A working likelihood approach to support vector regression with a data-driven insensitivity parameter

Jinran Wu and You-Gan Wang

School of Mathematical Sciences, Queensland University of Technology, 2 George St., Brisbane, 4001, Queensland, Australia.

*Corresponding author(s). E-mail(s): you-gan.wang@qut.edu.au;
Contributing authors: j73.wu@qut.edu.au;

Abstract

The insensitivity parameter in support vector regression determines the set of support vectors that greatly impacts the prediction. A data-driven approach is proposed to determine an approximate value for this insensitivity parameter by minimizing a generalized loss function originating from the likelihood principle. This data-driven support vector regression also statistically standardizes samples using the scale of noises different from conventional response scaling method. Statistical standardization together with probabilistic regularization based on a working likelihood function produces data-dependent values for the hyperparameters including the insensitivity parameter. The exact asymptotical solutions are provided when the noises are normally distributed. Nonlinear and linear numerical simulations with three types of noises (ϵ-Laplacian distribution, normal distribution, and uniform distribution), and in addition, five real benchmark data sets, are used to test the capacity of the proposed method. Based on all the simulations and the five case studies, the proposed support vector regression using a working likelihood, data-driven insensitivity parameter is superior and has lower computational costs.

Keywords: Approximate loss function, Parameter estimation, Prediction, Working likelihood

1 Introduction

In the machine learning field, support vector regression (SVR) has been popular in management and engineering applications [1–3], due to its solid theoretical foundation [4–6] and insensitivity to the dimensionality of the samples [7]. As recommended by [8], the parameter settings in SVR modelling contribute the generalization of the predictive performance. However, practitioners applying SVR in real-world applications often cannot obtain the most effective model. There are two key approaches to setting the hyperparameter. One option is to use the k-cross validation to choose the parameters for SVR [9, 10]. The other approach is to set the parameter as a constant, based on the empirical practice developed by [5]. In particular, the researchers suggested that the regularization parameter C and the insensitivity parameter ϵ be set at 1.0 and 0.1, respectively. However, although the tuning parameter setting provides an acceptable generalization in most conditions, there is still a huge gap between this solution and the best SVR using the optimal parameters.

1.1 Literature review

For the insensitivity parameter ϵ that controls the number of support vectors [11], [12] used the parameter ν to effectively control the number of support vectors to eliminate the free parameter, ϵ. However, one drawback is that the choice of ν has an impact on the generalization of the model [13]. Furthermore, insensitivity parameter estimation methods that consider the noises in observations have been developed. [14] proposed to estimate the insensitivity parameter in two steps. The first step is to estimate the regression errors by the SVR at $\epsilon = 0$. Then, the ϵ value is updated by $c\hat{\sigma}$ with an empirical constant c and the estimated standard deviation of the noise $\hat{\sigma}$. In the absence of outliers, the standard deviation can be calculated based on all the regression errors, and c is set as 1.98. Otherwise, a trimmed estimator is obtained by removing $5 - 10\%$ of samples at both ends to achieve robustness, and c is recommended to be fixed at 3. Obviously, although [14]'s method aims to incorporate data size in the estimation, the empirical settings make the method unable to recognize the noise level to estimate the insensitivity parameter ϵ. Like [14]'s method, [15] incorporated sample size into the insensitivity parameter estimation. 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, when the sample size increases, this $\hat{\epsilon}$ would approach to 0, so this method does not recognize the noise level for the insensitivity parameter estimation. Now, more recent literature on tuning parameters in the SVR can be found in [16] and [17].

Different from tuning the insensitivity parameter ϵ directly, in the reference of [6], the authors propose to train ν-support vector regression (ν-SVR) where a new parameter ν is introduced for controlling the proportion of support vectors. In the framework of ν, with the parameter ν, the insensitivity parameter can be optimized with other parameters together. Apparently, the parameter ν-SVR would determine the selection of the support vectors but must be prior given. Therefore, cross-validation method based on a pre-set ν sequence with huge computational costs or an empirical setting is used for the implementation of ν-SVR.

Because the selection of the insensitivity parameter ϵ can be regarded as a complex optimization problem with several local mini-ma, meta-heuristic algorithms have been popularly used to tune the insensitivity parameter in ϵ-SVR [18] to overcome the problem of the gradient directed algorithms. One of the typical examples is the work on estimating the residential building energy consumption by [19] where a multi-verse optimizer is employed for tuning ϵ for ϵ-SVR with cross-validation. Considering actual applications, researchers have searched for the tuning ϵ in ϵ-SVR [18] with meta-heuristic algorithms, such as moth flame optimization (MFO) [20], whale optimization algorithm (WOA) [21], grey wolf optimizer (GWO) [22], grasshopper optimization algorithm (GOA) [23], flower pollination algorithm (FPA) [24], differential evolution [25], and particle swarm optimization [26]. This kind of combined method based on cross-validation often requires high computational costs to obtain a good optimum for the insensitivity parameter. Compared with cross-validation method, meta-heuristic algorithms are used to find the potential solution according to fitness function values during search process instead of a pre-set potential solution set. It should be noted that although meta-heuristic algorithms can provide a good solution to tune the insensitivity parameter, more computation costs are required in practice.

1.2 Contribution

To reduce the computational cost for tuning the insensitivity parameter, we in this paper will derive an elegant statistical formula to estimate the value of ϵ. As explained by [8], the insensitive loss function consists of the least modulus (LM) loss and the special Huber loss function when $\epsilon = 0$. Hence, in our study, considering the insensitive Laplacian distribution loss function inspired by [4] and [27], we focused on the insensitivity parameter ϵ and propose a novel SVR with a data-driven (D-D) insensitivity parameter. Like [14] and [15]'s work, our method is developed on the theoretical background of SVR instead of parameter estimation based on re-sampling. Motivated by [28], we propose designating the working likelihood to estimate the insensitivity parameter for SVR. In other words, the working likelihood method can estimate appropriate hyper-parameters to find the most appropriate ϵ-Laplacian distribution to the real noise distribution. Our working likelihood (or D-D) method works as a vehicle for the ϵ loss function parameter estimation. In addition, different from the computational standardization, the target in the proposed model is standardized in a statistical manner using the scale of the noise. Thus, our D-D method is more practicable and intelligent. In our simulations (linear and nonlinear), three types of error distributions were used to test the D-D insensitivity parameter estimation, namely, the insensitive Laplacian distribution, normal distribution, and uniform distribution. Furthermore, some case studies were applied to validate that our D-D SVR has novel generalization in real applications. The meaning of key symbols are clarified in Table 1.

1.3 Organization of the paper

This rest of this paper is organized as follows. Section 2 describes the basic framework of ϵ-SVR. Section 3 illustrates the working likelihood method for insensitivity parameter estimation in ϵ-SVR and present some asymptotic properties of our estimate of scale and insensitivity parameter. Numerical simulations for three different types of noise sources (the insensitive Laplacian distribution, normal distribution, and uniform distribution) were implemented, and Section 4 presents a discussion of the analyses of the simulation results, which illustrate the effectiveness of the working likelihood. Then, in Section 5, we validate the superiority of our D-D SVR on five real data sets: energy efficiency, Boston housing, yacht hydrodynamics, airfoil self-noise, and concrete compressive strength according to the forecasting accuracy and the computational cost. Finally, in Section 6, we summarize the results that indicate the working likelihood (D-D) method has superior performance on insensitivity parameter estimation based on the real noise information in SVR, indicating that our D-D SVR is very effective in handling forecasting problems.

2 The support vector regression (SVR)

Assume the training data $(x_1,\ y_1),\ldots,(x_n,\ y_n) \in \chi \times \mathbb{R}$, where χ denotes the space of the input patterns. The case of linear function $f(\cdot)$ can be formed as

$$f(x) = \langle \omega, x\rangle + b \quad \omega \in \chi, b \in R, \tag{1}$$

where $\langle \cdot, \cdot\rangle$ represents the dot product in χ. In ϵ-SVR, the target is to obtain a function $f(x)$ that has at most ϵ deviation from the actual obtained target y_i for all the training data, and at the same time, is as flat as possible [7, 29]. This means that smaller errors ($\leq \epsilon$) are ignored, and larger errors will be accounted for in the loss function. Flatness in Eq. (1) means finding a small ϵ. Now, the objective function for the basic SVR can be presented with a ridge penalty $\|\omega\|^2$ and an ϵ-Laplace loss r_ϵ with residuals $r_i = y_i - f(x_i)$ [29],

$$\min_{\omega,b} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n} |r_i|_\epsilon \tag{2}$$

where a regularization parameter C (a positive constant) is introduced to determine the trade-off between the flatness of f and the amount up to which deviations are larger than ϵ. Here, we define $|r_i|_\epsilon$ as $\max\{z^+, z^-\}$ with $z^+ = \max\{r_i - \epsilon, 0\}$ and $z^- = \max\{-r_i - \epsilon, 0\}$. Notice that the optimization problem is feasible; it means that there exists such a function f that approximates all pairs $(x_i,\ y_i)$ with ϵ precision. Then, the slack variables ξ_i and ξ_i^* are introduced to cope with the otherwise infeasible constraints of the optimization version in Eq. (2). Now, the formulation is shown as,

$$\min_{\omega,b,\xi_i,\xi_i^*} \quad \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$
$$s.t. \quad \begin{cases} y_i - \langle \omega, x_i\rangle - b \leqslant \epsilon + \xi_i, \\ \langle \omega, x_i\rangle + b - y_i \leqslant \epsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geqslant 0. \end{cases}$$

The primal problem of the basic SVR can be transformed to the corresponding dual problem as

Table 1 Nomenclature.

Notation	Description	Notation	Description	Notation	Description
s	scale of noise	ϵ	insensitivity parameter	C	regularization parameter
n	sample size	p	dimension of predictors	R^2	coefficient of determination
r_i	i-th residual	u_i	i-th standardized residual	x_i	features of the i-th sample
$V(\cdot)$	loss function	$g(\cdot)$	working density function	y_i	response of the i-th sample
ϵ^*	asymptotic ϵ value	$L(\cdot)$	Joint likelihood function	s^*	asymptotic σ value
$h(\cdot)$	true density function	$\mathbb{I}$	indicator function	MAE	mean absolute error
$\hat{\epsilon}$	estimated ϵ	$\hat{s}$	estimated s	RMSE	root mean square error
CV	cross validation	CM	Cherkassky& Ma's method	D-D	data-driven method
$\text{ratio}_{\text{MAE}}$	ratio of MAE	$\text{ratio}_{\text{RMSE}}$	ratio of RMSE	μ_i	i-th clean response
γ	kernel parameter				

follows [29]:

$$\begin{aligned}\max_{\alpha,\alpha^*} \quad & -\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle x_i,x_j\rangle \\ & -\epsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*) \\ s.t. \quad & \begin{cases}\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0, \\ \alpha_i,\alpha_i^*\in[0,C].\end{cases}\end{aligned}$$

Here, α_i and α_i^* are Lagrange multipliers for $\epsilon+\xi_i-y_i+\langle\omega,x_i\rangle+b$ and $\epsilon+\xi_i^*-\langle\omega,x_i\rangle-b+y_i$, respectively. This dual optimization has a general solution,

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

where the offset b can be estimated according to the KKT conditions, and $k(x_i,x)$ is the kernel function including linear function as a special case.

As illustrated by [8], three important parameter settings in SVR significantly impact the model's generalization: the regularization parameter C, the kernel parameter γ, and the insensitivity parameter ϵ. The first one, C, can be estimated by the 0.95 quantile of $|y_i|$ [15],

$$C_{CM}=|y_i|_{(0.95)}, i=1,\ldots,n.$$

In addition, [30] pointed out that when the dimension p of predictors are very large, the regularization parameter C can be of the order of $\sqrt{n/\log(p)}$.

Then, the second kernel parameter γ in kernel functions (e. g., radial basis function kernel and polynomial kernel) 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. The last one is the most important parameter, ϵ, which controls the number of support vectors. We will explore how to estimate the insensitivity parameter ϵ based on the loss function mechanism from a statistical perspective in the next section.

3 The data-driven SVR

3.1 Working likelihood for insensitivity parameter estimation

Suppose the training data set consists of n samples $(x_i,y_i),(i=1,2,\ldots,n)$, and the target y_i, is generated from the following model:

$$y_i=f(x_i)+r_i=f(x_i)+s\cdot u_i,$$

where $f(\cdot)$ represents the expected value, while the second component, r_i (which is decomposed as su_i) is the noise (s is the scale, and u_i is the noise after scaling s).

In ϵ-SVR, the loss function is defined as

$$\begin{aligned}V(r)&=|r|_\epsilon, \\ &=\begin{cases}r-\epsilon & r>\epsilon, \\ 0 & -\epsilon\leqslant r\leqslant\epsilon, \\ -r-\epsilon & r<-\epsilon,\end{cases}\end{aligned}\tag{3}$$

where $r = y - \langle \omega, x \rangle - b$ is the residual item. The corresponding density function for r_i is,

$$g(r;\ \epsilon) = \frac{1}{2(1+\epsilon)} \exp(-|r|_\epsilon),$$

which will correspond to the loss function given by Eq. (3) up to a constant.

Thus, suppose that all r_i are identically and independently distributed with a density function $g(\cdot)$. Let θ be a vector collecting all the unknown parameters (ϵ, s). The negative log-likelihood based on the training data is then

$$\begin{aligned} -\log L(\theta) = -\textstyle\sum_{i=1}^n \log\left(g\left(\frac{y_i - f(x_i)}{s}\right)\right) \\ +n\log(s). \end{aligned}$$

Once the SVR approach is adapted, we essentially assume r_i follows a density function that is proportional to $\exp(-V(r))$. Our working likelihood D-D method estimates all the parameters in θ by maximizing $L(\theta)$ [31]. We investigate the choice of the insensitivity parameter ϵ in the SVR approach. Clearly, the ϵ value that results by maximizing L is data dependent and expected to be more effective. Meanwhile, the scale of the noise s can also be estimated.

Next, recalling that $r_i = su_i$, assume that $r_1, r_2, \ldots r_n$ are independent and identically distributed random variables. Denote $(\epsilon, s) = \theta$. Their joint working likelihood function is

$$\begin{aligned} L(\theta) &= \prod_{i=1}^n \left(\frac{1}{s} g(\frac{r_i}{s};\ \epsilon, s)\right) \\ &= \left(\frac{1}{s}\right)^n \cdot \left(\frac{1}{2(1+\epsilon)}\right)^n \cdot \exp\left(-\sum_{i=1}^n |\frac{r_i}{s}|_\epsilon\right). \end{aligned}$$

Therefore, $L(\theta)$ is a likelihood function with parameters ϵ and s properly regularized.

Theorem. *Suppose that $(\hat{\epsilon}, \hat{s})$ are the estimates by minimizing L, and (ϵ^*, s^*) are the limiting values of $(\hat{\epsilon}, \hat{s})$. Under the mild assumption of $E(r_i^2) < +\infty$, we have*

$$\begin{cases} \epsilon^* = \dfrac{\displaystyle\int_0^{s^*\epsilon^*} \{h(r) + h(-r)\}dr}{\displaystyle\int_{s^*\epsilon^*}^{\infty} \{h(r) + h(-r)\}dr} \\ s^* = \displaystyle\int_{s^*\epsilon^*}^{\infty} (h(r) + h(-r)) \cdot rdr \end{cases} \tag{4}$$

where $h(\cdot)$ is the true density function of the noise term r_i.

Proof First, the estimators of θ can be achieved by minimizing the negative log-likelihood function,

$$\begin{aligned} -\log L(\theta) &= n\log s + n\log\left[2(1+\epsilon)\right] + \sum_{i=1}^n |\tfrac{r_i}{s}|_\epsilon \\ &= n\log s + n\log\left(2(1+\epsilon)\right) \\ &\quad + \sum_{i=1}^n \left((\tfrac{r_i}{s} - \epsilon) \cdot \mathbb{I}(\tfrac{r_i}{s} > \epsilon)\right) \\ &\quad + \sum_{i=1}^n \left((-\tfrac{r_i}{s} - \epsilon) \cdot \mathbb{I}(\tfrac{r_i}{s} < -\epsilon)\right). \end{aligned} \tag{5}$$

Next, the derivatives of $(-\log L(\theta))$ with respect to ϵ and s are given as

$$\begin{cases} \dfrac{\partial\left(-\log L(\theta)\right)}{\partial\epsilon} = \dfrac{n}{1+\epsilon} - \displaystyle\sum_{i=1}^n \mathbb{I}(|\frac{r_i}{s}| > \epsilon), \\ \dfrac{\partial\left(-\log L(\theta)\right)}{\partial s} = \dfrac{n}{s} - \dfrac{1}{s^2}\displaystyle\sum_{i=1}^n |r_i| \cdot \mathbb{I}(|\frac{r_i}{s}| > \epsilon). \end{cases}$$

The working likelihood approach to (ϵ, s) estimates is equivalent to solving the following equations,

$$\begin{cases} \frac{1}{\epsilon+1} = \frac{1}{n}\sum_{i=1}^n \mathbb{I}(|r_i| > \epsilon s), \\ s = \frac{1}{n}\sum_{i=1}^n |r_i| \cdot \mathbb{I}(|r_i| > \epsilon s). \end{cases} \tag{6}$$

Under the assumption of $E(r_i^2) < +\infty$, we have $E(|r_i|) < \sqrt{\{E(r_i^2)\}} < +\infty$, and $E\{|r_i|\mathbb{I}(|r_i| > \epsilon^* s^*)\}^2 \leqslant E(r_i^2) < +\infty$, the law of large numbers hence holds for the two terms on the right-hand side of Eq. (6). Taking the limit as $n \to +\infty$, we obtain

$$\begin{cases} \frac{1}{\epsilon^*+1} = E\mathbb{I}(|r_i| > \epsilon^* s^*) = \int_{s^*\epsilon^*}^{+\infty} \{h(r) + h(-r)\}dr \\ s^* = E|r_i| \cdot \mathbb{I}(|r_i| > \epsilon^* s^*) = \int_{s^*\epsilon^*}^{+\infty} (h(r) + h(-r))\, rdr \end{cases},$$

which is equivalent to Eq. (4).

□

Remark 1 According to Eq. (4), the meaning of (ϵ^*, s^*) is clear. This indicates that ϵ^* is the odds ratio of being inside the box ($\leqslant \epsilon^*$) versus outside the box ($\geqslant \epsilon^*$). The parameter s^* is the average distance of the support vectors, while the distance of non-support vectors is regarded as 0.

Corollary 1 Suppose that $(\hat{\epsilon}, \hat{s})$ are the estimates by minimizing L, and (ϵ^*, s^*) are the limiting values of $(\hat{\epsilon}, \hat{s})$. If the true density function $h(\cdot)$ is ϵ-Laplacian distribution ($\epsilon > 0$), there exists a unique solution of limiting values (ϵ^*, s^*).

Proof If the true probability density function of the noise is ϵ-Laplacian,

$$h(r) = \frac{1}{2\sigma(1+\epsilon)} \exp\left(-|\frac{r}{\sigma}|_\epsilon\right). \tag{7}$$

Plugging $h(r)$ from Eq. (7) into Eq. (4), and we can obtain

$$\begin{cases} \frac{1}{1+\epsilon^*} = \dfrac{1}{1+\epsilon} \cdot \exp\left(\dfrac{\sigma\epsilon - s^*\epsilon^*}{\sigma}\right), \\ s^* = \dfrac{s^*\epsilon^* + \sigma}{1+\epsilon} \cdot \exp\left(\dfrac{\sigma\epsilon - s^*\epsilon^*}{\sigma}\right). \end{cases}$$

From the first sub-equation we have $\exp(\frac{\sigma\epsilon - s^*\epsilon^*}{\sigma}) = \frac{1+\epsilon}{1+\epsilon^*}$, which can be plugged into the second sub-equation on the right hand side, which simplifies to $s^* = \sigma$. The ϵ^* can be obtained by solving

$$\frac{1+\epsilon}{\exp(\epsilon)} = \frac{1+\epsilon^*}{\exp(\epsilon^*)}, \epsilon > 0.$$

Denote $t(\epsilon^*) = \frac{1+\epsilon^*}{\exp(\epsilon^*)}$, the derivative of $t(\epsilon^*)$ with respect to ϵ^* can be given as $t'(\epsilon^*) = -\epsilon^* \exp(-\epsilon^*) < 0$. This means $t(\epsilon^*)$ is strictly monotonic. In general, if $t(\epsilon^*)$ is strictly monotonic, $t(\epsilon^*) = t(\epsilon)$ implies $\epsilon^* = \epsilon$. Therefore, ϵ is a unique solution of ϵ^*. □

Corollary 2 If the true density function of the noise $h(\cdot)$ is normally distributed with mean 0 and standard deviation $\sigma < +\infty$, the limiting values ϵ^* and s^* are 1.524 and 0.557σ, respectively. This implies that the corresponding limiting value of the insensitivity parameter for the raw residuals without standardization is 0.848σ.

Proof Substituting the normal density function to Eq. (4), we can obtain

$$\begin{cases} \dfrac{s^*}{\sigma} = \frac{2}{\sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2}\epsilon^{*2}\left(\dfrac{s^*}{\sigma}\right)^2\right), \\ \dfrac{1}{1+\epsilon^*} = 2\left(1 - \Phi\left(\epsilon^*\left(\dfrac{s^*}{\sigma}\right)\right)\right). \end{cases}$$

Clearly, the solution $s^* = \sigma\tau$ where τ is the solution when $\sigma = 1$, i.e., we have invariant property $s^*(\sigma) = \sigma \cdot s^*(1)$. Thus, let $\tau = s^*(1)$, and we have

$$\begin{cases} \tau = \frac{2}{\sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2}(\epsilon^*\tau)^2\right), \\ \dfrac{1}{1+\epsilon^*} = 2\left(1 - \Phi(\epsilon^*\tau)\right). \end{cases}$$

This shows $\epsilon^*(\sigma) = \epsilon^*(1)$ which is a constant free from σ. Furthermore, the solution of the equation can be achieved as $\epsilon^* = 1.524$ and $\tau = 0.557$. Therefore, we can have the final solution as $\epsilon^* = 1.524$ and $s^* = 0.557\sigma$. Finally, we can obtain the estimate of the insensitivity parameter for the raw residuals without standardization as $s^* \cdot \epsilon^* = 0.848\sigma$. □

When the variance σ changes, the s^* changes proportionally as $s^* = 0.557\sigma$ and the corresponding insensitivity tube also varies accordingly with the radius $s^* \cdot \epsilon^* = 0.848\sigma$ while keeping the standardized tube unchanged (ϵ^* does not change with σ). This means, if the target unit is changed from *cm* to *mm*, for example, the new σ becomes larger, as 10σ, our D-D method can adaptively control the width of the tube appropriately so that the same prediction results will be obtained by automatically updating the hyperparameters. Interestingly, according to the limiting result, for any normal distributed error, because of $s^* \cdot \epsilon^* = 0.848\sigma$, the proportion of support vectors is kept roughly as $2 - 2\Phi(0.848) = 0.396$.

Remark 2 It should be noted that the optimization objective (5) is non-convex and more than one solutions exist for Eq. (6). ($\epsilon = 0, s = \sum_{i=1}^{n} |r_i|$) always is a solution of Eq. (6). Therefore, to handle such an optimization problem, considering the popularity of normal distribution, we set the initial values of ϵ and s as 1.524 and 0.557, respectively, for our optimization in this paper where limited-memory BFGS [32] is employed as optimizer. In addition, we also recommend using some meta-heuristics algorithms, such as particle swarm optimization (PSO) method [33], and repeat the optimization procedure and report the best solution with the most smallest value of the optimization objective (5) from all candidate solutions.

Each paired $\theta = (\epsilon, s)$ value corresponds to a potential key to a real data set. We now propose obtaining the "best" key in the toolbox. Figure 1 shows some potential keys for inferring the unknown noise. This means the ϵ-Laplacian distribution can approximate the real noise distribution by adapting the scale parameter s and the insensitivity parameter ϵ.

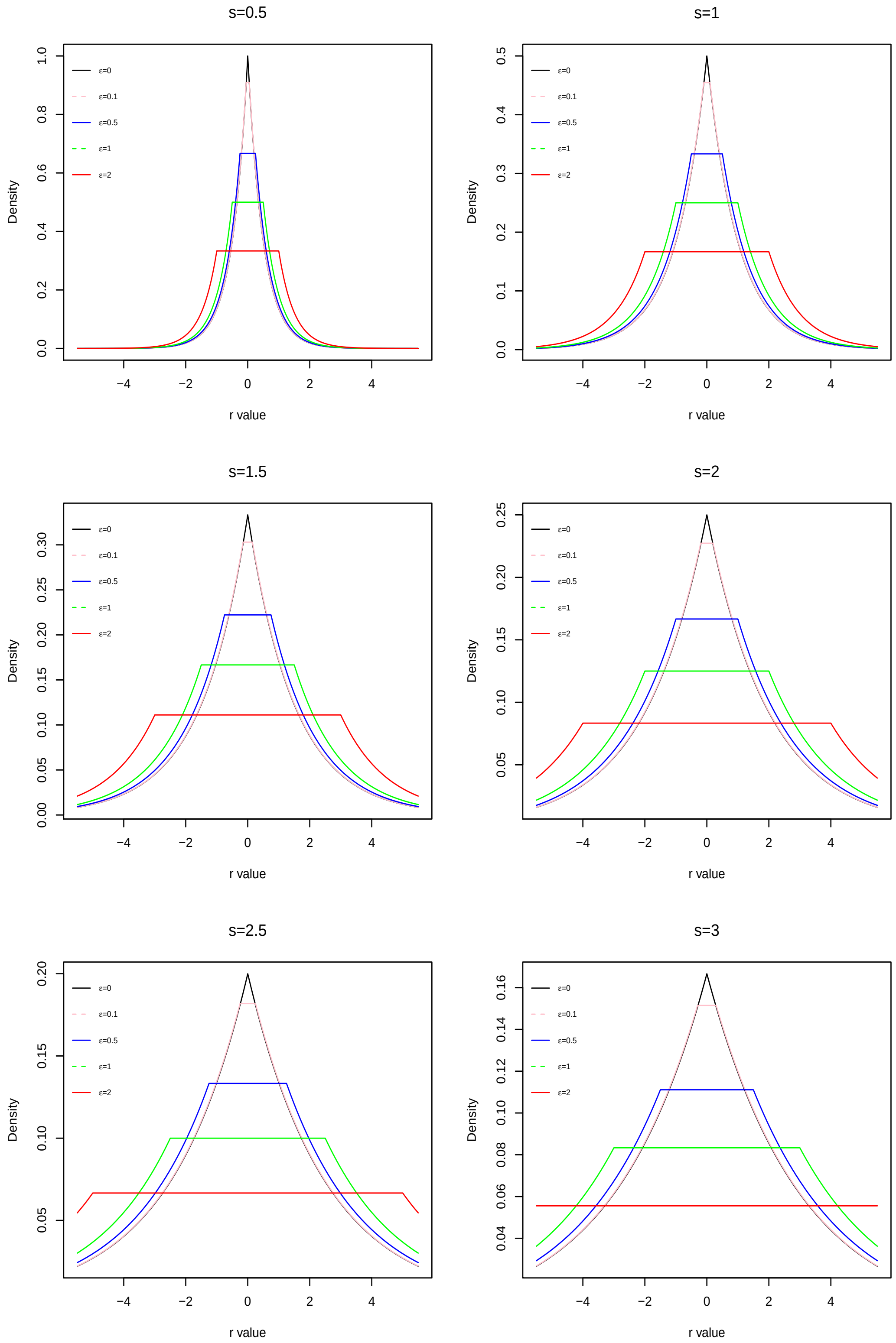

Fig. 1 Working likelihood functions with different insensitivity parameters at different scales.

3.2 The training procedure of our D-D SVR

Now, the full objective function for our proposed D-D SVR can be formulated as:

$$\min_{\omega,b,\epsilon,s} \tfrac{1}{2}\|\omega\|^2 \\ +C\left\{n\log s + n\log\left[2(1+\epsilon)\right] + \sum_{i=1}^{n}|\tfrac{r_i}{s}|_\epsilon\right\}.$$

In details, during the iterative training procedure, with given residuals r_i, the paired θ can be estimated as $(\hat{\epsilon},\hat{s})$ via minimizing Eq. (5). Then, a simplified objective function in our iterative procedure can be formulated as:

$$\min_{\omega,b} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}|\frac{r_i}{\hat{s}}|_{\hat{\epsilon}}. \tag{8}$$

Furthermore, Eq. (8) can be indirectly solved via R package ‘e1071’ [34] with scaled response $y_i/\hat{s}$ and the corresponding scaled regularization coefficient $C = |y_i/\hat{s}|_{(0.95)}$.

In brief, the pseudo code for our proposed SVR with D-D insensitivity parameters is given in Algorithm 1. To implement our D-D SVR, the maximum number of iteration t_{Max} and the threshold of the change of mean square error Δ_{Min} must be given. Moreover, the computational complexity for our proposed D-D method is affected by the basic ϵ-SVR part and the hyperparameter estimation part. The complexity of ϵ-SVR is $O(n^2 \times p + n^3)$ with the number of feature p and the complexity for estimating hyperparameter is f_{hp}. Therefore, the computational complexity for our method is $O(T(n^2 \times p + n^3 + f_{\text{hp}}))$ where T is the number of iteration.

In our D-D SVR training, one or two iterations generally is adequate for real practice because the residual improvement of order $O_p(1/n)$ after one iteration. A similar point also has been found in the references of [35] and [36]. In addition, we also can conclude the point in our case studies where the convergence curves are reported.

Algorithm 1 The pseudo code for our proposed D-D SVR

```
Input: $(x_i, y_i)$, $i = 1, 2, \dots, n$
Output: $(\omega^f, b^f)$
$t = 0$
$\epsilon_0 = 0.1$
$C_0 = 1.0$
Normalize predictors $x_i$ as $x_i^z$, $i = 1, 2, \dots, n$
Normalize responses $y_i$ as $y_i^0$, $i = 1, 2, \dots, n$
Obtain $(\omega_0, b_0)$ via minimizing Formula (2) with default parameter setting $(\epsilon_0, C_0)$ by using standardized data $(x_i^z, y_i^0)$, $i = 1, 2, \dots, n$
Obtain initial residuals $r_i^0 = y_i - \hat{y}_i^0$ where $\hat{y}_i^0$ is from $\epsilon$-SVR with $(\omega_0, b_0)$
Calculate MSE$_0$ with $y_i$ and $\hat{y}_i^0$, $i = 1, 2, \dots, n$
for $t = 1, 2, \dots, t_{\text{Max}}$ do
    $t = t + 1$
    Update $(\hat{\epsilon}^t, \hat{s}^t)$ via minimizing Formula (5) with $r_i^t$, $i = 1, 2, \dots, n$
    Update regularization parameter $C^t = |y_i^t/\hat{s}^t|_{(0.95)}$
    Obtain $(\omega_t, b_t)$ via minimizing Formula (8) with $(\hat{\epsilon}_t, C_t)$ by using data $(x_i^z, y_i)$, $i = 1, 2, \dots, n$
    Update residuals $r_i^t = y_i - \hat{y}_i^t$ where $\hat{y}_i^t$ is from $\epsilon$-SVR with $(\omega_t, b_t)$, $i = 1, 2, \dots, n$
    Calculate MSE$_t$ with $y_i$ and $\hat{y}_i^t$, $i = 1, 2, \dots, n$
    $\Delta_t = |\text{MSE}_t - \text{MSE}_{t-1}|$
    if $\Delta_t \leqslant \Delta_{\text{Min}}$ then
        Break
    end if
end for
$\omega^f = \omega_t$ and $b^f = b_t$
```

4 Simulation experiments

To illustrate how the working likelihood produces D-D parameter estimation (D-D) and a prediction, we now consider three types of residuals generated from the uniform distribution, the norm distribution, and the ϵ-Laplacian distribution, respectively.

For comparison, we will investigate other three insensitivity parameter estimation methods for the ϵ-SVR. The first one is the tuning parameter setting (tuning) ($C = 1.0$ and $\epsilon = 0.1$) [5]. The second method, [15]'s empirical parameter approach (CM), is

$$\epsilon_{CM} = 3\sigma_{\text{noise}}\sqrt{\frac{\ln n}{n}},$$

where the standard deviation of noise σ_{noise} is obtained from the residuals using $\epsilon = 0$. The last one is the k-cross validation (k-CV), where k is fixed at 10, and 5 alternative ϵ settings are set as $0.01, 0.05, 0.1, 0.2$ and 0.3. Both mean absolute error (MAE) and root mean square error (RMSE) are calculated for comparison as

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|,$$

and

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

where $\hat{y}_i$ is the i-th prediction, and y_i is the i-th observation. For each method X using the tuning method as the benchmark approach, two ratios are defined as

$$\text{ratio}_{\text{RMSE}} = \frac{\text{RMSE}_{\text{tuning}}}{\text{RMSE}_X},$$

and

$$\text{ratio}_{\text{MAE}} = \frac{\text{MAE}_{\text{tuning}}}{\text{MAE}_X}.$$

It is obvious that the method X beats the tuning setting only if the ratio is larger than 1, and otherwise, it does not. The nonlinear simulations and linear simulations are applied to show the efficiency of our proposed D-D SVR.

4.1 Nonlinear regression

To demonstrate the performance of our D-D SVR for nonlinear system modelling, the univariate *sinc* target function from the SVR literature [7, 37–39] is considered as

$$y_i = a \cdot \frac{\sin(x_i)}{x_i} + s \cdot u_i, \quad i = 1, 2, \ldots, n,$$

where x_i is generated from the uniform distribution $unif[-10, 10]$; s is the scale of the noise level; and the standard noise u_i is generated from a known distribution (ϵ-Laplacian distribution, normal distribution $N(0, \sigma^2)$, and uniform distribution $unif[-bd, bd]$). In addition, to make our simulations more meaningful, the scale of nonlinear system a is set as 5, 4, and 6 from insensitive-Laplacian noises, normal noises, and uniform noises, respectively. Also, we generate n simulation samples, and then the samples are divided into two groups of the same size. All experiments are repeated 100 times to calculate the average performance of the benchmark SVRs and our proposed D-D SVR. The kernel of the SVR is the default radial basic function $k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$ with $\gamma = 1$ [34]. It should be noted that, for our comparison, the ratio is calculated based on the gap between the prediction $\hat{y}_i$ and the μ_i ($\mu_i = a\sin(x_i)/x_i$). This can show the performance of our D-D SVR at eliminating the interruption from noise and model a real system. All the nonlinear simulation results are displayed in Table 2 (insensitive Laplacian distribution), Table 3 (normal distribution), and Table 4 (uniform distribution).

As illustrated in Table 2, compared with the CM and 10-CV, the ratios of the D-D from both RMSE and MAE are significantly greater than 1, indicating that our proposed SVR allowed for remarkable improvements in the forecasting performance for all 27 simulations. However, the insensitivity parameter ϵ tends to be underestimated. The main reason for this is that, as shown in Figure 1, the scale mainly contributes to the working likelihood function when the insensitivity parameter is small. Another reason is that the training sample size is not large enough to estimate the insensitivity parameter accurately. As the training set size enlarges, the estimated insensitivity parameter converges to the true ϵ.

Table 3 shows the second case, where the errors follow normal distributions. Our proposed method works well for approximating the best ϵ-Laplacian distribution, leading to significant improvements in the forecasting accuracy of all the simulation scenarios displayed in the Table. When the noise level is low (both s and σ are small), the superiority of the D-D approach is more prominent. For the simulation with noise settings (n 1000, s 0.7, and σ 0.5), the D-D's prediction achieves an amazing improvement (MAE, 64%, and RMSE, 48%), while both the CM and 10-CV methods each obtained only a slight increase. In the simulation setting with $n = 200$, $s = 1.1$, and $\sigma = 1.5$ (i.e., noises contribute more to responds), we have checked our simulations where one of the simulations are with plenty of large outliers. The performance of our method is heavily depended on the quality of data; as a result, our forecasting performance is not good.

The third nonlinear case also shows that our D-D method is an effective approach to data modelling with noises from the uniform distribution,

Table 2 Nonlinear case (ϵ-Laplacian distribution): Relative performance of the CM, 10-CV, and D-D methods in comparison to the tuning approach.

Noise settings			Parameters		CM		10-CV		D-D	
n	s	ϵ	$\hat{s}$	$\hat{\epsilon}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$
200	0.8	0.2	0.66	0.00	0.95	0.97	0.99	1.00	1.47	1.32
400	0.8	0.2	0.73	0.02	0.94	0.95	1.11	1.00	1.71	1.55
1000	0.8	0.2	0.77	0.01	0.95	0.95	1.11	1.01	1.86	1.69
200	0.8	0.5	0.70	0.00	0.96	0.97	1.07	1.01	1.44	1.36
400	0.8	0.5	0.77	0.04	0.95	0.96	1.09	1.00	1.66	1.52
1000	0.8	0.5	0.82	0.06	0.96	0.96	1.10	1.01	1.70	1.56
200	0.8	1.0	0.81	0.03	0.96	0.97	1.06	1.00	1.30	1.22
400	0.8	1.0	0.87	0.15	0.97	0.97	1.09	1.00	1.54	1.42
1000	0.8	1.0	0.87	0.53	0.99	0.99	1.10	1.01	1.71	1.56
200	1.0	0.2	0.85	0.00	0.96	0.97	1.10	1.01	1.47	1.34
400	1.0	0.2	0.93	0.01	0.94	0.95	1.10	1.00	1.66	1.51
1000	1.0	0.2	0.98	0.01	0.94	0.94	1.10	1.01	1.78	1.64
200	1.0	0.5	0.89	0.00	0.95	0.96	1.07	0.99	1.41	1.31
400	1.0	0.5	0.97	0.02	0.95	0.96	1.08	1.00	1.55	1.44
1000	1.0	0.5	1.03	0.05	0.96	0.96	1.08	1.01	1.65	1.54
200	1.0	1.0	0.99	0.04	0.98	0.98	1.05	1.00	1.24	1.18
400	1.0	1.0	1.08	0.15	0.97	0.98	1.05	1.00	1.42	1.35
1000	1.0	1.0	1.08	0.57	1.00	1.00	1.09	1.01	1.67	1.54
200	1.2	0.2	1.02	0.00	0.94	0.95	1.08	1.00	1.39	1.28
400	1.2	0.2	1.10	0.00	0.93	0.94	1.09	1.00	1.55	1.41
1000	1.2	0.2	1.17	0.01	0.93	0.93	1.11	1.01	1.73	1.58
200	1.2	0.5	1.08	0.02	0.95	0.95	1.06	1.00	1.26	1.19
400	1.2	0.5	1.16	0.01	0.95	0.95	1.07	0.99	1.45	1.34
1000	1.2	0.5	1.24	0.06	0.96	0.97	1.08	1.01	1.64	1.52
200	1.2	1.0	1.20	0.07	0.99	0.98	1.04	1.00	1.18	1.14
400	1.2	1.0	1.32	0.14	0.99	0.99	1.06	1.01	1.29	1.22
1000	1.2	1.0	1.31	0.49	1.00	1.01	1.09	1.02	1.52	1.42

Table 3 Nonlinear case (normal distribution): Relative performance of the CM, 10-CV, and D-D methods in comparison to the tuning approach.

Noise settings			Parameters		CM		10-CV		D-D	
n	s	σ	$\hat{s}$	$\hat{\epsilon}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$
200	0.7	0.5	0.22	0.00	0.96	0.98	1.00	1.00	1.37	1.16
400	0.7	0.5	0.24	0.00	0.97	0.97	1.01	1.01	1.67	1.46
1000	0.7	0.5	0.24	0.47	1.00	1.00	1.03	1.03	1.64	1.48
200	0.7	1.0	0.44	0.00	0.97	0.98	1.00	1.00	1.29	1.20
400	0.7	1.0	0.49	0.06	0.97	0.98	1.00	1.00	1.48	1.36
1000	0.7	1.0	0.45	0.88	0.98	0.99	1.01	1.01	1.50	1.38
200	0.7	1.5	0.66	0.03	0.98	0.98	1.00	1.00	1.17	1.12
400	0.7	1.5	0.70	0.31	0.97	0.98	1.00	1.00	1.38	1.29
1000	0.7	1.5	0.65	1.05	0.98	0.99	1.01	1.01	1.48	1.36
200	0.9	0.5	0.28	0.00	0.97	0.99	1.00	1.01	1.42	1.25
400	0.9	0.5	0.31	0.01	0.96	0.97	1.00	1.01	1.58	1.42
1000	0.9	0.5	0.30	0.68	0.99	0.99	1.02	1.02	1.59	1.45
200	0.9	1.0	0.57	0.00	0.97	0.97	1.00	1.00	1.17	1.10
400	0.9	1.0	0.62	0.17	0.97	0.98	1.00	1.01	1.33	1.23
1000	0.9	1.0	0.57	0.90	0.98	0.99	1.00	1.00	1.48	1.36
200	0.9	1.5	0.86	0.10	0.98	0.98	1.00	1.00	1.09	1.04
400	0.9	1.5	0.89	0.40	0.98	0.99	1.00	1.01	1.25	1.19
1000	0.9	1.5	0.81	1.15	0.99	0.99	1.01	1.01	1.37	1.28
200	1.1	0.5	0.35	0.00	0.96	0.97	1.00	1.00	1.36	1.24
400	1.1	0.5	0.38	0.04	0.97	0.97	1.00	1.00	1.56	1.41
1000	1.1	0.5	0.35	0.85	0.98	0.98	1.01	1.01	1.59	1.45
200	1.1	1.0	0.69	0.03	0.98	0.98	1.01	1.01	1.14	1.08
400	1.1	1.0	0.75	0.29	0.98	0.98	1.00	1.00	1.33	1.24
1000	1.1	1.0	0.68	1.03	0.99	0.99	1.00	1.00	1.42	1.31
200	1.1	1.5	1.03	0.17	1.00	1.00	1.00	1.00	1.01	0.98
400	1.1	1.5	1.06	0.56	0.98	0.98	1.00	1.00	1.17	1.11
1000	1.1	1.5	1.01	1.07	0.99	0.99	1.00	1.00	1.30	1.22

Table 4 Nonlinear case (uniform distribution): Relative performance of the CM, 10-CV, and D-D methods in comparison to the tuning approach.

Noise settings			Parameters		CM		10-CV		D-D	
n	s	bd	$\hat{s}$	$\hat{\epsilon}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$
200	3.0	0.8	0.86	0.35	0.99	0.99	1.01	1.01	1.26	1.22
400	3.0	0.8	0.69	2.15	1.00	1.01	1.01	1.01	1.74	1.62
1000	3.0	0.8	0.41	4.96	1.00	1.00	1.02	1.02	2.16	1.94
200	3.0	1.0	1.08	0.37	1.02	1.02	1.01	1.01	1.17	1.15
400	3.0	1.0	0.83	2.28	1.01	1.02	1.01	1.01	1.66	1.59
1000	3.0	1.0	0.51	4.91	1.01	1.01	1.02	1.02	2.23	2.06
200	3.0	1.2	1.23	0.80	1.05	1.06	1.02	1.02	1.27	1.26
400	3.0	1.2	0.99	2.36	1.01	1.02	1.01	1.02	1.67	1.59
1000	3.0	1.2	0.62	4.85	1.01	1.01	1.02	1.02	2.10	1.98
200	4.0	0.8	1.10	0.72	1.02	1.02	1.01	1.01	1.24	1.21
400	4.0	0.8	0.87	2.47	1.02	1.03	1.02	1.02	1.67	1.60
1000	4.0	0.8	0.55	4.87	1.01	1.01	1.02	1.02	2.17	2.02
200	4.0	1.0	1.34	0.87	1.07	1.07	1.02	1.02	1.22	1.20
400	4.0	1.0	1.11	2.25	1.03	1.03	1.01	1.02	1.62	1.56
1000	4.0	1.0	0.69	4.85	1.01	1.01	1.01	1.02	2.07	1.97
200	4.0	1.2	1.66	0.76	1.08	1.09	1.02	1.01	1.14	1.13
400	4.0	1.2	1.25	2.67	1.04	1.04	1.02	1.02	1.54	1.51
1000	4.0	1.2	0.84	4.78	1.01	1.02	1.02	1.02	2.01	1.93
200	5.0	0.8	1.38	0.63	1.05	1.05	1.02	1.01	1.17	1.14
400	5.0	0.8	1.05	2.52	1.03	1.04	1.02	1.02	1.59	1.53
1000	5.0	0.8	0.66	5.11	1.01	1.01	1.02	1.02	2.07	1.97
200	5.0	1.0	1.68	0.96	1.07	1.08	1.02	1.03	1.15	1.15
400	5.0	1.0	1.33	2.56	1.03	1.04	1.01	1.01	1.50	1.47
1000	5.0	1.0	0.85	4.97	1.02	1.02	1.02	1.02	2.12	2.03
200	5.0	1.2	1.94	1.11	1.11	1.10	1.03	1.03	1.16	1.14
400	5.0	1.2	1.47	2.86	1.04	1.04	1.01	1.01	1.48	1.45
1000	5.0	1.2	0.96	5.41	1.03	1.03	1.03	1.03	2.00	1.93

and the simulation results are given in Table 4. Obviously, two ratios from the proposed D-D method are notably greater than 1. For instance, compared with the CM and 10-CV methods, both ratios of the simulation from the D-D method with noise setting n 1000, s 5.0 and bd 1.2, are nearly 200% (MAE) and 193% (RMSE), respectively, so our D-D method obtained a nearly twofold improvement.

From the above three types of nonlinear simulations, it can be concluded that our proposed D-D method for ϵ-SVR noticeably improves the forecasting performance in nonlinear applications.

4.2 Linear regression

Now we consider the most popular linear model generated by the following:

$$y_i = \beta_0 + \beta_1 \cdot x_i + s \cdot u_i, \quad i = 1, 2, \ldots, n,$$

where $\beta_0 = 1$ and x_i is generated from the normal distribution $N(0, 1)$. Considering different noise levels for all simulations, we set β_1 as 2, 2, and 1 for noises generated from the ϵ-Laplacian distribution, normal distribution, and uniform distribution, respectively. In addition, the kernel of the ϵ-SVR is the linear function $k(x_i, x_j) = x_i' \cdot x_j$. All simulations are implemented 100 times to record the average performance. The linear simulation results for the ϵ-Laplacian distribution, normal distribution $N(0, \sigma^2)$, and uniform distribution $unif[-bd, bd]$ are listed in Tables 5, 6 and 7, respectively.

First, in the linear simulation for residuals generated from the ϵ-Laplacian distribution, the estimated insensitivity parameter $\hat{\epsilon}$ and the estimated scale parameter $\hat{s}$ all approximate to the real settings with our D-D method in different noise levels, as shown in Table 5. For comparison of the accuracy for the forecasting performance, in the linear regression with $n = 300$ and $R^2 = 0.38$, our proposed D-D SVR performed better than the CM and the 10-CV, with a more than 68% improvement with MAE and a 69% improvement with RMSE. In addition, according to simulation results with $n = 100$, $s = 0.5$, and $\epsilon = 1.0$, we can find our proposed methods are like the 10-CV method much better than the CM method and the basic tuning method. Here, it is noted that more computational costs in CV method are required to find a proper parameter from a pre-set sequence of ϵ. Overall, our D-D method

Table 5 Linear case (ϵ-Laplacian distribution): Relative performance of the CM, 10-CV, and D-D methods in comparison to the tuning approach.

Noise settings				Parameters		CM		10-CV		D-D	
n	s	ϵ	R^2	$\hat{s}$	$\hat{\epsilon}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$
100	0.5	0.8	0.87	0.51	0.71	0.80	0.79	0.99	0.98	1.13	1.13
200	0.5	0.8	0.87	0.51	0.71	0.81	0.83	0.97	0.97	1.21	1.21
300	0.5	0.8	0.87	0.49	0.86	1.39	1.35	1.14	1.12	1.46	1.41
100	0.5	1.0	0.85	0.52	0.74	0.96	0.96	1.07	1.06	1.07	1.09
200	0.5	1.0	0.85	0.51	0.93	1.18	1.16	1.16	1.15	1.26	1.25
300	0.5	1.0	0.86	0.50	0.95	1.15	1.14	1.08	1.08	1.23	1.22
100	0.5	1.2	0.86	0.52	1.12	0.99	1.03	1.08	1.11	1.30	1.32
200	0.5	1.2	0.86	0.50	1.21	1.34	1.35	1.16	1.15	1.38	1.38
300	0.5	1.2	0.84	0.52	1.07	1.20	1.19	1.08	1.07	1.33	1.31
100	1.0	0.8	0.65	0.97	0.76	0.98	0.96	1.08	1.07	1.33	1.28
200	1.0	0.8	0.62	1.01	0.73	0.96	0.92	1.04	0.99	1.52	1.53
300	1.0	0.8	0.62	1.00	0.76	1.19	1.20	1.10	1.11	1.20	1.21
100	1.0	1.0	0.61	1.00	0.98	0.76	0.75	1.02	1.03	1.20	1.18
200	1.0	1.0	0.61	1.02	0.95	1.18	1.16	1.11	1.10	1.31	1.28
300	1.0	1.0	0.60	1.01	0.95	1.30	1.27	1.15	1.15	1.52	1.50
100	1.0	1.2	0.58	0.99	1.27	1.32	1.27	1.12	1.11	1.39	1.37
200	1.0	1.2	0.57	1.00	1.17	1.46	1.42	1.22	1.19	1.55	1.50
300	1.0	1.2	0.58	1.02	1.14	1.37	1.36	1.12	1.10	1.62	1.60
100	1.5	0.8	0.43	1.48	0.78	0.89	0.86	1.09	1.07	1.28	1.27
200	1.5	0.8	0.42	1.47	0.77	1.04	1.05	1.03	1.03	1.27	1.28
300	1.5	0.8	0.42	1.49	0.79	1.19	1.20	1.21	1.22	1.25	1.27
100	1.5	1.0	0.42	1.44	1.09	1.07	1.06	1.09	1.09	1.24	1.23
200	1.5	1.0	0.41	1.49	0.99	1.23	1.23	1.14	1.13	1.37	1.35
300	1.5	1.0	0.40	1.48	1.05	1.25	1.23	1.14	1.13	1.30	1.26
100	1.5	1.2	0.38	1.53	1.15	1.37	1.38	1.30	1.30	1.73	1.74
200	1.5	1.2	0.38	1.60	1.04	1.17	1.20	1.07	1.05	1.44	1.40
300	1.5	1.2	0.38	1.50	1.19	1.32	1.29	1.11	1.11	1.68	1.69

can precisely improve forecasting performance by auto-adapting the insensitivity parameter.

The second linear simulation, shown Table 6, is the regression with noises from the normal distribution $N(0, \sigma^2)$. The simulation results show that with R^2 from 0.40 to 0.86, all the ratio$_{\text{MAE}}$ and ratio$_{\text{RMSE}}$ for D-D are all significantly greater than 1. In other words, our proposed method can auto-recognize a limited scale and obtain a limiting insensitivity parameter to approach real noises; as a result, the forecasting performance is superior. It is interesting that corresponding to the type of noise, the scale is also auto adapted to match the most approximate ϵ in the insensitive Laplacian distribution. In the simulation setting with $n = 300$, $s = 2.0$, and $\sigma = 1.2$, the noises contribute more as 60% to the response, thus, the data are with high randomization. We still can find our forecasting performances are like 10-CV with less computational costs. Overall, according to the reported table, we can find our proposed method can beat other two methods in almost simulations Therefore, our method can make ϵ-SVR more efficient in the linear model with Gaussian noises.

The final simulation, shown in Table 7, illustrates that our D-D method can obtain surprisingly good improvements. This is because the ratios from our D-D method are quite large, indicating that our proposed method can model the linear model with perfect accuracy. The most interesting finding in the parameter estimation analysis is that with an increasing number of samples, our D-D method approaches approximating the ϵ-Laplacian loss function by increasing ϵ and decreasing s; two parameter estimations will converge to limiting values. To sum up, for the noise from uniform distribution, our method is still a powerful tool for improving the linear regression forecasting.

Furthermore, for the mechanism exploration of our D-D method, compared with the CM in linear simulations, which is motivated by the noise following the normal distribution, our D-D's forecasting performance is close, but still is better when addressing the noise from the normal distribution shown in Table 6, while in Tables 5 and 7, our D-D method's performance can significantly improve the forecasting accuracy. This illustrates that our D-D method can auto-adapt the parameters to approximate any unknown noise

Table 6 Linear case (normal distribution): Relative performance of the CM, 10-CV, and D-D methods in comparison to the tuning approach.

Noise settings				Parameters		CM		10-CV		D-D	
n	s	σ	R^2	$\hat{s}$	$\hat{\epsilon}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$
100	1.0	0.8	0.86	0.48	1.28	1.13	1.13	1.08	1.09	1.27	1.27
200	1.0	0.8	0.88	0.46	1.39	1.09	1.08	1.05	1.04	1.37	1.34
300	1.0	0.8	0.86	0.47	1.41	1.01	1.04	1.03	1.06	1.26	1.27
100	1.0	1.0	0.81	0.59	1.20	0.95	0.95	1.06	1.08	1.17	1.18
200	1.0	1.0	0.79	0.58	1.43	1.17	1.20	1.11	1.13	1.35	1.36
300	1.0	1.0	0.80	0.58	1.39	1.10	1.10	0.99	1.00	1.34	1.35
100	1.0	1.2	0.74	0.72	1.25	0.85	0.81	0.97	0.97	1.35	1.32
200	1.0	1.2	0.74	0.73	1.26	1.03	1.04	1.00	1.00	1.12	1.12
300	1.0	1.2	0.75	0.68	1.51	1.09	1.09	1.01	1.00	1.21	1.21
100	1.5	0.8	0.75	0.71	1.35	1.04	1.03	1.25	1.24	1.22	1.20
200	1.5	0.8	0.73	0.71	1.32	1.03	1.03	1.00	1.00	1.22	1.19
300	1.5	0.8	0.73	0.70	1.38	1.09	1.09	1.03	1.02	1.27	1.28
100	1.5	1.0	0.67	0.85	1.49	0.79	0.79	1.10	1.10	1.65	1.68
200	1.5	1.0	0.64	0.85	1.47	1.22	1.19	1.09	1.08	1.34	1.33
300	1.5	1.0	0.64	0.86	1.48	1.17	1.19	1.17	1.19	1.37	1.39
100	1.5	1.2	0.56	1.03	1.57	1.09	1.10	1.05	1.04	1.20	1.21
200	1.5	1.2	0.55	1.03	1.48	1.04	1.05	0.97	0.96	1.26	1.25
300	1.5	1.2	0.56	1.03	1.41	1.16	1.16	1.09	1.08	1.24	1.25
100	2.0	0.8	0.62	0.90	1.48	1.23	1.20	0.98	0.97	1.25	1.22
200	2.0	0.8	0.61	0.88	1.65	1.15	1.17	1.05	1.04	1.46	1.46
300	2.0	0.8	0.61	0.91	1.54	1.14	1.12	1.14	1.11	1.43	1.39
100	2.0	1.0	0.52	1.13	1.50	1.07	1.09	1.03	1.04	1.21	1.23
200	2.0	1.0	0.51	1.15	1.42	1.12	1.12	1.03	1.03	1.35	1.32
300	2.0	1.0	0.51	1.11	1.54	1.14	1.12	1.08	1.07	1.45	1.41
100	2.0	1.2	0.43	1.37	1.43	1.51	1.62	1.14	1.15	1.82	1.94
200	2.0	1.2	0.42	1.35	1.48	1.20	1.19	1.05	1.06	1.22	1.22
300	2.0	1.2	0.40	1.36	1.52	0.98	1.00	1.02	1.02	1.01	1.03

Table 7 Linear case (uniform distribution): Relative performance of the CM, 10-CV, and D-D methods in comparison to the tuning approach.

Noise settings				Parameters		CM		10-CV		D-D	
n	s	bd	R^2	$\hat{s}$	$\hat{\epsilon}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$	ratio$_{\text{MAE}}$	ratio$_{\text{RMSE}}$
100	1.0	0.8	0.81	0.19	4.06	1.25	1.23	1.11	1.09	2.72	2.69
200	1.0	0.8	0.82	0.12	6.29	1.21	1.20	1.14	1.14	3.25	3.23
300	1.0	0.8	0.83	0.09	8.25	1.00	1.00	1.13	1.12	3.14	3.09
100	1.0	1.0	0.77	0.22	4.19	1.42	1.43	1.08	1.09	2.42	2.46
200	1.0	1.0	0.76	0.14	7.15	1.20	1.19	1.13	1.13	3.00	2.96
300	1.0	1.0	0.76	0.12	8.15	1.16	1.17	1.16	1.17	3.89	3.89
100	1.0	1.2	0.68	0.25	4.54	1.47	1.47	1.15	1.15	2.56	2.57
200	1.0	1.2	0.67	0.18	6.61	1.14	1.16	1.09	1.09	3.63	3.65
300	1.0	1.2	0.68	0.14	8.37	1.18	1.20	1.19	1.21	3.61	3.58
100	1.5	0.8	0.66	0.28	4.19	1.13	1.13	1.04	1.03	2.34	2.29
200	1.5	0.8	0.68	0.16	7.52	1.07	1.07	1.01	1.02	3.02	3.05
300	1.5	0.8	0.68	0.15	8.06	1.20	1.20	1.16	1.15	3.36	3.32
100	1.5	1.0	0.58	0.33	4.47	1.64	1.63	1.09	1.09	2.45	2.44
200	1.5	1.0	0.57	0.21	7.30	1.21	1.23	1.10	1.10	3.91	3.92
300	1.5	1.0	0.57	0.17	8.69	1.09	1.08	1.10	1.10	4.08	4.16
100	1.5	1.2	0.47	0.35	5.50	1.23	1.20	1.06	1.05	2.08	2.07
200	1.5	1.2	0.50	0.24	7.55	1.17	1.19	1.05	1.05	3.73	3.90
300	1.5	1.2	0.48	0.21	8.53	1.17	1.19	1.13	1.13	4.13	4.14
100	2.0	0.8	0.53	0.30	5.29	1.42	1.40	1.16	1.12	3.06	3.06
200	2.0	0.8	0.54	0.21	7.85	1.18	1.18	1.15	1.16	3.17	3.20
300	2.0	0.8	0.53	0.19	8.45	1.05	1.02	1.06	1.04	4.27	4.13
100	2.0	1.0	0.43	0.34	5.88	1.16	1.20	0.96	0.98	2.46	2.48
200	2.0	1.0	0.44	0.27	7.43	1.05	1.05	1.00	0.99	2.85	2.83
300	2.0	1.0	0.43	0.22	9.25	1.14	1.14	1.17	1.17	4.17	4.18
100	2.0	1.2	0.35	0.46	5.03	1.38	1.40	1.08	1.08	3.22	3.39
200	2.0	1.2	0.35	0.33	7.58	1.12	1.10	1.04	1.02	3.20	3.14
300	2.0	1.2	0.34	0.26	9.20	1.09	1.07	1.10	1.10	4.32	4.20

distribution and improve the SVR's performance, while the CM method focuses on the normal distribution. Moreover, the computational cost of the 10-CV method with five alternative parameter settings is over 10 times more than our D-D method. In addition, because of the parameter setting for the cross validation, the 10-CV method cannot guarantee its superior performance with high computational costs. Therefore, we can conclude that our D-D method can auto-adapt the ϵ-Laplacian loss function to guarantee the steadiness of a linear model with high levels of accuracy. Furthermore, because it is determined by the type of noise, the scale and the insensitivity parameter will converge to true values (the noise is generated from the ϵ-Laplacian distribution) or limiting values (the noise is from any other distribution).

5 Case studies

In the section, our D-D ϵ-SVR is evaluated with five case studies: energy efficiency (768 samples, eight attributes, and two responses) [40], yacht hydrodynamics (308 samples, six attributes, and one response) [41], airfoil self-noise (1503 samples, five attributes, and one response) [42], concrete compressive strength (1030 samples, eight attributes, and one response) [43] from the UCI Machine Learning Repository [44], and Boston housing prices (506 samples, 14 attributes, and one response) from the StatLib collection [45].

Each benchmark data set was randomly divided into two groups: the training set (70% of each data set) and the test set (the remaining data from each set). Then, each experiment was repeated 100 times to obtain the average performance of our proposed SVR. In this section, the execution time is added to show the efficiency of our proposed method as well. Because the scale of each attribute is different, the standard normalization was applied for attribute pre-processing before the training. The general radial basic function is selected as the kernel. In addition, the 10-CV [9] was applied in the insensitivity parameter selection with the same alternative parameter settings as the former simulations. In addition, according to our literature review, we employed three recent meta-heuristics method with 10-CV to tune the insensitivity parameter for the ϵ-SVR: whale optimization algorithm (WOA) [21], grey wolf optimizer (GWO) [22], multi-verse optimizer (MVO) [19] with 10 search agents. In addition, all algorithms are performed on an Intel i7-8700 CPU with 16.0 GB of RAM.

The ϵ and σ for the five benchmark data sets were estimated using our proposed method, and the convergence curves of our proposed method are shown in Figure 2 for one repeated experiment. According to convergence curves of MSE index for all investigated cases, we can find the procedure converges through one or two iterations. We then display the work likelihood functions for each case from one repeated experiment in Figure 3. Moreover, the corresponding negative log-likelihood function values with different ϵ values at the estimated scale in one of experiments for five cases are displayed in Figure 4. It is obvious that the specific ϵ-Laplacian loss function is data-driven by the real data sets. Different from the original ϵ-SVR, our proposed "scale" ϵ-SVR can auto-recognize the scale of noise in real data sets and self-adapt the insensitivity parameter accordingly.

The prediction performance for all five cases is listed in Table 8. Obviously, our proposed method can improve the accuracy of predictions based on the ratios. The most obvious cases are the MAE (tuning 3.90 vs. CM 4.11 vs. 10-CV 4.18 vs. D-D **2.70**) and RMSE (tuning 6.96 vs. CM 6.83 vs. 10-CV 6.83 vs. D-D **5.05**) for the yacht hydrodynamics. Compared with the tuning, 10-CV, and CM methods, the MAE and RMSE in the rest of the data sets (energy efficiency, Boston housing, airfoil self-noise, and concrete compressive strength) achieved around 10% improvements. In addition, compared with three meta-heuristic algorithms (WOA, MVO, and GWO), our proposed D-D method still can achieve good forecasting performance with less computational costs. For example, for modelling cooling load data, the forecasting performances are very similar, but the D-D method is more efficient (WOA: 90.87mins, MVO: 84.77mins, GWO: 85.78mins, and D-D: 10.55mins). Furthermore, according to comparisons in the datasets of Boston housing, yacht hydrodynamics, and concrete compressive strength), although three meta-heuristic algorithms need more computational costs, our D-D method still can beat them with highly accurate preferences. To show the significance of our forecasting results, a Wilcoxon signed-rank test is

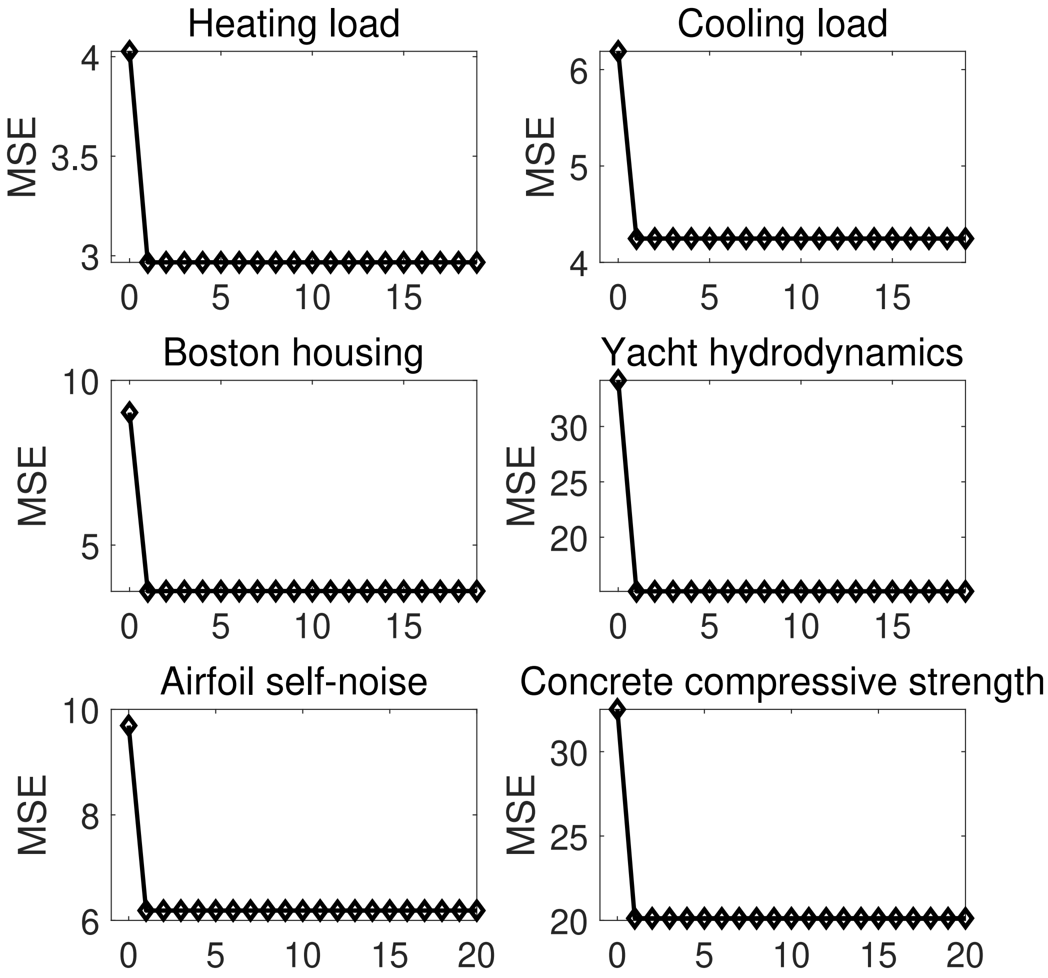

Fig. 2 The convergence curves of our proposed method in our case studies. The x-axis is the number of iterations.

used with MAE indexes from 100 repeated experiments for all case studies. The results are recorded in Table 8 where '1' represents the forecasting results are significant to our D-D method and '0' represents the forecasting results are insignificant.

To summarize, our proposed D-D method can auto-adapt the insensitivity parameter in the ϵ-Laplacian distribution approach to the real noise distribution; this means our working likelihood method can push the ϵ-Laplacian density function to seek the approximate likelihood function. As a result, our D-D SVR has an excellent performance in real applications.

6 Conclusion

The SVR with ϵ-Laplacian loss distribution is a mainstream algorithm for regression modelling, where the insensitivity parameter ϵ determines the support vector. However, to date, after inputs and target scaling, three types of strategies for parameter selection are used: the k-cross validation, which requires huge computational costs, the tuning parameter, which cannot make the SVR work more efficiently, and the empirical statistical estimation, the CM method that is based on normal distribution with some empirical settings. Obviously, the mentioned parameter settings are

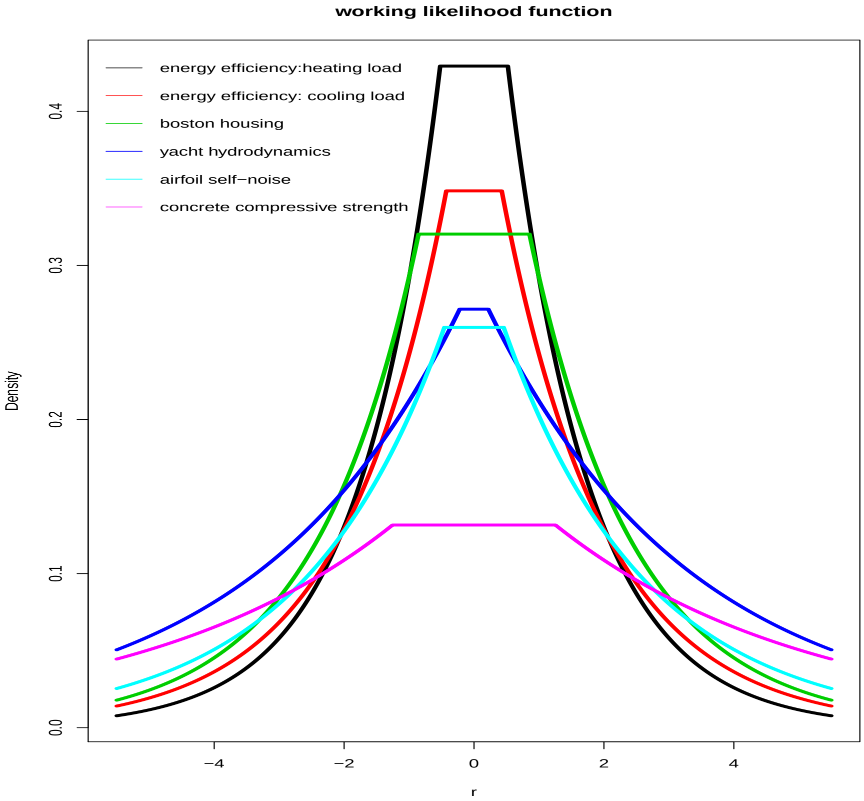

Fig. 3 Six working likelihood D-D functions for five case studies.

not the most appropriate hyper-parameters for SVR in most conditions, so, in this paper, we propose optimization of the insensitivity parameter based on the working likelihood function developed by [28], which is a D-D method, to estimate appropriate hyper-parameters for finding the most appropriate ϵ-Laplacian distribution to the real noise distribution to guarantee generalization in test sets. In addition, the D-D support vector regression is standardized by the scale of the noise in a more meaningful field. In nonlinear and linear simulations conducted with different types of noises (ϵ-Laplacian distribution, normal distribution, and uniform distribution), our proposed method demonstrated that it can automatically estimate the scale and the insensitivity parameter. As a result, our D-D SVR showed significantly improved forecasting accuracy in the test sets. Moreover, our D-D algorithm can estimate the approximate likelihood function in five real benchmark applications, and furthermore, the proposed method had dramatically improved performance in unknown sets. Therefore, our proposed D-D SVR is a more intelligent and powerful technique for the regression problem.

Here, it must be noted that we have no guarantee that the optimization (Formula (5)) has the only one global minimization, but we never experienced the problem in both numerical simulations and case studies. Additionally, tuning regularization parameter C and kernel parameter γ in an elegant way also are important but challenging. Interestingly, in the reference of [3], an insensitive linear-linear loss function was proposed for support vector regression to minimize the economic cost for load scheduling. Particularly, different penalties for over-prediction and under-prediction are given in the optimization objective from the real economic loss. Thus, the work [3] is different from our current work. However, it is of interest to develop a data-driven method to tune the insensitive parameter in the insensitive linear-linear loss function instead of the CV method used in [3]. Similarly, in machine learning modelling, our D-D

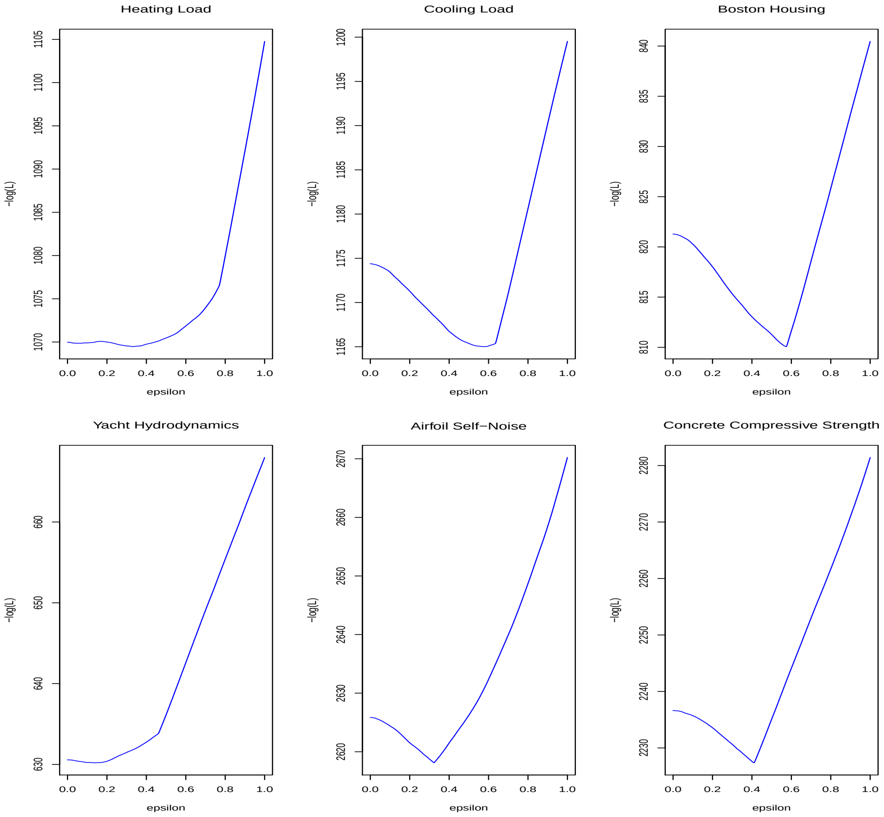

Fig. 4 The insensitivity parameter-negative likelihood function value plots for five cases

method using the framework of working likelihood is a viable general strategy for parameter estimations such as the twin SVR [46] and the general robust loss function [47]. For example, we can incorporate the explored lncosh loss function into SVR framework to improve the work [39].

Acknowledgments. The authors would like to thank the five reviewers for their constructive comments and suggestions, which have led to a much-improved paper. This work was supported in part by the Australian Research Council project DP160104292 and the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS), under grant number CE140100049.

Data availability. A demo of the proposed D-D SVR will be available on-line when the work is published.

Declarations. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Chen BJ, Chang MW, et al. Load forecasting using support vector machines: A study on EUNITE competition 2001. IEEE Transactions on Power Systems. 2004;19(4):1821–1830.

[2] Artemiou A, Dong Y, Shin SJ. Real-time sufficient dimension reduction through principal least squares support vector machines. Pattern Recognition. 2021;112:107768.

[3] Wu J, Wang YG, Tian YC, Burrage K, Cao T. Support vector regression with asymmetric loss for optimal electric load forecasting. Energy. 2021;223:119969.

[4] Vapnik V, Golowich SE, Smola AJ. Support vector method for function approximation,

Table 8 Results for four case studies with different methods*.(The unit of execution time: Minute)

Dataset	Index	Tuning	CM	10-CV	WOA [21]	MVO [19]	GWO [22]	D-D
Heating load	MAE	1.51	1.45	1.51	1.20	1.20	1.20	1.19
	RMSE	2.32	2.36	2.32	1.82	1.82	1.82	1.87
	Execution time	-	-	76.38	90.30	92.54	93.70	12.82
	Significance	1	1	1	0	0	0	-
Cooling load	MAE	1.87	1.84	1.84	1.54	1.54	1.54	1.57
	RMSE	2.72	2.72	2.73	2.30	2.30	2.30	2.34
	Execution time	-	-	82.56	90.87	84.77	85.78	10.55
	Significance	1	1	1	0	0	0	-
Boston housing	MAE	2.42	2.44	2.42	3.71	3.71	3.71	2.22
	RMSE	4.03	4.03	4.05	6.15	6.15	6.15	3.53
	Execution time	-	-	15.85	18.69	19.69	19.68	4.32
	Significance	1	1	1	1	1	1	-
Yacht hydrodynamics	MAE	3.90	4.11	4.18	4.87	4.83	4.83	2.70
	RMSE	6.96	6.83	6.83	9.82	9.83	9.82	5.05
	Execution time	-	-	10.09	12.62	12.26	12.28	1.57
	Significance	1	1	1	1	1	1	-
Airfoil self-noise	MAE	2.42	2.42	2.43	1.93	1.93	1.93	1.97
	RMSE	3.33	3.33	3.33	2.78	2.78	2.78	2.80
	Execution time	-	-	197.12	212.81	242.15	228.48	35.25
	Significance	1	1	1	0	0	0	-
Concrete compressive strength	MAE	4.98	4.98	4.96	5.36	5.36	5.36	4.37
	RMSE	6.82	6.82	6.84	7.94	7.94	7.94	6.11
	Execution time	-	-	47.48	53.94	60.21	62.68	13.15
	Significance	1	1	1	1	1	1	-

* The execution time with tuning method and CM method are not reported because their computational costs are very low.

regression estimation and signal processing. Advances in Neural Information Processing Systems. 1996;9:281–287.

[5] Chang CC, Lin CJ. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST). 2011;2(3):1–27.

[6] Chang CC, Lin CJ. Training v-support vector regression: theory and algorithms. Neural Computation. 2002;14(8):1959–1977.

[7] Drucker H, Burges CJ, Kaufman L, Smola A, Vapnik V. Support vector regression machines. Advances in Neural Information Processing Systems. 1996;9:155–161.

[8] Vapnik V. The nature of statistical learning theory. Springer Science & Business Media; 2013.

[9] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media; 2009.

[10] Ito K, Nakano R. Optimizing support vector regression hyperparameters based on cross-validation. In: Proceedings of the International Joint Conference on Neural Networks, 2003.. vol. 3. IEEE; 2003. p. 2077–2082.

[11] Schölkopf B, Bartlett P, Smola A, Williamson RC. Shrinking the tube: a new support vector regression algorithm. Advances in Neural Information Processing Systems. 1999;11:330–336.

[12] Schölkopf B, Smola AJ, Williamson RC, Bartlett PL. New support vector algorithms. Neural Computation. 2000;12(5):1207–1245.

[13] Schölkopf B, Bartlett PL, Smola AJ, Williamson RC. Support vector regression with automatic accuracy control. In: International Conference on Artificial Neural Networks. London: Springer; 1998. p. 111–116.

[14] Jeng JT, Chuang CC, Su SF. Support vector interval regression networks for interval regression analysis. Fuzzy Sets and Systems. 2003;138(2):283–300.

[15] Cherkassky V, Ma Y. Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks. 2004;17(1):113–126.

[16] Wen Z, Li B, Kotagiri R, Chen J, Chen Y, Zhang R. Improving Efficiency of SVM k-fold Cross-validation by Alpha Seeding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31; 2017. p. 2768–2774.

[17] Hsia JY, Lin CJ. Parameter selection for linear support vector regression. IEEE Transactions on Neural Networks and Learning Systems. 2020;31(12):5639–5644.

[18] Wu CH, Tzeng GH, Lin RH. A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. Expert Systems with Applications. 2009;36(3):4725–4735.

[19] Tabrizchi H, Javidi MM, Amirzadeh V. Estimates of residential building energy consumption using a multi-verse optimizer-based support vector machine with k-fold cross-validation. Evolving Systems. 2021;12(3):755–767.

[20] Zhou J, Qiu Y, Zhu S, Armaghani DJ, Li C, Nguyen H, et al. Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate. Engineering Applications of Artificial Intelligence. 2021;97:104015.

[21] Zhou J, Zhu S, Qiu Y, Armaghani DJ, Zhou A, Yong W. Predicting tunnel squeezing using support vector machine optimized by whale optimization algorithm. Acta Geotechnica. 2022;p. 1–24.

[22] Liu M, Luo K, Zhang J, Chen S. A stock selection algorithm hybridizing grey wolf optimizer and support vector regression. Expert Systems with Applications. 2021;179:115078.

[23] Algamal ZY, Qasim MK, Lee MH, Ali HTM. Improving grasshopper optimization algorithm for hyperparameters estimation and feature selection in support vector regression. Chemometrics and Intelligent Laboratory Systems. 2021;208:104196.

[24] Li W, Kong D, Wu J. A new hybrid model FPA-SVM considering cointegration for particular matter concentration forecasting: a case study of Kunming and Yuxi, China. Computational Intelligence and Neuroscience. 2017;2017.

[25] da Silva Santos CE, Sampaio RC, dos Santos Coelho L, Bestard GA, Llanos CH. Multi-objective adaptive differential evolution for SVM/SVR hyperparameters selection. Pattern Recognition. 2021;110:107649.

[26] Kalita DJ, Singh S. SVM hyper-parameters optimization using quantized multi-PSO in dynamic environment. Soft Computing. 2020;24(2):1225–1241.

[27] Bartlett PL, Boucheron S, Lugosi G. Model selection and error estimation. Machine Learning. 2002;48(1-3):85–113.

[28] Fu L, Wang YG, Cai F. A working likelihood approach for robust regression. Statistical Methods in Medical Research. 2020;29(12):3641–3652.

[29] Smola AJ, Schölkopf B. A tutorial on support vector regression. Statistics and Computing. 2004;14(3):199–222.

[30] Wu Y, Wang L. A survey of tuning parameter selection for high-dimensional regression.

Annual Review of Statistics and Its Application. 2020;7:209–226.

[31] Wang YG, Lin X, Zhu M, Bai Z. Robust estimation using the Huber function with a data-dependent tuning constant. Journal of Computational and Graphical Statistics. 2007;16(2):468–481.

[32] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Mathematical Programming. 1989;45(1):503–528.

[33] Poli R, Kennedy J, Blackwell T. Particle swarm optimization. Swarm Intelligence. 2007;1(1):33–57.

[34] Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F, Chang CC, et al. Package 'e1071'. The R Journal. 2019;p. 1–66.

[35] Lipsitz SR, Fitzmaurice GM, Orav EJ, Laird NM. Performance of generalized estimating equations in practical situations. Biometrics. 1994;50(1):270–278.

[36] Brown BM, Wang YG. Standard errors and covariance matrices for smoothed rank estimators. Biometrika. 2005;92(1):149–158.

[37] Chu W, Keerthi SS, Ong CJ. Bayesian support vector regression using a unified loss function. IEEE Transactions on Neural Networks. 2004;15(1):29–44.

[38] Singla M, Ghosh D, Shukla K, Pedrycz W. Robust twin support vector regression based on rescaled Hinge loss. Pattern Recognition. 2020;105:107395.

[39] Karal O. Maximum likelihood optimal and robust support vector regression with lncosh loss function. Neural Networks. 2017;94:1–12.

[40] Tsanas A, Xifara A. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. Energy and Buildings. 2012;49:560–567.

[41] Ortigosa I, Lopez R, Garcia J. A neural networks approach to residuary resistance of sailing yachts prediction. In: Proceedings of the International Conference on Marine Engineering MARINE. vol. 2007; 2007. p. 250.

[42] Lau K, López R, Oñate E, Ortega E, Flores R, Mier-Torrecilla M, et al. A neural networks approach for aerofoil noise prediction; 2006.

[43] Yeh IC. Analysis of strength of concrete using design of experiments and neural networks. Journal of Materials in Civil Engineering. 2006;18(4):597–604.

[44] Dua D, Graff C.: UCI Machine Learning Repository. Available from: http://archive.ics.uci.edu/ml.

[45] Fan RE.: LIBSVM Data: Regression. Available from: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html.

[46] Peng X. TSVR: an efficient twin support vector machine for regression. Neural Networks. 2010;23(3):365–372.

[47] Barron JT. A general and adaptive robust loss function. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society; 2019. p. 4326–4334.