Robust Extreme Learning Machine With its Application to Indoor Positioning

Xiaoxuan Lu, Han Zou, Hongming Zhou, Lihua Xie, *Fellow, IEEE*, and Guang-Bin Huang, *Senior Member, IEEE*

***Abstract*—The increasing demands of location-based services have spurred the rapid development of indoor positioning system and indoor localization system interchangeably (IPSs). However, the performance of IPSs suffers from noisy measurements. In this paper, two kinds of robust extreme learning machines (RELMs), corresponding to the close-to-mean constraint, and the small-residual constraint, have been proposed to address the issue of noisy measurements in IPSs. Based on whether the feature mapping in extreme learning machine is explicit, we respectively provide random-hidden-nodes and kernelized formulations of RELMs by second order cone programming. Furthermore, the computation of the covariance in feature space is discussed. Simulations and real-world indoor localization experiments are extensively carried out and the results demonstrate that the proposed algorithms can not only improve the accuracy and repeatability, but also reduce the deviation and worst case error of IPSs compared with other baseline algorithms.**

***Index Terms*—Indoor positioning system (IPS), robust extreme learning machine (RELM), second order cone programming (SOCP).**

I. Introduction

Due to the nonline-of-sight transmission channels between a satellite and a receiver, wireless indoor positioning has been extensively studied and a number of solutions have been proposed in the past two decades. Unlike other wireless technologies, such as ultrawideband and radio frequency identification, which require the deployment of extra infrastructures, the existing IEEE 802.11 network infrastructures, such as WiFi routers, are widely available in large numbers of commercial and residential buildings. In addition, nearly every mobile device now is equipped with a WiFi receiver [1].

The WiFi-based machine learning (ML) approaches are becoming popular in indoor positioning in recent years [2]. Fingerprinting method based on WiFi received signal strength (RSS), in particular, has received a lot of attentions. The fingerprinting localization procedure usually involves two stages: 1) offline calibration stage and 2) online matching stage. During the offline stage, a cite survey is conducted and signal strengths received at each location from various access points (APs) are recorded in a radio map. During the online stage, users' positions can be estimated by matching the online RSSs with the fingerprints stored in the radio map. Online matching strategy according to the relationships between physical locations and RSS map modeled by different ML algorithms is crucial for the performance of indoor positioning systems (IPSs). Neural network (NN) and support vector machines (SVM) [3], as two sophisticated ML techniques, have both been utilized in fingerprinting-based indoor positioning [4].

However, either NN or SVM-based IPSs face two challenges. On one hand, NN and SVM are time-consuming, and this issue becomes more serious in fingerprinting-based positioning systems, because large amount of training data are required for generating a radio map. Their high computational costs leave us little leeway, especially for some large-scale scenarios, to improve the performance and robustness of ML-based IPSs. On the other hand, noisy measurements are inevitable, considering that manual observational errors of calibrated points happen throughout the calibration phase. In addition, signal variation and ambient dynamics also affect the signals received by APs. These adverse factors can be considered as uncertainties, which may degrade the performance of IPSs. Many researchers bypass optimizing ML methods to enhance the robustness of IPSs since it will aggravate the situation of slow training rate. Kothari *et al*. [5] utilized the integration of complementary localization algorithms of dead reckoning and WiFi signal strength fingerprinting to achieve robust indoor localization, nevertheless, a disadvantage of dead reckoning is that the errors are cumulative, since new positions are calculated solely from previous ones. Meng *et al*. [6] proposed a robust noniterative three-step location sensing method, but its capability of reducing the worst case error (WCE) and variance is comparatively limited. Other robust indoor localization algorithms demand either extra infrastructure or users' interaction during calibration phases, which is not cost-efficient in reality.

These undesirable results motivate us to reconsider the problem: can we find a ML technique which is fast in training and has the capability of handling the robustness issue in IPSs? As a novel learning technique, extreme learning machines (ELM) has been demonstrated with its outstanding performance in training speed, prediction accuracy, and generalization ability [7], [8]. Several IPSs have already leveraged ELM to deliver accurate location estimation with fast training speed [1], [9], [10].

Manuscript received September 4, 2014; revised December 13, 2014; accepted January 25, 2015. Date of publication February 24, 2015; date of current version December 14, 2015. This work was supported in part by the National Research Foundation of Singapore under Grant NRF2011NRF-CRP001-090 and Grant NRF2013EWT-EIRP004-012, and in part by the Natural Science Foundation of China under NSFC 61120106011. This paper was recommended by Associate Editor X. Wang.

The authors are with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798 (e-mail: xlu010@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2399420

Extended from ELM, this paper proposes two robust ELMs (RELMs), which can be implemented in the random-hidden-nodes form or kernelization form depending on the situation, to boost the robustness of IPSs.

The problem of uncertainty and robustness has been intensively studied in recent years. Wang *et al.* [11] proposed an ELM tree model-based on the heuristics of uncertainty reduction and computationally lightweight for big data classification. Fuzzy integral method is adopted to study the probabilistic feed-forward neural networks [12]. Horata *et al.* [13] proposed an approach, which is also named RELM to improve the computational robustness by extended complete orthogonal decomposition and outlier robustness by reweighted least squares. Unlike these works, considering the noises in IPS as discussed above, we propose our algorithm under a stochastic framework. It is worthwhile to mention that RELMs are based on second order cone programming (SOCP), which is widely adopted in robust convex optimization problems. Simulation and real-world experimental results both demonstrate that RELMs-based IPSs outperform other baseline algorithms-based IPSs in terms of accuracy, repeatability (REP), and WCE.

An outline of this paper is as follows. In Section II, we introduce the preliminaries for this paper, including basic components of a WiFi-based IPS, backgrounds for ELM, and its comparison with SVR. Two second order moment constraints, i.e., close to mean (CTM) and small residual (SR) constraints, with their geometric interpretations are given in Section III. The random-hidden-nodes and kernelized formulations of RELMs are derived in Sections IV and V, respectively. How to calculate the covariance in the feature space is studied in Section VI. In Section VII, the proposed algorithms are evaluated by both simulation and real-world IPSs. The conclusion is drawn in Section VIII.

II. Preliminaries

A. WiFi Indoor Positioning

An enormous body of indoor positioning problems fall into a sort of regression problem. As shown in Table I, the input variable $\mathbf{x}(x_1, x_2, \ldots, x_d)$ is a vector of RSS received from APs in the environment, and $\mathbf{t}(t_1, t_2)$ is the indoor 2-D physical coordinates of a target's location. When an AP is undetectable in a position, its corresponding RSS is taken as -100 dBm. The problem here is to train and approximate the regression model.

Although in some works, the procedure of collecting signal strength involves physically moving a wireless device all around the target area, as in [14] and [15], we only pick out some spatially representative locations, i.e., reference (calibration) points, from the target area, and conduct sampling at each reference point for a period of time to build up a radio map.

B. Introduction to ELM

Originally inspired by biological learning to overcome the challenging issues faced by back propagation (BP) learning algorithms, ELM is a kind of ML algorithm based on a generalized single-hidden layer feedforward NN (SLFN) architecture [16]. It has been demonstrated to provide good generalization performance at an extremely fast learning speed [17]–[19].

TABLE I
Input Variable: RSS ($\mathbf{x}$) and Output: Location ($\mathbf{t}$)

Location	AP_1	AP_2	AP_3	AP_4	AP_5	AP_6	AP_7
$\mathbf{t}_1$	-53	-87	-73	-80	-79	-87	-89
$\mathbf{t}_2$	-53	-62	-73	-80	-74	-84	-92
$\mathbf{t}_3$	-76	-72	-83	-71	-75	-79	-83
$\vdots$				$\ddots$			
(t_1, t_2)	x_1	x_2	x_3	x_4	x_5	x_6	x_7

Let $\Upsilon = \{(\mathbf{x}_i, \mathbf{t}_i); i = 1, 2, \ldots, N\}$ be a training set consisting of patterns, where $\mathbf{x}_i \in \mathbf{R}^{1\times d}$ and $\mathbf{t}_i \in \mathbf{R}^{1\times m}$, then the goal of regression is to find the relationship between $\mathbf{x}_i$ and $\mathbf{t}_i$. Since the only parameters to be optimized are the output weights, the training of ELM is equivalent to solving a least squares problem [20].

In the training process, the first stage is that the hidden neurons of ELM map the inputs onto a feature space

$$\mathbf{h} : \mathbf{x}_i \rightarrow \mathbf{h}(\mathbf{x}_i) \tag{1}$$

where $\mathbf{h}(\mathbf{x}_i) \in \mathbf{R}^{1\times L}$.

We denote $\mathbf{H}$ as the hidden layer output matrix (randomized matrix)

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \mathbf{h}(\mathbf{x}_2) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix}_{N\times L} \tag{2}$$

with L the dimension of the feature space and $\boldsymbol{\beta} \in \mathbf{R}^{L\times m}$ as the output weight matrix that connects the hidden layer with the output layer. Then, each output of ELM is given by

$$\mathbf{t}_i = \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta}, \quad i = 1, 2, \ldots, N. \tag{3}$$

ELM theory aims to reach the smallest training error but also the smallest norm of output weight [16]

$$\begin{aligned} \min_{\boldsymbol{\xi}, \boldsymbol{\beta} \in \mathbf{R}^{L\times m}} \quad & L_P = \frac{1}{2}\|\boldsymbol{\beta}\|_{\varpi_1}^{\varrho_1} + \frac{C}{2}\sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \|\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} - \mathbf{t}_i\|_{\varpi_2}^{\varrho_2} = \xi_i \quad i = 1, 2, \ldots, N \end{aligned} \tag{4}$$

where $\varrho_1 > 0, \varrho_2 > 0, \varpi_1, \varpi_2 = 0, 1/2, 1, 2, \ldots, +\infty$,[1] C is the penalty coefficient on the training errors and $\xi_i \in \mathbf{R}^m$ is the error vector with respect to the ith training pattern.

A simplest example of the above is basic ELM [17]

$$\begin{aligned} \min_{\boldsymbol{\beta} \in \mathbb{R}^{L\times m}} \quad & L_P = \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \|\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} - \mathbf{t}_i\|^2 = \xi_i \quad i = 1, 2, \ldots, N \end{aligned} \tag{5}$$

which can be solved by the least squares method

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{6}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of $\mathbf{H}$.

[1] Unless explicitly specified, $\varpi_1 = \varpi_2 = 2$ for all norm notations in this paper.

Extended from basic ELM, [21] proposed an optimization-based ELM (OPT-ELM) for the binary classification problem by introducing inequality constraints. We follow from [21] to give a form of OPT-ELM for regression problems:

$$\begin{aligned} \min_{\boldsymbol{\xi},\boldsymbol{\beta}\in\mathbf{R}^{L\times m}} \quad & L_P = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & \|\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} - \mathbf{t}_i\| \le \varepsilon + \xi_i \\ & \xi_i \ge 0 \quad i = 1, 2, \ldots, N \end{aligned} \tag{7}$$

where ε is a slack variable. This formulation is very similar to support vector regression (SVR) in a nonlinear case [3], [22], which is in the following form:

$$\begin{aligned} \min_{\boldsymbol{\xi},\mathbf{w},b} \quad & L_{P_{\text{SVM}}} = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & \|\mathbf{w}\cdot\phi(\mathbf{x}_i) + b - \mathbf{t}_i\| \le \varepsilon + \xi_i \\ & \xi_i \ge 0 \quad i = 1, 2, \ldots, N \end{aligned} \tag{8}$$

where $\phi(\cdot)$ is the nonlinear feature mapping function in SVR, $\mathbf{w}$ is the output weights and b is the approximation (output) bias. ε and ξ_i are as defined in the OPT-ELM case.

Detailed comparison between ELM and SVM for classification problems are given in [21] and [23], and in the next section we further this comparison to regression problems. For convenience of description, we henceforth follow from [16] to refer to the formulation of (7) as OPT-ELM, while basic ELM stands for the formulation of (5). The terminology ELM in the rest of this paper has more broad meaning, which can be considered as the gathering of basic ELM and its random-hidden-nodes-based variants.[2]

C. Comparisons Between ELM and SVR

Both formulations of ELM and SVR are within the scope of quadratic programming, however, the decision variable b, i.e., the bias term, is not existent in ELM.

SVR and its variants emphasize the importance of bias b in their implementation. The reason is that the separation capability of SVM was considered more important than its regression capability when SVM was first proposed to handle binary classification applications. Under this background, its universal approximation capability may somehow have been neglected [3]. Due to the inborn reason that the feature mapping $\phi(\cdot)$ in SVR is unknown, it is difficult to study the universal approximation capability of SVR without the explicitness of feature mapping. Since $\phi(\cdot)$ is unknown and may not have universal approximation capability, given a target function $f(\cdot)$ and any small ε precision, there may not exist a $\mathbf{w}$ such that $\|\mathbf{w}\cdot\phi(\mathbf{x}) - f(\mathbf{x})\| < \varepsilon$. In other words, there may exist some system errors even if SVM and its variants with appropriate kernels can classify different classes well, and these system errors need to be absorbed by the bias b. This may be the reason why in principle the bias b has to remain in the optimization constraints [16].

On the other hand, all the parameters of the ELM mapping $\mathbf{h}(\mathbf{x})$ are randomly generated, and $\mathbf{h}(\mathbf{x})$ is known to users finally. According to [17]–[19], ELM with almost any nonlinear piecewise continuous function $\mathbf{h}(\mathbf{x})$ has the universal approximation capability. Therefore, the bias b is not necessary in the output nodes of ELM.

In addition, from the optimization point of view, less decision variables to be determined implies less computational costs, and this computational superiority becomes more obvious when the scale of the training data gets larger.

Kernel ELM is somehow superior to SVR for the sake of flexibility in kernels. Namely, the feature mapping to form the kernels can be unknown mapping or random feature mapping. More introduction about kernel ELM will be given in Section V.

Huang [16] pointed out that the "redundant" b renders SVR sub-optimal compared with ELM if same kernels are both used in them, because the feasible solution space of SVR is a subset of ELM feasible solution space.

We shall indicate that the main difference between ELM and SVR is their different account of starting points. SVR [24] was developed at first as an extension of SVM. As mentioned above, SVM was designed for binary classification at first, and the subsequent variants for regression problems were developed on the basis of SVM without addressing the problem caused by b. By contrast, ELM was originally proposed for regression, the feature mappings $\mathbf{h}(\mathbf{x})$ are known, and universal approximation capability was considered at the first place. Thus, in ELM, the approximation error tends to be zero and b should not be present [16], [21], [23].

III. Robust ELM

A. Uncertainties of Input and Output Data

RELM is proposed under a stochastic framework. Assume that both input $\mathbf{x}$ and output data $\mathbf{t}$ are perturbed by noises. Since $\mathbf{H}$ is the feature space after nonlinear mapping from the input space, if the input data is contaminated, $\mathbf{H}$ is also mixed with disturbances. We follow from [25] to assume the disturbances in the feature space are additive:

$$\begin{aligned} \mathbf{h}(\mathbf{x}_i) &= \mathbf{h}(\mathbf{x}_i)_{\text{true}} + (\iota_1)_i \\ \mathbf{t}_i &= (\mathbf{t}_i)_{\text{true}} + (\iota_2)_i \end{aligned} \tag{9}$$

where $(\iota_1)_i$ and $(\iota_2)_i$ are uncorrelated perturbations in the feature space and output space with proper dimensions, respectively. The new vector $\mathbf{y}_i \in \mathbf{R}^{1\times(L+m)}$ is the ith input and ith output observation, i.e., $\mathbf{y}_i = [\mathbf{h}(\mathbf{x}_i), \mathbf{t}_i]$. And now we give the following definitions:

$$\begin{aligned} &\bar{\mathbf{h}}(\mathbf{x}_i) = \mathbf{E}(\mathbf{h}(\mathbf{x}_i)), \quad \bar{\mathbf{t}}_i = \mathbf{E}(\mathbf{t}_i) \\ &\Sigma_{hh}^i = \mathbf{Cov}(\mathbf{h}(\mathbf{x}_i), \mathbf{h}(\mathbf{x}_i)), \quad \Sigma_{tt}^i = \mathbf{Cov}(\mathbf{t}_i, \mathbf{t}_i) \end{aligned} \tag{10}$$

where $\mathbf{E}(\cdot)$ and $\mathbf{Cov}(\cdot)$ denote expectation and covariance operators for random variables, respectively. Since, the perturbations in the feature space $(\iota_1)_i$ and output space $(\iota_2)_i$ are uncorrelated, i.e., $\Sigma_{ht}^i = 0$, we have

$$\begin{aligned} \bar{\mathbf{y}}_i &= \mathbf{E}([\mathbf{h}(\mathbf{x}_i), \mathbf{t}_i]) = \left[\bar{\mathbf{h}}(\mathbf{x}_i), \bar{\mathbf{t}}_i\right] \\ \Sigma_{yy}^i &= \mathbf{Cov}(\mathbf{y}_i, \mathbf{y}_i) = \begin{bmatrix} \Sigma_{hh}^i & 0 \\ 0 & \Sigma_{tt}^i \end{bmatrix}_{(L+m)\times(L+m)}. \end{aligned} \tag{11}$$

[2] We particularly avoid including kernel ELM and its variants in the above gathering, given the fact that they do not possess the most significant property of ELM—random feature mapping.

The ith prediction error is denoted by $\mathbf{e}_i \in \mathbf{R}^{1\times m}$ and its expectation $\bar{\mathbf{e}}_i$ is defined as follows:

$$\mathbf{e}_i = \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} - \mathbf{t}_i, \bar{\mathbf{e}}_i = \bar{\mathbf{h}}(\mathbf{x}_i)\boldsymbol{\beta} - \bar{\mathbf{t}}_i. \tag{12}$$

It follows from [25] and [26] that, by inserting CTM and SR constraints into SVR, the predictions can be robust to perturbations in the data set.

CTM is a criterion on that we require the prediction errors to be insensitive to the distribution of the noises in input and output data

$$\Pr_{x_i,y_i}\{|e_i - \bar{e}_i| \geq \theta_i\} \leq \eta \quad i = 1, 2, \ldots, N \tag{13}$$

x_i, y_i here are the input and output data, and θ_i means the confidence threshold while η denotes the maximum tolerance of the deviation.

An alternative way to boost the robustness is restricting the residual to be small, which leads to the SR constraint

$$\Pr_{x_i,y_i}\{|e_i| \geq \xi_i + \varepsilon\} \leq \eta \tag{14}$$

where ξ_i corresponds to the prediction error and ε is a slack variable. Compared with the CTM constraint, the SR constraint requires the estimator to be robust in terms of deviations which lead to larger estimation error rather than centering. In fact, both CTM and SR constraints are robust constraints utilized to bound probabilities of highly deviated errors subject to second order moment constraints.

B. Sufficient Condition of CTM Constraint

It should be pointed out that, the above two robust constraints only consider a scalar output case, however, the outputs of IPSs are usually vectors. Moreover, ELM or kernel ELM algorithms are inherently different from SVR, therefore different constraints should be provided for our problem setting. We now give our CTM constraint for this paper

$$\Pr_{\mathbf{h}(\mathbf{x}_i),\mathbf{t}_i}\left\{\|\mathbf{e}_i - \bar{\mathbf{e}}_i\|^2 \geq \theta_i^2\right\} \leq \tau \quad i = 1, 2, \ldots, N \tag{15}$$

where θ_i is still a confidence threshold and τ here stands for some probability. Nevertheless, CTM constraints in this form are intractable. Multidimensional Chebyshev's inequality is leveraged to convert the original constraints into tractable ones.

Lemma 1 [27]: Let $\mathbf{z}$ be an m-dimensional random row vector with expected value $\bar{\mathbf{z}}$ and positive-definite covariance Σ, then

$$\Pr\left\{(\mathbf{z} - \bar{\mathbf{z}})\Sigma^{-1}(\mathbf{z} - \bar{\mathbf{z}})^T \geq \theta^2\right\} \leq \frac{m}{\theta^2}. \tag{16}$$

Proposition 1: For $\mathbf{z}$ and Σ defined in Lemma 1, if $\|\mathbf{z}\|^2 \geq \epsilon\|\Sigma\|$, then $\mathbf{z}\Sigma^{-1}\mathbf{z}^T \geq \epsilon$.

Proof: Since Σ is a real-valued symmetric matrix, it can be diagonalized as $\Sigma = P^{-1}\Lambda P$. Λ here is a real-valued matrix with eigenvalues of Σ on its diagonal. It can be shown that

$$\Lambda \leq \|\Sigma\|\mathbf{I} \Rightarrow \Lambda^{-1} \geq \|\Sigma\|^{-1}\mathbf{I} \tag{17}$$

which leads to

$$\mathbf{z}\Sigma^{-1}\mathbf{z}^T = \mathbf{z}P^{-1}\Lambda^{-1}P\mathbf{z}^T \geq \frac{\mathbf{z}\mathbf{z}^T}{\|\Sigma\|} \tag{18}$$

and (18) gives rise to

$$\|\mathbf{z}\|^2 \geq \epsilon\|\Sigma\| \Rightarrow \mathbf{z}\Sigma^{-1}\mathbf{z}^T \geq \epsilon. \tag{19}$$

■

Proposition 1 also implies

$$\Pr\left\{\|\mathbf{z}\|^2 \geq \epsilon\|\Sigma\|\right\} \leq \Pr\left\{\mathbf{z}\Sigma^{-1}\mathbf{z}^T \geq \epsilon\right\}. \tag{20}$$

Theorem 1: Let $\boldsymbol{\beta} \in \mathbf{R}^{L\times m}$ and $\boldsymbol{\omega} = [\boldsymbol{\beta}^T, -\mathbf{1}]^T \in \mathbf{R}^{(L+m)\times m}$ and Σ_{yy}^i is defined in (11), then a sufficient condition for (15) is

$$\left\|\left(\Sigma_{yy}^i\right)^{\frac{1}{2}}\boldsymbol{\omega}\right\| \leq \theta_i\sqrt{\tau/m} \tag{21}$$

where $-\mathbf{1}$ is a vector of all entries of -1 with proper length.

Proof: Substitute $\mathbf{e}_i, \theta_i$ for $\mathbf{z}, \theta$ into (16), we have

$$\Pr_{\mathbf{h}(\mathbf{x}_i),\mathbf{t}_i}\left\{(\mathbf{e}_i - \bar{\mathbf{e}}_i)\left(\Sigma_{ee}^i\right)^{-1}(\mathbf{e}_i - \bar{\mathbf{e}}_i)^T \geq \theta_i^2\right\} \leq \frac{m}{\theta_i^2} \tag{22}$$

which together with (20), leads to

$$\begin{aligned}\Pr_{\mathbf{h}(\mathbf{x}_i),\mathbf{t}_i}&\left\{\|\mathbf{e}_i - \bar{\mathbf{e}}_i\|^2 \geq \theta_i^2\right\}\\ &\leq \Pr_{\mathbf{h}(\mathbf{x}_i),\mathbf{t}_i}\left\{(\mathbf{e}_i - \bar{\mathbf{e}}_i)\left(\Sigma_{ee}^i\right)^{-1}(\mathbf{e}_i - \bar{\mathbf{e}}_i)^T \geq \frac{\theta_i^2}{\|\Sigma_{ee}^i\|}\right\}\\ &\leq \frac{m\|\Sigma_{ee}^i\|}{\theta_i^2}.\end{aligned} \tag{23}$$

Thus, $m\|\Sigma_{ee}^i\|/\theta_i^2 \leq \tau$ is a sufficient condition for (15). By taking into account that

$$\Sigma_{ee}^i = \boldsymbol{\omega}^T\Sigma_{yy}^i\boldsymbol{\omega} \tag{24}$$

inserting (24) into $m\|\Sigma_{ee}^i\|/\theta_i^2 \leq \tau$ and then taking the square root on both sides, (21) follows. ■

C. Sufficient Condition of SR Constraint

The sufficient condition of SR constraint can be derived in the same fashion. The SR constraint in our case is

$$\Pr_{\mathbf{h}(\mathbf{x}_i),\mathbf{t}_i}\left\{\|\mathbf{e}_i\|^2 \geq (\xi_i + \varepsilon)^2\right\} \leq \tau \quad i = 1, 2, \ldots, N. \tag{25}$$

Theorem 2: Let $\boldsymbol{\beta} \in \mathbf{R}^{L\times m}$, $\boldsymbol{\omega} = [\boldsymbol{\beta}^T, -\mathbf{1}]^T \in \mathbf{R}^{(L+m)\times m}$ and Σ_{yy}^i is defined in (11), then a sufficient condition for (25) is

$$\left\|\begin{matrix}\left(\Sigma_{yy}^i\right)^{\frac{1}{2}}\boldsymbol{\omega}\\ \bar{\mathbf{h}}(\mathbf{x}_i)\boldsymbol{\beta} - \bar{\mathbf{t}}_i\end{matrix}\right\| \leq (\xi_i + \varepsilon)\sqrt{\tau/m} \tag{26}$$

where $-\mathbf{1}$ is a vector of all entries of -1 with proper length.

Proof: Taking $\mathbf{e}_i\mathbf{e}_i^T \in \mathbf{R}$ as a random variable, from Markov's inequality, we have

$$\begin{aligned}\Pr_{\mathbf{h}(\mathbf{x}_i),\mathbf{t}_i}\left\{\|\mathbf{e}_i\|^2 \geq (\xi_i + \varepsilon)^2\right\} &= \Pr_{\mathbf{h}(\mathbf{x}_i),\mathbf{t}_i}\left\{\mathbf{e}_i\mathbf{e}_i^T \geq (\xi_i + \varepsilon)^2\right\}\\ &\leq \frac{\mathbf{E}(\mathbf{e}_i\mathbf{e}_i^T)}{(\xi_i + \varepsilon)^2}.\end{aligned}$$

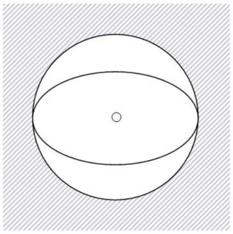

Fig. 1. Shadow area indicates the possible region the random variable may fall into.

Denote $\mathrm{tr}(\cdot)$ as the trace operator of a matrix

$$\begin{aligned}\mathbf{E}\left(\mathbf{e}_i\mathbf{e}_i^T\right) &= \mathbf{E}\left\{\mathrm{tr}\left(\mathbf{e}_i^T\mathbf{e}_i\right)\right\}\\ &= \mathbf{E}\left\{\mathrm{tr}\left(\mathbf{e}_i^T\mathbf{e}_i - \bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i\right)\right\} + \mathrm{tr}\left(\bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i\right)\\ &= \mathrm{tr}\left(\Sigma_{ee}^i + \bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i\right). \qquad (27)\end{aligned}$$

Since Σ_{ee}^i and $\bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i$ are both positive semi-definite, which implies that $\Sigma_{ee}^i + \bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i$ is positive semi-definite. Since

$$\left\|\Sigma_{ee}^i + \bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i\right\| = \max\{\lambda_1, \ldots, \lambda_m\} \qquad (28)$$

where λ_i stands for an eigenvalue of $\Sigma_{ee}^i + \bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i$, we have

$$\mathrm{tr}\left(\Sigma_{ee}^i + \bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i\right) \le m\left\|\Sigma_{ee}^i + \bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i\right\| \qquad (29)$$

which leads to

$$m\left\|\Sigma_{ee}^i + \bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i\right\| = m\left\|\begin{matrix}\left(\Sigma_{yy}^i\right)^{\frac{1}{2}}\boldsymbol{\omega}\\ \bar{\mathbf{h}}(\mathbf{x}_i)\boldsymbol{\beta} - \bar{\mathbf{t}}_i\end{matrix}\right\|^2. \qquad (30)$$

By letting

$$\frac{m}{(\xi_i+\varepsilon)^2}\left\|\begin{matrix}\left(\Sigma_{yy}^i\right)^{\frac{1}{2}}\boldsymbol{\omega}\\ \bar{\mathbf{h}}(\mathbf{x}_i)\boldsymbol{\beta} - \bar{\mathbf{t}}_i\end{matrix}\right\|^2 \le \tau \qquad (31)$$

and taking square root on both sides, we claim that (26) is a sufficient condition for (25). ■

D. Geometric Interpretation

The geometric interpretations of the above claims are as follows:

1) Proposition 1 can be interpreted as that the chance of a random variable lying outside a sphere with radius $\sqrt{\epsilon\|\Sigma\|}$ is greater than that of a random variable lying outside an ellipsoid with radius $\sqrt{\epsilon}$ and covariance matrix Σ. This is intuitive because the largest length of semi-axe of the ellipsoid is equal to the radius of the sphere and they share the same center. Fig. 1 shows the illustration when the ellipsoid and sphere are projected onto a 2-D space.

2) The above CTM robust criterion can be understood as a restriction that each training data $\mathbf{y}_i$ picked from the ellipsoid $\Psi_i(\bar{\mathbf{y}}_i, \Sigma_{yy}^i, (m/\tau)^{1/2})$ satisfies the inequality

$$\|\mathbf{e}_i - \bar{\mathbf{e}}_i\| \le \theta_i \qquad (32)$$

where

$$\Psi_i\left(\bar{\mathbf{y}}_i, \Sigma_{yy}^i, \sqrt{\frac{m}{\tau}}\right) \triangleq \left\{\mathbf{y}_i \middle| \left(\mathbf{y}_i - \bar{\mathbf{y}}_i\right)\left(\Sigma_{yy}^i\right)^{-1}\left(\mathbf{y}_i - \bar{\mathbf{y}}_i\right)^T \le \frac{m}{\tau}\right\}. \qquad (33)$$

From Theorem 1, we have

$$\sqrt{m/\tau}\left\|\Sigma_{yy}^{i\ \frac{1}{2}}\boldsymbol{\omega}\right\| \le \theta_i. \qquad (34)$$

Further, by noting that

$$\begin{aligned}\|\mathbf{e}_i - \bar{\mathbf{e}}_i\| &= \|(y_i - \bar{y}_i)\,\boldsymbol{\omega}\|\\ &= \left\|(y_i - \bar{y}_i)\,\Sigma_{yy}^{i\ -\frac{1}{2}}\Sigma_{yy}^{i\ \frac{1}{2}}\boldsymbol{\omega}\right\|\\ &\le \left\|(y_i - \bar{y}_i)\,\Sigma_{yy}^{i\ -\frac{1}{2}}\right\|\left\|\Sigma_{yy}^{i\ \frac{1}{2}}\boldsymbol{\omega}\right\|\\ &\le \sqrt{\frac{m}{\tau}}\left\|\Sigma_{yy}^{i\ \frac{1}{2}}\boldsymbol{\omega}\right\|. \qquad (35)\end{aligned}$$

It is obvious that the above geometric interpretation for the CTM constraint holds.

3) A similar geometric interpretation can be given for the SR constraint. Let

$$\widehat{\Sigma}_{yy}^i = \Sigma_{yy}^i + \mathbf{y}_i^T\mathbf{y}_i \qquad (36)$$

a SR constraint enforces each training data $\mathbf{y}_i$ picked from the ellipsoid $\Psi_i(0, \widehat{\Sigma}_{yy}^i, \sqrt{m/\tau})$

$$\Psi_i\left(0, \widehat{\Sigma}_{yy}^i, \sqrt{\frac{m}{\tau}}\right) \triangleq \left\{\mathbf{y}_i|\mathbf{y}_i\left(\widehat{\Sigma}_{yy}^i\right)^{-1}\mathbf{y}_i^T \le \frac{m}{\tau}\right\} \qquad (37)$$

satisfies the following inequality:

$$\|\mathbf{e}_i\| \le \xi_i + \varepsilon. \qquad (38)$$

The procedure to verify this interpretation is in the same fashion of the CTM case

$$\begin{aligned}\|\mathbf{e}_i\| &= \|y_i\boldsymbol{\omega}\|\\ &= \left\|y_i\widehat{\Sigma}_{yy}^{i\ -\frac{1}{2}}\widehat{\Sigma}_{yy}^{i\ \frac{1}{2}}\boldsymbol{\omega}\right\|\\ &\le \left\|y_i\widehat{\Sigma}_{yy}^{i\ -\frac{1}{2}}\right\|\left\|\widehat{\Sigma}_{yy}^{i\ \frac{1}{2}}\boldsymbol{\omega}\right\|\\ &\le \sqrt{\frac{m}{\tau}}\left\|\widehat{\Sigma}_{yy}^{i\ \frac{1}{2}}\boldsymbol{\omega}\right\|. \qquad (39)\end{aligned}$$

From Theorem 2, we have

$$\begin{aligned}\left\|\begin{matrix}\left(\Sigma_{yy}^i\right)^{\frac{1}{2}}\boldsymbol{\omega}\\ \bar{\mathbf{h}}(\mathbf{x}_i)\boldsymbol{\beta} - \bar{\mathbf{t}}_i\end{matrix}\right\|^2 &= \left\|\Sigma_{ee}^i + \bar{\mathbf{e}}_i^T\bar{\mathbf{e}}_i\right\|\\ &= \left\|\boldsymbol{\omega}^T\left(\Sigma_{yy}^i + \mathbf{y}_i^T\mathbf{y}_i\right)\boldsymbol{\omega}\right\|\\ &= \left\|\boldsymbol{\omega}^T\left(\Sigma_{yy}^i + \mathbf{y}_i^T\mathbf{y}_i\right)\boldsymbol{\omega}\right\|\\ &\le \frac{\tau}{m}(\xi_i+\varepsilon)^2. \qquad (40)\end{aligned}$$

Taking square roots of (40) yields

$$\left\|\left(\widehat{\Sigma}_{yy}^i\right)^{\frac{1}{2}}\boldsymbol{\omega}\right\| \le \sqrt{\frac{\tau}{m}}(\xi_i+\varepsilon) \qquad (41)$$

which together with (39) implies

$$\|\mathbf{e}_i\| \le \xi_i + \varepsilon. \qquad (42)$$

IV. Robust ELM for Regression

Based on the preliminary results of last section, we now formulate CTM-constrained RELM (CTM-RELM) and SR-constrained RELM (SR-RELM) for noisy input and output data.

A. CTM-Based RELM

By adding second order moment constraints to the basic ELM formulation in Theorem 1, the CTM-RELM is formulated as

$$\begin{aligned} \min_{\boldsymbol{\beta},b,\theta,\boldsymbol{\xi}} \quad & L_P = b + C\sum_{i=1}^{N}\xi_i + D\sum_{i=1}^{N}\theta_i \\ \text{s.t.} \quad & \|\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} - \mathbf{t}_i\| \le \varepsilon + \xi_i \\ & \left\| \left(\Sigma_{yy}^{i}\right)^{\frac{1}{2}} \boldsymbol{\omega} \right\| \le \theta_i\sqrt{\tau/m} \\ & \xi_i \ge 0 \quad i = 1, 2, \ldots, N \\ & \|\boldsymbol{\beta}\| \le b \end{aligned} \tag{43}$$

where C is defined in (7), and D is a penalty coefficient to control the deviation of the prediction errors.

B. SR-Based RELM

Likewise, Theorem 2 also leads to a SOCP problem formulation

$$\begin{aligned} \min_{\boldsymbol{\beta},b,\boldsymbol{\xi}} \quad & L_P = b + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & \left\| \begin{matrix} \left(\Sigma_{yy}^{i}\right)^{\frac{1}{2}} \boldsymbol{\omega} \\ \bar{\mathbf{h}}(\mathbf{x}_i)\boldsymbol{\beta} - \bar{\mathbf{t}}_i \end{matrix} \right\| \le (\xi_i + \varepsilon)\sqrt{\tau/m} \\ & \xi_i \ge 0 \quad i = 1, 2, \ldots, N \\ & \|\boldsymbol{\beta}\| \le b. \end{aligned} \tag{44}$$

V. Kernelization for RELMs

As discussed in Section II-C, the kernel trick is adopted in SVR. In fact, the kernel trick can also be applied to ELM. We have indicated that, the explicit nonlinear feature mapping with random hidden nodes in ELM can bring about some advantages compared to SVR. Nevertheless, it does not mean that the kernel trick is useless for ELM. In reality, the capability of universal approximation of ELM can not be fulfilled due to the curse of dimensionality. Kernel methods enable access to the corresponding very high-dimensional, even infinite-dimensional, feature spaces at a low computational cost both in space and time [28]. In the case of a Gaussian kernel, the feature map lives in an infinite dimensional space, i.e., it has infinite number of hidden nodes L, which enables ELM to work as universal approximator [18]. Some related works have adopted the kernel method in ELM and produce desirable results [23], [29].[3] In this section, we slightly modify CTM and SR constraints and then incorporate them into the kernelized formulations of RELMs.

It follows from [23] that the optimal weight matrix $\boldsymbol{\beta}$ in ELM has the form:

$$\boldsymbol{\beta} = \mathbf{H}^T\mathbf{P} \tag{45}$$

where $\mathbf{P} \in \mathbf{R}^{N\times m}$. Once the model, i.e., β, is determined, we can make predictions by

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} = \sum_{i=1}^{N}\mathbf{h}(\mathbf{x})\mathbf{h}(\mathbf{x}_i)^T\mathbf{P}_i. \tag{46}$$

Based on the definition of ELM kernel, we have

$$f(\mathbf{x}) = \sum_{i=1}^{N} k(\mathbf{x}, \mathbf{x}_i)\mathbf{P}_i \tag{47}$$

where $k(\cdot,\cdot)$ is a kernel function. The kernel matrix of ELM is defined as [16]

$$\mathbf{K} = \mathbf{H}\mathbf{H}^T : \mathbf{K}_{i,j} = \mathbf{h}(\mathbf{x}_i)\cdot\mathbf{h}\left(\mathbf{x}_j\right)^T = k\left(\mathbf{x}_i, \mathbf{x}_j\right) \tag{48}$$

when the number of training samples is n, $\mathbf{K} \in \mathbf{R}^{N\times N}$.

The intrinsic modularity of kernel machines also means that any kernel function can be used provided it produces symmetric, positive semi-definite kernel matrices [28]. In our case, we restrict $\mathbf{K}$ not only to satisfy the modularity but also have all of its entries being real numbers. Thus, we can decompose $\mathbf{K}$ in such way

$$\mathbf{K} = \mathbf{K}^{\frac{1}{2}}\mathbf{K}^{\frac{1}{2}} \tag{49}$$

where $\mathbf{K}^{1/2}$ is real symmetric. From (45) and (48), we get

$$\boldsymbol{\beta}^T\boldsymbol{\beta} = \mathbf{P}^T\mathbf{K}\mathbf{P} = \left(\mathbf{K}^{\frac{1}{2}}\mathbf{P}\right)^T\mathbf{K}^{\frac{1}{2}}\mathbf{P} \tag{50}$$

which leads to $\|\boldsymbol{\beta}\| = \|\mathbf{K}^{1/2}\mathbf{P}\|$.

We now give the kernelized CTM constraint

$$\left\|\Sigma_{yy}^{i}\right\|^{\frac{1}{2}} \left\| \begin{matrix} \mathbf{K}^{\frac{1}{2}}\mathbf{P} \\ -\mathbf{1} \end{matrix} \right\| \le \theta_i\sqrt{\tau/m} \quad i = 1, 2, \ldots, N \tag{51}$$

where $-\mathbf{1}$ is a matrix of all entries of -1 with the dimension of $m \times m$. Note that (51) is a sufficient condition of (21) since

$$\left\|\left(\Sigma_{yy}^{i}\right)^{\frac{1}{2}}\boldsymbol{\omega}\right\| \le \left\|\Sigma_{yy}^{i}\right\|^{\frac{1}{2}}\|\boldsymbol{\omega}\| \le \left\|\Sigma_{yy}^{i}\right\|^{\frac{1}{2}} \left\| \begin{matrix} \mathbf{K}^{\frac{1}{2}}\mathbf{P} \\ -\mathbf{1} \end{matrix} \right\| \tag{52}$$

where $\boldsymbol{\omega} = [\boldsymbol{\beta}^T, -\mathbf{1}]^T$, and the kernelized CTM-RELM is of the form as

$$\begin{aligned} \min_{\mathbf{P},b,\theta,\boldsymbol{\xi}} \quad & L_P = b + C\sum_{i=1}^{N}\xi_i + D\sum_{i=1}^{N}\theta_i \\ \text{s.t.} \quad & \|\mathbf{K}_{i,:}\mathbf{P} - \mathbf{t}_i\| \le \varepsilon + \xi_i \\ & \left\|\Sigma_{yy}^{i}\right\|^{\frac{1}{2}} \left\| \begin{matrix} \mathbf{K}^{\frac{1}{2}}\mathbf{P} \\ -\mathbf{1} \end{matrix} \right\| \le \theta_i\sqrt{\tau/m} \\ & \xi_i \ge 0 \quad i = 1, 2, \ldots, N \\ & \left\|\mathbf{K}^{\frac{1}{2}}\mathbf{P}\right\| \le b. \end{aligned} \tag{53}$$

[3] For terminology consistency, we use kernel ELM to refer to the kernel trick-based ELM and its variants.

A similar fashion can be adopted to derive the kernelized SR-RELM formulation

$$\begin{aligned} \min_{\mathbf{P},b,\xi} \quad & L_P = b + C\sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \left\| \begin{matrix} \left\|\Sigma_{yy}^i\right\|^{\frac{1}{2}} \begin{bmatrix} \mathbf{K}^{\frac{1}{2}}\mathbf{P} \\ -\mathbf{1} \end{bmatrix} \\ \mathbf{K}_{i,:}\mathbf{P} - \mathbf{t}_i \end{matrix} \right\| \\ & \le (\xi_i + \varepsilon)\sqrt{\tau/m} \\ & \xi_i \ge 0 \quad i = 1, 2, \ldots, N \\ & \left\|\mathbf{K}^{\frac{1}{2}}\mathbf{P}\right\| \le b. \end{aligned} \tag{54}$$

VI. Covariance in the Feature Space

We firstly calculate the covariance when the nonlinear mapping functions are known explicitly. We write $\mathbf{h}(\mathbf{x})$ as follows:

$$\mathbf{h}(\mathbf{x}) = [G(\mathbf{a}_1, b, \mathbf{x}), \ldots, G(\mathbf{a}_L, b, \mathbf{x})] \tag{55}$$

where $\mathbf{a}_i$, b are randomly generated weights and bias connecting an input and the ith hidden node. $G(\mathbf{a}_i, b, \mathbf{x})$ is the activation function.

A statistical method is provided to derive the covariance theoretically in the feature space. For each input $\mathbf{x}_i$, we randomly generate Z samples $\{\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^Z\}$ according to the distribution of $\mathbf{x}_i$ with mean $\bar{\mathbf{x}}_i$ and covariance Σ_{xx}^i. Then the covariance matrix of $\mathbf{h}(\mathbf{x}_i)$ can be approximated by

$$\Sigma_{hh}^i = \frac{1}{Z}\sum_{z=1}^{Z} \tilde{\mathbf{h}}\left(\mathbf{x}_i^z\right)^T \tilde{\mathbf{h}}\left(\mathbf{x}_i^z\right) \tag{56}$$

where

$$\tilde{\mathbf{h}}\left(\mathbf{x}_i^z\right) = \mathbf{h}\left(\mathbf{x}_i^z\right) - \frac{1}{Z}\sum_{z=1}^{Z} \mathbf{h}\left(\mathbf{x}_i^z\right). \tag{57}$$

However, the covariance in the kernel case is more delicate and cannot be derived explicitly. Note that, in the kernelized cases of (53) and (54), only the norm of covariance Σ_{yy}^i is needed, that is

$$\Sigma_{yy}^i = \left\| \begin{matrix} \Sigma_{hh}^i & 0 \\ 0 & \Sigma_{tt}^i \end{matrix} \right\| = \max\left\{\left\|\Sigma_{hh}^i\right\|, \left\|\Sigma_{tt}^i\right\|\right\}. \tag{58}$$

$\|\Sigma_{tt}^i\|$ can be readily calculated, and we now give a solution to approximate $\|\Sigma_{hh}^i\|$. The $\mathcal{L}2$-norm of real symmetric matrix Σ_{hh}^i equals its largest eigenvalue. Let λ and $\mathbf{v}$ be an eigenvalue and its corresponding eigenvector

$$\lambda\mathbf{v} = \Sigma_{hh}^i\mathbf{v}. \tag{59}$$

It is been proved in [30] that λ of Σ_{yy}^i also satisfies

$$Z\lambda\boldsymbol{\alpha} = \tilde{\mathbf{K}}^i\boldsymbol{\alpha} \tag{60}$$

where $\tilde{\mathbf{K}}^i = \mathbf{K}^i - \mathbf{L}\mathbf{K}^i - \mathbf{K}^i\mathbf{L} + \mathbf{L}\mathbf{K}^i\mathbf{L}$ and $\mathbf{L} \in \mathbf{R}^{Z\times Z}$ with each entry $\mathbf{L}_{i,j} = 1/Z$. Here, the $Z \times Z$ matrix $\mathbf{K}$ is defined by

$$\mathbf{K}_{i,j}^i := k\left(\mathbf{x}_i^i, \mathbf{x}_i^j\right) = \left(\mathbf{h}\left(\mathbf{x}_i^i\right) \cdot \mathbf{h}\left(\mathbf{x}_i^j\right)\right). \tag{61}$$

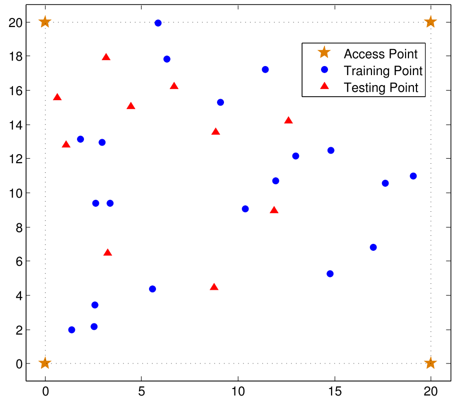

Fig. 2. Positions of the WiFi AP, offline calibration points, and online testing points in the simulated field.

Hence, we can compute the $\mathcal{L}_2$-norm of Σ_{hh}^i from the set of eigenvalues of $\tilde{\mathbf{K}}^i$

$$\left\|\Sigma_{hh}^i\right\| = \frac{1}{Z}\max\left\{\lambda\left(\tilde{\mathbf{K}}^i\right)\right\} \tag{62}$$

where $\lambda(\tilde{\mathbf{K}}^i)$ is the set of all the eigenvalues of $\tilde{\mathbf{K}}^i$.

VII. Performance Verification

A. Simulation Results and Evaluation

We develop a simulation environment using MATLAB R2013a in order to evaluate the performance of our proposed algorithms before any real-world experiment is conducted. As shown in Fig. 2, we assume a 20 × 20 m room where four WiFi APs are installed at the four corners of the room. The most commonly used path loss model for indoor environment is the ITU indoor propagation model [31]. Since it provides a relation between the total path loss PL (dBm) and distance d (m), it is adopted to simulate the WiFi signal generated from each WiFi AP. The indoor path loss model can be expressed as

$$\mathrm{PL}(d) = \mathrm{PL}_0 - 10\alpha\log(d) + X \tag{63}$$

where PL_0 is the path loss coefficient and it is set to be -40 dBm in our simulation. α is the path loss exponent and X represents some random noises.

The distribution of RSS indication from four real-world APs in our IPS is illustrated in Fig. 3. As shown in Fig. 3, the signals collected by one AP can be quite different even at a same location due to noises and outliers. Therefore, four different types of data with disturbances are generated based on (63), i.e., data mixed with the Gaussian noise $X \sim \mathcal{N}(0, 1)$, data mixed with the student's noise $X \sim \mathcal{T}(0, 1, 1)$, data mixed with the gamma noise $X \sim Ga(1, 1)$ and data contaminated by one-sided outliers (20% contamination rate),[4] to test the performance of RELMs. To make our simulation more practical, 100 testing samples are artificially generated at each training point and testing point, respectively using (63) with different perturbations.

We apply our RELMs to the simulated data, and compare our proposed algorithms with basic ELM, OPT-ELM,

[4] The strategy of adding outliers here is similar to the one of [13].

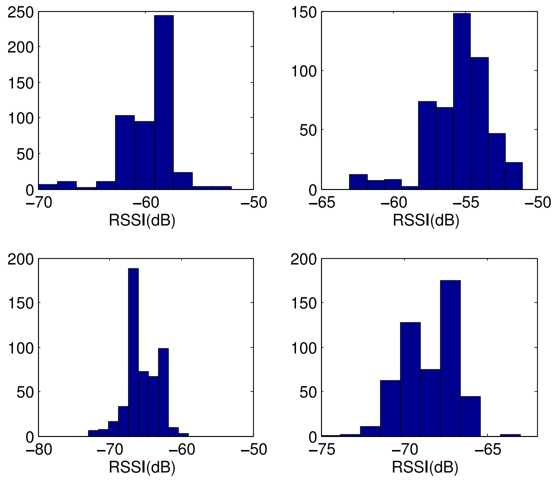

Fig. 3. RSS index of distribution of four APs at one position.

kernel ELM, and SVR [32]. In the CTM-RELM formulation, there are three hyperparameters, C, D, and τ to be tuned. C and D are both selected by grid method from the exponential sequence $[2^{-5}, 2^{-4}, \ldots, 2^{5}]$ utilizing fivefold cross-validation on the training data set. τ increases from 0.1 to 1 with a step size of 0.1. In SR-RELM case, there are two hyper-parameters, C and τ to be tuned, they are all selected with the same strategy as CTM-RELM. For both RELMs, the slack variable ε is empirically selected as 0.05. The SOCP problems are solved by CVX MATLAB toolbox [33]. Since the performances of ELM and its variants are not sensitive to the number of hidden nodes L as long as it is larger than some threshold [23], we fix L as 500 for our proposed algorithms, basic ELM and OPT-ELM to facilitate the comparison of computational costs. The width of Gaussian kernel λ used in SVR and kernel ELM are selected from the exponential sequence $[2^{-5}, 2^{-4}, \ldots, 2^{5}]$ utilizing fivefold cross-validation.

Four performance measures are introduced: mean root square error (MRSE), standard deviation (STD), WCE, and REP over r repeated realizations. Noted that MRSE, STD, and WCE in this case are taken from the mean over the r repeated realizations. REP is measured by the deviation of the MRSE over the repeated realizations, and this measure is proposed based on the fact that ELM with same parameters, e.g., the number of hidden nodes, in the same training data set may draw quite different results. r in our experiment is selected as 30

$$\mathrm{MRSE} = \frac{1}{r}\sum_{j=1}^{r}\left(\frac{1}{s}\sum_{i=1}^{s}\left\|\mathbf{t}_i - \mathbf{h}_i\hat{\boldsymbol{\beta}}\right\|\right)_j$$

$$\mathrm{STD} = \frac{1}{r}\sum_{j=1}^{r}\left(\sqrt{\sum_{i=1}^{s}\left(\left\|\mathbf{t}_i - \mathbf{h}_i\hat{\boldsymbol{\beta}}\right\| - \frac{1}{s}\sum_{i=1}^{s}\left\|\mathbf{t}_i - \mathbf{h}_i\hat{\boldsymbol{\beta}}\right\|\right)^2}\right)_j$$

$$\mathrm{WCE} = \frac{1}{r}\sum_{j=1}^{r}\left(\max_{i\in S}\left\|\mathbf{t}_i - \mathbf{h}_i\hat{\boldsymbol{\beta}}\right\|\right)_j$$

$$\mathrm{REP} = \sqrt{\frac{1}{r}\sum_{j=1}^{r}\left(\frac{1}{s}\sum_{i=1}^{s}\left\|\mathbf{t}_i - \mathbf{h}_i\hat{\boldsymbol{\beta}}\right\| - \mathrm{MRSE}\right)_j^2}$$

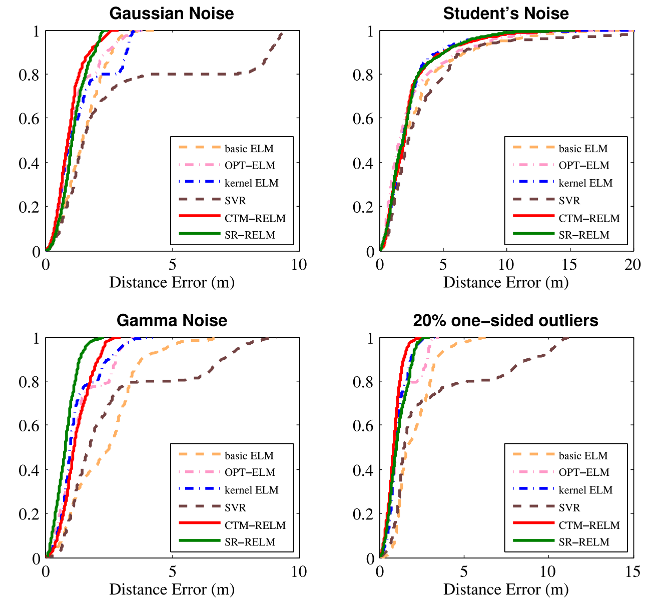

Fig. 4. Cumulative percentile of error distance for simulation data sets.

where s is the number of testing samples, S is the index set of testing samples like $[1, 2, \ldots, s]$.

As shown in Fig. 4, the proposed two algorithms outperform the other four algorithms in terms of accuracy and WCE. More exact number can be found in Table II, from which we see that the REP of the RELMs-based systems is improved compared with basic ELM and OPT-ELM-based ones. The enhancement of the REP is due to more constraints brought in our algorithms, which shrinks the size of solution searching space. Note that, the shrinking happening here is different from the one discussed in [21], in which the loss of solution searching freedom of SVR is caused by the redundant b [16].

B. Evaluation in Real-World IPSs

The system architecture of our WiFi-based IPS is shown in Fig. 5. The main components of this system consist of the existing commercial WiFi APs, mobile devices with WiFi function, a location server and a web-based monitoring system. The following is a brief operation procedure of our WiFi-based IPS. First of all, a data collection App for android devices was developed. After the mobile device turns on the WiFi module, it can collect RSS information from different APs every second and sends this information to a location server. The responsibility of the location server is to analyze the RSS, and calculate the estimated position of the mobile device. Then, the user can obtain his or her real time position through our web-based monitoring system directly on his or her mobile device.

We conducted real-world indoor localization experiments to evaluate the performance of the proposed RELM approaches. The testbed is the Internet of Things Laboratory in the School of Electrical and Electronic Engineering, Nanyang Technological University. The area of the test-bed is around 580 m^2 (35.1 × 16.6 m).

TABLE II
COMPARISON OF SIMULATION RESULTS

Performance measures	MRSE (m)	STD (m)	WCE (m)	REP (m)	Training Time (s)	Testing Time (s)
Gaussian Noise						
Basic ELM	1.87	0.79	4.58	0.37	0.48	8.9×10^{-3}
OPT-ELM	1.14	1.02	3.75	0.26	1.46	1×10^{-2}
Kernel ELM	1.22	0.77	3.81	0	0.28	4.5×10^{-2}
SVR	2.97	2.97	9.49	0.08	422.72	2.2×10^{-1}
CTM-RELM	0.92	0.52	2.88	0.07	93.74	6.3×10^{-3}
SR-RELM	1.13	0.58	2.47	0.04	9.09	7.4×10^{-3}
Student's Noise						
Basic ELM	3.09	3.04	23.85	0.25	0.24	6.7×10^{-3}
OPT-ELM	2.60	2.79	19.83	0.31	1.35	1×10^{-2}
Kernel ELM	2.42	2.47	21.13	0	0.31	4.8×10^{-2}
SVR	3.80	4.33	44.37	0.015	366.12	2.3×10^{-1}
CTM-RELM	2.45	2.28	15.86	0.07	94.41	8.3×10^{-3}
SR-RELM	2.39	2.11	15.69	0.063	9.05	7.8×10^{-3}
Gamma Noise						
Basic ELM	2.33	1.35	6.95	0.32	0.21	7.1×10^{-3}
OPT-ELM	1.33	0.88	3.52	0.27	1.45	1.1×10^{-2}
Kernel ELM	1.24	0.80	4.24	0	0.30	4.6×10^{-2}
SVR	2.69	2.34	9.13	0.04	375.21	2.6×10^{-1}
CTM-RELM	1.23	0.59	2.98	0.06	84.98	7.5×10^{-3}
SR-RELM	0.91	0.43	2.33	0.055	9.49	7.2×10^{-3}
20% one-sided outliers						
Basic ELM	2.08	1.67	8.74	0.38	0.25	6.9×10^{-3}
OPT-ELM	1.28	1.04	3.93	0.30	1.42	9.4×10^{-3}
Kernel ELM	1.08	0.55	3.31	0	0.28	4.1×10^{-2}
SVR	3.03	3.15	11.25	0.053	374.90	2.7×10^{-1}
CTM-RELM	0.95	0.45	2.92	0.05	80.45	6.2×10^{-3}
SR-RELM	1.11	0.61	2.96	0.061	8.84	7.8×10^{-3}

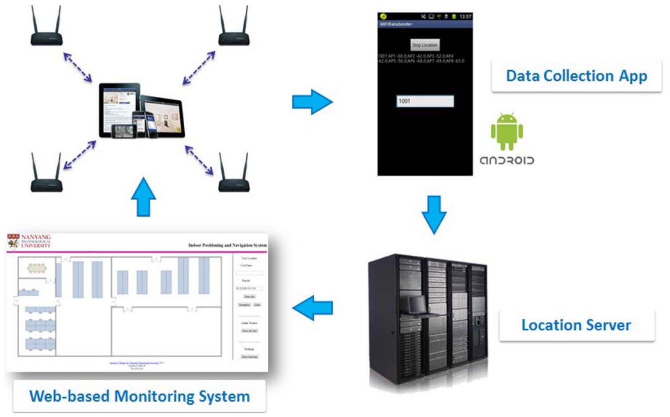

Fig. 5. System architecture of our WiFi-based IPS.

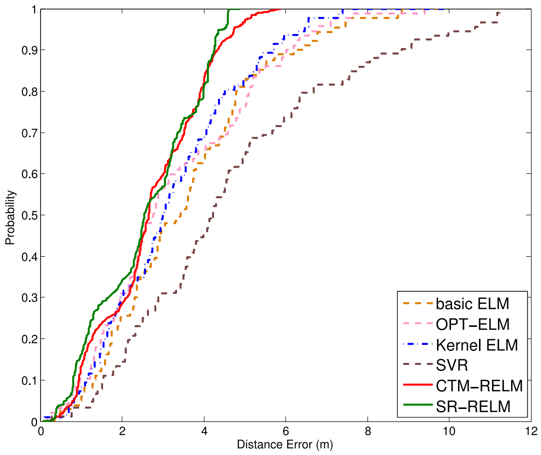

Fig. 6. Cumulative percentile of error distance for IPS testing results.

The layout of the testbed is shown in Fig. 7. Eight D-link DIR-605L WiFi cloud routers are utilized as WiFi APs for our experiments. The Android application is installed on a Samsung I929 Galaxy SII mobile phone. All the WiFi RSS fingerprints at offline calibration points and online testing points are collected using this phone for performance evaluation.

The RELM model was built up by the following steps. During the offline phase, 30 offline calibration points were selected and 200 WiFi RSS fingerprints were collected at each point. The positions of these 30 offline calibration points are demonstrated in Fig. 7. By leveraging these 6000 WiFi RSS fingerprints and their physical positions as training inputs and training targets (outputs) accordingly, the RELM model was constructed. During the online phase, we continued to collect WiFi RSS fingerprints at online testing points for five days. On each day, two distinct online testing points were selected in order to reflect the environmental dynamics. The positions

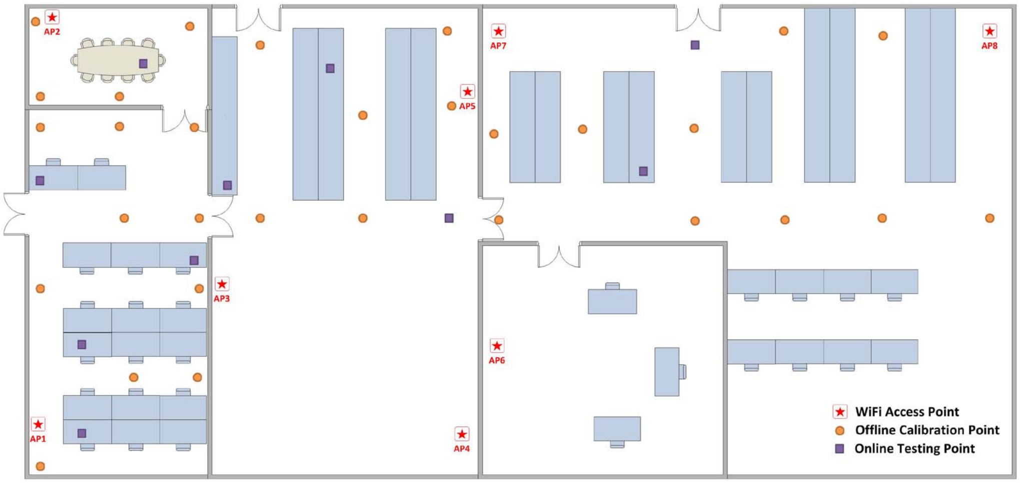

Fig. 7. Positions of the WiFi APs, offline calibration points, and online testing points in the test-bed.

TABLE III
COMPARISON OF EXPERIMENTAL TESTING RESULTS

Performance measures	MRSE (m)	STD (m)	WCE (m)	REP (m)	Training Time (s)	Testing Time (s)
Basic ELM	3.71	3.84	8.22	0.79	1.13	7.1×10^{-3}
OPT-ELM	3.32	1.98	9.65	0.56	2.84	8.8×10^{-3}
Kernel ELM	3.21	2.60	10.06	0	1.44	2.2×10^{-1}
SVR	4.66	2.64	10.93	0.024	4137.68	6.1×10^{-1}
CTM-RELM	2.96	1.26	5.61	0.12	141.65	9.7×10^{-3}
SR-RELM	2.94	1.53	4.47	0.22	18.88	9.0×10^{-3}

of these ten online testing points are also presented in Fig. 7. Two hundred WiFi RSS fingerprints are collected at each point. The parameter setting for the proposed and compared algorithms in this experiment is similar with the one introduced in Section VII-A, apart from the number of hidden units, which is set to 1000.

The testing results with respect to four performance measures given in Section VII-A are shown in Table III. Fig. 6 illustrates the comparison in terms of cumulative percentile of error distance, which shows that the proposed CTM-RELM can provide higher accuracy and have an obvious effect in reducing the STD compared to ELM and OPT-ELM. On the other hand, SR-RELM also gives an accuracy as good as CTM-RELM, and has better performance of confining the WCE. The above results are reasonable, since the two robust constraints have their different emphasis. In addition, both CTM-RELM and SR-RELM can give better performances in REP than basic ELM.

The proposed algorithms incur longer training time due to the introduction of second order moment constraints instead of linear constraints. However, a slightly longer training time is not a concern in IPSs, considering that it is the calibration phase, e.g., procedure of radio map generating, that accounts for the large body of time consumption. Besides, RELMs inherit the simpleness, e.g., random feature mapping, dispensation with bias b, and single layer structure from ELM, therefore its training time is still competitive compared with SVR and its variants.

VIII. CONCLUSION

Before concluding this paper, we provide some important discussions.

1) *Choice of the Measure for Accuracy:* It is noteworthy that, we adopt MRSE instead of the conventional root mean square error (RMSE) as our measure. It is because MRSE makes more practical sense than RMSE for IPSs, which has been widely adopted in indoor positioning contests [2]. The measure of REP is introduced in particular for ELM because it produces variation in repeated realizations, namely, with same parameters setting, e.g., the number of hidden nodes, of the same training set, ELM may draw different results. This is mainly due to the reason that the number of hidden units is not infinite so that the universal approximation using SLFNs with random nodes may not be accurate [18]. However, it is should be noted that, most iteratively tuning-based algorithms such as BP, actually also face the unreproducibility issue, and from the perspective of STD, ELM is even more stable.
2) *Abandonment of Kernelized RELMs:* Although we have proposed the kernelized CTM-RELM and SR-RELM, we did not adopt them in simulation and real-world experiment due to their limits in scaling. Firstly, the size of the decision variables in the kernelized CTM-RELM formulation is $N \times m + 2N + 1$, while the size of the CTM-RELM is $L \times m + 2N + 1$. Considering that the number of training data N is usually several times

larger than the number of hidden nodes L, we would encounter memory issue if we implement the Kernelized CTM-RELM. The same logic applies to the SR-RELM case. Secondly, the kernel-based algorithms enjoy computational efficiency in optimization problems when the dimension d of the feature is larger than N, while in our case, the size of feature is far fewer than the number of training samples, therefore it is not cost-effective to conduct training with kernels.[5] Thirdly, prediction by kernel-based methods takes $\mathcal{O}(Nd)$ time since it uses the dual variables, while prediction using random-hidden-nodes-based methods by primal variables, e.g., ELM, OPT-ELM, and RELMs only takes $\mathcal{O}(d)$ [28]. The testing time listed in Tables II and III is consistent with the above claim. Although a slightly longer training time is within the tolerance for IPSs, the fast prediction speed is highly demanded as IPSs' servers need to provide real-time positioning services for large crowds in some dense indoor environments such as shopping malls, cinemas and airports. However, when encountering small-scale data sets, or where the size of features is very large, kernelized RELMs can be leveraged.

3) *Implementation Tricks for RELMs:* How to calculate the covariance and mean is tricky for regression problems, since one has to use only one sample to approximate its corresponding statistics. In this paper, we take advantage of the specificity of the learning problem in IPSs—grouping. The whole data set can be divided into several groups by their belonging calibration points, and in any group, its members "theoretically" should have the same RSS (input) and coordinates (output). But in reality, it is impossible due to the uncertainties as discussed above. However, these members in one certain group can be intuitively used to calculate the mean and covariance needed to represent the group for problem formulations. By this "grouping" trick, we can further reduce the number of the constraints in (43) and (44) from n to N/g, where g is the size of a group the number of sampling at one calibration point. This trick can be directly extended to RELMs for classification problems.

4) *Assumption About Additive Noises in the Feature Space:* Though we assume that the noises lying in the feature space are additive, the simulation is conducted under the circumstances that the inputs were corrupted with additive disturbances. The simulation results demonstrate that RELMs are effective for these cases. In fact, assuming noises in the feature space are additive is conventionally adopted by a number of ML and optimization researchers [34]–[36]. It is possible that our assumption becomes invalid under some circumstances, e.g., input mixed with multiplicative noises. However, the case of multiplicative noises lying in RSS is rare in indoor environments [37]. When they are not significant, those multiplicative noises can be seen as outliers and Section VII-A has shown that RELMs can address outliers (20% contamination rate) well.

To sum up, this paper proposed CTM-RELM and SR-RELM to address the problem of noisy measurements in IPSs by introducing two CTM and SR constraints to the OPT-ELM, and further gave two SOCP-based formulations. The kernelized RELMs and the method to calculate the theoretical covariance matrix in the feature space were further discussed. Simulation results and real-world indoor localization experiments both demonstrated that the CTM-RELM-based IPS can provide higher accuracy and smaller STD than other algorithms-based IPSs; while the SR-RELM-based IPS can provide better accuracy and smaller WCEs. The REP of the proposed algorithms was also demonstrated to be better.

The future work will focus on how to reduce the computational costs of the proposed algorithms for IPSs with large data sets. Sparse matrix techniques will be leveraged to make it possible. Meanwhile, more performance testing for RELMs will be conducted for classification problems with different combinations of ϖ_1 and ϖ_2 for the norm.

References

[1] H. Zou, X. Lu, H. Jiang, and L. Xie, "A fast and precise indoor localization algorithm based on an online sequential extreme learning machine," *Sensors*, vol. 15, no. 1, pp. 1804–1824, Jan. 2015.

[2] Q. Yang, S. J. Pan, and V. W. Zheng, "Estimating location using Wi-Fi," *IEEE Intell. Syst.*, vol. 23, no. 1, pp. 8–13, Jan./Feb. 2008.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Mar. 1995.

[4] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[5] N. Kothari, B. Kannan, E. D. Glasgwow, and M. B. Dias, "Robust indoor localization on a commercial smart phone," *Proc. Comput. Sci.*, vol. 10, pp. 1114–1120, Aug. 2012.

[6] W. Meng, W. Xiao, W. Ni, and L. Xie, "Secure and robust Wi-Fi fingerprinting indoor localization," in *Proc. Int. Conf. Indoor Position. Indoor Nav. (IPIN)*, Guimarães, Portugal, Sep. 2011, pp. 1–7.

[7] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16, pp. 3056–3062, Oct. 2007.

[8] W. Xi-Zhao, S. Qing-Yan, M. Qing, and Z. Jun-Hai, "Architecture selection for networks trained with extreme learning machine using localized generalization error model," *Neurocomputing*, vol. 102, pp. 3–9, Feb. 2013.

[9] W. Xiao, P. Liu, W.-S. Soh, and Y. Jin, "Extreme learning machine for wireless indoor localization," in *Proc. 11th Int. Conf. Inf. Process. Sens. Netw.*, Beijing, China, Apr. 2012, pp. 101–102.

[10] J. Liu, Y. Chen, M. Liu, and Z. Zhao, "SELM: Semi-supervised ELM with application in sparse calibrated location estimation," *Neurocomputing*, vol. 74, no. 16, pp. 2566–2572, Sep. 2011.

[11] R. Wang, Y.-L. He, C.-Y. Chow, F.-F. Ou, and J. Zhang, "Learning ELM-tree from big data based on uncertainty reduction," *Fuzzy Sets Syst.*, vol. 258, pp. 79–100, Jan. 2015.

[12] J. Zhai, H. Xu, and Y. Li, "Fusion of extreme learning machine with fuzzy integral," *Int. J. Uncertain. Fuzz. Knowl.-Based Syst.*, vol. 21, pp. 23–34, Dec. 2013.

[13] P. Horata, S. Chiewchanwattana, and K. Sunat, "Robust extreme learning machine," *Neurocomputing*, vol. 102, pp. 31–44, Feb. 2013.

[14] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, Nov. 2004.

[15] H. Zou, H. Wang, L. Xie, and Q.-S. Jia, "An RFID indoor positioning system by using weighted path loss and extreme learning machine," in *Proc. 1st IEEE Int. Conf. Cyber-Phys. Syst. Netw. Appl. (CPSNA)*, Taipei, Taiwan, Aug. 2013, pp. 66–71.

[5] Indeed, kernel ELM possesses fast training speed, because it adopts normal equation method, i.e., it is equality constrained-optimization-based [16]. But when inequality constraints are added in the convex optimization setting (inequality constraints can bring about the benefit of sparsity in solutions [23], [29]), the normal closed-form method may not work anymore. Some recent work on ELM, e.g., sparse ELM [29] has already used the inequality constraints-based formulation. Thus, the above claim about the computational costs still holds for kernel ELM.

[16] G.-B. Huang, "An insight into extreme learning machines: Random neurons, random features and kernels," *Cogn. Comput.*, vol. 6, no. 3, pp. 1–15, Sep. 2014.

[17] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[18] 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.

[19] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, pp. 306–314, Oct. 2005.

[20] G. Huang, S. Song, J. N. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.

[21] G.-B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1, pp. 155–163, Dec. 2010.

[22] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[23] 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.

[24] V. Vapnik, S. E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 281–287.

[25] P. K. Shivaswamy, C. Bhattacharyya, and A. J. Smola, "Second order cone programming approaches for handling missing and uncertain data," *J. Mach. Learn. Res.*, vol. 7, pp. 1283–1314, Jul. 2006.

[26] G. Huang, S. Song, C. Wu, and K. You, "Robust support vector regression for uncertain input and output data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1690–1700, Nov. 2012.

[27] J. Navarro, "A very simple proof for the multivariate Chebyshev inequality," *Commun. Stat. Theory Methods*, Dec. 2013.

[28] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[29] Z. Bai, G.-B. Huang, D. Wang, H. Wang, and M. B. Westover, "Sparse extreme learning machine for classification," *IEEE Trans. Cybern.*, vol. 25, no. 4, pp. 836–843, Apr. 2014.

[30] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[31] T. Chrysikos, G. Georgopoulos, and S. Kotsopoulos, "Site-specific validation of ITU indoor path loss model at 2.4 GHz," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw. Workshops (WoWMoM)*, Kos, Greece, Jun. 2009, pp. 1–6.

[32] J. A. Suykens *et al.*, *Least Squares Support Vector Machines*, vol. 4. River Edge, NJ, USA: World Scientific, 2002.

[33] M. C. Grant, S. P. Boyd, and Y. Ye. (Jun. 2014). *CVX: MATLAB Software for Disciplined Convex Programming (Web Page and Software)*. [Online]. Available: http://cvxr.com/cvx

[34] H. Xu, C. Caramanis, and S. Mannor, "Robustness and regularization of support vector machines," *J. Mach. Learn. Res.*, vol. 10, pp. 1485–1510, Jul. 2009.

[35] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Rev.*, vol. 53, no. 3, pp. 464–501, Aug. 2011.

[36] K. P. Bennett and E. Parrado-Hernández, "The interplay of optimization and machine learning research," *J. Mach. Learn. Res.*, vol. 7, pp. 1265–1281, Jul. 2006.

[37] A. Goldsmith, *Wireless Communications*. Cambridge, NY, USA: Cambridge Univ. Press, 2005.

Xiaoxuan Lu received the B.Eng. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. He is currently pursuing the M.Eng. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

His current research interests include machine learning, mobile computing, signal processing, and their applications to energy reduction in buildings.

Han Zou received the B.Eng. (First Class Honors) degree from Nanyang Technological University, Singapore, in 2012, where he is currently pursuing the Ph.D. degree from the School of Electrical and Electronic Engineering.

He is currently a Graduate Student Researcher with the Berkeley Education Alliance for Research in Singapore Limited, Singapore. His current research interests include wireless sensor networks, mobile computing, indoor positioning and navigation systems, indoor human activity sensing and inference, and occupancy modeling in buildings.

Hongming Zhou received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2009 and 2014, respectively.

He is currently a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University. His current research interests include classification and regression algorithms such as extreme learning machines, neural networks, and support vector machines as well as their applications including heating, ventilation and air conditioning system control applications, biometrics identification, image retrieval, and financial index prediction.

Lihua Xie (F'07) received the B.E. and M.E. degrees from the Nanjing University of Science and Technology, Nanjing, China, in 1983 and 1986, respectively, and the Ph.D. degree from the University of Newcastle, Callaghan, NSW, Australia, in 1992, all in electrical engineering.

Since 1992, he has been at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. From 1986 to 1989, he was a Teacher at the Department of Automatic Control, Nanjing University of Science and Technology. From 2006 to 2011, he was a Changjiang Visiting Professor at the South China University of Technology, Guangzhou, China. From 2011 to 2014, he was a Professor and the Head of Division of Control and Instrumentation at Nanyang Technological University, Singapore. His current research interests include robust control and estimation, networked control systems, multiagent networks, and unmanned systems. He has published over 260 journal papers and co-authored two patents and six books.

Prof. Xie has served as an Editor of IET Book Series in Control and an Associate Editor of a number of journals including the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Automatica*, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-II.

Guang-Bin Huang (SM'04) received the B.Sc. degree in applied mathematics and M.Eng. degree in computer engineering from Northeastern University, Shenyang, China, in 1991 and 1994, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 1999.

He was at the Applied Mathematics Department and Wireless Communication Department of Northeastern University. From 2001, he was an Assistant Professor and an Associate Professor (with tenure) at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is the Principal Investigator of several industrial sponsored research and development projects. He has also led/implemented several key industrial projects including the Chief Architect/Designer and the Technical Leader of Singapore Changi Airport Cargo Terminal 5 Inventory Control System Upgrading Project. His current research interests include big data analytics, human computer interface, brain computer interface, image processing/understanding, machine-learning theories and algorithms, extreme learning machine, and pattern recognition. He was the Highly Cited Researcher listed in 2014—The World's Most Influential Scientific Minds by Thomson Reuters. He was also invited to give keynotes on numerous international conferences.

Dr. Huang was the recipient of the Best Paper Award from the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS in 2013. He is currently serving as an Associate Editor of *Neurocomputing*, *Cognitive Computation*, *Neural Networks*, and the IEEE TRANSACTIONS ON CYBERNETICS.