Pattern Recognition xxx (xxxx) 108989

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com

A new algorithm for support vector regression with automatic selection of hyperparameters

You-Gan Wang [a, b, 1], Jinran Wu [a, 2, *], Zhi-Hua Hu [c, 3], Geoffrey J. McLachlan [d, 4]

[a] Queensland University of Technology, Brisbane, QLD 4001, Australia
[b] Australian Catholic University, Brisbane, QLD 4000, Australia
[c] Shanghai Maritime University, Shanghai 201306, China
[d] The University of Queensland, St Lucia, QLD 4072, Australia

ARTICLE INFO

Article history:
Received 20 December 2021
Received in revised form 19 July 2022
Accepted 16 August 2022

Keywords:
Automatic selection
Loss functions
Noise models
Parameter estimation
Probability regularization

ABSTRACT

The hyperparameters in support vector regression (SVR) determine the effectiveness of the support vectors with fitting and predictions. However, the choice of these hyperparameters has always been challenging in both theory and practice. The ν-support vector regression eliminates the need to specify an ϵ value elegantly, but at the cost of specifying or postulating a ν value. We propose an extended primal objective function arising from probability regularization leading to an automatic selection of ϵ, and we can express ν as an explicit function of ϵ. The resultant hyperparameter values can be interpreted as 'working' values required only in training but not testing or prediction. This regularized algorithm, namely ϵ^*-SVR, automatically provides a data-dependent ϵ and is found to have a close connection to the ν-support vector regression in the sense that ν as a fraction is a sensible function of ϵ. The ϵ^*-SVR automatically selects both ν and ϵ values. We illustrate these findings with some public benchmark datasets.

1. Introduction

The support vector machine (SVM) has been found to be very effective at prediction, together with many other tools in machine learning. Akin to statistical regression, an SVM establishes the estimates of the regression or classification parameters using a symmetric loss function, which equally penalizes over- and under-predictions. The most popular version of support vector regression (SVR) is ϵ-SVR with a threshold ϵ value: predictions with errors in the ϵ tube are regarded as perfect; otherwise, a loss of the deviation from this ϵ boundary will occur. For a given ϵ, the solution usually is obtained via the corresponding dual problem, and its computational complexity does not depend on data size. Notice that the solution and thus subsequent predictions do depend on the choice of ϵ. Typically, regression is regarded as a generalization of the classification problem to include cases where the target is real. As noted by Chang and Lin [1], the hyperparameters have some subtle differences in regression and classification. In this paper, we will focus on the ϵ-SVR where the target values are continuous.

To avoid choosing an ϵ in ϵ-SVR, Schölkopf et al. [2] introduced a fraction parameter ν that essentially also controls the number of support vectors leading to an automatic selection of ϵ. The relationship between ϵ-SVR and ν-SVR was well established by Chang and Lin [1]. The insensitivity parameter ϵ that controls the number of support vectors uses the parameter ν to effectively control the number of support vectors to eliminate the free parameter, ϵ [2]. The main idea of adding a penalty $\nu\epsilon$ in the loss function can be interpreted as adding $\nu\epsilon$ to every individual observation. This explains why, for given constants ν and ϵ, the SVR is unchanged. The other interpretation is that we select a proportion of ν observations (outside the tube) and impose an additional loss of ϵ to each of these observations. An appropriate ϵ can be chosen by minimizing the total extended loss, resulting in an ϵ value so that the proportion of support vectors is about ν. However, one drawback is that the choice of ν has an impact on the generalization of the model [3].

Considering the selection of a parameter ϵ may seriously affect a model's performance [4], practitioners generally have three key approaches to setting the hyperparameter. One option is to use the k-fold cross validation (or leave-one-out) to choose the parameters for the SVR [5] by grid search. When computationally feasible, a grid search can be carried out at pre-defined sets; see Chang and Lin [1] and Gupta and Gupta [6], among many others.

* Corresponding author.
E-mail address: j73.wu@qut.edu.au (J. Wu).
[1] Orcid: 0000-0003-0901-4671
[2] Orcid: 0000-0002-2388-3614
[3] Orcid: 0000-0003-4099-3310
[4] Orcid: 0000-0002-5921-3145

https://doi.org/10.1016/j.patcog.2022.108989
0031-3203/© 20XX

The second approach is to set the parameter as a function of estimated noise from the data. In the context of interval estimation, Jeng et al. [4] proposed using the ϵ based on the standard deviation of the errors. The proportionality constant is roughly equivalent to a quantile of errors, and hence the proportion of errors outside the ϵ-tube. Like the method used by Jeng et al. [4], Cherkassky and Ma [7] proposed the insensitivity parameter as a function of the sample size and data noise. As explored by them, the empirical formulation for $\hat{\epsilon}$ is calculated by the product of the empirical constant 3, the standard deviation of the noise, and an empirical coefficient $\sqrt{\ln n/n}$ (n is the sample size). However, if the sample size is increased, this $\hat{\epsilon}$ would approach 0, so this method does not recognize the noise level for insensitivity parameter estimation. More recently, Wen et al. [8] and Hsia and Lin [9] provided additional insightful guidelines on how these hyperparameters can be selected practically.

Different from former two approaches, the metaheuristic-based approach can adaptively search the tuning value based on minimizing objective function via updating search agents. The approach now is popular in tuning the parameter for ϵ-SVR. For example, Santos et al. [10] presented a Multi-Objective Differential Evolution (APMT-MODE) where they transformed the selection of hyperparameters as a multi-objective optimization problem where the proportion of support vectors and the error index are used as two objective functions that can balance model complexity and modelling performance to improve the generalization of ϵ-SVR. In the application of stock market prediction, Houssein et al. [11] proposed a combined forecasting model where the Equilibrium Optimizer (EO) [12] is used to optimize the hyperparameters in ϵ-SVR via minimizing forecasting error with cross-validation method. Similarly, another two recent combined models can be found in Wang et al. [13] and Cao et al. [14], where the Marine Predators Algorithm (MPA) [15] and the Henry Gas Solubility Optimization (HGSO) [16] are combined with SVR, respectively. It should be noted that the approach can tune all hyperparameters instantaneously with given search space for each hyperparameter [17]. We shall note that the approach is more intelligent than grid search but involves huge computation costs.

Up to now, as far as we know, how to set the ϵ or ν parameter values has been a primary difficulty for practitioners despite many successful applications of the SVR. This paper makes a threefold contribution, 1) establishing an extended primal objective function based on probability regularization leading to a data dependent proportion parameter ν and ϵ; 2) this regularization establishes the equivalence of the ϵ-SVR and ν-SVR in which ν is specified as a proportion of $\epsilon^{-1}\log(1+\epsilon)$; and 3) it proposes values that are equivalent to the maximum likelihood estimates under the distributional assumptions by the loss function; ν is an explicit function of ϵ. In addition, according to forecasting performance in the investigated datasets, our new algorithm can provide highly accurate predictions with less computational costs.

The rest of this paper is organized as follows. Section 2 reviews the framework of SVR. Section 3 presents an extension to scale issues, further extends the primal objective function based on probability regularization, which allows for automatic selection of the ϵ vale for any given dataset, establishes the connection of this approach to the well-known ν-SVR, and provides a dual solution for computation purposes. Section 4 provides examples to illustrate how the new algorithm connects with ν-SVR and compares it with five benchmark algorithms. Finally, concluding remarks and follow-up research problems for future studies are presented in Section 5.

2. Support vector regression (SVR)

Suppose that the training data consist of $(x_1, y_1), \ldots, (x_n, y_n) \in \chi \times \mathbb{R}$, where χ denotes the space of the input patterns. We consider the following noise model that generates the target y_i,

$$y_i = f(x_i) + u_i,$$

where u_i is the noise. The linear case corresponds to

$$f(x) = \langle \omega, x \rangle + b \quad \omega \in \chi, b \in \mathbb{R},$$

where $\langle .,. \rangle$ represents the dot product in χ.

In statistical regression models, $f(\cdot)$ can also be modeled as a nonlinear but known function with some unknown parameters as, for example, in growth studies. In the machine learning literature, a 'transformed' linear relationship is often assumed: first map x_i to $\phi(x_i)$, and then a linear relationship is adopted between y_i and $\phi(x_i)$. Here, $\phi(x_i)$ is also known as the characteristic vector. The resultant solution for the prediction of y at x, $f(x)$, can always be written as a linear combination of $k(x, x_i) = \phi(x)^{\mathrm{T}}\phi(x_i)$ via a kernel function k. The commonly used kernel functions include Gaussian, power, and polynomial functions. For the sake of simplicity, we will only consider the linear case as the extension to the nonlinear cases is straightforward after transforming x.

Given a model governed by a set of parameters, the maximum likelihood approach finds a 'best' model among all the feasible density function members. The traditional statistical estimation and inference will assume that the observed data are indeed generated by this family of distributions.

To proceed, let us denote the predicted or fitted values after model fitting (training) as $\hat{y}_i$ and the corresponding residuals $y_i - \hat{y}_i$ as r_i, which becomes the noise u_i in the model when $\hat{y}_i$ is the ideal value $f(x_i)$ (evaluated at the true ω values). The general SVR obtains the parameter estimates by minimizing the Euclidean norm, $\|\omega\|^2$, combined with a loss function $l(r_i)$ [18],

$$\text{minimize } \mathscr{L}(\omega, b) = \frac{1}{2}\|\omega\|^2 + C\left(n^{-1}\sum_{i=1}^{n} l(r_i)\right). \tag{1}$$

Here C is a parameter that controls the trade-off between the generalization ability and model fitness. As $C \to \infty$, when $l(r_i) = |r_i|$ the primal function becomes equivalent to the traditional median regression model without the penalty term $\|\omega\|^2$.

In general, when $l(r_i) = r_i^2$, the primal function (i.e., least square SVR) is closely related to the ridge regression where the primal objective is often written as

$$\left(n^{-1}\sum_{i=1}^{n} r_i^2\right) + \lambda\|\omega\|^2.$$

When $\|\omega\|^2$ is replaced by the L_1 norm $\|\omega\|$ in the parameter regularization, sparse solutions can be obtained (i.e., some parameters in ω will be shrunk to exactly 0). This is widely known as the least absolute shrinkage and selection operator (LASSO), which has been found to be very useful in variable selection, especially when there is many predictors [19].

Here, we will use the function $\mathscr{L}$ generically. The variables in $\mathscr{L}(.)$ indicate that minimization can be executed with respect to these variables. In ϵ-SVR, the methodology is based on finding an $f(x)$ so that its generation of the ϵ-tube contains as many observations as possible, and at the same time, is as flat as possible due to the term $\|\omega\|^2$ in the primal [18]. This is because the small errors ($\leq \epsilon$) are not counted in the loss function, and only larger errors will be accounted for in the loss function. The primal objective function can be written in the general form

$$\mathscr{L}(\omega, b|\epsilon) = \frac{1}{2}\|\omega\|^2 + C\left(n^{-1}\sum_{i=1}^{n}|r_i|_\epsilon\right), \tag{2}$$

where $|r_i|_\epsilon = \max\left(|r_i| - \epsilon, 0\right) = \{0, \text{ if } |r_i| < \epsilon, \text{ otherwise } |r_i| - \epsilon\}$. The notation $|\epsilon$ on the left-hand side of Eq. (2) means that it is given an ϵ value.

Note that the function $\mathscr{L}(\omega, b)$ in Eq. (2) is monotonic (decreasing) in ϵ, and it cannot be used in finding a reasonable ϵ value. To this end, Schölkopf et al. [20] proposed adding the term $\nu\epsilon$, which means assuming there is a fraction of ϵ cost associated with every observation ($0 \le \nu \le 1$),

$$\mathscr{L}(\omega, b; \epsilon|\nu) = \frac{1}{2}\|\omega\|^2 + C\left(\nu\epsilon + n^{-1}\sum_{i=1}^{n}|r_i|_\epsilon\right). \tag{3}$$

The new parameter ν controls the number of support vectors. More specifically, ν is an upper bound on the fraction of margin errors (outside the tube), and the fraction of support vectors must be $\ge \nu$. This simple but elegant approach will bypass specification of an ϵ value. Schölkopf et al. [20] suggested the ϵ parameter can be obtained by minimizing $\mathscr{L}(\omega, b; \epsilon|\nu)$ with respect to ϵ. It has been shown that ν is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors [20]. However, ν must be chosen *a priori*.

As illustrated by Vapnik [21], three important parameter settings in ϵ-SVR significantly impact the model's generalization: the regularization parameter C, the kernel parameters, and the insensitivity parameter ϵ. The first one, C, can be estimated as [7],

$$C_{CM} = \max\left(\left|\bar{y} + 3\sigma_y\right|, \left|\bar{y} - 3\sigma_y\right|\right). \tag{4}$$

Some researchers suggest using $C_{\max} = \max\left(|y_i|, i = 1, 2, \ldots, n\right)$ [2, 20].

The second setting is the most important parameter, ϵ, which controls the number of support vectors. We will explore how to estimate the insensitivity parameter ϵ from a statistical perspective, based on the loss function mechanism, in the next section. Third, the kernel parameter is applied to adjust the mapping from the original space to the high-dimensional space; this is decided by the type of kernel function and the application domain. Obviously, this is not an issue in linear cases.

The regularization parameter C, again, determines the trade-off between the flatness of f and the amount up to which deviations are greater than ϵ. The choice of C is a challenging one. In practice, the most widely used approach is minimizing the prediction accuracy calculated based on the cross-validation technique [22].

The constant $1/C$ is equivalent to the tuning/regularization parameter in LASSO literature. According to Wu and Wang [23], in the high-dimensional setting (the dimension of X_i, p, becomes large), C should be of the order of $\sqrt{n/\log(p)}$. While the choice of C is challenging enough, the additional task of finding a sensible ϵ value in ϵ-SVR makes this approach much less attractive from a practical perspective. Kaneko and Funatsu [24] proposed cross-validation together with grid search for these hyperparameters, but this approach is highly computing intensive.

3. The proposed new algorithm

3.1. Extension to a scale parameter

As pointed out by Chang and Lin [1], scaling of the target values is generally necessary so that the error ϵ is meaningful in all cases, although this is less a problem for classification, which is essentially based on the signs. When $\|\omega\|^2$ is involved in the primal objective function, as in Eq. (2), it is more sensible to standardize the predictors x_i so that the corresponding coefficient ω_i is not unit dependent. If some of the units are changed in any of the x_i variables, we wish to maintain the same prediction. For instance, if the second predictor x_2 is multiplied by 100 (the unit is changed from m to cm), the original ω^2 will be equivalent to $10000\,\omega_1^2 + \omega_2^2 + \ldots$ which should be quite different from using $\omega_1^2 + \omega_2^2 + \ldots$. A simple approach to achieving this is to divide each continuous predictor x_i by its standard deviation. For binary variables or more generally categorical variables, there is no need to do so because their values are already scaled and have simple interpretations. In fact, it is imperative to have predictions that are free of affine transformations of the data. The same concern applies to the target variable. It is awkward to have different results or predictions when a different unit is used for the target variable. To this end, we allow a scale parameter σ (a scalar) and replace r_i with r_i/σ in the loss function.

$$\mathscr{L}(\omega, b|\sigma) = \frac{1}{2}\|\omega\|^2 + Cn^{-1}\sum_{i=1}^{n} l\left(r_i/\sigma; \epsilon\right). \tag{5}$$

We can regard the r_i/σ as standardized residuals. The parameter ϵ becomes unit free and hence its meaning does not change in different examples. In fact, if we take σ as any positive value, σ_0, the optimal solution of ω does not change if we replace ϵ with $\sigma_0\epsilon$ and C with C/σ_0. We shall see this in the dual formation in Section VI. Our algorithm will take advantage of this invariant property and obtain an explicit solution for ϵ based on probability regularization. A naïve choice of σ is the standard deviation of y_i. However, we will show later that σ will be automatically chosen, also based on probability regularization, and it is related to the standard deviation of the noise u_i. Probability regularization enables us to apply maximum likelihood when the data are truly generated from the corresponding probability density function; more generally, the density function can be regarded as a working tool for approximating the true likelihood using the 'optimal' hyperparameters.

The loss in the ith prediction, $l\left(r_i/\sigma; \epsilon\right)$, can be regarded as the negative log-likelihood in which a constant is often ignored. The log scale is used so that the total loss is additive (and computationally convenient as well).

3.2. Extended primal function

In statistics, some distributional assumptions on u_i are often made in order to obtain estimates of the parameters in $f(\cdot)$. Instead of adopting a likelihood function that supposedly generated the noise u_i, we nominate a distribution $g_\theta(u)$ for u_i. We do not assume the $\{u_i, i = 1, 2, \ldots, n\}$ are generated from this distribution family $g_\theta(\cdot)$, but certain assumptions (such as mean 0) are needed to ensure good properties of the predictions.

Suppose a is a constant. If the loss function l with $l^* = l + a$ is replaced in Eq. (5), the solutions of (ω, b) via minimization of $\mathscr{L}$ will remain the same. This means, for any constant a, $\mathscr{L}(\omega, b|\sigma)$ is equivalent to the following extended primal objective function,

$$\begin{aligned}\mathscr{L}(\omega, b; \epsilon, \sigma) &= \frac{1}{2}\|\omega\|^2 + C\left[n^{-1}\sum_{i=1}^{n}\{l(r_i/\sigma; \epsilon) + a\}\right] \\ &= \frac{1}{2}\|\omega\|^2 + C\left\{n^{-1}\sum_{i=1}^{n} l\left(r_i/\sigma; \epsilon\right)\right\} + Ca.\end{aligned} \tag{6}$$

We now suggest a sensible choice of a that leads to automatic choice of ϵ and σ. We will apply probability regularization to l so that the $\int e^{-l^*(u/\sigma;\epsilon)}du = 1$. This leads to

$$\begin{aligned}a &= \log\left\{\int e^{-l(u/\sigma;\epsilon)}du\right\} \\ &= \log(\sigma) + \log\left[\int e^{-l(u;\epsilon)}du\right].\end{aligned}$$

In the case of the ϵ-SVR, simple algebra leads to

$$a = \log(2) + \log(1+\epsilon) + \log(\sigma).$$

The extended primal function now becomes [ignoring the constant $C\log(2)$]

$$\mathscr{L}(\omega, b; \epsilon, \sigma) = \frac{1}{2}\|\omega\|^2 + Cn^{-1}\sum_{i=1}^{n}\left|\frac{r_i}{\sigma}\right|_\epsilon + C \log(\sigma) + C\log(1+\epsilon). \tag{7}$$

With probability regularization, the loss function is equivalent to the maximum likelihood. The resultant estimates are equivalent to those via maximum likelihood under the distributional assumptions implied by the loss function. Its derivative can therefore be used as a statistical score function for the maximum likelihood estimate. This is the benefit of this extended primal function: it provides a working likelihood solution of ϵ when we do not actually assume the data are generated by this 'working' likelihood function [25,26].

By setting the derivative for ϵ to 0, we can obtain the automatic choice of ϵ as ϵ^* as

$$\epsilon^* = \frac{\sum_{i=1}^{n} I\left(|r_i| \le \sigma\epsilon^*\right)}{\sum_{i=1}^{n} I\left(|r_i| > \sigma\epsilon^*\right)}, \tag{8}$$

where I is the standard indicator function (taking a value of either 0 or 1).

Similarly, we set the derivative of $\mathscr{L}$ with respect to σ to 0, we have

$$\sigma^* = \frac{\sum_{i=1}^{n}\left|r_i\right| \cdot I\left(|r_i| > \sigma\epsilon^*\right)}{n}. \tag{9}$$

Thus, both parameters ϵ and σ can be obtained by minimizing Eq. (7) or jointly solving Eqs. (8) and (9). Akin to solving any nonlinear equations, an iterative approach starting with initial values is often needed. Our proposed new algorithm is trained by an iterative procedure on new computed residuals. Our experience indicates that one or two iterations are often adequate for practical use. This may be because the residual improvement of order $O_p(1/n)$ can be achieved after one iteration. For example, Lipsitz et al. [27] found that this is the case in updating the working correlation parameters, and Brown and Wang [28] also found this is the case in updating the hyper-parameter for smoothness in a rank estimation. Here, we generate the initial residuals with the ϵ-SVR with default setting which is recommended by Chang and Lin [29], and we use two iterations in our case studies.

As $n \to \infty$, we can obtain the limiting values of ϵ and σ for a given distribution of noise u_i. Suppose that $g(\cdot)$ is the density function of the noise term u_i. Asymptotically, Eq. (8) becomes,

$$\begin{cases} \frac{1}{\epsilon^*+1} = \mathrm{P}\left(|u_i| > \sigma\epsilon^*\right), \\ 1 = \int_{\epsilon^*}^{+\infty} u\left\{g(u) + g(-u)\right\} du. \end{cases} \tag{10}$$

The geometric meaning of (ϵ, σ) now becomes clear; $\epsilon^*/(1+\epsilon^*)$ is the proportion of errors within the tube, or in other words, $1/(1+\epsilon^*)$ is the proportion of support vectors. The parameter σ is the average distance of the support vectors, while the distance of non-support vectors is regarded as 0.

3.3. Connection to ν-SVR

We have now obtained an ϵ^* via probability regularization. The resultant ϵ^* will achieve the minimum value of given by Eq. (7). This result can also be regarded as the ν-SVR with $\nu_\epsilon = \log(1+\epsilon)/\epsilon$, shown in Fig. 1. For this given ν_ϵ, the same estimates of $(\omega, b; \epsilon, \sigma)$ will be obtained by minimizing

$$\mathscr{L}\left(\omega, b; \epsilon, \nu_\epsilon, \sigma\right) = \tfrac{1}{2}\|\omega\|^2 + C\left(\nu(\epsilon)\,\epsilon + n^{-1}\textstyle\sum_{i=1}^{n}\left|\frac{r_i}{\sigma}\right|_\epsilon + \log(\sigma)\right). \tag{11}$$

This primal function takes the same form as the ν-SVR given by Eq. (3). This observation establishes a close relationship of ϵ^*-SVR and ν-SVR and provides a sensible choice of $\nu = \nu^*$ in ν-SVR. It is easy to show that $0 \le \nu^* \le 1$, as required in ν-SVR. As for the scale parameter σ, we propose obtaining the estimated value using Eq. (9).

3.4. The dual solution

In general, we introduce the slack variables ξ_i and ξ_i^* to cope with the discontinuous nature of the derivatives, which will also lead to convenience in incorporating constraints on the parameters as well. Most literature on SVR concentrates on the dual optimization solutions. For completeness, we will briefly present the dual solutions. Also,Chapelle [30] indicated that the primal can be more efficient than the dual in optimization, or or vice versa, depending on the sample size n and the feature size p.

The optimization problem, Eq. (7), can be transformed to its dual problem as follows [1]:

$$\begin{aligned} \max\ G(\alpha_i, \alpha_i^*) = {} & -\tfrac{1}{2}\textstyle\sum_{i,j=1}^{n}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)\left\langle x_i, x_j\right\rangle \\ & + \textstyle\sum_{i=1}^{n} y_i\left(\alpha_i - \alpha_i^*\right) \\ \text{subject to} \quad & \begin{cases} \sum_{i=1}^{n}\left(\alpha_i - \alpha_i^*\right) = 0, \\ \sum_{i=1}^{n}\left(\alpha_i + \alpha_i^*\right)/\sigma \le \nu\breve{C}, \\ \alpha_i/\sigma, \alpha_i^*/\sigma \in \left[0, \breve{C}\right], \end{cases} \end{aligned} \tag{12}$$

with $\breve{C} = C/\sigma$. Here, α_i and α_i^* are Lagrange multipliers for $\epsilon + \xi_i - \left(y_i - f\left(x_i\right)\right)/\sigma$ and $\epsilon + \xi_i^* + \left(y_i - f\left(x_i\right)\right)/\sigma$, respectively. This dual optimization has a general solution,

$$f(x) = \sum_{i=1}^{n}\left(\alpha_i - \alpha_i^*\right)k\left(x_i, x\right) + b, \tag{13}$$

where the dual optimization is subjected to the constraints $0 \leqslant \alpha_i/\sigma, \alpha_i^*/\sigma \leqslant \breve{C}$, and $k\left(x_i, x\right)$ is the kernel function including the linear function as a special case.

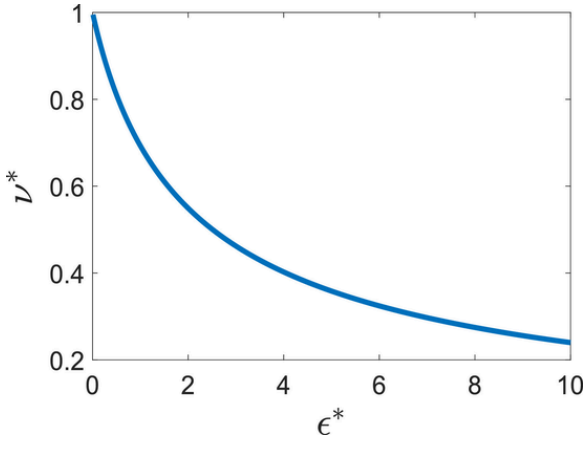

Fig. 1. The relationship between ν^* and ϵ^*.

4. Case studies

In this section, we design two experiments to show the connection with ν-SVR and validate the effectiveness of our new algorithm with benchmark algorithms, respectively.

4.1. Experiment I: Connection with ν-SVR

In this section, to show the connection with ν-SVR, we employed the proposed automatic selection of hyperparameters in three datasets from the UCI Machine Learning Repository [31], including the QSAR aquatic toxicity dataset (qsar), the yacht hydrodynamics dataset (yachts), and the concrete compressive strength dataset (concrete). Brief descriptions of these datasets are given in Table 1.

In each case study, the dataset is randomly divided into two groups: a training set (80% of each set) and a test set (the remaining 20%). Furthermore, to illustrate the efficiency of our algorithm, ν-SVR with 10 alternative ν values (0.1, 0.2, $\cdots$, 1.0) are selected as benchmark algorithms. For our parameter settings, following the experiment of Schölkopf et al. [20], the radial basic function $k(x,y) = \exp\left(-\gamma \|x-y\|^2\right)$ is set as the kernel function for the SVR training with the width $\gamma = 1/(0.3 \cdot N)$ (N is the dimensionality of predictors). Moreover, as recommended by Cherkassky and Ma [7], the regularization parameter C is set as $|y_i|_{(0.95)}$ for ν-SVR, while the regularization parameter $\check{C}$ in the dual problem was set as $|y_i/\sigma|_{(0.95)}$ in our proposed algorithm.

Table 1
Description of experimental datasets in experiment I.

Dataset	Sample size	Predictors	Response range
qsar	546	8	[0.12,10.05]
yachts	308	6	[0.01,62.42]
concrete	1030	8	[2.33,82.60]

Additionally, due to the different scale of the predictors in each set, the standard normalization is applied in predictor pre-processing before modelling. All experiments are implemented 2000 times to evaluate the performance of our proposed automatic algorithm.

To evaluate the performance of two indexes for the algorithms of interest, the mean of the root mean square error (RMSE) and the standard deviation of RMSE (RMSE.STD) were calculated in test sets for 2000 experiments. Also, the corresponding average ϵ and the average fraction of support vector (SVs) were obtained. The forecasting performances are displayed in Fig. 2. All experiments are implemented in R 3.6.2 on HPC with ncpus 8 and 32GB memory.

As shown in Fig. 2, the new SVR significantly improves forecasting accuracy in the three case studies. In addition, the performance of our algorithm shows good stability. To illustrate the underlying mechanism for its superior performance, some summarized indices and the execution time of our experiments is listed in Table 2.

Table 2 shows that our proposed algorithm for SVR can automatically select the hyperparameters ν^* and ϵ^* and can achieve superior performance with high accuracy in these cases. In other words, the method can recognize the pattern of a dataset and obtain a proper ν^* and ϵ^* based on the working likelihood approach. Furthermore, among all investigated datasets, our proposed algorithm with data-driven hyperparameters has the best performance. In the yachts forecasting, the RMSE with the proposed algorithm is 4.42, while that of the benchmark models ranged from 6.58 to 14.33. Additionally, in the SVR framework, it should be noted that the parameter ν can determine the insensitivity parameter ϵ to control the performance of the SVR. In our experimental settings, although 10 alternative ν values are given, it is still difficult to obtain a proper insensitivity parameter ϵ in real data. For example, in the concrete dataset, the new algorithm obtains 1.47 for ϵ ($\epsilon = \sigma^* \cdot \epsilon^*$) with $SVs = 0.71$, and the closest ϵ from ν-SVR with $\nu = 0.5$ (among 10 different ν values) is estimated as 1.80 with $SVs = 0.68$. More interestingly, based on the discussion in Section 3.3, the corresponding ν^* values can be estimated at 0.88, 0.98, and 0.81, respectively, for the qsar yachts, and concrete datasets.

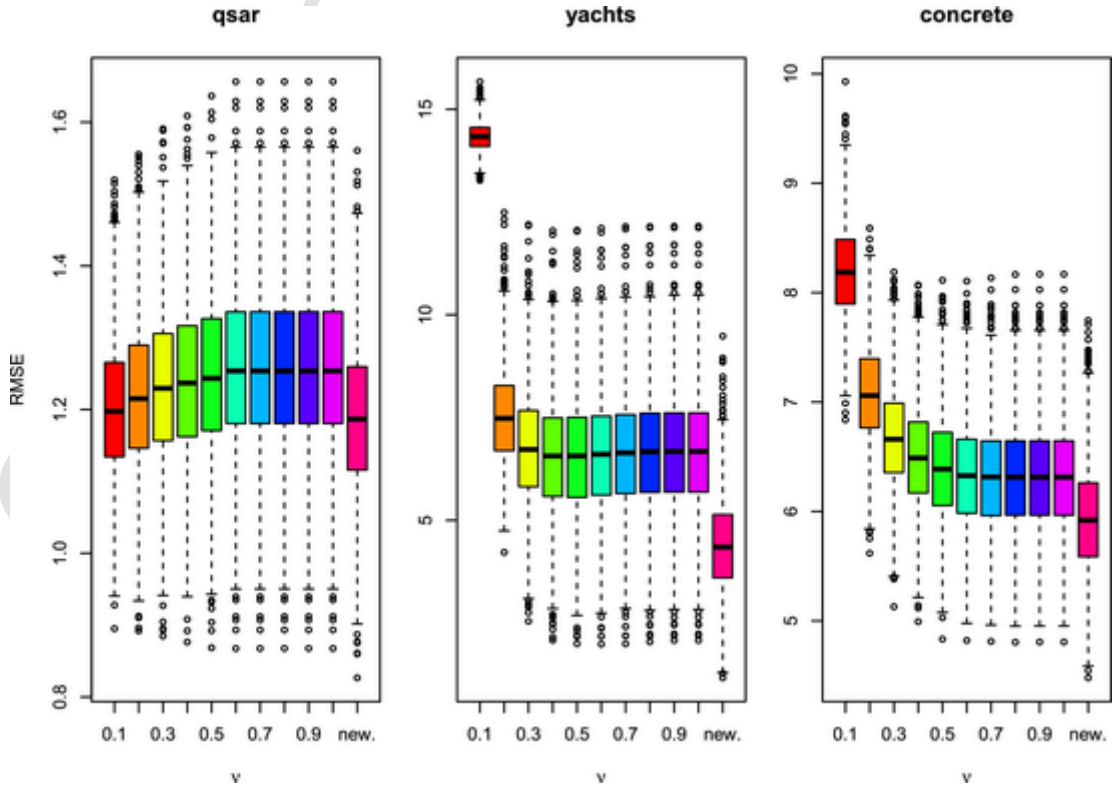

Fig. 2. The forecasting performances on test data (2000 random samples) in three cases.

Table 2
Results of experiments from the three cases.

Dataset		ν-SVR										New
qsar	ν	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	0.88[a]
	ϵ	0.87	0.54	0.33	0.19	0.08	0.00	0.00	0.00	0.00	0.00	0.14[b]
	SVs	0.38	0.54	0.68	0.81	0.92	1.00	1.00	1.00	1.00	1.00	0.85
	RMSE	1.20	1.22	1.23	1.24	1.25	1.26	1.26	1.26	1.26	1.26	1.19
	RMSE.STD	0.10	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
yachts	ν	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	0.98[a]
	ϵ	14.61	4.88	1.75	0.70	0.32	0.19	0.12	0.05	0.00	0.00	0.14[b]
	SVs	0.26	0.32	0.45	0.58	0.69	0.79	0.87	0.97	1.00	1.00	0.89
	RMSE	14.33	7.56	6.78	6.58	6.58	6.62	6.65	6.68	6.69	6.69	4.42
	RMSE.STD	0.34	1.17	1.38	1.43	1.44	1.45	1.45	1.46	1.46	1.46	1.18
concrete	ν	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	0.81[a]
	ϵ	10.28	6.26	4.22	2.90	1.80	0.96	0.34	0.01	0.00	0.00	1.47[b]
	SVs	0.20	0.33	0.45	0.57	0.68	0.79	0.89	0.99	1.00	1.00	0.71
	RMSE	8.20	7.08	6.67	6.50	6.40	6.34	6.32	6.32	6.32	6.32	5.93
	RMSE.STD	0.43	0.46	0.48	0.49	0.50	0.51	0.51	0.51	0.51	0.51	0.52

a Note that here ν^* is under the scale of response. b Because of the scale in our new algorithm, to compare with ν-SVR, the insensitivity parameter ϵ is back-transformed as: $\epsilon = \sigma^* \cdot \epsilon^*$.

To summarize our case studies, it can be concluded that our performances are close to the best performances that can be achieved by selecting a different ν in a test set. Furthermore, according to our proposed new algorithms, the insensitivity parameter ϵ^* and its corresponding parameter ν^* can be estimated based on the pattern of a dataset to efficiently improve the performance of the SVR with real data.

These examples illustrate that the new algorithm based on probability regularization can lead to sensible ϵ and ν values, leading to potentially improved results. Further, the meaning of ν, ϵ, and the scale parameter σ become very clear, and their easy interpretation is a bonus for data analysis. For the resulting ϵ^* value, there is a proportion of $1/(1+\epsilon^*)$ of support vectors and the corresponding ν value is $\log(1+\epsilon^*)/\epsilon^*$, a lower bound of the fraction of SVs. This is because it is easy to show that $(1+\epsilon)/\epsilon \log(1+\epsilon)$ is a monotonically increasing function in ϵ>0.

4.2. Experiment II: Comparison with benchmark algorithms

To show the effectiveness of our proposed new algorithm, we compare it with five benchmark algorithms for ϵ-SVR: CM Method [7], APMT-MODE [10], EO [12], MPA [15], and HGSO [16]. Here, we shall note that APMT-MODE, EO, MPA, and HGSO are recently state-of-the-art metaheuristic-based approaches for tuning hyperparameters in ϵ-SVR. In addition, we further investigate another five datasets (abalone, housing, mg, spacega, and triazines) from the UCI Machine Learning Repository [31] and LIBSVM Data: Regression. The description of these additional datasets is detailed in Table 3.

In this experiment, we fix the regularization parameter C and the kernel width γ with the tuning results with the recently APMT-MODE algorithm to fairly compare forecasting performance because our work focuses on the selection of the insensitivity parameter ϵ. Here, in our experiment, each dataset is divided into 80% for the training set and the remaining 20% for the test set. In details, following the work of Santos et al. [10], we first conduct the APMT-MODE algorithm to obtain Pareto Fronts and find the best solution of (C, γ, ϵ). After the tuning solution of (c, γ) with the APMT-MODE algorithm is obtained, we only optimize the insensitivity parameter ϵ by training ϵ-SVR in training set and obtain the tuning results of ϵ according to RMSE index with EO, MPA, and HGSO, respectively. The search ranges for these metaheuristic-based algorithms are set as $\gamma : \exp([-10 : 10])$, $\epsilon : (0, 10]$, and $C : \exp([-10, 10])$ according to Santos et al. [10]. Furthermore, in our new algorithm, we scale the response with the estimated scale σ^*, thus the regularization coefficient is set as C/σ^*. Here, taking an example, we record the results of tuning parameter (C, γ, ϵ) with all considered benchmark algorithms in one of our repeated experiments in Appendix A. To measure the efficiency of our new algorithm, we report on the RMSE and RMSE.STD in test sets as well as the execution time (Time) in Table 4.

Table 3
Description of experimental datasets in experiment II.

Dataset	Sample size	Predictors	Response range
abalone	4177	9	[1,29]
housing	506	13	[5,50]
mg	1385	6	[0.42,1.32]
spacega	3107	6	[-3.06,0.10]
triazines	186	60	[0.10,0.90]

As illustrated in Table 4, the proposed new algorithm is more effective than the considered five benchmark algorithms (CM Method, APMT-MODE, EO, MPA, and HGSO) with high accuracy in test set and less computational costs. Two points can be obtained as follows. First, in most datasets, the forecasting performance of the new algorithm is superior based on RMSE. For example, as for the housing dataset, the index of RMSE with the new algorithm is 4.13 while those of RMSE with CM Method, APMT-MODE. EO, MPA, and HGSO are 4.19, 5.34, 4.18, 4.18, and 4.17, respectively. Another point on computational cost makes clear that our new algorithm is more efficient than three recently state-of-the-art algorithms (EO, MPA, HGSO) and like CM Method and APMT-MODE for tuning hyperparameters. For instance, in the spacega dataset with 3107 samples, compared with three metaheuristic-based algorithms only for tuning insensitivity parameter, our proposed algorithm speeds up computation time as: 2392.75 (EO), 4526.50s (MPA), 4910.90s (HGSO), and 2283.25s (New), respectively.

Finally, the rank-based statistical test is introduced to show the significance among our new algorithm, and all considered benchmark algorithms in test sets. After calculation based on our results for eight cases, the average ranks based on RMSE are recorded in Table 4.

Table 5 shows the results based on the recorded RMSE ranks for the six algorithms. The Friedman χ^2 test gives a χ^2 value of 27.335 with $(6-1=5)$ degrees of freedom. The corresponding p-value is $4.91e-05$ indicating that these six algorithms are very different. Compared with other five benchmark models, our new algorithm performs the best in *all* cases leading to an average rank 1.00, while the average rank for APMT-MODE is 5.5. The other models produced the average ranks of 3.25, 1.88, 1.88 and 1.75 for CM Method, EO, MPA and HGSO, respectively. The high significance of the test is probably due to the superior performance of the new algorithm ϵ^*-SVR and the worse performance of APMT-MODE, quite consistently across all the cases except trazines in which APMT-MODE performs the second best).

Table 4
The numerical results with different benchmark algorithms for ϵ-SVR (the unit of execution time: second).

Dataset		CM Method [7]	APMT-MODE [10]	EO [12]	MPA [15]	HGSO [16]	New
abalone	RMSE	2.13	2.76	2.13	2.13	2.13	2.13
	RMSE.STD	0.08	0.03	0.08	0.08	0.08	0.08
	Time	1339.42	1334.20	2538.60	3772.80	3157.40	1342.98
concrete	RMSE	5.57	5.62	5.56	5.56	5.56	5.50
	RMSE.STD	0.62	0.39	0.61	0.61	0.61	0.58
	Time	676.25	675.52	1186.68	1725.72	1245.83	676.68
housing	RMSE	4.19	5.34	4.18	4.18	4.17	4.13
	RMSE.STD	0.97	0.63	0.99	0.99	0.98	0.98
	Time	47.00	46.98	57.53	67.53	59.39	47.52
mg	RMSE	0.20	0.23	0.20	0.20	0.20	0.19
	RMSE.STD	0.03	0.01	0.03	0.03	0.03	0.03
	Time	57.23	57.19	93.64	127.17	97.36	57.36
qsar	RMSE	1.17	1.39	1.19	1.19	1.19	1.16
	RMSE.STD	0.09	0.08	0.08	0.08	0.08	0.09
	Time	109.36	109.34	117.84	129.53	120.04	109.87
spacega	RMSE	0.11	0.14	0.11	0.11	0.11	0.11
	RMSE.STD	0.01	0.01	0.01	0.01	0.01	0.01
	Time	2281.74	2172.30	2392.75	4626.50	4910.90	2283.25
triazines	RMSE	15.43	15.41	15.41	15.41	15.41	15.30
	RMSE.STD	2.72	2.55	2.55	2.55	2.55	2.79
	Time	30.10	30.08	35.36	41.02	35.52	30.14
yachts	RMSE	2.66	3.27	1.49	1.49	1.49	1.44
	RMSE.STD	0.31	0.47	0.40	0.40	0.40	0.37
	Time	462.00	464.72	870.89	1247.53	922.27	463.70

Table 5
RMSE rankings of different tuning methods for ϵ-SVR.

Dataset	CM Method [7]	APMT-MODE [10]	EO [12]	MPA [15]	HGSO [16]	New
abalone	1	6	1	1	1	1
concrete	5	6	2	2	2	1
housing	5	6	3	3	2	1
mg	1	6	1	1	1	1
qsar	2	6	3	3	3	1
spacega	1	6	1	1	1	1
triazines	6	2	2	2	2	1
yachts	5	6	2	2	2	1
Average	3.25	5.50	1.88	1.88	1.75	1.00

To summarize the experimental results, we conclude our new SVR can provide good predictions with less computational costs. Our proposed method is based on probability regularization dependent on the quality of the investigated data. Thus, we can find in some datasets, some results are not always the best. The performance evaluated depends on several factors: firstly, the noise distribution is the most important, and secondly, the test data. If computation is not a concern, grid search can always find the best hyperparameters. This is also true for any optimization problem. However, 'guided' search is usually more efficient. Our approach can be regarded as a parametric approach instead of nonparametric grid search or semiparametric metaheuristic approaches. Considering the corresponding computational cost, our predictions are still effective compared with other benchmark methods.

5. Conclusion and future work

The key functionality of ν is to initiate a sensible automatic estimate of the ϵ value when ν is chosen *a priori*. In summary, our major contributions include 1) imposing the probability regularization on the parameter ϵ for automatic selection of ϵ, which leads to the discovery of ϵ^*; 2) a connection between ν and ϵ, specifically, $\nu^* = \log(1+\epsilon^*)/\epsilon^*$, and 3) we also propose standardizing the noise, which will make all the hyperparameters more meaningful to interpret. The examples also clearly show that substantial improvement may arise when the scale parameter σ is introduced. For any given ϵ value, the traditional SVR, ν-SVR, and proposed ϵ^*–SVR are all equivalent and produce the same results.

Accuracy and variability of predictions are affected by several factors, indeed, all factors in the model. Although the ϵ-Laplacian loss distribution is a mainstream algorithm for regression modelling, it should be of great interest to explore other loss functions. Other loss functions may be more justifiable based on the performance and interpretability, and sometimes even robustness, when the presence of outliers becomes a concern [7].

Computation is always a concern, especially when analysing large datasets. This is because it is related to feasibility and use of computing resources. The proposed ϵ^*-SVR does not need to evaluate the errors based on cross-validation or many subsamples (randomly sampled from the whole sample available), and thus it is computationally simpler and potentially more applicable to larger datasets.

In addition, the new algorithm also standardizes the scale based on the estimated noise variance (automatically) in a more meaningful field. The examples also demonstrate the capability of the new algorithm to automatically estimate both the scale and the insensitivity parameters with improved forecasting accuracy. Therefore, we hope more applications can be explored to test this new algorithm.

It is well known that the value of C should be proportional to the input noise level [7]. Practically, the range or sample variance of the target values is often calculated to assist in the choice of C. Our framework has incorporated a scale parameter σ, and it seems we can consider C being proportional to σ (instead of the sample standard deviation, which is often much larger). Future work can also include the choice of L_1 norm (absolute error) as with LASSO. The performance of our automatic choice of hyperparameters in L_1-SVR [32], especially, in variable selection in the cases of high dimension, is of great practical interest. Our algorithm still requires input values from other hyperparameters. Another limitation is that we have only focused on the SVR. It is not clear if a similar technique can be developed for classification. There is also a need for us to generalize to extreme learning machines and neural networks with multi-layers, which have been found to achieve better generalization performance [33]. Here, we did not investigate the impact of different kernels and kernel parameter (such as γ) values in this work but focused on ϵ and ν. The excessive computation burden is another concern.

An important generalization to the SVR is by Peng [34], who proposed the twin support vector regression involving a pair of nonparallel functions corresponding to the ϵ-insensitive down- and up-bounds of

the unknown regressors. The number of regressions and hyperparameters are doubled. Much progress has been made in developing more efficient algorithms [35]. Moreover, our proposed approach can be generalized to improved $I\nu$-TWSVM, $I\nu$-TWSVM in Khemchandani et al. [36], and ν-TWSVR in Rastogi et al. [37]. Further novel generalization by Xu et al. [38] using pinball loss function asymmetric ν-twin SVR can handle the asymmetric noise and outliers, making the prediction more efficient and robust as well. Gupta and Gupta [6] further improved this algorithm based on regularization and the structural risk minimization principle. Also, our proposed method has the potential to handle the hyperparameter selection in the general robust loss function by Barron [39] and re-scaled hinge loss by Singla et al. [40]. Several realistic scenarios show the superiority of their elegant algorithm. It will be of much interest to establish the convergence rate and to develop some analytical solutions for any of the parameters so that their framework can be used in large-scale data analysis.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests nor personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the Australian Research Council projects DP160104292 and CE140100049.

Appendix A. Tuning results of the hyperparameters

Table A.1

Table A.1
The tuning results with different benchmark algorithms for ϵ-SVR.

	APMT-MODE [10]			CM Method [7]	EO [12]	MPA [15]	HGSO [16]	New
Datasets	ϵ	C	γ	ϵ	ϵ	ϵ	ϵ	ϵ^1
abalone	4.99	153.91	6.87	0.29	1.68	1.65	1.55	0.20
concrete	4.99	2626.60	2.74	1.21	0.87	0.85	0.85	1.65
housing	9.81	89.14	7.47	1.44	1.17	1.17	1.18	0.56
mg	0.49	0.02	0.00	0.02	0.11	0.11	0.11	0.03
qsar	3.00	207.80	4.13	0.23	0.30	0.30	0.30	0.16
spacega	0.49	630.12	14.60	0.02	0.08	0.08	0.08	0.02
triazines	0.46	15.38	0.00	0.06	1.41	4.90	2.60	0.01
yachts	4.52	2053.80	4.28	3.41	0.45	0.44	0.45	0.28

1 The reported corresponding insensitivity parameter with scale for the new algorithm is calculated as $\epsilon = \epsilon^* \cdot \sigma^*$.

References

[1] C.-C. Chang, C.-J. Lin, Training v-support vector regression: theory and algorithms, Neural Comput. 14 (8) (2002) 1959–1977.
[2] B. Schölkopf, A.J. Smola, R.C. Williamson, P.L. Bartlett, New support vector algorithms, Neural Comput. 12 (5) (2000) 1207–1245.
[3] B. Schölkopf, P. Bartlett, A. Smola, R. Williamson, Support vector regression with automatic accuracy control. International Conference on Artificial Neural Networks, Springer, 1998, pp. 111–116.
[4] J.-T. Jeng, C.-C. Chuang, S.-F. Su, Support vector interval regression networks for interval regression analysis, Fuzzy Sets Syst. 138 (2) (2003) 283–300.
[5] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Science & Business Media, 2009.
[6] U. Gupta, D. Gupta, An improved regularization based lagrangian asymmetric ν-twin support vector regression using pinball loss function, Appl. Intell. 49 (10) (2019) 3606–3627.
[7] V. Cherkassky, Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, Neural Netw. 17 (1) (2004) 113–126.
[8] Z. Wen, B. Li, R. Kotagiri, J. Chen, Y. Chen, R. Zhang, Improving efficiency of SVM k-fold cross-validation by alpha seeding. Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 2768–2774.
[9] J.-Y. Hsia, C.-J. Lin, Parameter selection for linear support vector regression, IEEE Trans. Neural Netw. Learn. Syst. 31 (12) (2020) 5639–5644.
[10] C.E.d.S. Santos, R.C. Sampaio, L. dos Santos Coelho, G.A. Bestard, C.H. Llanos, Multi-objective adaptive differential evolution for SVM/SVR hyperparameters selection, Pattern Recognit. 110 (2021) 107649.
[11] E.H. Houssein, M. Dirar, L. Abualigah, W.M. Mohamed, An efficient equilibrium optimizer with support vector regression for stock market prediction, Neural Comput. Appl. 34 (4) (2022) 3165–3200.
[12] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: a novel optimization algorithm, Knowl. Based Syst. 191 (2020) 105190.
[13] G. Wang, X. Zeng, G. Lai, G. Zhong, K. Ma, Y. Zhang, Efficient subject-independent detection of anterior cruciate ligament deficiency based on marine predator algorithm and support vector machine, IEEE J. Biomed. Health Inform. (2022).
[14] W. Cao, X. Liu, J. Ni, Parameter optimization of support vector regression using henry gas solubility optimization algorithm, IEEE Access 8 (2020) 88633–88642.
[15] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A.H. Gandomi, Marine predators algorithm: a nature-inspired metaheuristic, Expert Syst. Appl. 152 (2020) 113377.
[16] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: a novel physics-based algorithm, Future Gener. Comput. Syst. 101 (2019) 646–667.
[17] R.S. Rao, A.R. Pais, P. Anand, A heuristic technique to detect phishing websites using TWSVM classifier, Neural Comput. Appl. 33 (11) (2021) 5733–5752.
[18] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.
[19] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (Methodological) 58 (1) (1996) 267–288.
[20] B. Schölkopf, P.L. Bartlett, A.J. Smola, R.C. Williamson, Shrinking the tube: a new support vector regression algorithm. Advances in Neural Information Processing Systems, 1999, pp. 330–336.
[21] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2013.
[22] K. Ito, R. Nakano, Optimizing support vector regression hyperparameters based on cross-validation. Proceedings of the International Joint Conference on Neural Networks, vol. 3, IEEE, 2003, pp. 2077–2082.
[23] Y. Wu, L. Wang, A survey of tuning parameter selection for high-dimensional regression, Annu. Rev. Stat. Appl. 7 (2020) 209–226.
[24] H. Kaneko, K. Funatsu, Fast optimization of hyperparameters for support vector regression models with highly predictive ability, Chemom. Intell. Lab. Syst. 142 (2015) 64–69.
[25] Y.-G. Wang, X. Lin, M. Zhu, Z. Bai, Robust estimation using the Huber function with a data-dependent tuning constant, J. Comput. Graph. Stat. 16 (2) (2007) 468–481.
[26] L. Fu, Y.-G. Wang, F. Cai, A working likelihood approach for robust regression, Stat. Methods Med. Res. 29 (12) (2020) 3641–3652.
[27] S.R. Lipsitz, G.M. Fitzmaurice, E.J. Orav, N.M. Laird, Performance of generalized estimating equations in practical situations. Biometrics, 1994, pp. 270–278.
[28] B.M. Brown, Y.-G. Wang, Standard errors and covariance matrices for smoothed rank estimators, Biometrika 92 (1) (2005) 149–158.
[29] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst.Technol. (TIST) 2 (3) (2011) 27.
[30] O. Chapelle, Training a support vector machine in the primal, Neural Comput. 19 (5) (2007) 1155–1178.
[31] D. Dua, C. Graff, UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.
[32] J. Zhu, S. Rosset, R. Tibshirani, T.J. Hastie, 1-norm support vector machines. Advances in Neural Information Processing Systems, 2004, pp. 49–56.
[33] S. Balasundaram, D. Gupta, On optimization based extreme learning machine in primal for regression and classification by functional iterative method, Int. J. Mach. Learn. Cybern. 7 (5) (2016) 707–728.
[34] X. Peng, Tsvr: an efficient twin support vector machine for regression, Neural Netw. 23 (3) (2010) 365–372.
[35] Q. Hou, J. Zhang, L. Liu, Y. Wang, L. Jing, Discriminative information-based nonparallel support vector machine, Signal Process. 162 (2019) 169–179.
[36] R. Khemchandani, P. Saigal, S. Chandra, Improvements on ν-twin support vector machine, Neural Netw. 79 (2016) 97–107.
[37] R. Rastogi, P. Anand, S. Chandra, A ν-twin support vector machine based regression with automatic accuracy control, Appl. Intell. 46 (3) (2017) 670–683.
[38] Y. Xu, Z. Yang, X. Pan, A novel twin support-vector machine with pinball loss, IEEE Trans. Neural Netw. Learn. Syst. 28 (2) (2016) 359–370.
[39] J.T. Barron, A general and adaptive robust loss function. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4331–4339.
[40] M. Singla, D. Ghosh, K.K. Shukla, W. Pedrycz, Robust twin support vector regression based on rescaled hinge loss, Pattern Recognit. 105 (2020) 107395.

You-Gan Wang received the PhD degree in statistics from Oxford University, Oxford, U.K., in 1991. He is currently a Professor with the Australian Catholic University, Brisbane, QLD, Australia. His research interests include developing statistical methodology for correlated data analysis, robust inferences and model selection and applying advanced techniques that help to solve important problems in medical sciences, environmental research, and natural resource management. In applied statistics, he has been working with multidiscipline teams on a wide range of problems. Their findings have significant impacts in resource management (fisheries and hydrology) and clinical trials (biostatistics). Professor Wang has been invited on several occasions to organize/speak at international conferences and to review journal papers. His work has substantial impacts and scientific innovations in statistical modelling and data science. He is on the Editorial Board of Biometrics, Scientific Reports, and Environmental Modelling and Assessment.

Jinran Wu received his PhD degree in statistics from Queensland University of Technology (QUT), Brisbane QLD, Australia, in 2022. He is currently an associate lecturer with the School of Mathematical Sciences, QUT. He is interested in statistical machine learning and optimizations with applications in power systems, environmental science, bioinformatics, and management science. He was awarded the Australian Government Research Training Program Scholarship (International) in 2018 and was a recipient of the 2021 Chinese Government Award for Outstanding Self-financed Students Abroad. He is a Review Editor on the Editorial Board of Machine Learning and Artificial Intelligence (specialty section of Frontiers in Big Data and Frontiers in Artificial Intelligence).

Zhi-Hua Hu received his PhD degree in control science and engineering from Donghua University, China, in 2009. Before that, he had worked for about ten years as Software Developer and Project Manager with information technology industry. From 2009 to now, he was a Researcher with the Logistics Research Center, Shanghai Maritime University. Since 2014, he has been a professor with management science and engineering. He is the author of more than 150 journal articles. His research interests include logistics operations optimization, big data system and management, artificial intelligence, and algorithms.

Geoffrey J. McLachlan received the BSc (Hons.) and PhD degrees from the University of Queensland, Brisbane QLD, Australia, in 1969 and 1973, respectively. Since 1975, he has been a Faculty Member with the Department of Mathematics, University of Queensland. He has authored over 270 research articles, including six monographs. Dr. McLachlan is a fellow of the American Statistical Association, the Royal Statistical Society, and the Australian Mathematical Society. He was a recipient of the DSc in 1994, an Australian Research Council in 2007 Professorial Fellowship, the Pitman Medal of the Statistical Society of Australia in 2010, the research medal of the International Federation of Classification Societies (IFCS), and the IEEE ICDM Research Contributions Award in 2011. He was elected as a fellow of the Australian Academy of Science in 2015. He is on the Editorial Board of several international journals and has served on the Program Committee for many international conferences. He has been a member of the College of Experts of the Australian Research Council and a past President of the IFCS.