IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 20, 2019, accepted January 6, 2020, date of publication January 9, 2020, date of current version January 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2965150

Temperature Compensation of Piezo-Resistive Pressure Sensor Utilizing Ensemble AMPSO-SVR Based on Improved Adaboost.RT

JI LI[1], CHENTAO ZHANG[2,4], XUKUN ZHANG[1], HONGLIN HE[1], WENGUANG LIU[1], AND CAISEN CHEN[3]

[1]School of Aeronautical Manufacturing Engineering, Nanchang Hangkong University, Nanchang 330003, China
[2]Fujian (Quanzhou)–HIT Research Institute of Engineering and Technology, Quanzhou 362000, China
[3]Military Exercise and Training Center, Army Academy of Armored Forces, Beijing 100072, China
[4]School of Aerospace Engineering, Xiamen University, Xiamen 361102, China

Corresponding author: Chentao Zhang (zctyyy@163.com)

This work was supported in part by the National Science Foundation of China under Grant 51965042, Grant 51665040, and Grant U1836101, in part by the Foundation of Jiangxi Education Department, China, under Grant DA201903159, and in part by the Ph.D. Starting Foundation of Nanchang Hangkong University under Grant EA201803220.

ABSTRACT As the silicon material is severely influenced by the ambient temperature, the silicon piezo-resistive pressure sensor remarkably suffers from a strong nonlinearity in the response characteristic as the ambient temperature varies. To address this crucial issue, an adaptive mutation particle swarm optimization optimized support vector regression (AMPSO-SVR) combined with improved AdaBoost.RT algorithm is presented. The opposition-based learning initialization and Levy mutation is applied in the adaptive mutation particle swarm optimization (AMPSO) to achieve the appropriate model selection task which directly determines the performance of SVR. The performance of original AdaBoost.RT is improved by a dynamical modification approach for threshold and quoted error criterion. In order to verify the effectiveness of the proposed temperature compensation approach, several additional optimization methods such as Cuckoo search (CS), dragonfly algorithm (DA), multi-verse optimizer (MVO), conventional particle swarm optimization (PSO), Levy flight improved particle swarm optimization (Levy-PSO) and the AMPSO combined with SVR are investigated. The minimum quoted error, maximum quoted error, the mean quoted error and the variance of the quoted error over testing data obtained by the proposed method are 6.8764×10^{-5}, 6.4463×10^{-4}, 3.2619×10^{-4} and 2.5714×10^{-8} respectively, which are superior to the corresponding indices obtained by other methods. The analysis of simulation results indicates the method proposed in this research is applicable, effective and efficient for industrial application.

INDEX TERMS Pressure sensor, temperature compensation, support vector regression, particle swarm optimization, Adaboost.RT.

I. INTRODUCTION

The measurement principle of piezo-resistive pressure sensor is the Wheatstone bridge, each arm of it constitutes a silicon resistor. The actual response characteristic of the silicon resistor is mainly determined by the doping concentration and the temperature coefficient of piezo-resistive coefficient of silicon. In addition to the resistor, some unsatisfied aspects in the manufacturing process of the pressure sensor such as thermal expansion coefficient of the packaging materials and the performance of electronics are also affected by temperature. As the ambient temperature changes, all the aforementioned factors may result in strong nonlinear response characteristics of the sensor.

To eliminate the unexpected influence of temperature, some compensation methods focus on hardware tries to linearize the response characteristics by regulating circuit parameters [1]–[5]. Nevertheless, the debugging complexity, considerable cost and limited compensation precision would restrict the generalization of it. In opposition to the hardware compensation, the software compensation methods refer to the analytical approaches and the artificial

The associate editor coordinating the review of this manuscript and approving it for publication was Min Xia.

intelligence approaches possess more flexibility. The analytical approaches such as look-up table, interpolation and surface fitting [6]–[8] are easy to be implemented in sensor circuits while it may confront two kinds of dilemma: the number of interpolation nodes dramatically increases with the requirement of measurement precision and the ill-conditioned problem in solving normal equations when the fitting order increases. The artificial intelligence approaches involve BP neural networks [9]–[11] and support vector machine [12]–[14]. The empirical risk minimum (ERM) principle and gradient descent iteration are the cornerstones of BP neural networks, which may lead the modeling process fall into some pitfalls as the curse of dimensionality, local minimum, under-fitting or over-fitting, etc., [15]–[17]. Vapnik developed the support vector machine (SVM) which rooted in structural risk minimum (SRM) can obtain the global optimal solution by solving a convex optimization problem [18]. A diversity of function approximation applications are resorted to the support vector regression (SVR), which replaces the hard margin in conventional SVM with the soft margin [19], [20]. Without loss of any generation, temperature compensation task can be treated as a regression problem. Hence, the SVR is employed in this research to reveal the mapping relationship between the input and the output of the pressure sensor.

Since the generalization performance of SVR critically relies on the setting of parameters, optimization algorithms including particle swarm optimization (PSO), cuckoo search (CS), firefly algorithm (FA) and shuffle frog leaping algorithm (SFLA) are studied recently to specify the optimal parameters set of SVR [21]–[24]. In order to overcome the overfitting problem of single model in dealing with regression task, Solamatine and Shrestha proposed a promising ensemble method called AdaBoost.RT [25]. Then main idea of AdaBoost.RT is to use an absolute relative error (ARE) threshold φ to convert the regression problem to a binary classification problem. A predictor is labeled as correct if ARE of a sample is less than φ, otherwise, it is deemed as incorrect. Then the weight updating parameters are calculated by the total error rate of each predictor (weak learner). Consequently, the training data distribution at next iteration is updated by these parameters until all predictors are trained. Although AdaBoost.RT is a simplified, robust and implemented conveniently ensemble method, its performance is sensitive to the constant φ that need to be manually fixed at the very beginning. According to the experiment results obtained by Solamatine and Shrestha, AdaBoost.RT is stable when φ varies in the range of [0,0.4] [26]. Tian and Mao [27] proposed a modified AdaBoost.RT to predict the temperature of molten steel in ladle furnace. The approach incorporates self-adaptive mechanism supervised by the root mean square error trend in the iteration process to ensure the good prediction performance. Liu *et al.* [28] developed a new modified AdaBoost.RT technique by weighting the wrong predicted sample and loss function, which is proved as an efficient way to enhance the predictive ability of XXT.

In this study, a compensation approach in the context of ensemble technique is proposed, which combined SVR with AdaBoost.RT. Aiming at configuring the optimal parameters of SVR, an adaptive mutation particle swarm optimization (AMPSO) benefits from the opposition-based learning and Levy flight is presented. Moreover, a self-adaptive modification process of threshold φ is proposed to improve regression performance of original AdaBoost.RT. Finally, the temperature compensation model is established by ensemble AMPSO-SVR based on improved AdaBoost.RT.

II. BASIC ALGORITHM

A. SUPPORT VECTOR REGRESSION (SVR)

Given an independent and identical distributed sample as $(x_1, y_1)\cdots, (x_i, y_i), \cdots (x_m, y_m) \in R^n \times R$, ($x_i$ is input, y_i is the corresponding output). The regression problem can be formed as:

$$d_i = \omega^T x_i + b \tag{1}$$

where d_i is the prediction value of x_i obtained by SVR. By introducing ε-loss function as the loss function and denoting the slack variables by ξ_i and ξ_i^*, the regression problem can be depicted equally as follows:

$$\begin{aligned} &min \frac{1}{2}\omega^T\omega + C\sum\nolimits_{i=1}^{m}(\xi_i+\xi_i^*) \\ &s.t. \begin{cases} y_i - \omega^T x_i - b \le \varepsilon + \xi_i, \\ \omega^T x_i + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i \ge 0, \\ \xi_i^* \ge 0. \end{cases}, \end{aligned} \tag{2}$$

Constructing Lagrange function from Eq. (1) and Eq. (2):

$$\begin{aligned} &L\left(\omega, b, \alpha_i, \alpha_i^*, \xi_i, \xi_i^*, \gamma_i, \gamma_i^*\right) \\ &\quad = \frac{1}{2}\omega^T\omega + C\sum\nolimits_{i=1}^{m}\left(\xi_i+\xi_i^*\right) \\ &\qquad - \sum\nolimits_{i=1}^{m}(\gamma_i\xi_i + \gamma_i^*\xi_i^*) - \sum\nolimits_{i=1}^{m}\alpha_i(\omega^T x_i + b - y_i \\ &\qquad + \varepsilon + \xi_i) - \sum\nolimits_{i=1}^{m}\alpha_i^*(y_i - \omega^T x_i - b + \varepsilon + \xi_i^*) \end{aligned} \tag{3}$$

According to the Karush-Kuhn-Tucker (KKT) condition, taking the derivative with respect of all variables of the Lagrange function:

$$\frac{\partial L}{\partial \omega} = 0, \quad \frac{\partial L}{\partial b} = 0, \frac{\partial L}{\partial \xi_i} = 0, \frac{\partial L}{\partial \xi_i^*} = 0 \tag{4}$$

The results obtained are as follows:

$$\begin{cases} \omega = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i \\ \sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0 \\ \gamma_i = C - \alpha_i, \quad i = 1, 2, \cdots, m \\ \gamma_i^* = \alpha_i^* - C, \quad i = 1, 2, \cdots, m \end{cases} \tag{5}$$

Substituting Eq. (5) into Eq. (3) to convert the primal problem to the corresponding dual problem:

$$max\frac{1}{2}\sum_{i,j=1}^{m}\left(\alpha_i-\alpha_i^*\right)\left(\alpha_j-\alpha_j^*\right)x_i^Tx_j+\sum_{i=1}^{m}\alpha_i\left(y_i-\varepsilon\right)-\sum_{i=1}^{m}\alpha_i^*\left(y_i+\varepsilon\right)$$

$$\text{s.t.}\begin{cases}\sum_{i=1}^{m}\left(\alpha_i-\alpha_i^*\right)=0\\ 0\leq\alpha_i,\alpha_i^*\leq C\end{cases}\tag{6}$$

According to the KKT condition, the dual variables which satisfy the constraints of inequality in primal problem should be diminished:

$$\begin{cases}\alpha_i\left(\varepsilon+\xi_i-y_i+d_i\right)=0\\ \alpha_i^*\left(\varepsilon+\xi_i+y_i-d_i\right)=0\\ \left(\alpha_i-C\right)\xi_i=0\\ \left(\alpha_i^*-C\right)\xi_i^*=0\\ \alpha_i\alpha_i^*=0\\ \xi_i\xi_i^*=0\end{cases}\tag{7}$$

Samples which locate in the insensitive region corresponding $\alpha_i = 0$ and $\alpha_i^* = 0$; others locate outside the insensitive region corresponding $\alpha_i = C$ and $\alpha_i^* = C$; the rest sample corresponding $\xi_i = 0$ and $\xi_i^* = 0, \alpha_i, \alpha_i^* \in (0, C)$ are on the boundary between insensitive and sensitive region. Thus, b can be estimated by the following equations in the manner of sequential minimal optimization (SMO) [29]:

$$\begin{cases}b=d_i-w^Tx_i-\varepsilon, & 0<\alpha_i<C\\ b=d_i-w^Tx_i+\varepsilon, & 0<\alpha_i<C\end{cases}\tag{8}$$

Generally, the essence of the undetermined regression relationship is nonlinear. According to the Cover theorem, a nonlinear problem is more linearly separable by mapping the vector x in the primal space into a high-dimensional space based on a function set $\{\varphi_j(x)\}_{j=1}^{\infty}$ constructed by kernel tricks [30]. The SVR regression function therefore can be denoted as:

$$F=\left\{f|f\left(x\right)=\omega^T\cdot\varphi\left(x\right)+b,\omega\in R^n\right\}\tag{9}$$

where ω is the weight vector of hyperplane, b is the bias. Suppose $K\left(x_i,x_j\right)$ is a kernel function to map the variables in original space in to nonlinear high-dimensional space, which satisfies the Mercer theorem [31]:

$$K\left(x_i,x_j\right)=\varphi\left(x_i\right)\varphi\left(x_j\right)\tag{10}$$

Substituting Eq. (10) into Eq. (5), Eq. (9) becomes:

$$F=\left\{f|f\left(x\right)=\sum_{i=1}^{m_s}\left(\alpha_i-\alpha_i^*\right)K\left(x_i,x\right)+b\right\}\tag{11}$$

where m_s is the number of support vector. Some kernel functions obey the Mercer's condition are linear kernel: $K\left(x,x_i\right) = x^Tx_i$, polynomial kernel: $K\left(x,x_j\right) = \left(x^Tx_i+1\right)^p$, RBF kernel: $K\left(x,x_i\right) = exp\left(-\|x-x_i\|^2/2\sigma^2\right)$ and multilayer perception kernel (MLP): $K\left(x,x_i\right) = tanh(\beta_0x^Tx_i+\beta_1)$. The RBF kernel function is taken as the research object for its extensive utilization; however, the presented optimizing algorithm is also suitable to other Mercer kernels.

B. ADAPTIVE MUTATION PARTICLE SWARM OPTIMIZATION (AMPSO)

Kennedy is inspired by the birds' hunting behavior and proposed the particle swarm optimization algorithm (PSO) [32], [33]. The PSO algorithm can be described as: A particle swarm consists of m particles in an n dimensional searching space, the speed state vector is made of four parts:$x_i = (x_{i1}, x_{i2}\cdots, x_{ij}, \cdots, x_{in})^T, i = 1, 2, \cdots, m$, which means a particle's present position in the searching space; $v_i = (v_{i1}, \cdots, v_{ij}, \cdots, v_{in})^T$, which denotes the speed of each particle; $p_i = (p_{i1}, \cdots, p_{ij}, \cdots, p_{in})^T$,which is the best position of each particle from the beginning of searching; a vector depicted by $p_g = (p_{g1}, \cdots, p_{gj}, \cdots, p_{gn})^T$ records the best position of the particle swarm at every iteration from the start of searching. Particles' behavior in conventional PSO is as follows:

$$v_{ij}^{k+1}=wv_{ij}^k+c_1r_1^k\left(p_{ij}^k-x_{ij}^k\right)+c_2r_2^k\left(p_{gj}^k-x_{ij}^k\right)\tag{12}$$

$$x_{ij}^{k+1}=x_{ij}^k+v_{ij}^{k+1}\tag{13}$$

where v_{ij}^k is the speed of the ith particle's jth component at the kth iteration, w is the flying weight which linearly decreases with the iteration process, c_1 and c_2 are social factor and cognitive factor, r_1^k and r_2^k are two random numbers generated in an uniform distribution interval of (0, 1).

For the PSO is a kind of population-based searching scheme, the mechanism of which will yields stagnation or premature in the searching process. Hence, an adaptive mutation strategy is required in the latter stage of the search to guarantee the diversity of the population until the stopping criterion is satisfied. Among several random strategies to achieve this goal, Levy flight is a prominent choice. The reason for choosing it as the adaptive mutation strategy is that the population variance grows faster in Levy flight than in other random search behavior, which enables it efficiently find the optimal solution in a relative large space [34]. Specifically, the Levy flight is defined as:

$$L\left(s\right)\sim|s|^{-1-\beta},\quad 0<\beta\leq2\tag{14}$$

where s is the step of Levy flight which is derived from the Levy distribution. To obtain a formal representation of the step, a method in a previous research performed by Mategna [35] is adopted in this study, which takes the form as:

$$s=\frac{u}{|v|^{1/\beta}}\tag{15}$$

where u and v are two independent random variables generated from different normal distributions. The normal distributions are:

$$u \sim N(0,\sigma_u^2),\ v \sim N(0,\sigma_v^2), \tag{16}$$

where

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma\left[\frac{1+\beta}{2}\right]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_v = 1. \tag{17}$$

If the trail time of failing to improve the solution quality for the ith particle reaches a predefined number at the kth generation, a solution should be updated through an equation as:

$$\begin{aligned} x_{ij}^k &= x_{ij}^k + 0.01 * step_j * N(0,1), \quad j = 1, 2, \cdots, r, \\ r &= round\,(n * U(0,1)). \end{aligned} \tag{18}$$

where k is the generation number, i is the particle index in the population, j is the dimension index of the particle to be updated, r is the number of specified dimension to be updated whenever the adaptive mutation condition occurs, n is the number of dimension of each particle, *step* is calculated by Eq. (15), $N(0,1)$ and $U(0,1)$ are normally distributed and uniformly distributed random numbers between 0 and 1 respectively. The constant 0.01 used here is to reduce the aggressive step size of Levy flight which could lead the particle fly outside the searching region. The mean square error (MSE) between the predicted value derived from the SVR and the actual value is adopted as the fitness function. From a straightforward perspective, the convergence speed of the searching process depends on the distribution uniformity of population in the searching space to some extent. Nevertheless, the solutions that far away from the optimal position may slow down the searching process if distribution uniformity of the population is the only considered factor. In the light of the analysis of the population initialization mentioned before, the opposition-based learning [36] is employed in the initialization period. To be more specific, each dimension of the opposition particle denoted by x_{i_oppo} can be found in the predefined solution space as follows:

$$x_{ij\_oppo} = u_{ij} + v_{ij} - x_{ij}, \quad x_{ij} \in \left[u_{ij}, v_{ij}\right]. \tag{19}$$

where i and j are defined in the introduction in the original PSO, u_{ij} and v_{ij} are the lower and upper limitation of the jth dimension of the ith particle in the population, respectively.

The details of the adaptive mutation particle swarm optimization (AMPSO) are summarized as follows:

(1) Initialize all the algorithm parameters: maximum iteration number M; size of the particle swarm m; cognition factor c_1 and social factor c_2; the limits of the velocity and the position in each dimension as $[v_{i\_min}, v_{i\_max}]$ and $[x_{i\_min}, x_{i\_max}]$; the maximum limit *Fail_Time* of the time that unable to improve individual particle's solution quality; the parameter β utilized in the calculation of the step size of the Levy flight.

(2) Initialize the particle swarm within the solution space by opposition-based learning. Evaluate the fitness of the population, set x_i to be p_i and the particle with the best fitness to be p_g.

(3) Set the time t_i that unable to improve the ith particle solution quality to zero. Update the particle velocity and position according to Eq. (12) and Eq. (13). Evaluate fitness function of each particle and compare it with p_i, if the ith fitness fit_i is less than p_i then x_i is set to be p_i, otherwise, t_i is added by one. If the ith fitness fit_i is less than p_g then x_i is set to be p_g and reset t_i to be zero.

(4) If t_i is large than *Fail_Time* then update the particle position by Eq. (18). Reset t_i to be zero and go back to step 3.

(5) If the stopping criterion is satisfied the optimization process is over.

The pseudo codes of the AMPSO algorithm are provided in Figure 1.

```
Initialize the particle swarm population Xi (i=1,2,...,n) by the Eq. (19)
Evaluate the fitness of each particle, record the position p_i_local and fitness
local_fitness fit_i_local of local best, record the position p_g and fitness global_fitness
fit_g of global best
Set the fail time to update the local best of ith particle fail_time_i as 0
while k<M
    Calculate the weight by w = (k - M)/M
    for each particle i
        Calculate the weight by the Eq. (12)
        Update the particle position by the Eq. (13)
        Check if the ith particle flies beyond the solution space and amend it
        Evaluate the fitness of the ith particle fit_i
        if fit_i < fit_i_local
            Update p_i_local and fit_i_local
            fail_time_i=0
        else
            fail_time_i=fail_time_i+1
        end
        if fail_time_i >= M/5
            fail_time_i=0
            Update the ith particle by the Eq. (14) ~ the Eq. (18)
            Evaluate the fitness of the ith particle fit_i
            if fit_i < fit_i_local
                Update p_i_local and fit_i_local
            end
        end
        if fit_i < fit_g
            Update the p_g and fit_g
        end
    end
    k=k+1
end
return p_g and fit_g
```

FIGURE 1. **Pseudo-code of the AMPSO algorithm.**

C. IMPROVED ADABOOST.RT

Instead of the invariable φ in original AdaBoost.RT, a dynamically modification strategy is proposed to change the value of φ at every iteration. Given the sample set with m elements as $(x_1,y_1), \ldots, (x_i,y_i), \ldots (x_m,y_m) \in \mathrm{R}^n \times \mathrm{R}$, in which $x_i \in \mathrm{R}^n$ is the input and $y_i \in \mathrm{R}$ is the output. The concrete procedures of original AdaBoost.RT are described as follows: (1) Identify the weak learner, the maximum iteration times (weak learner number) and the threshold $\varphi \in (0, 1)$.

(2) Initialize the iteration number $t = 1$. Set each the sample weight as $D_t(i) = 1/m$. The error rate of sample set is specified as $\varepsilon_t = 0$.

(3) Calculate the absolute relative error (ARE) for each sample:

$$\mathrm{ARE}_i = \left| \frac{f_t(x_i) - y_i}{y_i} \right| \tag{20}$$

where $f_t(x_i)$ is the prediction of the input x_i by the tth weak learner.

(4) Calculate the error rate ε_t for the tth weak learner:

$$\varepsilon_t = \sum D_t(i), \quad \mathit{if}\ \mathrm{ARE}_i > \varphi \tag{21}$$

(5) Set $\beta_t = \varepsilon_t^n$(n=1,2 or 3), update the weight distribution for all sample data:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t, & \mathrm{ARE}_i \le \varphi, \\ 1, & \mathit{otherwise}. \end{cases} \tag{22}$$

where $Z_t = \sum_i D_t(i)$ is to ensure the $D_{t+1}(i)$ conforms a probability distribution.

(6) The final ensemble predictor for each sample can be represented by:

$$f(x_i) = \sum\nolimits_t \log\left(\frac{1}{\beta_t}\right) f_t(x_i) / \sum\nolimits_t \mathit{log}\left(\frac{1}{\beta_t}\right) \tag{23}$$

The error weight β_t is a kind of measurement to evaluate the confidence of each weak learner, the importance of the tth weak learner in the final ensemble hypothesis is inverse proportional to the change of β_t. Furthermore, the exponential power n is used to emphasize the influence of error rate on the weak learner.

The threshold φ plays a role of arbitrator, responsibility of which is to evaluate the importance of each sample. A reasonable choice of φ is significant to the performance of ensemble selection in AdaBoost.RT. If φ is too low, the improvement of AdaBoost.RT cannot be guaranteed for it is difficult to generate enough correct sample in relative limited iteration epochs. On the other side, if φ is too high, some outliers are boosted repetitively and other sample data are neglected by the iterative process at the mean time. Consequently, the overfitting of AdaBoost.RT is triggered. Therefore, a dynamically modification strategy for φ during the modeling process is definitely required. In the view point of industrial application, quoted error (QE) is more comprehensive to illustrate the measurement performance than ARE, which takes the form as:

$$\mathrm{QE}_i = \left| \frac{f_t(x_i) - y_i}{y_{FS}} \right| \tag{24}$$

where $f_t(x_i)$ is the prediction of the input x_i by the tth weak learner, y_{FS} is the full scale of the output. The zero output or

```
Initialize the number of sample set m, the number of weak learner N (N≥2), threshold
0 < φ_1 = φ_2 < 1
while t<N
    for each input x_i
        D_t(i)=1/m
        Calculate QE_i of x_i by the Eq. (24)
    end
    Calculate the mean square error of the training set e^t_MSE by Eq. (26)
    Calculate the error rate ε_t for the tth weak learner by Eq. (21)
    Update the weight distribution for all sample data by Eq. (22)
    if t ≥ 2
        Calculate the regulation parameter λ by Eq. (27)
        if e^t_MSE < e^(t-1)_MSE
            φ_(t+1) = φ_t(1 + λ)
        else
            φ_(t+1) = φ_t(1 − λ)
        end
        if t ≥ 3
            if max(|QE_i(t) − QE_i(t − 1)|) < σ
                φ_(t+1) = φ_t
            end
        end
        if φ_(t+1) ∉ (min_i QE_i , max_i QE_i)
            Update φ_(t+1) by Eq. (28)
        end
    end
    t=t+1
end
return f(x) by Eq. (23)
```

FIGURE 2. Pseudo-code of the improved AdaBoost.RT algorithm.

output besides zero intuitively leads to a large ARE even prediction accuracy is satisfied. To deal with this issue, the zero output or output besides zero is replaced by a predefined small number. A self-adaptive modification method of φ is proposed in the improved AdaBoost.RT, which is described as follows:

(1) Set the original threshold φ_1 and the 2nd iteration threshold $\varphi_2 = \varphi_1$,

(2) The φ remains unchanged on the premise that $\max(|\mathrm{QE}_i(t) - \mathrm{QE}_i(t-1)|) < \sigma$, where σ is a relative small value. Otherwise, φ is updated by:

$$\varphi_{t+1} = \begin{cases} \varphi_t(1+\lambda), & e^t_{MSE} < e^{t-1}_{MSE}, \\ \varphi_t(1-\lambda), & e^t_{MSE} \ge e^{t-1}_{MSE}. \end{cases} \tag{25}$$

where e_{MSE} denotes the mean square error with respect to QE_i is calculated by:

$$e_{MSE} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left(\frac{f_t(x_i) - y_i}{y_{FS}} \right)^2 \tag{26}$$

the regulation parameter λ in terms of e_{MSE} is given by:

$$\lambda = \frac{1}{2} \left| (e^t_{MSE} - e^{t-1}_{MSE}) / e^t_{MSE} \right| \tag{27}$$

To achieve the goal of maintaining enough large size and diversity of the sample, the threshold φ is adaptively regulated by taking consideration of the difference of e_{MSE} between two continuous iteration.

(3) In order to maintain the validation of the variable φ at every iteration, if $\varphi_{t+1} \notin (\mathop{min}\limits_i QE_i, \mathop{max}\limits_i QE_i)$, a restriction rule to keep the weak learner form the over-training or

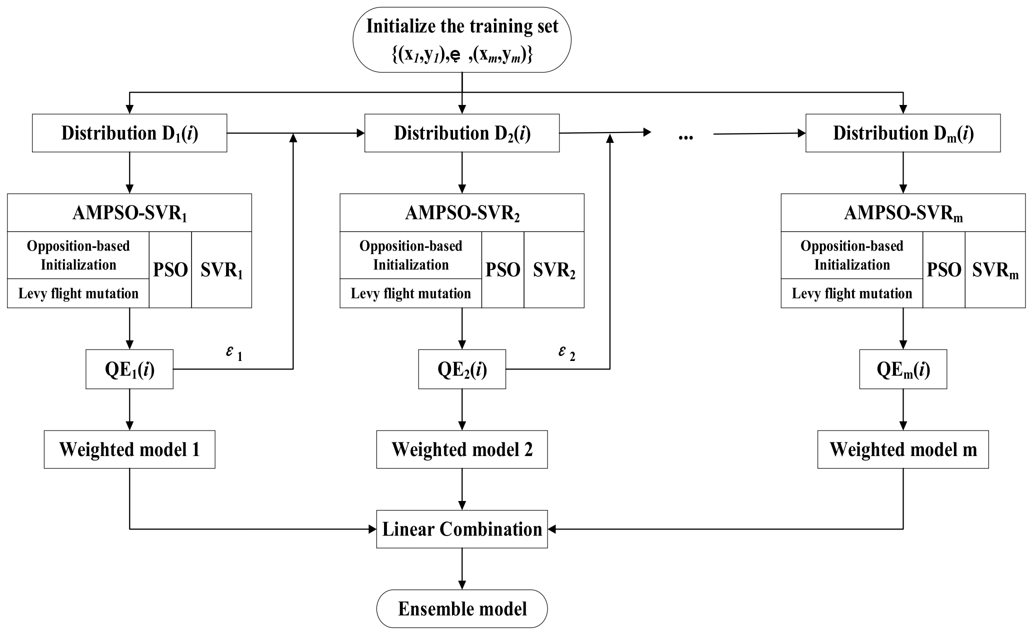

FIGURE 3. Ensemble AMPSO-SVR based on improved AdaBoost.RT.

under-training is given:

$$\varphi_{t+1} = \min\left(\frac{1}{m}\sum\nolimits_{i=1}^{m} QE_i, \varphi_1\right) \tag{28}$$

The pseudo codes of improved AdaBoost.RT are provided in Figure 2.

III. ENSEMBLE AMPSO-SVR BASED ON IMPROVED ADABOOST.RT

In this study, the presented optimization is applied to identify the optimal parameters set and even the support vectors number of SVR. Furthermore, the 3-fold cross validation is applied to prevent the SVR from over-fitting [37]. On the other hand, the application of the 3-fold cross validation can also efficiently evaluate the learning ability and generalizing performance of the SVR model. In the context of the temperature compensation issue, the input vector is formed by environmental temperature (T) and measured output voltage (V) while the output is the pressure value (P') predicted by the presented method. Firstly, the parameter set (C, γ, ε) of SVR model is selected by the AMPSO and calibration data $((T_1, V_1, P'_1), \cdots, (T_j, V_j, P'_j), \cdots, (T_m, V_m, P'_m))$. Then the improved AdaBoost.RT, which including a dynamical modification threshold strategy based on QE criterion, takes the optimized SVR as the weak learner. The mathematical description of the presented temperature compensation method can be given by substituting the trained SVR into Eq. (23):

$$P'_j = \sum\nolimits_t \log\left(\frac{1}{\beta_t}\right)[\sum\nolimits_{i=1}^{m_s{}^t}\left(\alpha_i^t - \alpha_i^{*t}\right)K^t[(T_i, V_i), (T_j, V_j)] + b^t]/\sum\nolimits_t log\left(\frac{1}{\beta_t}\right) \tag{29}$$

where the label t denotes the model parameters of the tth weak learner, i.e. AMPSO-SVR.

Details of the ensemble AMPSO-SVR based on improved AdaBoost.RT are exhibited in Figure 3.

IV. EXPERIMENT AND RESULT ANALYSIS

A. CALIBRATION EXPERIMENTS

A calibration experiment of a piezo-resistive pressure sensor is conducted to evaluate the performance of the proposed method. The measurement range of the pressure sensor is from −40000 Pa to 40000 Pa, and working permission temperature range is restricted in −20°C ∼ 70°C. The output of the pressure sensor on the working condition is designed from 1 V to 5V according to industrial requirement. The pressure sensor assembled with gas hoses and electric wires is mounted in a temperature chamber during the experiment, as is shown in Figure 4. The standard input pressure is represented by P, furthermore, the ambient temperature and the output of the piezo-resistive pressure sensor are denoted as T and U. T and U are placed together to form the input of the presented temperature compensation method, while the output is the predict pressure value P'.

The calibration experiment contains five temperature preservation processes at five different temperature levels as −20°C, 0°C, 20°C, 50°C and 70°C. Every temperature preservation process lasts about 3 hours. The pressure is sampled every other 5000 Pa from −40000 Pa to 40000 Pa in every temperature preservation process. A total of 17*5=85 input-output pairs form the experiment data, which is tabulated in Table 1. The temperature effect of response characteristic of the piezo-resistive pressure sensor can be seen in Figure 3(a). It can be seen in Figure 5(a), the sensor's output varies notably along with the temperature variation.

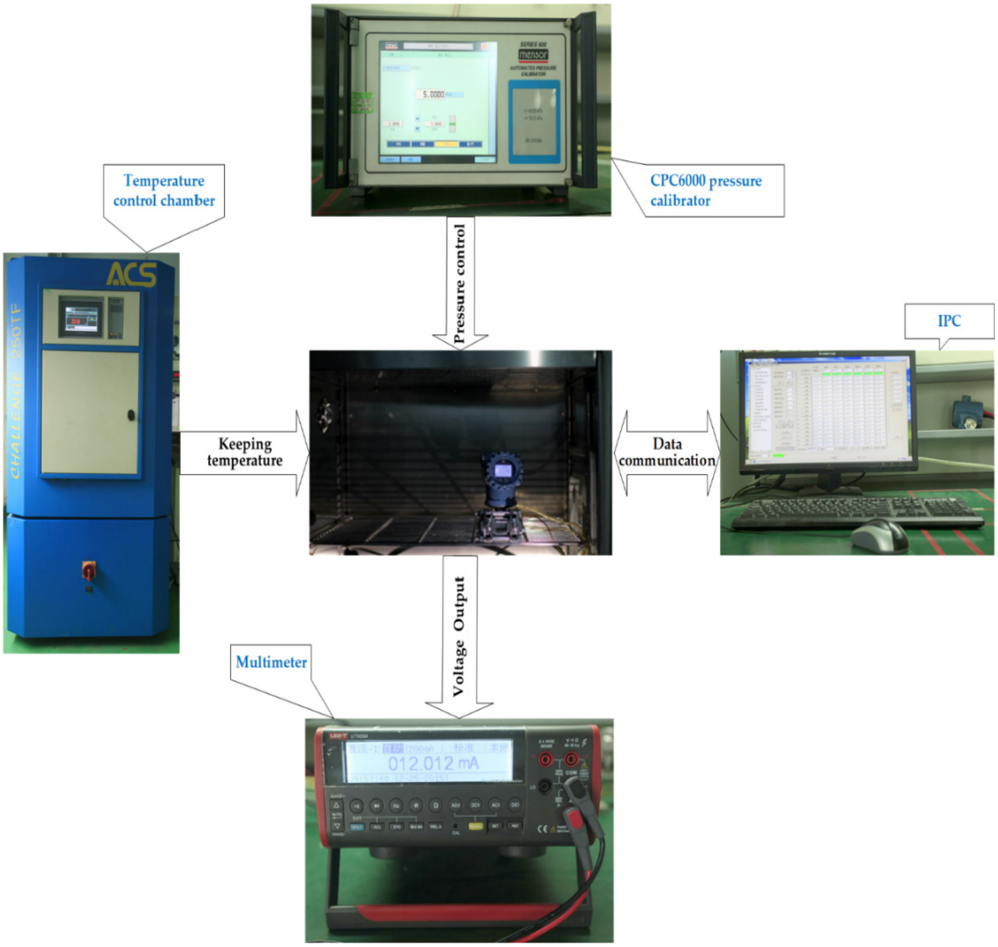

FIGURE 4. Temperature experiment system.

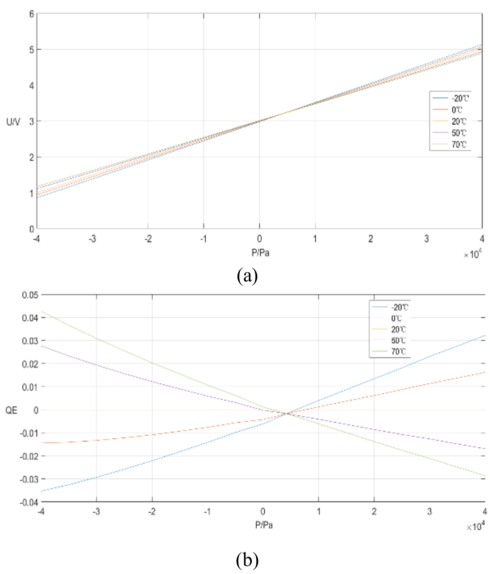

FIGURE 5. Temperature effect of the piezo-resistive pressure sensor's response characteristic.

Furthermore, the maximum value of all quoted errors (QE) mentioned in Eq. (24) are depicted in Figure 5(b), which reaches 4.27% with respect to the sensor's output at the benchmark temperature of 20°C. The dramatic temperature effect leads to nonlinear output charateristic of pressure sensor, which forces sensor producers to develop an efficient temperature compensation scheme.

B. ALGORITHM PARAMETERS SETTING

To test the validity of the proposed method, the pressure prediction in a straightforward way by optimized SVR and the presented ensemble AMPSO-SVR based on improved AdaBoost.RT model are both investigated. In particular, six optimization choices as cuckoo search (CS) [38], dragonfly algorithm [39], multi-verse optimizer [40], the conventional particle swarm optimization (PSO), particle swarm optimization improved with Levy flight (Levy-PSO) [41] and the proposed AMPSO are taken into consideration. The optimal parameters selection of the SVR model is an unconstraint optimization problem in essence, which may carry a practical burden of time cost. In order to refine the parameters searching process, the lower limitations of all the SVR parameters (C, γ, ε) are defined as 2.2204×10^{-16}. On the other hand, the upper limitations of which can be calculated according to related research conducted by Ortiz-Garcia *et al.* [42] and Varewyck and Martens [43]. The details of parameters setting for all the optimization algorithms are demonstrated

TABLE 1. Calibration experiment data.

P/Pa	T/℃ −20	0	20	50	70
	U/V				
−40000	0.8590	0.9422	1.0003	1.1110	1.1710
−35000	1.1203	1.1940	1.2504	1.3436	1.3969
−30000	1.3829	1.4469	1.5002	1.5774	1.6238
−25000	1.6466	1.7011	1.7499	1.8124	1.8520
−20000	1.9113	1.9562	1.9999	2.0486	2.0812
−15000	2.1769	2.2125	2.2498	2.2857	2.3116
−10000	2.4434	2.4695	2.5001	2.5236	2.5427
−5000	2.7105	2.7274	2.7501	2.7624	2.7748
0	2.9784	2.9859	3.0030	3.0020	3.0076
5000	3.2465	3.2448	3.2502	3.2420	3.2409
10000	3.5155	3.5046	3.4997	3.4830	3.4753
15000	3.7846	3.7646	3.7498	3.7243	3.7099
20000	4.0536	4.0246	4.0000	3.9656	3.9447
25000	4.3228	4.2850	4.2503	4.2073	4.1799
30000	4.5919	4.5452	4.4997	4.4490	4.4152
35000	4.8606	4.8051	4.7502	4.6907	4.6504
40000	5.1289	5.0648	4.9999	4.9322	4.8855

TABLE 2. Initial parameters of PSO for searching the optimal parameters of SVR.

Parameters	CS	DA	MVO	PSO	Levy-PSO	AMPSO
Population size	30	30	30	30	30	30
Maximum number of iteration	50	50	50	50	50	50
Minimum inertial weight/Minimum		0.4	0.2	0.4	0.4	0.4
Maximum inertial weight/Maximum		0.9	1	0.9	0.9	0.9
Learning factor c_1				2	2	2
Learning factor c_2				2	2	2
F_{lim}					10	10
Beta(Levy flight)		1.5			(0,2]	(0,2]
Discovery rate of alien solutions	0.25					
p			6			

in Table 2. The whole coding is implemented on MATLAB platform (2018b) with the help of libsvm toolbox [44].

C. COMPENSATION RESULTS ANALYSIS AND DISCUSSION

Aim to verify the improvement effectiveness of the proposed AMPSO, All calibration data thus is divided into training set and testing set. The sample at temperature levels of −20°C, 20°C and 70°C forms the training set and the rest experiment data is taken as the testing set. The searching convergence characteristic of all the investigated optimizations including the best fitness and the average fitness of the population is illustrated in Figure 6 and Figure 7. It can be inferred that the searching ability of CS is somewhat inferior to other algorithms but the average fitness of the population is maintain at a relative high level. The PSO is also suffer from a premature problem as the CS as it shows in the best fitness comparison with other methods even the average fitness of PSO seems to get smaller as the iteration proceeding. DA and MVO possess a relative decent searching ability than CS and PSO. Furthermore, the convergence performance of DA and MVO is between CS and PSO from the best fitness aspect. Nonetheless, their performance in average fitness can be deemed as lack of searching capability in the later stage of the optimization. Both the proposed scheme and the Levy-PSO demonstrate the ability to perform a more detail searching. Moreover, the opposition-based learning not only provides the proposed scheme an ideal beginning of iteration, but also leads to a faster convergence speed than the Levy-PSO.

After training with the data collected from calibration for 30 times, model parameters and averaged support vectors (SV) number of the trained SVR are tabulated in Table 3. As it can be seen from the statistical data in Table 3, all the compensation methods except Levy-PSO-SVR and AMPSO-SVR are captured by a local optimal since the value

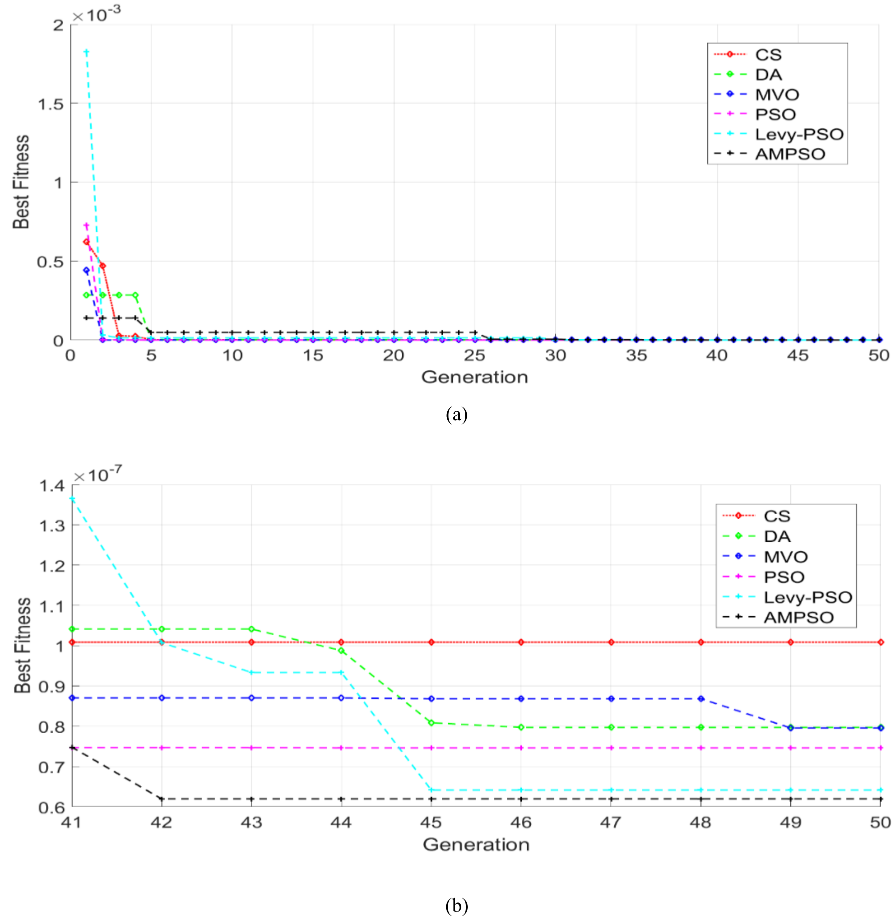

FIGURE 6. Best fitness obtained by each algorithm. (a)Best fitness obtained from the start to 50th generations; (b) best fitness obtained from 41st to 50th generations.

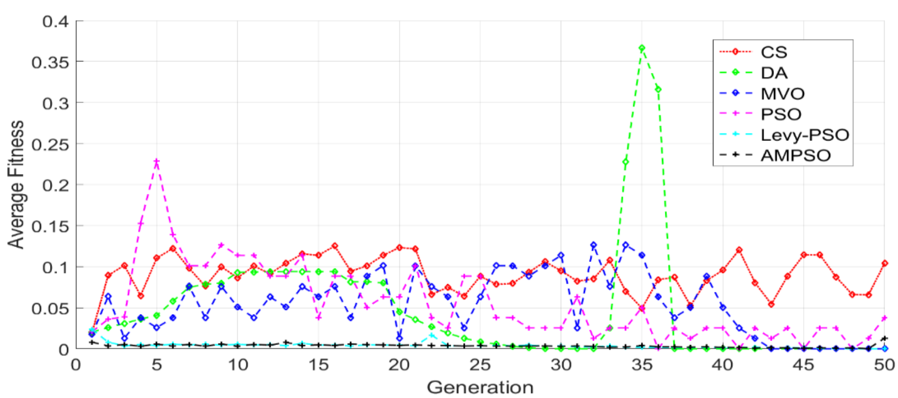

FIGURE 7. Average fitness obtained by each algorithm.

of epsilon (ε) does not change any more during the learning period of the modeling. The parameter C is also called the penalty factor that mainly accounts for the generalization ability, the lower the value of which the better the generalization ability of the trained model. The presented method has the lowest C value which means the model constructed by it would be provided with the best generalization ability.

The parameter gamma (γ) is defined as the width of the kernel function which represents the sparseness of the kernel function in the primal space. The smaller the value of gamma is the more likely the kernel function acts as a local function and devotes more approximation ability to the points around the support vector. If the support vector really stands for the key information about the model, the more intensity of

TABLE 3. Hyper-parameters obtained by different methods.

Single Temperature compensation model	C_{mean}	C(var)	γ(mean)	γ(var)	ε(mean)	ε(var)	SV
CS-SVR	6.5353×10^{4}	1.1478×10^{9}	6.0893×10^{-2}	1.4270×10^{-3}	2.2204×10^{-16}	0	49
DA-SVR	5.2265×10^{4}	1.3503×10^{9}	1.6166×10^{-1}	1.2797×10^{-1}	2.2204×10^{-16}	0	49
MVO-SVR	5.1239×10^{4}	1.0324×10^{8}	3.4948×10^{-2}	1.5478×10^{-5}	2.2204×10^{-16}	0	49
PSO-SVR	5.0834×10^{4}	1.4451×10^{9}	3.5010×10^{-2}	1.1736×10^{-2}	2.2204×10^{-16}	0	49
Levy-PSO-SVR	4.0200×10^{4}	6.2161×10^{8}	3.9186×10^{-2}	4.9014×10^{-4}	4.7845×10^{-5}	1.0176×10^{-9}	41
AMPSO-SVR	2.4352×10^{4}	3.2469×10^{8}	3.3223×10^{-2}	3.0849×10^{-5}	5.2762×10^{-5}	1.4514×10^{-9}	40

TABLE 4. Compensation results of training set by different methods.

Models	QE_{train} (min)	QE_{train} (max)	QE_{train} (mean)	QE_{train} (var)
CS-SVR	5.9585×10^{-4}	2.0757×10^{-3}	9.5948×10^{-4}	1.1937×10^{-7}
DA-SVR	2.6888×10^{-4}	1.0084×10^{-3}	5.5320×10^{-4}	3.4974×10^{-8}
MVO-SVR	3.8831×10^{-4}	1.9919×10^{-3}	8.8961×10^{-4}	1.2469×10^{-7}
PSO-SVR	4.6115×10^{-4}	1.9506×10^{-3}	9.8056×10^{-4}	1.3728×10^{-7}
Levy-PSO-SVR	2.3207×10^{-4}	7.9027×10^{-4}	3.9056×10^{-4}	1.6145×10^{-8}
AMPSO-SVR	1.2405×10^{-4}	6.1527×10^{-4}	2.4727×10^{-4}	1.0332×10^{-9}
AdaBoost.RT-AM PSO-SVR	5.3955×10^{-5}	2.3719×10^{-4}	1.2202×10^{-4}	2.9158×10^{-9}

TABLE 5. Compensation results of testing set by different methods.

Models	QE_{test}(min)	QE_{test} (max)	QE_{test} (mean)	QE_{test} (var)
CS-SVR	4.1947×10^{-4}	2.5386×10^{-3}	1.1618×10^{-3}	2.8142×10^{-7}
DA-SVR	4.8538×10^{-4}	3.6272×10^{-3}	1.9946×10^{-3}	8.6216×10^{-7}
MVO-SVR	3.1591×10^{-4}	1.8705×10^{-3}	9.7193×10^{-4}	1.7325×10^{-7}
PSO-SVR	4.7692×10^{-4}	2.2455×10^{-3}	9.2787×10^{-4}	1.4220×10^{-7}
Levy-PSO-SVR	2.6193×10^{-4}	1.1603×10^{-3}	5.8015×10^{-4}	6.0403×10^{-8}
AMPSO-SVR	1.2696×10^{-4}	7.4849×10^{-4}	4.2026×10^{-4}	3.7460×10^{-8}
AdaBoost.RT-AMP SO-SVR	6.8764×10^{-5}	6.4463×10^{-4}	3.2619×10^{-4}	2.5714×10^{-8}

the effort draws on the support vector can help to reduce the redundant support vectors without any loss in the model generalization ability. So in this case the value of gamma the smaller the better and the presented method can satisfied the requirement. The parameter epsilon is the width of the hyper-plane in the mapped Reproduced Kernel Hilbert Space (RKHS) which indicates the acceptance of the training data into the ε-tube. Generally, the value of epsilon should not be too large nor too small to tradeoff the number of the support vectors and the generalization ability of the model. From the he model simplicity viewpoint, support vector number of the trained SVR should be as less as possible without any the unsatisfied effect on the model generation ability.

The quoted error described in Eq. (26) is employed as validation criterion of compensation algorithm for pressure sensor. With the evaluation of each compensation methods, a compensation result set consists of 85 averaged compensated pressure value is obtained. Four performance indices of the compensation result set as minimum (min), maximum (max), mean (mean) and variance (var) of the averaged quoted error are investigated to evaluate the performance of different models, and the details are summarized in Table 4 and Table 5. Even the performance of CS-SVR is inferior to other conventional methods optimized SVR (DA-SVR, MVO-SVR, PSO-SVR) in training set, the generation ability is not the worst as its mean quoted error is smaller than DA-SVR and MVO-SVR. Taking into consideration of

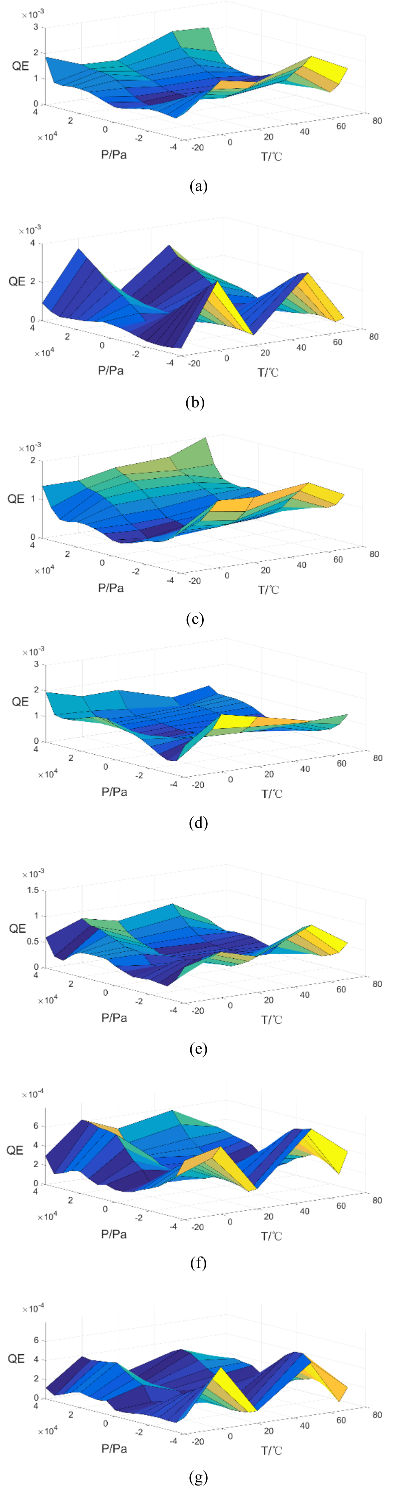

FIGURE 8. **Temperature compensation error surface of testing set. (a) Compensation results obtained by CS-SVR; (b) Compensation results obtained by DA-SVR; (c) Compensation results obtained by MVO-SVR; (d) Compensation results obtained by PSO-SVR; (e) Compensation results obtained by Levy-PSO-SVR; (f)Compensation results obtained by AMPSO-SVR; (g) Compensation results obtained by ensemble AMPSO-SVR based on improved AdaBoost.RT.**

the performance in modeling, the DA-SVR is likely one of the promising choices for the temperature compensation, but as the error data in the testing set shows, the model constructed by it may not describes the complicate relation between all the variables thoroughly. The MVO-SVR and PSO-SVR also confront the overfitting problem as same as DA-SVR, which is the significant factor to limit the implementation of them in this research. Obviously, remarkable improvement obtained by Levy-PSO-SVR and the presented method are derived from the introduction of Levy flight. Overall, the investigated indices indicate that the AMPSO-SVR reaches a good balance among the training ability, the generalization ability, the stability and the simplicity of the model.

The compensation performance and compensated relative error surface of different methods are shown in (a) $\sim$ (g) of Figure 8 respectively. It can be observed the maximum and overall quoted error of the proposed model outperforms the other six models. This result benefits from two aspects. On one side, the inherent feature of the AdaBoost avoids the bias between the constructed approximated model and actual model. On the other side, the learning process of critical sample is emphasized by the presented regulation strategy to adapt the threshold φ dynamically.

V. CONCLUSION

A temperature compensation method for pressure sensor in combination of AMPSO optimized SVR and improved AdaBoost.RT is presented in this research. Since the performance of SVR is sensitive to model parameters, a new adaptive mutation particle swarm optimization is proposed to train the SVR. What is more, the convergence process of AMPSO is accelerated by the opposition-based learning and Levy flight. In order to overcome the overfitting problem of single SVR model, an ensemble approach refers to AdaBoost.RT is presented. With the help of dynamically modification strategy of the threshold and quoted error criterion, the AdaBoost.RT is more suitable for industrial evaluation and its performance is improved.

To accomplish the process of learning the mapping relationship between input and output of the pressure sensor, training data for SVR is acquired through a calibration experiment. The presented AMPSO algorithm not only avoids the local optimal model parameters, but also simplified the architecture of SVR. The simulation result demonstrates the proposed method outperforms other optimized SVR methods to a certain extent. Furthermore, the improved AdaBoost.RT takes advantages from the chosen weak learner, i.e. SVR, is superior to the single trained model. The randomness of the relative error in full scale exhibits the certain effect of temperature has been found by the presented strategy. According to the analysis of the calibration experiment and the simulation results, it proves the presented ensemble AMPSO-SVR based on improved AdaBoost.RT is an efficient and practical temperature compensation approach.

REFERENCES

[1] A. H. González, J. R. Beltrán, J. T. Cerón, and E. C. Rodríguez, "New architecture for intelligent pressure sensors with analog and digital outputs," *IEEE Latin Amer. Trans.*, vol. 12, no. 2, pp. 116–121, Mar. 2014.

[2] M. Aryafar, M. Hamedi, and M. Ganjeh, "A novel temperature compensated piezoresistive pressure sensor," *Measurement*, vol. 63, pp. 25–29, Mar. 2015.
[3] S.-K. Yoo, K.-Y. Shin, T.-B. Lee, S.-O. Jin, and J. Kim, "Development of a radial pulse tonometric (RPT) sensor with a temperature compensation mechanism," *Sensors*, vol. 13, no. 1, pp. 611–625, Jan. 2013.
[4] S.-R. Chiu, L.-T. Teng, J.-W. Chao, C.-Y. Sue, C.-H. Lin, H.-R. Chen, and Y.-K. Su, "An integrated thermal compensation system for MEMS inertial sensors," *Sensors*, vol. 14, no. 3, pp. 4290–4311, Mar. 2014.
[5] R. Y. Que, R. Zhu, Q. Z. Wei, and Z. Cao, "Temperature compensation for thermal anemometers using temperature sensors independent of flow sensors," *Meas. Sci. Technol.*, vol. 22, no. 8, Aug. 2011, Art. no. 085404.
[6] C.-S. Chae, J.-H. Kwon, and Y.-H. Kim, "A study of compensation for temporal and spatial physical temperature variation in total power radiometers," *IEEE Sensors J.*, vol. 12, no. 6, pp. 2306–2312, Jun. 2012.
[7] J. Yu, J. Li, Q. Dai, D. Li, X. Ma, and Y. Lv, "Temperature compensation and data fusion based on a multifunctional gas detector," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 1, pp. 204–211, Jan. 2015.
[8] S. Fan, Q. Zhang, and J. Qin, "Temperature compensation of pressure sensor based on the interpolation of splines," *Beijing Hangkong Hangtian Daxue Xuebao/J. Beijing Univ. Aeronaut. Astronaut.*, vol. 32, pp. 684–686, 2006.
[9] H. Wang, W. Zhang, L. You, G. Yuan, Y. Zhao, and Z. Jiang, "Back propagation neural network model for temperature and humidity compensation of a non dispersive infrared methane sensor," *Instrum. Sci. Technol.*, vol. 41, no. 6, pp. 608–618, Nov. 2013.
[10] D. Xia, L. Kong, Y. Hu, and P. Ni, "Silicon microgyroscope temperature prediction and control system based on BP neural network and fuzzy-PID control method," *Meas. Sci. Technol.*, vol. 26, no. 2, Feb. 2015, Art. no. 025101.
[11] A. Shokri, O. B. Haddad, and M. A. Mariño, "Multi-objective quantity–quality reservoir operation in sudden pollution," *Water Resour Manage*, vol. 28, no. 2, pp. 567–586, Jan. 2014.
[12] Y.-S. Zhang and T. Yang, "Modeling and compensation of MEMS gyroscope output data based on support vector machine," *Measurement*, vol. 45, no. 5, pp. 922–926, Jun. 2012.
[13] X. Wang, F. Huang, and Y. Cheng, "Super-parameter selection for Gaussian-Kernel SVM based on outlier-resisting," *Measurement*, vol. 58, pp. 147–153, Dec. 2014.
[14] H. Pang, D. Chen, M. Pan, S. Luo, Q. Zhang, and F. Luo, "Nonlinear temperature compensation of fluxgate magnetometers with a least-squares support vector machine," *Meas. Sci. Technol.*, vol. 23, no. 2, Feb. 2012, Art. no. 025008.
[15] F. Zhang, L. Taishan, and Z. Liping, "Temperature compensation modeling of infrared methane detector based on BP and RBF neural network," Tech. Rep., 2008.
[16] J. Ding, J. Zhang, W. Huang, and S. Chen, "Laser gyro temperature compensation using modified RBFNN," *Sensors*, vol. 14, no. 10, pp. 18711–18727, Oct. 2014.
[17] Y. Sun, Y. Liu, and H. Liu, "Temperature compensation for six-dimension force/torque sensor based on radial basis function neural network," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Jul. 2014, pp. 789–794.
[18] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.
[19] Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, and Y. Bai, "An adaptive SVR for high-frequency stock price forecasting," *IEEE Access*, vol. 6, pp. 11397–11404, 2018.
[20] C. Sun, L. Yang, L. Chen, and J. Zhang, "SVR based blind signal recovery for convolutive MIMO systems with high-order QAM signals," *IEEE Access*, vol. 7, pp. 23249–23260, 2019.
[21] M. Barman and N. B. Dev Choudhury, "Season specific approach for short-term load forecasting based on hybrid FA-SVM and similarity concept," *Energy*, vol. 174, pp. 886–896, May 2019.
[22] W. Liu, G. Guo, F. Chen, and Y. Chen, "Meteorological pattern analysis assisted daily PM2.5 grades prediction using SVM optimized by PSO algorithm," *Atmos. Pollut. Res.*, vol. 10, no. 5, pp. 1482–1491, Sep. 2019.
[23] L. Ma, G. Chen, and H. Wang, "Gearbox fault diagnosis method based on SVM trained by improved SFLA," in *Computational Intelligence, Networked Systems and Their Applications*, vol. 462, M. Fei, C. Peng, Z. Su, Y. Song, and Q. Han, Eds. 2014, pp. 257–263.
[24] X. Zhang, J. Wang, and Y. Gao, "A hybrid short-term electricity price forecasting framework: Cuckoo search-based feature selection with singular spectrum analysis and SVM," *Energy Econ.*, vol. 81, pp. 899–913, Jun. 2019.
[25] D. P. Solomatine and D. L. Shrestha, "AdaBoost.RT: A boosting algorithm for regression problems," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vols. 1–4, Jul. 2004, pp. 1163–1168.
[26] D. L. Shrestha and D. P. Solomatine, "Experiments with AdaBoost.RT, an improved boosting scheme for regression," *Neural Comput.*, vol. 18, no. 7, pp. 1678–1710, Jul. 2006.
[27] H.-X. Tian and Z.-Z. Mao, "An ensemble ELM based on modified AdaBoost.RT algorithm for predicting the temperature of molten steel in ladle furnace," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 73–80, Jan. 2010.
[28] S. Liu, J. Xu, J. Zhao, X. Xie, and W. Zhang, "Efficiency enhancement of a process-based rainfall–runoff model using a new modified AdaBoost.RT technique," *Appl. Soft Comput.*, vol. 23, pp. 521–529, Oct. 2014.
[29] S. Abe, "Fusing sequential minimal optimization and Newton's method for support vector training," *Int. J. Mach. Learn. Cyber.*, vol. 7, no. 3, pp. 345–364, Jun. 2016.
[30] S. Haykin, *Neural Networks and Learning Machines*. Beijing, China: Pearson, 2009.
[31] W. Keqi, X. Yonghua, and S. Liping, "Study on nonlinear compensation of eddy current sensor based on support vector machine," in *Proc. IEEE Int. Symp. Ind. Electron.*, Seoul, South Korea, Jul. 2009, pp. 133–137.
[32] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2007, pp. 120–127.
[33] Y. Chen, L. Li, J. Xiao, Y. Yang, J. Liang, and T. Li, "Particle swarm optimizer with crossover operation," *Eng. Appl. Artif. Intell.*, vol. 70, pp. 159–169, Apr. 2018.
[34] X.-S. Yang, "Firefly algorithm, Lévy flights and global optimization," *Res. Develop. Intell. Syst.*, vol. 26, pp. 209–218, 2010.
[35] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of L'evy stable stochastic processes," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 49, no. 5, pp. 4677–4683, Jul. 2002.
[36] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Inf. Sci.*, vol. 181, no. 20, pp. 4699–4714, Oct. 2011.
[37] T.-T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognit.*, vol. 48, no. 9, pp. 2839–2846, Sep. 2015.
[38] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.
[39] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, May 2016.
[40] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.
[41] H. Haklı and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Appl. Soft Comput.*, vol. 23, pp. 333–345, Oct. 2014.
[42] E. G. Ortiz-García, S. Salcedo-Sanz, Á. M. Pérez-Bellido, and J. A. Portilla-Figueras, "Improving the training time of support vector regression algorithms through novel hyper-parameters search space reductions," *Neurocomputing*, vol. 72, nos. 16–18, pp. 3683–3691, Oct. 2009.
[43] M. Varewyck and J.-P. Martens, "A practical approach to model selection for support vector machines with a Gaussian kernel," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 41, no. 2, pp. 330–340, Apr. 2011.
[44] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

JI LI received the B.E. degree in mechanical engineering from Nanchang Hangkong University, in 2010, and the Ph.D. degree in mechanical engineering from Xiamen University, in 2017. His research interests include machine learning, meta-heuristic algorithm, and advanced sensing technology.

CHENTAO ZHANG received the B.E. degree in precision instruments and machinery from the Harbin Institution of Technology, in 2010, and the Ph.D. degree in instrument science and technology from Xiamen University, in 2017. He is currently as the Vice Technical Director with Fujian (Quanzhou)–HIT Research Institute of Engineering and Technology. His research interests include machine vision, intelligent robot, and advanced sensing technology.

XUKUN ZHANG received the B.E. degree in agricultural machinery from Jiangxi Agricultural University, in 1985, and the Ph.D. degree in agricultural engineering from China Agricultural University, in 2005. His research interest is mechatronics and energy-saving drying technology.

HONGLIN HE received the B.E. degree in mechanical engineering from Northeastern University, China, in 1988, the M.D. degree in mechanical manufacturing from the Huazhong University of Science and Technology, in 1996, and the Ph.D. in mechatronics engineering from the Nanjing University of Aeronautics and Astronautics, in 2007. His research interest is mechatronics design and precise driving and control.

WENGUANG LIU received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics, China. He is currently working as an Associate Professor with Nanchang Hangkong University, China. His research interest is failure modes analysis of mechatronic equipment.

CAISEN CHEN was born in 1983. He received the master's degree in computer software and theory and the Ph.D. degree in network security technology from the Ordnance Engineering College, in 2009 and 2011, respectively. He is currently a Lecturer of the Academy of Army Armored Forces, Beijing. His current research interests include information security and implementation attack analysis on Cryptosystems. He also researches the security of mobile device.