator, $e_k = t_k - \boldsymbol{\beta}^T g_k$ is the estimation error. $\mathbf{H}$ denotes the hidden layer output matrix, where $g_{ki} \in \mathbf{H}(k = 1, ..., N; i = 1, ..., \tilde{N})$ is the activation value of the ith hidden neuron for the kth input vector $g_{ki} = g(\mathbf{x}_k; a_i, b_i)$. $T = [t_1, \cdots, t_k, \cdots, t_N]^T$ is the desired output vector. A pseudoinverse operation yields the unique l_2 solution of (2), that is $\boldsymbol{\beta} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T T$.

Now, an alternative optimality criterion, the sparse least mean p-power (SLMP), has been applied in our study to improve the robust performance in realistic scenarios with more compact model than those of the ELM.

B. SPARSE LEAST MEAN p-POWER

Let $e_k = t_k - f(\mathbf{x}_k)$ be the estimation error. Then the sparse least mean p-power (SLMP) cost is defined as ($p \in \mathbb{R}^+$),

$$J_{SLMP} = \frac{1}{N}\sum_{k=1}^{N} \lambda^{N-k}|e_k|^p + \rho S_N \tag{3}$$

The first term, $\frac{1}{N}\sum_{k=1}^{N} \lambda^{N-k}|e_k|^p$, is the least mean p-power (LMP) error criterion and the MSE criterion (2) is a special case with $p = 2$. The LMP criterion is computationally simple, and has been proven successful in various applications [22], [25], [29], [50]–[53]. It has been pointed out that the LMP has some useful properties such that it may produce a better solution if the performance function has different optimum solutions for various p, instead of the MSE. While the datum is non-Gaussian light-tailed distribution, steepest descent algorithm based on LMP error criterion with $p > 2$ (especially when $p = 4$) may have better convergence performance (i.e., achieve either faster convergence speed or lower misadjustment). The learning algorithm based on LMP error criterion with $p < 2$ (e.g. when $p = 1$) is robust to non-Gaussian heavy-tailed distribution noises. λ is commonly referred to as forgetting factor to deemphasize data from the remote past. It is a non-negative constant and usually set in the range (0, 1).

S_N denotes a sparsity penalty constraint term of the output weights, which can improve the robustness and generalization of the algorithm, besides the compactness of the model. ρ is a regularization parameter that balance the tradeoff between LMP and sparsity penalty. The different value for ρ will lead to different performance of the algorithm and the details are shown in the simulation. Several sparsity penalty terms are introduced to the algorithms (e.g. l_0-norm or l_1-norm). The l_0-norm is an optimal SPC. However, the optimization of the l_0-norm is an NP-hard problem. For this reason, various approximations of l_0-norm are usually utilized as the SPC in the literatures. The l_1-norm is a popular one of such approximations. In our study, l_1-norm is selected as the regularization term.

III. SPARSE RECURSIVE LEAST MEAN p-POWER EXTREME LEARNING MACHINE

An empirical sparse least mean p-power related online extreme learning machine (SRLMP-ELM) is developed in this section. The SRLMP-ELM is based on the primitive ELM algorithm which is randomly setting the parameters of a SLFN. However, a sequential updating procedure based on the sparse recursive least mean p-power error criterion replaces the ELM learning operation. In this section, we will derive the algorithm to update the weight vector of the ELM under the SLMP error criterion (3). In the following parts, we will present the detail process of the SRLMP-ELM algorithm.

A. SPARSE RECURSIVE LEAST MEAN p-POWER (SRLMP)

According to the description of ELM in preliminary, the output of an ELM can be seen as a general linear system $\boldsymbol{\beta}^T g = t$, where $\boldsymbol{\beta} \in \mathbb{R}^{\tilde{N}}$, $g \in \mathbb{R}^{\tilde{N}}$ and $t \in \mathbb{R}$. For this general linear system, the SRLMP algorithm is the extension of the recursive least square (RLS) algorithm with cost function (2) [54]–[56]. The cost function of SRLMP algorithm is defined as regularizing LMP error criterion by a sparsity penalty term,

$$J_{SLMP} = \frac{1}{N}\sum_{k=1}^{N}\lambda^{N-k}|e_k|^p + \rho S_N \tag{4}$$

where e_k is the error in kth sample time and $e_k = t_k - \boldsymbol{\beta}_N^T g_k$. S_N denotes a sparsity penalty constraint (SPC) and l_1-norm is selected as the SPC here,

$$S_N = \|\boldsymbol{\beta}_N\|_1 \tag{5}$$

Substituting (5) into (4) yields,

$$J_{SLMP} = \frac{1}{N}\sum_{k=1}^{N}\lambda^{N-k}|e_k|^p + \rho\|\boldsymbol{\beta}_N\|_1 \tag{6}$$

In theory, it has been proved by some results of convex function in literature [57] that the every minimum of LMP error criterion $\frac{1}{N}\sum_{k=1}^{N}|e_k|^p$ is a global minimum while $p \geq 1$. Thus the performance function J_{SLMP} has a global minimum while S_N is a convex function. Since l_1-norm is a convex function, J_{SLMP} has a global minimum. The optimal solution $\boldsymbol{\beta}_N$ for minimizing J_{SLMP} can be obtained by differentiating (6) with respect to $\boldsymbol{\beta}_N$ and setting the derivatives to zero. The derivatives are,

$$\begin{aligned}\frac{\partial J_{SLMP}}{\partial \boldsymbol{\beta}_N} &= \frac{1}{N}\sum_{k=1}^{N}\lambda^{N-k}\frac{\partial|e_k|^p}{\partial \boldsymbol{\beta}_N} + \rho\frac{\partial\|\boldsymbol{\beta}_N\|_1}{\partial \boldsymbol{\beta}_N}\\ &= \frac{1}{N}\sum_{k=1}^{N}\lambda^{N-k}\frac{\partial|e_k|^p}{\partial e_k}\cdot\frac{\partial e_k}{\partial \boldsymbol{\beta}_N} + \rho\frac{\partial\|\boldsymbol{\beta}_N\|_1}{\partial \boldsymbol{\beta}_N}\end{aligned} \tag{7}$$

Also because

$$|e_k|^p = \begin{cases} e_k^p & p: even\\ sgn(e_k)e_k^p & p: odd\end{cases} \tag{8}$$

the following expression is obtained,

$$\begin{aligned}\frac{\partial|e_k|^p}{\partial e_k} &= \begin{cases} pe_k^{p-1} & p: even\\ psgn(e_k)e_k^{p-1} & p: odd\end{cases}\\ &= p|e_k|^{p-2}e_k\end{aligned} \tag{9}$$

where $sgn(e_k) = e_k/|e_k|$. Thus (7) can be written as,

$$\frac{\partial J_{SLMP}}{\partial \boldsymbol{\beta}_N} = \frac{1}{N}\sum_{k=1}^{N}\lambda^{N-k}p|e_k|^{p-2}e_k\frac{\partial e_k}{\partial \boldsymbol{\beta}_N} + \rho\frac{\partial\|\boldsymbol{\beta}_N\|_1}{\partial \boldsymbol{\beta}_N} \tag{10}$$

Substituting $e_k = t_k - \boldsymbol{\beta}_N^T g_k$ into (10) yields,

$$\frac{\partial J_{SLMP}}{\partial \boldsymbol{\beta}_N} = \frac{1}{N}\sum_{k=1}^{N}\lambda^{N-k}p|e_k|^{p-2}(t_k - \boldsymbol{\beta}_N^T g_k)g_k + \rho\frac{\partial\|\boldsymbol{\beta}_N\|_1}{\partial \boldsymbol{\beta}_N} \tag{11}$$

Setting $\frac{\partial J_{SLMP}}{\partial \boldsymbol{\beta}_N} = 0$ and (11) can be further written as,

$$\sum_{k=1}^{N}\lambda^{N-k}|e_k|^{p-2}g_k g_k^T\boldsymbol{\beta}_N = \sum_{k=1}^{N}\lambda^{N-k}|e_k|^{p-2}t_k g_k + \rho\frac{\partial\|\boldsymbol{\beta}_N\|_1}{\partial \boldsymbol{\beta}_N} \tag{12}$$

Letting

$$\Psi_N = \sum_{k=1}^{N}\lambda^{N-k}|e_k|^{p-2}g_k g_k^T \tag{13}$$

and

$$\Phi_N = \sum_{k=1}^{N}\lambda^{N-k}|e_k|^{p-2}t_k g_k \tag{14}$$

Here, we set $G_N = [g_1, \ldots, g_N]$, then Ψ_N and Φ_N can be rewritten as,

$$\Psi_N = G_N\begin{bmatrix}\lambda^{N-1}|e_1|^{p-2} & \ldots & 0\\ & \vdots & \\ 0 & \ldots & |e_N|^{p-2}\end{bmatrix}G_N^T \tag{15}$$

and

$$\Phi_N = G_N \begin{bmatrix} \lambda^{N-1}|e_1|^{p-2} & \ldots & 0 \\ & \vdots & \\ 0 & \ldots & |e_N|^{p-2} \end{bmatrix} T \tag{16}$$

Here, Ψ_N and Φ_N can be called as the sparse p-Power correlation matrix of G_N and the sparse p-Power cross-correlation vector of G_N and T, respectively. They serve similar purpose as the conventional correlation matrix of G_N and the cross-correlation vector of G_N and T. Furthermore, we set

$$\Upsilon_N = \Phi_N + \rho \frac{\partial \|\boldsymbol{\beta}_N\|_1}{\partial \boldsymbol{\beta}_N} \tag{17}$$

Considering Eqs(12)-(17), the following relation can be obtained,

$$\Psi_N \boldsymbol{\beta}_N = \Upsilon_N \tag{18}$$

The optimal solution $\boldsymbol{\beta}_N$ is,

$$\boldsymbol{\beta}_N = \Psi_N^{-1}\Upsilon_N \tag{19}$$

Equation (13), (14) and (17) can be further written as,

$$\begin{aligned}
\Psi_N &= \lambda \sum_{k=1}^{N-1} \lambda^{N-1-k}|e_k|^{p-2} g_k g_k^T + |e_N|^{p-2} g_N g_N^T \\
&= \lambda \Psi_{N-1} + |e_N|^{p-2} g_N g_N^T \qquad (20) \\
\Phi_N &= \lambda \sum_{k=1}^{N-1} \lambda^{N-1-k}|e_k|^{p-2} t_k g_k + |e_N|^{p-2} t_N g_N \\
&= \lambda \Phi_{N-1} + |e_N|^{p-2} t_N g_N \qquad (21) \\
\Upsilon_N &= \lambda \Phi_{N-1} + |e_N|^{p-2} t_N g_N + \rho \frac{\partial \|\boldsymbol{\beta}_N\|_1}{\partial \boldsymbol{\beta}_N} \\
&= \lambda \Phi_{N-1} + \lambda\rho \frac{\|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N} + |e_N|^{p-2} t_N g_N + \rho \frac{\partial \|\boldsymbol{\beta}_N\|_1}{\partial \boldsymbol{\beta}_N} \\
&\quad - \lambda\rho \frac{\|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N} \\
&= \lambda \Upsilon_{N-1} + |e_N|^{p-2} t_N g_N + \rho \frac{\partial \|\boldsymbol{\beta}_N\|_1}{\partial \boldsymbol{\beta}_N} - \lambda\rho \frac{\|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N} \qquad (22)
\end{aligned}$$

To this end, we assume that $\partial\|\boldsymbol{\beta}_N\|_1/\partial\boldsymbol{\beta}_N$ do not change significantly in a single time step, i.e., $\partial\|\boldsymbol{\beta}_N\|_1/\partial\boldsymbol{\beta}_N$ approach to $\partial\|\boldsymbol{\beta}_{N-1}\|_1/\partial\boldsymbol{\beta}_N$. Hence, we approximate (22) by

$$\Upsilon_N = \lambda \Upsilon_{N-1} + |e_N|^{p-2} t_N g_N + \rho(1-\lambda) \frac{\partial \|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N} \tag{23}$$

Substituting (23) into (19), we can get,

$$\boldsymbol{\beta}_N = \Psi_N^{-1}[\lambda \Upsilon_{N-1} + |e_N|^{p-2} t_N g_N + \rho(1-\lambda) \frac{\partial \|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N}] \tag{24}$$

Considering (20) and applying the matrix inversion lemma [58],

$$(A + \mu \boldsymbol{x}\boldsymbol{y}^T)^{-1} = A^{-1}(I - \frac{\mu \boldsymbol{x}\boldsymbol{y}^T A^{-1}}{1 + \mu \boldsymbol{y}^T A^{-1}\boldsymbol{x}}) \tag{25}$$

Letting $\Psi_{N-1} = A$, $\boldsymbol{x} = \boldsymbol{y} = g_N$, $\mu = |e_N|^{p-2}$, and we can get,

$$\Psi_N^{-1} = \lambda^{-1}\Psi_{N-1}^{-1}(I - \frac{|e_N|^{p-2} g_N g_N^T \Psi_{N-1}^{-1}}{\lambda + |e_N|^{p-2} g_N^T \Psi_{N-1}^{-1} g_N}) \tag{26}$$

For a simple description of (26), we introduce Ω_N and K_N as

$$\begin{aligned}
\Omega_N &= \Psi_N^{-1} \\
K_N &= \frac{|e_N|^{p-2}\Omega_{N-1} g_N}{\lambda + |e_N|^{p-2} g_N^T \Omega_{N-1} g_N}
\end{aligned} \tag{27}$$

Then we obtain

$$\Omega_N = \lambda^{-1}(I - K_N g_N^T)\Omega_{N-1} \tag{28}$$

where Ω_N and K_N are the extended kalman gain vectors similar to those in RLS. Thus (24) can be rewritten as

$$\begin{aligned}
\boldsymbol{\beta}_N &= \lambda^{-1}(I - K_N g_N^T)\Omega_{N-1}[\lambda \Upsilon_{N-1} + |e_N|^{p-2} t_N g_N \\
&\quad + \rho(1-\lambda)\frac{\partial \|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N}] \\
&= (I - K_N g_N^T)[\Omega_{N-1}\Upsilon_{N-1} + \lambda^{-1}\Omega_{N-1}|e_N|^{p-2} t_N g_N \\
&\quad + \rho\lambda^{-1}(1-\lambda)\Omega_{N-1}\frac{\partial \|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N}] \\
&= \Omega_{N-1}\Upsilon_{N-1} - K_N \Omega_{N-1}\Upsilon_{N-1} \\
&\quad + \lambda^{-1}|e_N|^{p-2}(I - K_N g_N^T)\Omega_{N-1} t_N g_N \\
&\quad + \rho\lambda^{-1}(1-\lambda)(I - K_N g_N^T)\Omega_{N-1}\frac{\partial \|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N} \\
&= \boldsymbol{\beta}_{N-1} + K_N(t_N - g_N^T \boldsymbol{\beta}_{N-1}) \\
&\quad + \rho\lambda^{-1}(1-\lambda)(I - K_N g_N^T)\Omega_{N-1}\frac{\partial \|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N}
\end{aligned} \tag{29}$$

The equation for updating $\boldsymbol{\beta}_N$ can be gotten,

$$\begin{aligned}
\boldsymbol{\beta}_N &= \boldsymbol{\beta}_{N-1} + e_N K_N \\
&\quad + \rho\lambda^{-1}(1-\lambda)(I - K_N g_N^T)\Omega_{N-1}\frac{\partial \|\boldsymbol{\beta}_{N-1}\|_1}{\partial \boldsymbol{\beta}_N}
\end{aligned} \tag{30}$$

Furthermore, the derivative $\partial\|\boldsymbol{\beta}_N\|_1/\partial\boldsymbol{\beta}_N$ is $sign(\boldsymbol{\beta}_N)$. $sign(\cdot)$ is the sign function. The function for updating $\boldsymbol{\beta}_N$ can be obtained,

$$\begin{aligned}
\boldsymbol{\beta}_N &= \boldsymbol{\beta}_{N-1} + e_N K_N \\
&\quad + \rho\lambda^{-1}(1-\lambda)(I - K_N g_N^T)\Omega_{N-1} sign(\boldsymbol{\beta}_{N-1})
\end{aligned} \tag{31}$$

B. SPARSE RECURSIVE LEAST MEAN p-POWER - ELM ALGORITHM (SRLMP-ELM)

Considering again the description of ELM in preliminary, there is a standard SLFN and N arbitrary distinct samples $(\mathbf{x}_k, t_k)$ in the algorithm. The SLFN with $\tilde{N}$ hidden nodes with activation function $g(x)$ and the hidden layer output matrix is $g_k = [g(\mathbf{x}_k; a_1, b_1), g(\mathbf{x}_k; a_2, b_2), \ldots, g(\mathbf{x}_k; a_{\tilde{N}}, b_{\tilde{N}})]^T$. Now, the SRLMP-ELM algorithm can be summarized as follows.

SRLMP-ELM Algorithm:

1) Assign random input weights a_i and bias b_i (for additive hidden nodes) or center a_i and impact factor b_i (for RBF

hidden nodes), $i = 1, \cdots, \tilde{N}$. Initialize $\boldsymbol{\beta}_0 = 0$, $\Omega_0 = \mathbf{I}_{\tilde{N}\times\tilde{N}}$, λ,ρ and p. Set the training step $k = 1$.

2) Obtain the current training data $(\mathbf{x}_k, t_k)$
3) Calculate the hidden layer output matrix
$g_k = [g(\mathbf{x}_k;\ c_1, a_1)\ g(\mathbf{x}_k; c_2, a_2), \ldots, g(\mathbf{x}_k;\ c_{\tilde{N}}, a_{\tilde{N}})]^T$
4) Calculate the error term
$e_k = t_k - \boldsymbol{\beta}_{k-1}^T \mathbf{g}_k$
5) Calculate the gain vector $K_k = \dfrac{|e_k|^{p-2}\Omega_{k-1}g_k}{\lambda + |e_k|^{p-2}g_k^T\Omega_{k-1}g_k}$
6) Calculate the output weight $\boldsymbol{\beta}_k$
$\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} + e_k K_k + \rho\lambda^{-1}(1-\lambda)(I - K_k g_k^T)\Omega_{k-1} sign(\boldsymbol{\beta}_{k-1})$
7) Update Ω_k
$\Omega_k = \lambda^{-1}(I - K_k g_k^T)\Omega_{k-1}$
8) If there is any new training data, set $k = k + 1$ and go to 2. Otherwise, the algorithm is terminated.

Remark: We can further discuss the computation complexity of the proposed SRLMP-ELM algorithms. For the $\tilde{N}$ hidden units and N-length training sequence, the total training complexity of the SRLMP-ELM is of $O(N\tilde{N}^2)$. The same computation complexity can thus be observed comparing that of $O(N\tilde{N}^2)$ in the primitive ELM matrix inversion [11] and of $O(N\tilde{N}^2)$ in the OS-ELM [14], [59]. But since the data is processed sequentially in the SRLMP-ELM and OS-ELM, they cost more time than the ELM algorithm. However, the more compact model can obtained by SRLMP-ELM through the sparse penalty constraint. Thus, the running time can be rapidly deduced in the testing phase that is illustrated by the following simulation results.

IV. PERFORMANCE EVALUATION

In this section, the performance of the proposed SRLMP-ELM learning algorithm is compared with ELM, OS-ELM and RLMP-ELM on a few regression problems. To confirm the validity of the proposed SRLMP-ELM with different p and ρ value, we utilize training samples with the noises of several different distributions for illustrating that the better performance could be achieved through choosing p and ρ value according to the features of the noises distribution.

The symmetric alpha-stable ($S\alpha S$) distribution is a classic non-Gaussian distribution, which can model impulsive type of noises with heavy-tailed distributions [31]. In many literatures, the impulsive characteristics of physical noise sources have been modeled by the $S\alpha S$ [26], [28]–[30], [33], [34]. Generally, a $S\alpha S$ random distribution can be described conveniently by its characteristic function [31], [60]

$$\phi(t) = exp(j\mu t - \gamma|t|^{\alpha}) \tag{32}$$

where $\alpha \in (0, 2]$ is the characteristic exponent and completely determines the shape of the distribution, i.e., the thickness of the tail in the distribution. This family of distributions comprises the particular case of Gaussian with $\alpha = 2$. The second-order and higher-order statistics of the symmetric alpha-stable distribution ($\alpha \neq 2$) are infinity. μ is the location parameter (and assumed to be zero here). γ is the dispersion of the distribution and similar to the variance of Gaussian random variable. In practice, the signal of semi-conducting electrical devices in communication and radar systems is subject to internal thermal Gaussian noises. Hence a sum of independent $S\alpha S$ and Gaussian random process appears in a variety of practical situations mentioned above, namely, a $S\alpha SG$ distribution [55], [61]–[63]. The process is easily presented in the characteristic function

$$\phi(t) = exp(-\gamma_{S\alpha S}|t|^{\alpha} - \gamma_G|t|^{\alpha}) \tag{33}$$

where $\gamma_{S\alpha S} > 0$ and $\gamma_G = \sigma_G^2/2 > 0$ are the dispersions of $S\alpha S$ and Gaussian random variables. σ_G^2 is related to the variance of the Gaussian component.

In order to effectively illustrate the good performance of SRLMP-ELM algorithm, Gaussian and non-Gaussian datasets are considered in the study. For Gaussian dataset, Gaussian noises are added to the noise free training set or real data to generate training samples, called as Gaussian training set. Some non-Gaussian dataset, such as symmetry alpha-stable ($S\alpha S$) noise, sum of independent $S\alpha S$ and Gaussian random noise ($S\alpha SG$), and Uniform noises are used to create training samples. They are called as $S\alpha S$ training set, $S\alpha SG$ training set and Uniform training set, respectively. Furthermore, all the simulations are carried out in MATLAB R2013a environment running in an Intel(R) CORE(TM) i5 CPU, 1.80GHz, 8GB RAM. The details of validation process are shown in the following sections.

A. *SinC*

In this section, a popular example in literatures, *SinC* function, is presented to confirm the theoretical analysis of the proposed SRLMP-ELM algorithm. Here *SinC* is given as,

$$y(x) = \begin{cases} \sin(x)/x & x \neq 0 \\ 0 & x = 0 \end{cases} \tag{34}$$

We randomly create 5000 data for the training and validation sets, respectively, where the input x is the uniform distribution on the interval $[-10, 10]$.

For illustrating the compact size of the proposed network model, we make model selection procedure firstly for each type of dataset to determine the optimal architecture, that is the number of the hidden nodes. Then we illustrate the performance of SRLMP-ELM algorithm by comparing with ELM, OS-ELM and RLMP-ELM algorithms.

1) MODEL SELECTION

The estimation of optimal architecture of the classic ELM network is called as model selection in the literature. It is problem specific and has to be predetermined. For ELM, OS-ELM and RLMP-ELM algorithms, the optimal number of hidden units needs to be determined. And what's more, the initial network size of SRLMP-ELM should be determined by the model selection. In order to illustrate the good performance of SPLMP-ELM algorithm, the number of hidden units of OS-ELM, RLMP-ELM and the initial number of

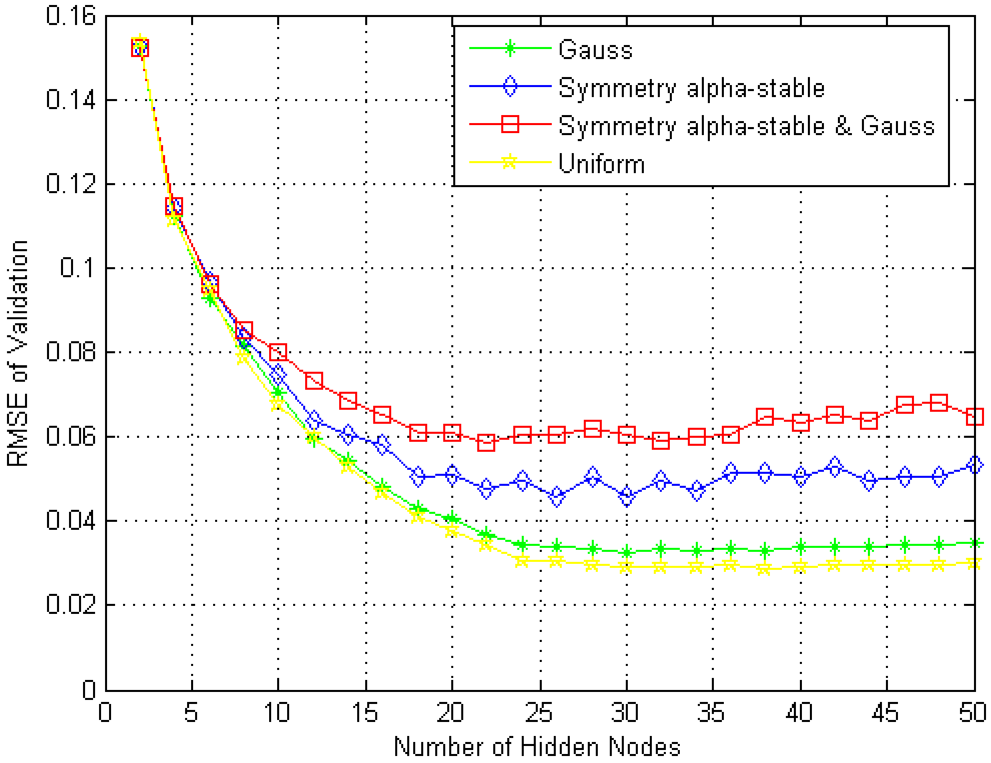


FIGURE 1. **ELM model selection for *SinC* based on four types of training sets.**

hidden nodes in this paper are selected as same as the one of ELM algorithm. Thus the model selection procedure is focus on the performance of ELM algorithm with different hidden nodes while the training datasets are Gaussian or non-Gaussian separately.

For ELM algorithm with every training dataset, such as Gaussian set, $S\alpha S$ set, $S\alpha SG$ set and Uniform set, the training process is performed with different number of hidden nodes which is chosen from the range [2, 50] with the interval 2, while the Gaussian activation function is selected here for the hidden nodes. Here Monte Carlo method is used and over 200 trials are conducted for each number of hidden nodes. The result of the model selection is shown in Figure 1.

For ELM algorithm with Gaussian training dataset, random zero mean Gaussian noises with variance 0.16 are created and added to all training samples to generate the Gaussian training set in each trial. After each trial, the testing set without any noises are used to validate the performance of the algorithm. The average performance is calculated after over 200 trials and shown with green curve in Figure 1. The Root Mean Square Error (RMSE) of the testing set is used as the criterion of the ELM's performance. For other three training datasets, the model selection procedures are the same as that of Gaussian training set. But different type of noises are added on the training samples to create corresponding training set as mentioned above. For $S\alpha S$ training set, Symmetry alpha-stable random noise ($\alpha = 1.2$ and the dispersion $\gamma_{S\alpha S} = 0.04$) are used. For $S\alpha SG$ training set, the sum of independent $S\alpha S$ ($\alpha = 1.2$, $\gamma_{S\alpha S} = 0.04$) and Gaussian (zero mean, the variance is 0.16) random noises are used. For Uniform training set, the large uniform noise distributed in $[-0.5, 0.5]$ has been added to all the training samples. The performances of ELM with these three different training datasets are illustrated in blue, red and yellow curves in Figure 1, separately.

As observed from the figure, the lowest validation errors are achieved when the number of hidden nodes of ELM is above 24 for the Gaussian and Uniform training sets. It can also be seen that RMSE curves for these two training datasets are smooth. It implies that ELM algorithm is not sensitive to the network size while the outputs of training data are stained by Gaussian and Uniform noises. For $S\alpha S$ and $S\alpha SG$ training sets, the curves are not smooth and ELM algorithm is a little sensitive to the network size for the outputs with $S\alpha S$ and $S\alpha SG$ noises. But the lowest validation errors are achieved when the number of hidden nodes of ELM is in the range [20, 34]. According to the result of the model selection, 30 hidden units are chosen for ELM, OS-ELM, RLMP-ELM and the initial hidden units of SRLMP-ELM algorithm.

2) PERFORMANCE EVALUATION OF SRLMP-ELM ALGORITHM

In this part, the performance of SRLMP-ELM algorithms with different values of p and ρ is discussed. According to the analysis above, 30 is selected as the optimal number

TABLE 1. Performance comparison of SRLMP-ELM, RLMP-ELM, ELM and OS-ELM algorithms for *SinC* case based on four types of training sets.

Noise Type	Algorithms			Training			Validation			#nodes
				RMSE	Dev	Time(s)	RMSE	Dev	Time(s)	
Gauss	ELM			0.3996	0.0045	0.0310	0.0326	0.0045	0.0183	30
	OS-ELM			0.4031	0.0049	0.7198	0.0411	0.0065	0.0216	30
	SRLMP-ELM	$p = 1.6$	$\rho = 0$ RLMP-ELM	0.4000	0.0038	0.7539	0.0279	0.0064	0.0198	30
			$\rho = 0.3$	0.3999	0.0041	0.7839	0.0307	0.0039	0.0146	24
			$\rho = 1.2$	0.4007	0.0043	0.7620	0.0313	0.0069	0.0106	20
		$p = 2.0$	$\rho = 0$ RLMP-ELM	0.3994	0.0045	0.7609	0.0307	0.0053	0.0187	30
			$\rho = 0.3$	0.3996	0.0037	0.7638	0.0292	0.0042	0.0145	25
			$\rho = 1.2$	0.3998	0.0039	0.7740	0.0331	0.0058	0.0096	21
$S\alpha S$	ELM			0.6320	0.4170	0.0313	0.0503	0.0275	0.0153	30
	OS-ELM			0.7096	0.4403	0.7873	0.0527	0.0278	0.0146	30
	SRLMP-ELM	$p = 1.6$	$\rho = 0$ RLMP-ELM	0.6750	0.4413	0.8134	0.0151	0.0027	0.0164	30
			$\rho = 0.3$	0.6016	0.4473	0.8006	0.0182	0.0046	0.0113	19
			$\rho = 1.2$	0.7186	0.4607	0.8706	0.0203	0.0035	0.0102	17
		$p = 2.0$	$\rho = 0$ RLMP-ELM	0.5984	0.4130	0.8000	0.0445	0.0220	0.0155	30
			$\rho = 0.3$	0.5620	0.3966	0.8813	0.0459	0.0125	0.0115	20
			$\rho = 1.2$	0.6113	0.3927	0.8682	0.0468	0.0144	0.0096	17
$S\alpha SG$	ELM			0.7648	0.3589	0.0302	0.0603	0.0237	0.0149	30
	OS-ELM			0.7096	0.4403	0.8173	0.0752	0.0278	0.0156	30
	SRLMP-ELM	$p = 1.6$	$\rho = 0$ RLMP-ELM	0.7883	0.3587	0.7895	0.0274	0.0046	0.0166	30
			$\rho = 0.3$	0.7020	0.3330	0.8453	0.0297	0.0048	0.0130	24
			$\rho = 1.2$	0.7192	0.3168	0.8433	0.0319	0.0049	0.0117	22
		$p = 2.0$	$\rho = 0$ RLMP-ELM	0.8317	0.4088	0.8828	0.0617	0.0234	0.0157	30
			$\rho = 0.3$	0.7674	0.3528	0.8730	0.0631	0.0249	0.0135	25
			$\rho = 1.2$	0.7441	0.3248	0.8858	0.0644	0.0227	0.0118	22
Uniform	ELM			0.3452	0.0021	0.0315	0.0292	0.0043	0.0165	30
	OS-ELM			0.3456	0.0027	0.8153	0.0317	0.0068	0.0156	30
	SRLMP-ELM	$p = 2.0$	$\rho = 0$ RLMP-ELM	0.3458	0.0021	0.8469	0.0289	0.0043	0.0142	30
			$\rho = 0.3$	0.3461	0.0023	0.8002	0.0295	0.0059	0.0122	26
			$\rho = 1.2$	0.3467	0.0033	0.8163	0.0306	0.0062	0.0111	21
		$p = 4.0$	$\rho = 0$ RLMP-ELM	0.3467	0.0021	0.7973	0.0261	0.0052	0.0153	30
			$\rho = 0.3$	0.3472	0.0021	0.7892	0.0275	0.0060	0.0123	22
			$\rho = 1.2$	0.3481	0.0028	0.7955	0.0313	0.0069	0.0101	17

of hidden nodes for ELM, OS-ELM and RLMP-ELM algorithms, in addition to the initial number of hidden units for SRLMP-ELM. The forgetting factor λ is set as 0.995.

The details of the comparison about SRLMP-ELM algorithm with different values p and ρ, ELM, OS-ELM and RLMP-ELM algorithms are summarized in the follow-up table. The averaged results over 200 independent trials on each algorithm in terms of the running time, the RMSE and the variance of the RMSE of the training and testing process are presented in Table 1. The number of hidden nodes is included in the table.

The RLMP-ELM algorithm proposed in our previous work is the specific case of the SRLMP-ELM algorithm with $\rho = 0$. As observed from Table 1, the accuracies of SRLMP-ELM with different ρ and p, ELM and OS-ELM based on Gaussian training dataset are similar to each other. All algorithms are robust to the Gaussian distribution data. There is an obvious difference that the training time cost by ELM is much less than those cost by other algorithms. Just as the above analysis, the computation complexity of ELM, OS-ELM, RLMP-ELM and SRLMP-ELM algorithms are same, but the last three algorithms cost more running time than ELM due to conducting data one by one. Another obvious difference is that the number of hidden nodes of SRLMP-ELM with some values of ρ is smaller than 30. The SRLMP-ELM with $p = 1.6$ and $\rho = 1.2$ has only 20 hidden units while its accuracy is almost equal to that of the ELM. Similarly, there is only 21 hidden units in the SRLMP-ELM with $p = 2$ and $\rho = 1.2$. Thus, the corresponding algorithm has more compact model than that of the ELM, OS-ELM and RLMP-ELM algorithm while the accuracies of all the algorithms are almost same. Furthermore, the testing time of the SRLMP-ELM with these two compact models is less than that of the other algorithms just as thought in advance.

The performances of all algorithms for the $S\alpha S$ training dataset are also shown in Table 1. The validation RMSE of SRLMP-ELM algorithm with $p = 1.6$ and ρ in the range of [0, 1.2] are much better than that of other algorithms. Just as mentioned in our previous work, the algorithms with least mean square criterion are sensitive to the data with impulsive characteristic, while RLMP-ELM with $p = 1.6$ are more robust to impulsive training data used here. The SRLMP-ELM algorithm with $p = 1.6$ and $\rho = 0$, i.e., RLMP-ELM with $p = 1.6$, obtains the lowest testing root-mean-square error (RMSE) 0.0151 while the criteria of ELM and OS-ELM are both above 0.05. The validation RMSE of SRLMP-ELM algorithm with $p = 1.6, \rho = 0.3$ and $p = 1.6, \rho = 1.2$

are 0.0182 and 0.0203, separately. This is a little larger than that of RLMP-ELM with $p = 1.6$. However, the number of hidden units for these two algorithms are 19 and 17. Thus these two algorithms have more compact models than that of RLMP-ELM with $p = 1.6$ which has 30 hidden nodes and is the same as that of ELM and OS-ELM. On the other hand, the testing time of the SRLMP-ELM with these two compact models is less than those of the other algorithms. The accuracies of SRLMP-ELM algorithm with $p = 2$ and ρ in the range of $[0, 1.2]$ are almost same with the performance of ELM and OS-ELM. However the number of hidden units for SRLMP-ELM algorithm with $p = 2$, $\rho = 0.3$ and $p = 2$, $\rho = 1.2$ are smaller than 30. Also these two algorithms require less testing time. In conclusion, for the data with impulsive characteristic, the SRLMP-ELM algorithm with $p = 1.6$ and ρ value in the range of $[0.3, 1.2]$ can obtain better accuracy and more compact model than other algorithms in comparison, that is, ELM, OS-ELM and RLMP-ELM algorithms.

Table 1 also illustrates the performances of all algorithms based on $S\alpha SG$ training dataset. The validation RMSEs of ELM, OS-ELM, RLMP-ELM algorithms with $p = 2$ and SRLMP-ELM algorithm with $p = 2$ and ρ value in the range of $[0.3, 1.2]$ are almost the same. SRLMP-ELM algorithm with $p = 1.6$, $\rho = 0$, that is RLMP-ELM algorithm with $p = 1.6$, still obtains the lowest testing RMSE 0.0274. However, the SRLMP-ELM algorithm with $p = 1.6$ and ρ value in the range of $[0.3, 1.2]$ can obtain the accuracy in the range of $[0.0297, 0.0319]$ while the model size is in the range of $[22, 24]$ and the testing time is less than 0.0117 second. From algorithm accuracy and model complexity, the performance of SRLMP-ELM algorithm with $p = 1.6$ and ρ value in the range of $[0.3, 1.2]$ are better than other algorithms for the data with $S\alpha SG$ distribution, the sum of impulsive and Gaussian data.

Finally, the performances of all algorithms based on Uniform training dataset are shown in the bottom of Table 1. The SRLMP-ELM algorithm with $p = 4, \rho = 0$, i.e., RLMP-ELM algorithm with $p = 4$ obtain the lowest testing RMSE 0.261 because the Uniform data are bounded. The best accuracy is only slightly better than those of other algorithms. It is not obvious. However, the SRLMP-ELM algorithm with $p = 4$ and ρ value in the range of $[0.3, 1.2]$ obtains more compact model and less testing running time than those of other algorithms.

From the simulation results of SinC case, we have observed that SRLMP-ELM algorithm with appropriate p and ρ value can obtain better accuracy, more compact model and less testing time on non-Gaussian dataset than ELM and OS-ELM algorithms. The proposed algorithm with ρ in the range of $[0.3, 1.2]$ can have fewer hidden nodes and less testing time than RLMP-ELM with the same p value, while their learning accuracies are similar. In order to further illustrate the good performance of proposed algorithm, we have conducted the detailed simulation on the two real datasets. One is the non-stationary time-series prediction problem, predicting time series value of the internet traffic. The other is predicting the Altitude value of some location in 3D road networks. The details are given in the following sections.

B. TIME SERIES OF INTERNET TRAFFIC

A real internet traffic dataset is considered in this example and we get it from a researcher Paulo Cortez's home page, http://www3.dsi.uminho.pt/pcortez/series/A5M.txt. The goal is to predict the value of the current sample using the previous ten consecutive samples. All the datasets are normalized into $[0, 1]$.

In this experiment, the number of training observation samples is 4000 and the number of testing observation samples is 2000. The same Gaussian and non-Gaussian noises as described above are added to the 4000 training data in each trial. According to the model selection presented above, 18 is selected as the optimum number of hidden units for ELM, OS-ELM, RLMP-ELM and the initial hidden nodes of SRLMP-ELM. For each type of training data set, the average results over 200 trails are shown in Table 2.

The detailed performances of each algorithm for Gaussian training dataset are illustrated in Table 2. As can be observed from the table, almost all algorithms obtain similar accuracy, except SRLMP-ELM algorithms with $p = 1.6, \rho = 1.2$ and $p = 2, \rho = 1.2$. The validation RMSE of these two SRLMP-ELM algorithms are both above 0.06 that is larger than that of ELM 0.0394 since there is only 10 hidden nodes for these two algorithms. SRLMP-ELM algorithms with $p = 1.6$, $\rho = 0.3$ and $p = 2$, $\rho = 0.3$ can obtain the similar accuracy as ELM while these two algorithms only have 15 hidden nodes and less testing time.

As can be observed from Table 2, in case of $S\alpha S$ training dataset, the testing RMSEs of SRLMP-ELM with $p = 1.6$ and ρ in the range of $[0, 1.2]$ are less than those of other algorithms, ELM, OS-ELM and SRLMP-ELM algorithm with $p = 2$. The lowest testing RMSE is obtained by SRLMP-ELM with $p = 1.6, \rho = 0$, that is RLMP-ELM algorithm with $p = 1.6$. The testing accuracy of SRLMP-ELM algorithm with $p = 1.6, \rho = 0.3$ is 0.0303. This is almost half of that of ELM while the number of hidden unites is only 11 and the testing time is 0.0048 second. SRLMP-ELM algorithms with $p = 1.6$, $\rho = 1.2$ can obtain the testing accuracy 0.0377 with 8 hidden nodes and 0.0035 second testing time.

Table 2 shows the performances of all algorithms for the $S\alpha SG$ training dataset. The testing RMSEs of SRLMP-ELM with $p = 1.6$ and ρ in the range of $[0, 1.2]$ are a little less than those of other algorithms for the case of Gaussian random noises. The lowest testing RMSE is obtained by SRLMP-ELM with $p = 1.6, \rho = 0$, i.e., RLMP-ELM algorithm with $p = 1.6$. The good accuracy and more compact model are obtained by SRLMP-ELM with $p = 1.6, \rho = 0.3$ and $p = 1.6, \rho = 1.2$. This is similar to $S\alpha S$ training dataset.

For Uniform training dataset, the testing accuracy of SRLMP-ELM with $p = 4$ and ρ in the range of $[0, 1.2]$

TABLE 2. Performance comparison of SRLMP-ELM, RLMP-ELM, ELM and OS-ELM algorithms for *Time series of Internet traffic* case based on four types of training sets.

Noise Type	Algorithms			Training			Validation			#nodes
				RMSE	Dev	Time(s)	RMSE	Dev	Time(s)	
Gauss	ELM			0.3991	0.0043	0.0249	0.0394	0.0064	0.0060	18
	OS-ELM			0.4063	0.0057	0.4805	0.0481	0.0335	0.0058	18
	SRLMP-ELM	$p = 1.6$	$\rho = 0$ RLMP-ELM	0.4004	0.0045	0.5031	0.0376	0.0064	0.0048	18
			$\rho = 0.3$	0.4015	0.0047	0.4885	0.0402	0.0084	0.0042	15
			$\rho = 1.2$	0.4023	0.0045	0.5008	0.0630	0.0089	0.0029	10
		$p = 2.0$	$\rho = 0$ RLMP-ELM	0.4010	0.0049	0.4849	0.0349	0.0058	0.0063	18
			$\rho = 0.3$	0.4006	0.0041	0.5182	0.0404	0.0065	0.0047	15
			$\rho = 1.2$	0.4035	0.0037	0.5008	0.0664	0.0078	0.0030	10
$S\alpha S$	ELM			0.6565	0.4449	0.0182	0.0589	0.0318	0.0069	18
	OS-ELM			0.6748	0.4523	0.4750	0.0588	0.0321	0.0075	18
	SRLMP-ELM	$p = 1.6$	$\rho = 0$ RLMP-ELM	0.6861	0.4654	0.4883	0.0260	0.0105	0.0067	18
			$\rho = 0.3$	0.5800	0.4119	0.5028	0.0303	0.0196	0.0048	11
			$\rho = 1.2$	0.6710	0.4541	0.4856	0.0377	0.0220	0.0035	8
		$p = 2.0$	$\rho = 0$ RLMP-ELM	0.6656	0.4324	0.4783	0.0462	0.0220	0.0065	18
			$\rho = 0.3$	0.6952	0.4639	0.4581	0.0578	0.0209	0.0043	11
			$\rho = 1.2$	0.6696	0.4467	0.4863	0.0659	0.0251	0.0032	6
$S\alpha SG$	ELM			0.7836	0.3596	0.0162	0.0669	0.0281	0.0053	18
	OS-ELM			0.7978	0.3690	0.4941	0.0695	0.0362	0.0060	18
	SRLMP-ELM	$p = 1.6$	$\rho = 0$ RLMP-ELM	0.8245	0.3936	0.4952	0.0382	0.0069	0.0065	18
			$\rho = 0.3$	0.8422	0.4149	0.4850	0.0438	0.0087	0.0044	16
			$\rho = 1.2$	0.8162	0.4018	0.4933	0.0640	0.0115	0.0039	11
		$p = 2.0$	$\rho = 0$ RLMP-ELM	0.8348	0.3765	0.4941	0.0498	0.0233	0.0061	18
			$\rho = 0.3$	0.8569	0.4028	0.4847	0.0531	0.0204	0.0048	16
			$\rho = 1.2$	0.7458	0.3377	0.4984	0.0696	0.0234	0.0036	11
Uniform	ELM			0.3462	0.0025	0.0170	0.0384	0.0096	0.0059	18
	OS-ELM			0.3460	0.0026	0.4950	0.0390	0.0036	0.0056	18
	SRLMP-ELM	$p = 2.0$	$\rho = 0$ RLMP-ELM	0.3466	0.0025	0.4947	0.0369	0.0071	0.0057	18
			$\rho = 0.3$	0.3469	0.0024	0.4879	0.0388	0.0072	0.0041	16
			$\rho = 1.2$	0.3500	0.0028	0.5862	0.0438	0.0084	0.0033	10
		$p = 4.0$	$\rho = 0$ RLMP-ELM	0.3474	0.0023	0.5655	0.0336	0.0064	0.0062	18
			$\rho = 0.3$	0.3475	0.0027	0.5140	0.0361	0.0074	0.0044	16
			$\rho = 1.2$	0.3411	0.0030	0.5218	0.0416	0.0092	0.0029	10

are slightly better than those of other algorithms. The lowest testing RMSE is obtained by SRLMP-ELM algorithm with $p = 4$, $\rho = 0$, that is RLMP-ELM with $p = 4$. The similar accuracy and more compact model are obtained by SRLMP-ELM with $p = 4$, $\rho = 0.3$ and $p = 4$, $\rho = 1.2$, as in the case of SinC. The details are illustrated in Table 2.

C. 3D ROAD NETWORK

This dataset is constructed by a 3D road network in North Jutland, Denmark. Each sample includes longitude, latitude and altitude. This 3D road network dataset can be used by any applications that require to know very accurate elevation information of a road network to perform more accurate routing for eco-routing, cyclist routes etc. For the data mining and machine learning community, this dataset also can be used as ground-truth validation in spatial mining techniques and satellite image processing. This dataset can be achieved on http://archive.ics.uci.edu/ml/datasets.html. In our experiment, the inputs are longitude and latitude. The output is altitude.

Here, 5000 and 1000 samples of 3D road network dataset are randomly chosen for training and testing at each trial. The procedure of creating training dataset is totally same as that in the SinC case. According to the model selection procedure, 60 is selected as the optimal number of hidden units for ELM, OS-ELM, RLMP-ELM and the initial number of hidden units for SRLMP-ELM. For each type of training dataset, the average results over 200 trails are shown in Table 3.

For this problem, the performances of all algorithms based on different types of training dataset are similar with those in the above two cases. For Gaussian training dataset, the performances of all algorithms are substantially similar, but SRLMP-ELM with $p = 1.6$, $\rho = 0.3$ and $p = 1.6$, $\rho = 1.2$ have more compact model and less testing time, as observed from Table 3. From the table, it can be seen that SRLMP-ELM with $p = 1.6$ and ρ value in the range of $[0.3, 1.2]$ have less RMSE than ELM and OS-ELM in case of $S\alpha S$ and $S\alpha SG$ training dataset. The lowest testing RMSE is obtained by SRLMP-ELM with $p = 1.6$ and $\rho = 0$, that is RLMP-ELM with $p = 1.6$ in both of these two training data set. The good accuracy and more compact model are obtained by SRLMP-ELM with $p = 1.6$, $\rho = 0.3$ and $p = 1.6$, $\rho = 1.2$. The performance details of all the algorithms for Uniform training set are illustrated in the bottom of Table 3. From the table, it can be seen that SRLMP-ELM algorithm with $p = 4$ and $\rho = 0.3$ has slightly less RMSE than those of

TABLE 3. Performance comparison of SRLMP-ELM, RLMP-ELM, ELM and OS-ELM algorithms for 3*D Road Network* case based on four types of training sets.

Noise Type	Algorithms			Training			Validation			#nodes
				RMSE	Dev	Time(s)	RMSE	Dev	Time(s)	
Gauss	ELM			0.4044	0.0039	0.0690	0.0832	0.0024	0.0063	60
	OS-ELM			0.4044	0.0040	3.2274	0.0842	0.0025	0.0068	60
	SRLMP-ELM	$p = 1.6$	$\rho = 0$ RLMP-ELM	0.4082	0.0045	3.3227	0.0836	0.0018	0.0064	60
			$\rho = 0.3$	0.4011	0.0041	3.3559	0.0901	0.0015	0.0046	41
			$\rho = 1.2$	0.4042	0.0042	3.3242	0.1042	0.0018	0.0034	33
		$p = 2.0$	$\rho = 0$ RLMP-ELM	0.4093	0.0043	3.3102	0.0847	0.0022	0.0069	60
			$\rho = 0.3$	0.4012	0.0039	3.3049	0.0911	0.0020	0.0047	39
			$\rho = 1.2$	0.4075	0.0038	3.3169	0.0959	0.0024	0.0035	33
$S\alpha S$	ELM			0.6058	0.4150	0.0671	0.0979	0.0286	0.0063	60
	OS-ELM			0.6901	0.4307	3.4879	0.1016	0.0316	0.0066	60
	SRLMP-ELM	$p = 1.6$	$\rho = 0$ RLMP-ELM	0.6313	0.4597	3.4486	0.0585	0.0015	0.0061	60
			$\rho = 0.3$	0.5712	0.4020	3.4726	0.0641	0.0014	0.0036	35
			$\rho = 1.2$	0.5994	0.4339	3.4677	0.0711	0.0012	0.0033	22
		$p = 2.0$	$\rho = 0$ RLMP-ELM	0.6409	0.3986	3.6075	0.0975	0.0154	0.0065	60
			$\rho = 0.3$	0.6516	0.4529	3.5218	0.1057	0.0223	0.0039	35
			$\rho = 1.2$	0.6529	0.3720	3.4851	0.1081	0.0131	0.0035	21
$S\alpha SG$	ELM			0.7784	0.3578	0.0703	0.1097	0.0278	0.0062	60
	OS-ELM			0.7713	0.3556	3.2292	0.1033	0.0214	0.0072	60
	SRLMP-ELM	$p = 1.6$	$\rho = 0$ RLMP-ELM	0.8012	0.4133	3.3611	0.0769	0.0026	0.0065	60
			$\rho = 0.3$	0.7959	0.3891	3.3460	0.0895	0.0026	0.0048	44
			$\rho = 1.2$	0.8323	0.3866	3.4665	0.0962	0.0023	0.0035	30
		$p = 2.0$	$\rho = 0$ RLMP-ELM	0.7812	0.3808	3.3532	0.1085	0.0177	0.0062	60
			$\rho = 0.1$	0.7630	0.3300	3.3628	0.1091	0.0113	0.0051	45
			$\rho = 1.3$	0.7972	0.3505	3.3515	0.1105	0.0089	0.0038	33
Uniform	ELM			0.3522	0.0022	0.0670	0.0807	0.0043	0.0061	60
	OS-ELM			0.3525	0.0024	3.4777	0.0813	0.0048	0.0062	60
	SRLMP-ELM	$p = 2.0$	$\rho = 0$ RLMP-ELM	0.3577	0.0024	3.5145	0.0783	0.0027	0.0061	60
			$\rho = 0.3$	0.3586	0.0021	3.5045	0.0901	0.0031	0.0053	47
			$\rho = 1.2$	0.3615	0.0024	3.4310	0.0987	0.0023	0.0034	34
		$p = 4.0$	$\rho = 0$ RLMP-ELM	0.3573	0.0024	3.6153	0.0605	0.0022	0.0063	60
			$\rho = 0.3$	0.3602	0.0025	3.5996	0.0754	0.0019	0.0048	46
			$\rho = 1.2$	0.3604	0.0028	3.5342	0.0862	0.0021	0.0033	36

ELM and OS-ELM. Furthermore, the more compact model and less testing time are achieved by the SRLMP-ELM with $p = 4$, $\rho = 0.3$ and $p = 4$, $\rho = 1.2$.

V. CONCLUSION

An efficient and accurate online sequential learning algorithm with more compact structure, for single-hidden layer feedforward neural networks (SLFNs) is proposed in this paper. It is called sparse recursive least mean p-power extreme learning machine (SRLMP-ELM). Same as ELM and OS-ELM, the activation functions for hidden units here can be any bounded nonconstant piecewise continuous functions for additive nodes and any integrable piecewise continuous functions for RBF nodes. The SRLMP-ELM algorithm maintains the computationally simple ELM structure but the sum criterion of a least mean p-power (LMP) error and an sparse penalty constraint, aiming to improve the generalization and compact the model while minimize the p powers of the error, provides a mechanism to update the output weights sequentially. Under the same architecture, SRLMP-ELM has the same computational complexity as those of ELM and OS-ELM. The real world benchmark regression and non-stationary time-series prediction problems are presented to show that the proposed SRLMP-ELM algorithm can obtain better performance in non-Gaussian situations than ELM and OS-ELM algorithms. The details are as follows,

1) For non-Gaussian and Gaussian distributed data, the SRLMP-ELM algorithm with some p values and ρ $(0.3 \leq \rho < 1.2)$ can achieve more compact model and less testing time.
2) For non-Gaussian heavy-tailed distributed data, such as alpha-stable noises and the sum of alpha-stable and Gaussian noises, the SRLMP-ELM algorithm with $p = 1.6$ and ρ $(0.3 \leq \rho < 1.2)$ can obtain better generalization performance, more accurate results, more compact models and less testing time.
3) As for non-Gaussian light-tailed distributed data, such as uniform noise, the SRLMP-ELM algorithm with $p = 4$ and ρ $(0.3 \leq \rho < 1.2)$ can get slightly better generalization performance with more compact structure and less testing time.
4) For Gaussian distributed data, the SRLMP-ELM algorithm with $p = 2.0$ and ρ $(0.3 \leq \rho < 1.2)$ can obtain the almost same generalization performance as ELM and OS-ELM, but the network structure is more compact.

REFERENCES

[1] F. Golestaneh, P. Pinson, and H. B. Gooi, "Very short-term nonparametric probabilistic forecasting of renewable energy generation—With application to solar energy," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3850–3863, Sep. 2016.

[2] P. C. Chang and C. Y. Fan, "A hybrid system integrating a wavelet and TSK fuzzy rules for stock price forecasting," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 6, pp. 802–815, Nov. 2008.

[3] Y. Li, Z. Jia, and X. Li, "Task scheduling based on weather forecast in energy harvesting sensor systems," *IEEE Sensors J.*, vol. 14, no. 14, pp. 3763–3765, Nov. 2014.

[4] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 22–32, Jan. 2012.

[5] B. Chen, S. Zhao, P. Zhu, and J. C. Prı́ncipe, "Quantized kernel recursive least squares algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1484–1491, Sep. 2013.

[6] R. Meir and V. E. Maiorov, "On the optimality of neural-network approximation using incremental algorithms," *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 323–337, Mar. 2000.

[7] S. Ferrari and R. F. Stengel, "Smooth function approximation using neural networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 24–38, Jan. 2005.

[8] M. Hou and X. Han, "Constructive approximation to multivariate function by decay rbf neural network," *IEEE Trans. Neural Netw.*, vol. 21, no. 9, pp. 1517–1523, Sep. 2010.

[9] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

[10] L. C. Yann, B. Leon, B. O. Genevieve, and K. R. Müller, *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 1998, pp. 9–50.

[11] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theorey and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[12] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[13] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[14] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

[15] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran, "Online sequential fuzzy extreme learning machine for function approximation and classification problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 4, pp. 1067–1072, Aug. 2009.

[16] Y. Ye, S. Squartini, and F. Piazza, "Online sequential extreme learning machine in nonstationary environments," *Neurocomputing*, vol. 116, pp. 94–101, Sep. 2013.

[17] W.-Y. Deng, Q.-H. Zheng, and Z.-M. Wang, "Cross-person activity recognition using reduced kernel extreme learning machine," *Neural Netw.*, vol. 53, pp. 1–7, May 2014.

[18] T. Matias, D. Gabriel, F. Souza, R. Araújo, and J. C. Pereira, "Fault detection and replacement of a temperature sensor in a cement rotary kiln," in *Proc. IEEE 18th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2013, pp. 1–8.

[19] J.-S. Lim, S. Lee, and H.-S. Pang, "Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to nonstationary system estimations," *Neural Comput. Appl.*, vol. 22, nos. 3–4, pp. 569–576, 2013.

[20] S. G. Soares and R. Araújo, "An adaptive ensemble of on-line extreme learning machines with variable forgetting factor for dynamic system prediction," *Neurocomputing*, vol. 171, pp. 693–707, Jan. 2016.

[21] J. Yang, Y. Shi, and H.-J. Rong, "Random neural q-learning for obstacle avoidance of a mobile robot in unknown environments," *Adv. Mech. Eng.*, vol. 8, no. 7, pp. 1–15, 2016.

[22] J. Yang, P. Chen, H.-J. Rong, and B. Chen, "Least mean p-power extreme learning machine for obstacle avoidance of a mobile robot," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, 2016, pp. 1968–1976.

[23] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York, NY, USA: Springer-Verlag, 2006.

[24] J. Luo, C.-M. Vong, and P.-K. Wong, "Sparse Bayesian extreme learning machine for multi-classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 4, pp. 836–843, Apr. 2014.

[25] Y. Jing, Y. Feng, H.-J. Rong, and B. Chen, "Recursive least mean p-power extreme learning machine," *Neural Netw.*, vol. 91, pp. 22–33, Jul. 2017.

[26] S. Liao and A. C. S. Chung, "Feature based nonrigid brain MR image registration with symmetric alpha stable filters," *IEEE Trans. Med. Imag.*, vol. 29, no. 1, pp. 106–119, Jan. 2010.

[27] M. Zimmermann and K. Dostert, "Analysis and modeling of impulsive noise in broad-band powerline communications," *IEEE Trans. Electromagn. Compat.*, vol. 44, no. 1, pp. 249–258, Feb. 2002.

[28] N. C. Beaulieu and S. Niranjayan, "New UWB receiver designs based on a Gaussian-Laplacian noise-plus-MAI model," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 4128–4133.

[29] F. Wen, "Diffusion least-mean P-power algorithms for distributed estimation in alpha-stable noise environments," *Electron. Lett.*, vol. 49, no. 21, pp. 1355–1356, Oct. 2013.

[30] J. Lee and C. Tepedelenlioglu, "Distributed detection in coexisting large-scale sensor networks," *IEEE Sensors J.*, vol. 14, no. 4, pp. 1028–1034, Apr. 2014.

[31] C. L. Nikias and M. Shao, *Signal Processing With Alpha-Stable Distributions and Applications*. New York, NY, USA: Wiley, 1995.

[32] M. Bouvet and S. C. Schwartz, "Comparison of adaptive and robust receivers for signal detection in ambient underwater noise," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 37, no. 5, pp. 621–626, May 1989.

[33] P. Kismode, "Alpha-stable distributions in signal processing of audio signals," in *Proc. 41st Conf. Simulation Modelling*, Sep. 2000, pp. 87–94.

[34] X. Zhong, A. B. Premkumar, and A. S. Madhukumar, "Particle filtering for acoustic source tracking in impulsive noise with alpha-stable process," *IEEE Sensors J.*, vol. 13, no. 2, pp. 589–600, Feb. 2013.

[35] G. Mao, "A timescale decomposition approach to network traffic prediction," *IEICE Trans. Commun.*, vol. E88B, no. 10, pp. 3974–3981, 2005.

[36] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015.

[37] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, nos. 16–18, pp. 3056–3062, 2007.

[38] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3460–3468, Oct. 2008.

[39] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 8, pp. 1352–1357, Aug. 2009.

[40] Y. Yang, Y. Wang, and X. Yuan, "Bidirectional extreme learning machine for regression problem and its learning effectiveness," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 9, pp. 1498–1505, Sep. 2012.

[41] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally pruned extreme learning machine," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 158–162, Jan. 2010.

[42] Y. Miche, M. van Heeswijk, P. Bas, O. Simula, and A. Lendasse, "TROP-ELM: A double-regularized ELM using LARS and Tikhonov regularization," *Neurocomputing*, vol. 74, no. 16, pp. 2413–2421, 2011.

[43] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, no. 1, pp. 359–366, 2008.

[44] R. Zhang, Y. Lan, G.-B. Huang, and Z.-B. Xu, "Universal approximation of extreme learning machine with adaptive growth of hidden nodes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 365–371, Feb. 2012.

[45] Z. Bai, G.-B. Huang, D. Wang, H. Wang, and M. B. Westover, "Sparse extreme learning machine for classification," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1858–1870, Oct. 2014.

[46] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 3125–3128.

[47] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4013–4025, Aug. 2010.

[48] E. M. Eksioglu and A. K. Tanc, "RLS algorithm with convex regularization," *IEEE Signal Process. Lett.*, vol. 18, no. 8, pp. 470–473, Aug. 2011.

[49] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*. Hoboken, NJ, USA: Wiley, 2010.

[50] W. Ma, H. Qu, J. Zhao, B. Chen, and G. Gui, "Sparsity aware normalized least mean p-power algorithms with correntropy induced metric penalty," in *Proc. IEEE Int. Conf. Digit. Signal Process. (DSP)*, Jul. 2015, pp. 638–642.

[51] B. Chen, L. Xing, Z. Wu, J. Liang, J. C. Príncipe, and N. Zheng, "Smoothed least mean *p*-power error criterion for adaptive filtering," *Digit. Signal Process.*, vol. 40, pp. 154–163, May 2015.

[52] Y. Xiao, Y. Tadokoro, and K. Shida, "Adaptive algorithm based on least mean p-power error criterion for Fourier analysis in additive noise," *IEEE Trans. Signal Process.*, vol. 47, no. 4, pp. 1172–1181, Apr. 1999.

[53] B. Chen, Y. Zhu, J. Hu, and J. C. Príncipe, *System Parameter Identification: Information Criteria and Algorithms*. Amsterdam, The Netherlands: Elsevier, 2013.

[54] S.-C. Chan and Y.-X. Zou, "A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: Fast algorithm and convergence performance analysis," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 975–991, Apr. 2004.

[55] M. Z. A. Bhotto and A. Antoniou, "Robust recursive least-squares adaptive-filtering algorithm for impulsive-noise environments," *IEEE Signal Process. Lett.*, vol. 18, no. 3, pp. 185–188, Mar. 2011.

[56] D. Zha, "Robust multiuser detection method based on least *p*-norm state space criterion," *Wireless Pers. Commun., Int. J.*, vol. 40, no. 2, pp. 191–204, 2006.

[57] S.-C. Pei and C.-C. Tseng, "Least mean p-power error criterion for adaptive FIR filter," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 9, pp. 1540–1547, Dec. 1994.

[58] P. S. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Norwell, MA, USA: Kluwer, 2008.

[59] G.-B. Huang, N.-Y. Liang, H.-J. Rong, P. Saratchandran, and N. Sundararajan, "On-line sequential extreme learning machine," in *Proc. IASTED Int. Conf. Comput. Intell. (CI)*, Calgary, AB, Canada, 2005, pp. 232–237.

[60] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," *Proc. IEEE*, vol. 81, no. 7, pp. 986–1010, Jul. 1993.

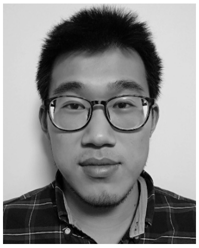
[61] G. Samorodnitsky and M. S. Taqqu, "Stable non-Gaussian random processes: Stochastic models with infinite variance," *J. Amer. Stat. Assoc.*, vol. 90, no. 430, pp. 805–806, 1995.

[62] J. Ilow, D. Hatzinakos, and A. N. Venetsanopoulos, "Performance of FH SS radio networks with interference modeled as a mixture of Gaussian and alpha-stable noise," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 509–520, Apr. 1998.

[63] R. Brcich and A. Zoubir, "Estimation and detection in a mixture of symmetric alpha stable and Gaussian interference," in *Proc. IEEE Signal Process. Workshop Higher-Order Stat.*, Jun. 1999, pp. 219–223.

JING YANG (M'16) received the B.S. and M.S. degrees in control science and engineering and the Ph.D. degree in pattern recognition and intelligent systems from Xi'an Jiaotong University, China, in 1999, 2002, and 2010, respectively.

From 1999 to 2003, she was a Research Assistant with the Institute of Automation, Xi'an Jiaotong University. Since 2003, she has been an Assistant Professor with the Department of Automation Science and Technology, Xi'an Jiaotong University. She is currently a visiting scholar with the Computational NeuroEngineering Laboratory, University of Florida. Her research interests include machine learning, reinforcement learning, and information theory and their applications to intelligent systems such as autonomous vehicles. Since 2004, she has been a member of Intelligent Vehicles Team, Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University.



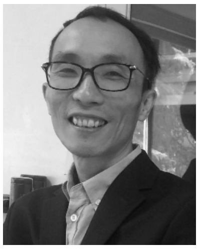
YI XU (S'18) received the B.S. degree in automation from Xi'an Jiaotong University, Xi'an, Shannxi, China, in 2017, where he is currently pursuing the master's degree in control science and engineering. His research interests include machine learning, artificial intelligence, model-based reinforcement learning and their applications to intelligent systems.

In 2015, he was the Team Leader of National Training Programs of Innovation and Entrepreneurship for Undergraduates. In 2016, he received third place at University Internet Plus Contest.

HAI-JUN RONG (M'14) received the B.Eng. degree in precision instrument from Xi'an Technological University, China, in 2000, the M.Eng. degree in control theory and control engineering from Xi'an Jiaotong University, China, in 2003, and the Ph.D. degree in intelligent control from Nanyang Technological University, Singapore, in 2008.

From 2006 to 2008, she was a Research Associate and Research Fellow with Nanyang Technological University. She has been an Associate Professor with the School of Aerospace, Xi'an Jiaotong University. Her research interests include neural networks, fuzzy systems, pattern recognition, and intelligent control. She is an Associate Editor of the *Evolving Systems* journal (Springer).

SHAOYI DU received the B.S. degrees in computational mathematics and in computer science, the M.S. degree in applied mathematics, and the Ph.D. degree in pattern recognition and intelligence system from Xi'an Jiaotong University, China, in 2002, 2005, and 2009 respectively. He was a Post-Doctoral Fellow with Xi'an Jiaotong University from 2009 to 2011 and was with The University of North Carolina at Chapel Hill from 2013 to 2014. He is currently a Professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, machine learning, and pattern recognition.



BADONG CHEN (M'10–SM'13) received the B.S. and M.S. degrees in control theory and engineering from Chongqing University, in 1997 and 2003, respectively, and the Ph.D. degree in computer science and technology from Tsinghua University in 2008. He was a Post-Doctoral Researcher with Tsinghua University from 2008 to 2010, and a Post-Doctoral Associate with the Computational NeuroEngineering Laboratory, University of Florida, from 2010 to 2012. In 2015, he was a Visiting Research Scientist with the Nanyang Technological University. In 2017, He was a Senior Research Fellow with The Hong Kong Polytechnic University. He is currently a Professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. He has published two books, four chapters, and over 200 papers in various journals and conference proceedings. His research interests include signal processing, information theory, machine learning, and their applications to cognitive science and neural engineering. He is a Technical Committee Member of the IEEE SPS Machine Learning for Signal Processing and the IEEE CIS Cognitive and Developmental Systems. He is an Associate Editor of the IEEE Transactions on Cognitive and Developmental Systems, the IEEE Transactions on Neural Networks and Learning Systems, and the *Journal of The Franklin Institute*. He has been on the Editorial Board of *Entropy*.