Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Highlights

Augmented support vector regression with an autoregressive process via an iterative procedure

Applied Soft Computing xxx (xxxx) xxx

Jinran Wu, You-Gan Wang*, Hao Zhang

- An augmented support vector regression model with an autoregressive process is designed to model temporal data.
- A robust iterative procedure is developed for training the proposed SVR.
- The working likelihood method is used for parameter estimation.

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Augmented support vector regression with an autoregressive process via an iterative procedure

Jinran Wu [a], You-Gan Wang [a,*], Hao Zhang [b]

[a] School of Mathematical Sciences, Queensland University of Technology, Brisbane, QLD 4001, Australia
[b] Department of Statistics, Purdue University, West Lafayette, IN 47906, USA

ARTICLE INFO

Keywords:
Augmented regression
Parameter estimation
Temporal pattern
Forecasting

ABSTRACT

The Support Vector Regression (SVR) technique can approximate intricate systems by addressing learning and estimation challenges within a reproducing kernel Hilbert space, devoid of reliance on specific parameter assumptions. However, when dealing with correlated data like time series, the SVR method often falls short in accounting for underlying temporal structures, leading to limited enhancements in prediction efficiency. We introduce an enhanced SVR method that considers temporal correlations (TemporalSVR) to overcome this constraint. Our proposed method extends kernel functions to include additional linear kernels, facilitating learning temporal patterns. Additionally, we develope an iterative training procedure for the augmented regression model. During model training, we estimate the hyper-parameter in the corresponding loss function using a 'working' likelihood approach, enhancing the generalization capabilities of the proposed regression. To demonstrate superior forecasting performance, we conduct extensive numerical simulations on both linear and nonlinear systems and the TemporalSVR achieves improvements ranging from 8% to 114% based on the RMSE ratio from the AR-X model. Furthermore, we investigate the forecasting performance of three basic models (NARX-NN, Statistical SVR, and SVR-ARIMA) and four deep learning (DL) techniques (Transformer, Informer, Reformer, Autormer, and Autoformer) by using a WTI forecasting study. Our proposed TemporalSVR achieves the smallest RMSE at 2.22 and attains the highest success ratio of stock direction prediction at 71.30%. All these numerical results highlight the effectiveness and advantages of our TemporalSVR in handling temporal data and making accurate predictions.

1. Introduction

SVR proves to be a potent method for efficiently approximating intricate systems by framing learning and estimation challenges within a reproducing kernel Hilbert space, as highlighted by Blanchard et al. [1]. Nevertheless, this algorithm often assumes data independence implicitly, as noted in Smola and Schölkopf [2]. This implicit assumption can result in sub-optimal outcomes, particularly when dealing with dependent data, such as temporal data in financial time series forecasting [3] and energy demand prediction [4].

The conventional SVR method falls short of capturing temporal dependencies adequately, leading to subpar forecasting performance. In contrast, statistical linear regression models, particularly those incorporating auto-regressive processes to address temporal correlation, have been proposed in Shi and Tsai [5]. However, the simplicity of linear models may not suffice for effectively modeling complex real-world data, given the linear data assumption, resulting in unsatisfactory practitioner performance.

To address this limitation and enhance the statistical linear regression with an auto-regressive process, we introduce an augmented SVR with an auto-regressive process, termed TemporalSVR. In our method, we extend the kernel functions to include additional linear kernels that account for temporal correlations within the regression framework. Additionally, we present an iterative training procedure tailored to the proposed framework. Specifically, introducing a new kernel function in TemporalSVR facilitates the extraction of temporal patterns, making it highly effective for time-series modeling. To further boost the forecasting performance of TemporalSVR, we incorporate a technique inspired by Wu and Wang [6], adopting the working likelihood method [7] to tune the insensitivity parameter ϵ in the proposed loss function. By applying the 'working' likelihood principle, we estimate the hyper-parameter controlling the number of support vectors, thereby contributing to improved model tuning and enhanced forecasting accuracy. Moreover, definitions of some key notations are provided in Table 1.

* Corresponding author.
E-mail address: ygwangug2012@gmail.com (Y.-G. Wang).

https://doi.org/10.1016/j.asoc.2024.111549
Received 13 August 2023; Received in revised form 17 March 2024; Accepted 20 March 2024

Table 1
Definitions of some key notations.

Notation	Description	Notation	Description
$\langle \cdot, \cdot \rangle$	dot product	C	regularization parameter
r_i	sample i residual	ϵ	insensitivity parameter
α_i, α_i^*	Lagrange multipliers	ξ_i, ξ_i^*	slack variables
n	sample size	u_t	an auto-regressive process
v_t	white noise	P	order of autoregressive model
y_t	response variable at time t	σ	scale parameter
Φ	weight of temporal kernel	ω	weight of non-temporal kernel
T	training size	T'	test size

The paper's main contributions can be summarized as follows:

- We introduce an innovative framework called TemporalSVR, which is an augmented SVR model with an autoregressive process, specifically designed to model temporal data. In contrast to traditional SVR, our method extends the kernel functions to include additional linear kernels, allowing for learning temporal patterns through linear kernel approximation.
- We develop a robust iterative procedure for training the TemporalSVR framework. This iterative process effectively trains the kernel function, encompassing both the original kernels and the supplementary linear kernels, to yield accurate estimates. Furthermore, within this iterative procedure, we adaptively estimate the insensitivity parameter ϵ using the working likelihood method, significantly enhancing the performance of the TemporalSVR framework.
- We conduct some simulation studies to demonstrate the reliability of our proposed training procedure. In addition, we analyze a real-world case involving crude oil price forecasting and the results of these experiments highlight the superior performance and forecasting capabilities of the TemporalSVR model compared to existing methods, including four state-of-the-art DL techniques, i.e., Transformer [8], Informer [9], Reformer [10], and Autoformer [11].

The paper is structured as follows: Section 2 investigates some work on SVR and DL techniques for time series forecasting. In Section 3, we explore the fundamental framework of SVR and investigate the related parameter settings. Next, in Section 4, we present the details of our proposed method, namely, the augmented SVR with an autoregressive process (TemporalSVR), employing an iterative procedure. In Section 5, we conduct numerous numerical simulations with different temporal patterns to validate the effectiveness of the proposed TemporalSVR framework. In Section 6, we apply the proposed framework to a real-world case study of crude oil price forecasting, demonstrating its practical effectiveness. Finally, in Section 7, we provide a summary of the proposed TemporalSVR framework and conclude the paper.

2. Literature review

Machine learning methods recently have been popularly employed in time series forecasting in real applications, such as power systems [12,13], environmental engineering [14], and financial forecasting [15–17]. The machine learning methods generally can be divided into two major categories: SVR-based and neural network-based methods.

First of all, the SVR with the insensitive Laplace loss and the l_2 penalty for kernel training has good generalization capabilities to unknown data [18]. Since there are no assumptions about the data pattern, whether it exhibits dependence or independence, proper lagged observations can be used as inputs for model training to forecast future values when modeling temporal data [19]. This input setting has been successful in modeling temporal data using SVR [20]. Later, many combination forecasting strategies based on SVR have been developed according to specific characteristics of targeted time series, such as a decomposition-ensemble SVR method [21] and heteroscedastic SVR methods [22,23]. Furthermore, for autoregressive regressions with predictors, there are two general methods for establishing the corresponding SVR. In the first approach, both lagged observations and predictors are directly used as features to train the model [24]. An example of this approach is photovoltaic power output forecasting, where historical data from the previous moment and the weather report for the next day are combined as features for training the SVR model [25]. Although this approach can incorporate predictors to improve forecasting performance, the contribution of predictors might be overshadowed by the importance of lagged observations. The second approach involves developing an additive framework, where SVR is used first to obtain the mapping between predictors and the output, and then the temporal relationship is extracted by modeling the remaining residuals from the first model using another model, such as some parametric models [26]. For instance, Zhu and Wei [27] presented a two-step framework to extract temporal correlation from residuals generated by SVR with the auto-regressive integrated moving average model. However, the second method lacks the ease of providing subsequent statistical inferences, such as constructing confidence intervals.

On the other hand, neural network methods, especially DL techniques featuring intricate hidden layers, have emerged as widely used tools in the realm of time series forecasting. These DL techniques are currently advancing swiftly, particularly in their capability to handle large-scale time series datasets. A recurrent neural network (RNN) developed by Medsker and Jain [28] is a classical artificial neural network for time series modeling, characterized by the direction of the flow of information between its layers, and in the recent review work by Hewamalage et al. [29], they discussed the competitive role of RNNs in forecasting, acknowledging their success in the M4 competition. Following the RNNs, Hochreiter and Schmidhuber [30] delved into Long Short-Term Memory (LSTM) networks, offering a solution to the prolonged training times associated with RNNs by introducing reversible residual layers and enforcing constant error flow. Recently, an efficient DL technique, transformer [31], was developed based on the multi-head attention mechanism for sequence modeling, and the model does not contain any recurrent units and requires less training time than previous recurrent neural architectures. Han et al. [8] explored the application of Transformers in time series modeling, showcasing their prowess in capturing long-range dependencies, with various Transformer variants tailored to address specific challenges. Moving forward, Kitaev et al. [10] introduced the Reformer, an efficient transformer model designed to alleviate computational costs associated with large-scale training on extended sequences. Later, Zhou et al. [9] presented Informer, a transformer-based model specifically crafted for long-sequence time-series forecasting, incorporating novel self-attention mechanisms and a generative style decoder to enhance efficiency and accuracy. Meanwhile, Wu et al. [11] introduced Autoformer, a decomposition architecture with an Auto-Correlation mechanism, aiming to overcome limitations of prior Transformer models in handling intricate temporal patterns and information utilization bottlenecks for improved long-term series forecasting. It is important to highlight that, in contrast, to SVR-based methods, all the DL techniques mentioned necessitate an extensive sequence length during training due to the abundance of parameters, ensuring the development of a forecasting model with robust generalization, and as highlighted by Hewamalage et al. [29], DL techniques are less user-friendly.

Therefore, this work focuses on SVR, and to address these limitations and improve predictions while maintaining valid inferences, we propose a novel augmented regression framework incorporating statistical correlation structures into SVR.

3. Related work

We assume the training data $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \subset \chi \times \mathbb{R}$ (where $\chi = \mathbb{R}^d$ denotes the space of inputs) is generated from

$$y_i = f(x_i) + u_i = \langle \omega, x_i \rangle + b + u_i, \omega \in \chi, b \in \mathbb{R},$$

with the dot product $\langle \cdot, \cdot \rangle$ in χ, noise u_i, and $f(x_i) = \langle \omega, x_i \rangle + b$. In the framework of ϵ-SVR, the actual function $f(x_i)$ can be estimated using kernel techniques. These techniques enable the mapping of samples to a Hilbert space, where there exists a robust linear association between features and responses. Nevertheless, the typical kernel approximation tends to overlook sample noises, leading to overfitting. To address this issue, Drucker et al. [32] introduced an insensitivity parameter, ϵ, and proposed an insensitive Laplace loss, aiming to enhance the generalization capability for regression problems. Specifically, when the error is larger than ϵ, the error would be counted on; otherwise, it would be tolerated. Similar to the support vector classifier, the primal objective for ϵ-SVR is combined with the l_2-norm penalty and an insensitive Laplace loss as

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

with regularization parameter C and residuals $r_i = y_i - \hat{y}_i$ where $\hat{y}_i$ is the fitted value. $|r_i|_\epsilon$ is defined as $\max\{u^+, u^-\}$ with $u^+ = \max\{r_i - \epsilon, 0\}$ and $u^- = \max\{-r_i - \epsilon, 0\}$. To easily solve the primal objective for ϵ-SVR training, the corresponding dual formulation is derived as

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

where α_i and α_i^* are the Lagrange multipliers for $y_i - \langle \omega, x_i \rangle - b - \epsilon - \xi_i$ and $\langle \omega, x_i \rangle + b - y_i - \epsilon - \xi_i^*$ with two slack variables ξ_i ($\geqslant 0$) and ξ_i^* ($\geqslant 0$), respectively. The prediction at x from the dual solution can be expressed as

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

where b can be estimated with Karush–Kuhn–Tucker (KKT) conditions. The dual problem in Formula (2) can be solved with quadratic programming and sequential minimal optimization. The details can be found in Smola and Schölkopf [2].

In ϵ-SVR, hyper-parameter selections often determine its performance in real applications, and there are two key hyper-parameters in the ϵ-SVR as shown in Formula (1). The first hyper-parameter ϵ is used to balance the data fitting and improve the generalization for SVR by controlling the number of support vectors. In the LibSVM library and R package 'e1071' [33], the default setting for the insensitivity parameter ϵ is 0.1 suggested by Chang and Lin [34]. Another one is the regularization parameter C, as illustrated in Formula (1), that determines the trade-off between the flatness of f and the amount up to which deviations larger than the toleration ϵ. There are two recommended settings for C. Cherkassky and Ma [35] recommended setting C as the 95% quantile of $|y_i|$, i.e., $|y_i|_{0.95}$. For high-dimensional problems, Wu and Wang [36] advocated the regularization parameter C as the order of $\sqrt{n/\log K}$ with inputs dimension K. Notice that when kernel $\langle x_i, x_j \rangle$ is $x_i' \cdot x_j$ (a linear kernel), ϵ-SVR can be regarded as a simple ridge linear regression with an insensitive Laplace loss. For nonlinear regression problems, some complex kernel functions are introduced, such as the radial basic function $\exp(-\gamma\|x_i - x_j\|^2)$, where the hyper-parameter in the kernel function needs to be tuned to train the model. For example, Scholkopf et al. [37] empirically set γ as $1/(0.3K)$ in their experiments. However, when the computation is feasible, a grid search can be very easily implemented to obtain the proper setting from a pre-set grid [38]. Recently, more insightful guidelines for hyper-parameter selections for real practices were provided by Wen et al. [39].

4. The proposed augmented SVR

4.1. The preliminary

In statistics, the well-known regression model with an auto-regressive process is represented as follows:

$$y_t = \sum_{k=1}^{K} \beta_k \cdot x_{tk} + u_t,$$

where u_t follows an auto-regressive process $u_t = \sum_{j=1}^{P} \Phi_j \cdot u_{t-j} + v_t$ with the order of the autoregressive model P. In this formulation, y_t represents the response variable, x_{tk} are the predictor variables, v_t denotes white noise, and β_k and Φ_j are the unknown parameters [40]. The parameter Φ_j can be estimated using the residuals u_t obtained from the first linear regression, and an iterative procedure is employed to update the two sets of parameters β_k and Φ_j iteratively until convergence is achieved [5].

4.2. The formulation of the augmented SVR

In our augmented SVR with an autoregressive process, we suppose a temporal observation set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)\} \subset \chi \times \mathbb{R}$ (i.e., $\chi = \mathbb{R}^d$), is generated from the following model $\mathcal{T}$ via two functions f and g:

$$\begin{cases} y_t = f(x_{t-1}) + u_t \\ u_t = g(\widetilde{u}_{t-1}) + \sigma v_t \end{cases}, \tag{3}$$

with $\widetilde{u}_{t-1} = (u_{t-1}, u_{t-2}, \ldots, u_{t-P})$ where the noise with scale σ is defined as σv_t, and x_{t-1} is a vector for all available information (predictors) at time $t-1$. For the temporal regression problem, the proposed augmented SVR via extended ϵ-SVR can be formulated as

$$y_t = \langle \omega, x_{t-1} \rangle + b + \langle \Phi, \widetilde{u}_{t-1} \rangle + r_t,$$

with residuals r_t where Φ is the weight parameter in the temporal kernel for learning the autoregressive process. Two kernel components $\langle \omega, x_{t-1} \rangle$ and $\langle \Phi, \widetilde{u}_{t-1} \rangle$ are designed to extract the pattern between the predictors and the response and the temporal pattern, respectively.

According to the statistical ϵ-SVR proposed by Wang et al. [41] where the insensitivity parameter ϵ can be self-adaptive with the working likelihood method [7], a new primal objective with scale parameter σ for the augmented SVR can be simply formulated as

$$\min_{\omega, \Phi} \frac{1}{2}\left(\|\omega\|^2 + \|\Phi\|^2\right) + C \sum_{t=1+P}^{T} \left(|\frac{r_t}{\sigma}|_\epsilon\right).$$

The corresponding density function for r_t/σ can be given as

$$p\left(\frac{r_t}{\sigma}|\epsilon, \sigma\right) = \frac{1}{\sigma} \cdot \frac{1}{2(1+\epsilon)} \exp\left(-|\frac{r_t}{\sigma}|_\epsilon\right).$$

It should be noticed that all r_t ($t = 1 + P, \ldots, T$) are independent and identically distributed random variables. Thus, the two parameters ϵ and σ can be estimated by minimizing the negative log-likelihood,

$$\begin{aligned} -\log L(\epsilon, \sigma) &= -\log \prod_{t=1+P}^{T} p\left(\frac{r_t}{\sigma}|\epsilon, \sigma\right) \\ &= (T-P)\log\sigma + (T-P)\log 2(1+\epsilon) \\ &\quad + \sum_{t=1+P}^{T} |\frac{r_t}{\sigma}|_\epsilon. \end{aligned} \tag{4}$$

The parameters can be calculated with the root of the partial derivatives of $-\log L(\epsilon,\sigma)$ concerning ϵ and σ as

$$\begin{cases} \frac{\partial -\log L(\epsilon,\sigma)}{\partial \epsilon} = \frac{T-P}{1+\epsilon} - \sum_{t=1+P}^{T} \mathbb{I}\left(|\frac{r_t}{\sigma}| > \epsilon\right) = 0 \\ \frac{\partial -\log L(\epsilon,\sigma)}{\partial \sigma} = \frac{T-P}{\sigma} - \frac{1}{\sigma^2} \sum_{t=1+P}^{T} |r_t| \cdot \mathbb{I}\left(|\frac{r_t}{\sigma}| > \epsilon\right) = 0 \end{cases}. \tag{5}$$

Remark 1. The initial value of σ is the standard deviation calculated by the initial residuals. In case the Eqs. (5) is infeasible, we can use the estimate of the innovation parameter from the time series analysis of the AR(P) model, and this estimate is a consistent estimator of σ. Moreover, we have not encountered the infeasible problem when estimating σ.

Therefore, the full primal objective with the l_2-norm penalty for the augmented SVR with an autoregressive process can be presented as

$$\begin{aligned} \min_{\omega,\Phi,\epsilon,\sigma} \frac{1}{2}\left(\|\omega\|^2 + \|\Phi\|^2\right) + C(T-P)\log\sigma + \\ C\left((T-P)\log 2(1+\epsilon) + \sum_{t=1+P}^{T}\left(|\frac{r_t}{\sigma}|_\epsilon\right)\right), \end{aligned} \tag{6}$$

where ω, Φ, ϵ, and σ can be optimized together. Because Φ, ϵ, and σ are estimated based on the series $U = \{u_2, \ldots, u_T\}$ and the final residual r_t, an iterative procedure is developed for the TemporalSVR training. Specifically, with estimated series $\hat{U}$, we can fix ϵ and σ as $\hat{\epsilon}$ and $\hat{\sigma}$, and update ω and Φ via

$$\min_{\omega,\Phi} \frac{1}{2}\left(\|\omega\|^2 + \|\Phi\|^2\right) + C\sum_{t=1+P}^{T}\left(|\frac{r_t}{\hat{\sigma}}|_{\hat{\epsilon}}\right). \tag{7}$$

We can update ϵ and σ by solving Eqs. (5) with estimated residuals $\hat{r}_t$. The detailed procedure for the TemporalSVR training for modeling temporal data is described in Section 4.4. We should emphasize that although (6) is non-convex, the proposed training procedure consistently yields reliable estimates in experiments.

4.3. The dual solution

Given the estimated parameters $\hat{\epsilon}$ and $\hat{\sigma}$, the convex optimization problem Formula (7) with slack variables ξ_t and ξ_t^* can be redefined as

$$\begin{aligned} \min_{\omega,\Phi,b,\xi_t,\xi_t^*} \quad & \frac{1}{2}\left(\|\omega\|^2 + \|\Phi\|^2\right) + \check{C}\sum_{t=1+P}^{T}\hat{\sigma}(\xi_t + \xi_t^*) \\ \text{s.t.} \quad & \begin{cases} y_t - \langle\omega, x_{t-1}\rangle - \langle\Phi, \widetilde{u}_{t-1}\rangle - b \leqslant \hat{\sigma}\hat{\epsilon} + \xi_t \\ \langle\omega, x_{t-1}\rangle + \langle\Phi, \widetilde{u}_{t-1}\rangle + b - y_t \leqslant \hat{\sigma}\hat{\epsilon} + \xi_t^* \\ \xi_t \geqslant 0, \xi_t^* \geqslant 0, t = 1+P, \ldots, T \end{cases}, \end{aligned}$$

with $\check{C} = |y_t/\hat{\sigma}|_{(0.95)}$. To solve the corresponding dual problem, a Lagrange function is constructed from the primal objective function and the corresponding constraints with a dual set of variables as

$$\begin{aligned} \mathcal{L} :=& \frac{1}{2}\left(\|\omega\|^2 + \|\phi\|^2\right) + \check{C}\sum_{t=1+P}^{T}\hat{\sigma}(\xi_t + \xi_t^*) \\ & - \sum_{t=1+P}^{T}\left(\eta_t\xi_t + \eta_t^*\xi_t^*\right) \\ & + \sum_{t=1+P}^{T}\alpha_t\left(\Delta_t - \hat{\sigma}\hat{\epsilon} - \xi_t\right) \\ & + \sum_{t=1+P}^{T}\alpha_t^*\left(-\Delta_t - \hat{\sigma}\hat{\epsilon} - \xi_t^*\right), \end{aligned}$$

with $\Delta_t = y_t - \langle\omega, x_{t-1}\rangle - \langle\Phi, \widetilde{u}_{t-1}\rangle - b$ where $\mathcal{L}$ is the Lagrangian, and $\eta_t, \eta_t^*, \alpha_t$, and α_t^* are the Lagrange multipliers. For convenience, we define $\alpha_t^{(*)} = (\alpha_t, \alpha_t^*), \eta_t^{(*)} = (\eta_t, \eta_t^*)$, and $\xi_t^{(*)} = (\xi_t, \xi_t^*)$, respectively. Then, the multipliers follow from the saddle point condition which can be calculated with the root of the partial derivatives of $\mathcal{L}$ concerning the primal variables $(\omega, \Phi, b, \xi_t^{(*)})$ as

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial\omega} = \omega + \sum_{t=1+P}^{T}(\alpha_t^* - \alpha_t)x_{t-1} = 0 \\ \frac{\partial\mathcal{L}}{\partial\Phi} = \Phi + \sum_{t=1+P}^{T}(\alpha_t^* - \alpha_t)\widetilde{u}_{t-1} = 0 \\ \frac{\partial\mathcal{L}}{\partial b} = \sum_{t=1+P}^{T}(\alpha_t^* - \alpha_t) = 0 \\ \frac{\partial\mathcal{L}}{\partial\xi_t^{(*)}} = \hat{\sigma}\check{C} - \eta_t^{(*)} - \alpha_t^{(*)} = 0 \end{cases}. \tag{8}$$

Moreover, the dual problem of the proposed TemporalSVR can be obtained as

$$\begin{aligned} \max_{\alpha,\alpha^*} & -\frac{1}{2}\sum_{t,t'=1+P}^{T}(\alpha_t - \alpha_t^*)(\alpha_{t'} - \alpha_{t'}^*)(\langle x_{t-1}, x_{t'-1}\rangle \\ & + \langle\widetilde{u}_{t-1}, \widetilde{u}_{t'-1}\rangle) - \hat{\epsilon}\sum_{t=1+P}^{T}\hat{\sigma}(\alpha_t + \alpha_t^*) \\ & + \sum_{t=1+P}^{T}(\alpha_t - \alpha_t^*)y_t \\ \text{s.t.} & \begin{cases} \sum_{t=1+P}^{T}(\alpha_t - \alpha_t^*) = 0 \\ 0 \leqslant \alpha_t^{(*)} \leqslant \hat{\sigma}\check{C}. \end{cases} \end{aligned} \tag{9}$$

The solution needs to satisfy the KKT condition

$$\begin{cases} \alpha_t\left(y_t - \langle\omega, x_{t-1}\rangle - \langle\Phi, \widetilde{u}_{t-1}\rangle - b - \hat{\sigma}\hat{\epsilon} - \xi_t\right) = 0 \\ \alpha_t^*\left(\langle\omega, x_{t-1}\rangle + \langle\Phi\widetilde{u}_{t-1}\rangle + b - y_t - \hat{\sigma}\hat{\epsilon} - \xi_t^*\right) = 0 \\ \alpha_t\alpha_t^* = 0, \xi_t\xi_t^* = 0 \\ \left(\hat{\sigma}\check{C} - \alpha_t^{(*)}\right)\xi_t^{(*)} = 0 \end{cases}.$$

Finally, according to Eqs. (8), the proposed TemporalSVR $\mathcal{T}(x_{t_0-1}, \widetilde{u}_{t_0-1}) = f(x_{t_0-1}) + g(\widetilde{u}_{t_0-1})$ with the Lagrange multipliers α_t and α_t^* can be estimated as

$$\begin{aligned} \mathcal{T}(x_{t_0-1}, \widetilde{u}_{t_0-1}) = & \sum_{t=1+P}^{T}(\alpha_t - \alpha_t^*)(\langle x_{t-1}, x_{t_0-1}\rangle \\ & + \langle\widetilde{u}_{t-1}, \widetilde{u}_{t_0-1}\rangle) + \widetilde{b}. \end{aligned}$$

In addition, the intercept $\widetilde{b}$ can be estimated by using all l observations from $S = \{(x_j, y_j) | 0 < \alpha_j^* < \hat{\sigma}\check{C}, j = 1+P, \ldots, T\}$ when the KKT condition holds as

$$\begin{aligned} \widetilde{b} = & \frac{1}{l}\sum_{j\in S}(y_j - \sum_{t=1+P}^{T}(\alpha_t - \alpha_t^*)(\langle x_{t-1}, x_{j-1}\rangle \\ & + \langle\widetilde{u}_{t-1}, \widetilde{u}_{j-1}\rangle) - \hat{\sigma}\hat{\epsilon}). \end{aligned} \tag{10}$$

It shall be noted that according to Proposition 2 of Carrasco et al. [42], the value of $\widetilde{b}$ can also be estimated by using α in $(0, \hat{\sigma}\check{C})$ and the KKT complementary condition.

4.4. The training procedure

In the proposed TemporalSVR, Φ is obtained with the provided $\widetilde{u}_{t-1}$ while two hyper-parameters ϵ and σ are estimated with the working likelihood method via residuals r_t. Thus, the pseudo-code for our proposed TemporalSVR framework via iterative learning is given in Algorithm 1. In practice, two parameters, the maximum number of the iteration Iter_{Max} and the threshold of change in mean square error Δ_{Iter}, need to be given for our iterative learning procedure, and it is empirically recommended to use $\text{Iter}_{\text{Max}} = 5$ and $\Delta_{\text{Iter}} = 0.01$ as the default values for these two parameters. (Some analysis is discussed in Remark 2.) In addition, we also visualize the training procedure for our proposed TemporalSVR in Fig. 1.

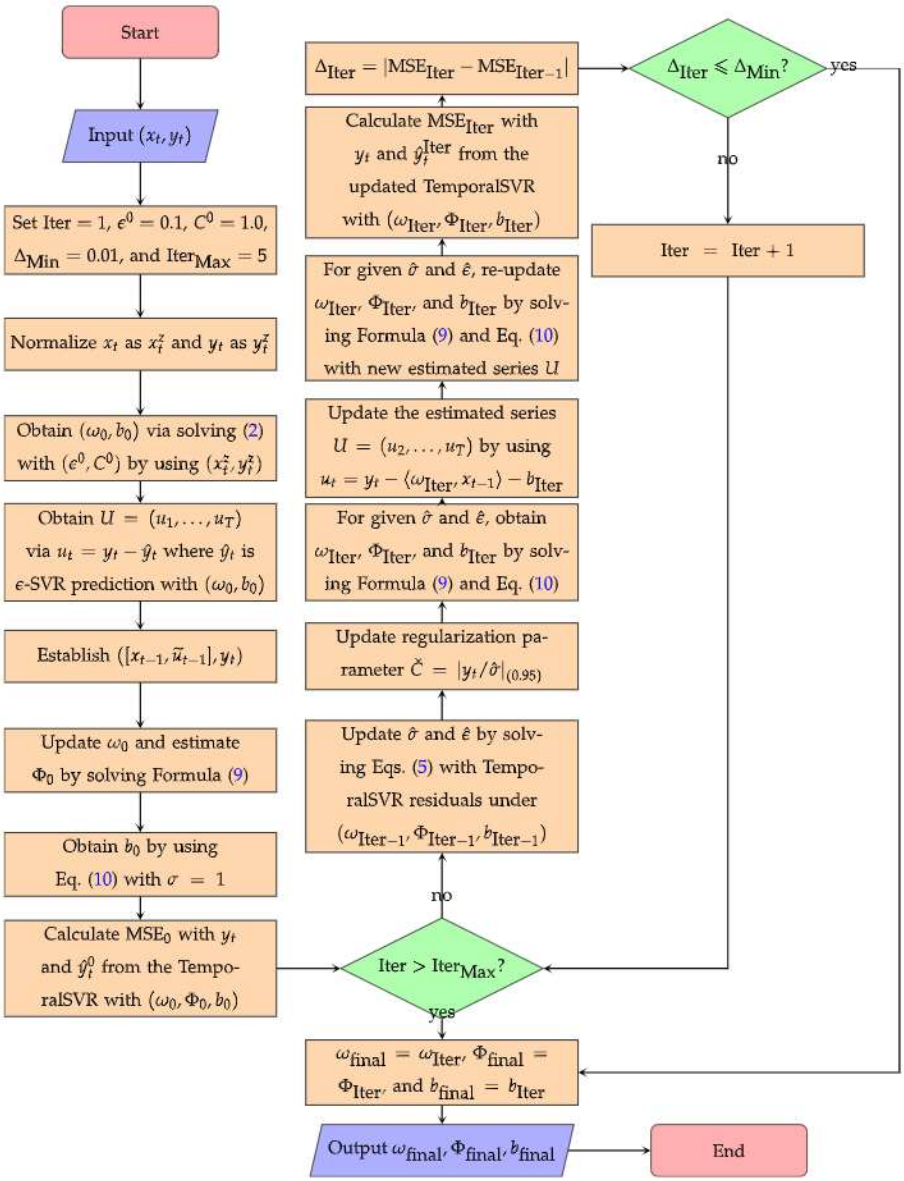

Fig. 1. The flowchart of the training procedure for the TemporalSVR framework.

Remark 2. An iterative procedure is employed for training the proposed TemporalSVR. Based on our experience, performing one or two iterations is often sufficient. This effectiveness may be attributed to the fact that a residual improvement of the order $O_d(1/n)$ can be attained after just one iteration, as exemplified by the findings of Brown and Wang [43] They observed a similar scenario when updating the hyperparameter for smoothness in rank estimation. Furthermore, akin to the classic SVR, the convexity of (7) is maintained for a given set of parameters $(\hat{\epsilon}, \hat{\sigma})$. The solution can be obtained by solving Formula (9) and Eq. (10). Moreover, concerning (4), as outlined by Wu and Wang [6], let us assume that $(\hat{\epsilon}, \hat{\sigma})$ represent the estimates obtained by minimizing (4), and (ϵ^*, σ^*) are the limiting values of $(\hat{\epsilon}, \hat{\sigma})$. We can derive the following insights: (1) a unique solution for limiting values (ϵ^*, σ^*) exists if the true density function of noise follows an ϵ-Laplacian distribution ($\epsilon > 0$), as indicated by their Corollary 1 (this point is also validated through our numerical simulations); and (2) the limiting values ϵ^* and σ^* are 1.524 and 0.557 s respectively, assuming the true density function of the noise is normally distributed with mean 0 and standard deviation $s < +\infty$, according to their Corollary 2. We acknowledge that this remark does not specifically pertain to numerical convergence and will be subject to further exploration in future research.

Remark 3. The computational complexity of our algorithm is mainly caused by calculating the initial estimator via basic ϵ-SVR with complexity $O(n^2 \times d + n^3)$, updating ω and b with the TemporalSVR with complexity $O(n^2 \times (d + P) + n^3)$ again, and updating $\hat{\sigma}$ and $\hat{\epsilon}$ by solving Eqs. (5) as n_h. The grid search is employed to find $\hat{\sigma}$ and $\hat{\epsilon}$, and the search range for $\hat{\sigma}$ is $[0.1, 0.2, \ldots, 2] \times \sigma^{\text{MAD}}$ where σ^{MAD} is the median absolute deviation (MAD) estimate from y_t-$\hat{y}_t^{\text{Iter}}$ and that for $\hat{\epsilon}$ is $[0.1, 0.2, \ldots, 0.5] \times (\max y_t)/\sigma^{\text{MAD}}, t = 1, 2, \ldots, T$, and we obtain $n_h = 100$ for the grid search. Therefore, the computational complexity is determinable as $O(L(n^2 \times (d + P) + n^3 + n_h) + n^2 \times d + n^3) = O((L + 1)n^3)$ with the iteration number L, leading to an overall order of n^3.

Remark 4. Here, it shall be noted that the second kernel's lag setting would determine the TemporalSVR framework's performance. The Akaike information criterion (AIC) is suggested for lag selection. In detail, the AIC incorporates a value of two for every degree of freedom, and only the changes in the AIC values are relevant. We can only tabulate the AIC differences to the best model for better visualization.

Algorithm 1: The pseudo-code for our proposed TemporalSVR

Require: (x_t, y_t), $t = 1, 2, \ldots, T$
Ensure: $(\omega_{\text{final}}, \Phi_{\text{final}}, b_{\text{final}})$
Iter = 1
$\epsilon^0 = 0.1$
$C^0 = 1.0$
$\Delta_{\text{Min}} = 0.01$
$\text{Iter}_{\text{Max}} = 5$
Normalize predictors x_t as x_t^z, $t = 1, 2, \ldots, T$
Normalize responses y_t as y_t^z, $t = 1, 2, \ldots, T$
Obtain initial estimates (ω_0, b_0) via solving Formula (2) with initial parameter setting (ϵ^0, C^0) by using standardized data (x_t^z, y_t^z), $t = 1, 2, \ldots, T$
Obtain initial estimated series $U = (u_1, \ldots, u_T)$ by calculating the residuals $u_t = y_t - \hat{y}_t$ where $\hat{y}_t$ is from ϵ-SVR with (ω_0, b_0)
Establish corresponding temporal inputs and output $([x_{t-1}, \widetilde{u}_{t-1}], y_t)$, $t = 2, \ldots, T$.
Update ω_0 and estimate Φ_0 by solving Formula (9)
Obtain b_0 by using Eq. (10) when assuming $\sigma = 1$
Calculate MSE_0 with y_t and $\hat{y}_t^0$, $t = 2, 3, \ldots, T$ from the TemporalSVR with (ω_0, Φ_0, b_0)
for $\text{Iter} = 1, 2, \ldots, \text{Iter}_{\text{Max}}$ **do**
  Update $\hat{\sigma}$ and $\hat{\epsilon}$ by minimizing Formula (4) or solving Eqs. (5) with residuals from the TemporalSVR with $(\omega_{\text{Iter}-1}, \Phi_{\text{Iter}-1}, b_{\text{Iter}-1})$
  Update regularization parameter $\check{C} = |y_t/\hat{\sigma}|_{(0.95)}$
  For given $\hat{\sigma}$ and $\hat{\epsilon}$, obtain ω_{Iter}, Φ_{Iter}, and b_{Iter} by solving Formula (9) and Eq. (10)
  Update the estimated series $U = (u_2, \ldots, u_T)$ by using $u_t = y_t - \langle \omega_{\text{Iter}}, x_{t-1} \rangle - b_{\text{Iter}}$
  For given $\hat{\sigma}$ and $\hat{\epsilon}$, re-update ω_{Iter}, Φ_{Iter}, and b_{Iter} by solving Formula (9) and Eq. (10) with new estimated series U
  Calculate MSE_{Iter} with y_t and $\hat{y}_t^{\text{Iter}}$, $t = 2, 3, \ldots, T$ from the updated TemporalSVR with $(\omega_{\text{Iter}}, \Phi_{\text{Iter}}, b_{\text{Iter}})$
  $\Delta_{\text{Iter}} = |\text{MSE}_{\text{Iter}} - \text{MSE}_{\text{Iter}-1}|$
  if $\Delta_{\text{Iter}} \leqslant \Delta_{\text{Min}}$ **then**
    Break
  end if
end for
$\omega_{\text{final}} = \omega_{\text{Iter}}$, $\Phi_{\text{final}} = \Phi_{\text{Iter}}$, and $b_{\text{final}} = b_{\text{Iter}}$

Another strategy for the lag choice is motivated by the shrinkage in l_1-SVR [44]. Specifically, given a large lag $P' (\gg P)$, the l_1-norm penalty can be applied in the second kernel instead of the l_2-norm penalty. Now, the lag P can be adaptively selected during the training procedure via the following objective function:

$$\min_{\omega, \Phi} \frac{1}{2} \left(\|\omega\|^2 + \|\Phi\| \right) + C \sum_{t=1+P'}^{T} \left(|\frac{r_t}{\hat{\sigma}}|_{\hat{\epsilon}} \right).$$

5. Numerical simulations

In this section, we consider two classical autoregressive processes, AR(1) and AR(2), in two types of regressions (linear regression and non-linear regression) for temporal data modeling. The proposed augmented SVR is employed for one-step-ahead forecasting. Notice that true lag order P is known in these simulations. Thus, it is fixed on the true value to validate the effectiveness of the proposed TemporalSVR framework. Furthermore, in our experiment design, we follow Eqs. (3) to generate simulation data. To simulate our data, we first generate the temporal part u_t by using innovation v_t (i.i.d noise) with scale parameter σ. Here, considering the stationary of the temporal part u_t, we drop out the first 1000 data points and keep the remaining u_t. Moreover, we combine the temporal part u_t with the non-temporal part $f(x_{t-1})$ to obtain our final observed responses y_t. In our simulation, we only can observe x_{t-1} and y_t to establish the temporal pattern with mathematical models. we consider the first 80% of the data as training and the remaining as testing. We calculate the error indexes based on the predicted values and their corresponding true values ($\mu = f + g$).

To evaluate the performance of the proposed temporal SVR framework, three indexes, the mean absolute error (MAE), the root mean square error (RMSE), and the mean relative error (MRE) are applied as

$$\text{MAE} = \frac{1}{T'} \sum_{i=1}^{T'} |\mu_i - \hat{y}_i|,$$

$$\text{RMSE} = \sqrt{\frac{1}{T'} \sum_{i=1}^{T'} \left(\mu_i - \hat{y}_i \right)^2},$$

and

$$\text{MRE} = \frac{1}{T'} \sum_{i=1}^{T'} |\frac{\mu_i - \hat{y}_i}{\mu_i}|,$$

with the prediction $\hat{y}_i$ and the true value μ_i in the ith test sample ($i = 1, 2, \ldots, T'$) with test size T'.

In our numerical simulation, the autoregressive regression with exogenous variable (AR-X) with the true lag order is used as the baseline. Here, based on these three indexes, MAE, RMSE, and MRE, three ratios are employed to measure the difference between the baseline and SVR-based models, X, i.e., the statistical SVR [6] and the proposed TemporalSVR, as

$$\text{ratio}_{\text{MAE}} = \frac{\text{MAE}_{\text{AR-X}}}{\text{MAE}_{\text{X}}},$$

$$\text{ratio}_{\text{RMSE}} = \frac{\text{RMSE}_{\text{AR-X}}}{\text{RMSE}_{\text{X}}},$$

and

$$\text{ratio}_{\text{MRE}} = \frac{\text{MRE}_{\text{AR-X}}}{\text{MRE}_{\text{X}}}.$$

Here, ratio values from the above equations greater than 1 will indicate the better performance of model X.

5.1. Linear regression with AR(P) process

In this subsection, we consider the linear system is generated from the formula

$$y_t = \beta_1 x_{t-1} + u_t, t = 1, 2, \ldots, T,$$

where x_{t-1} is generated from the normal distribution $N(0, 1)$ with $\beta_1 = 2$. It shall be noted that the expectation $E(Bx_{t-1})$ is still 0 and the variation $Var(Bx_{t-1})$ is B^2. In our simulation, we use the variation from different parts to describe each contribution, respectively, and the results are presented in Appendix A. The residual series u_t follows the AR(1) process and the AR(2) process, respectively. In the linear regression simulations, to generate a stationary time series process, ϕ_1 is set as 0.4, 0.6, and 0.8 for AR(1) and AR(2) processes, respectively, while ϕ_2 is fixed at 0.1 for the AR(2) process. (All the parameter settings are satisfied with the assumption of the stationarity.) The innovation to generate the AR(P) process is considered from two methods: the ϵ-Laplace distribution with a scale parameter σ and the mixture distribution (70% from ϵ-Laplace distribution and 30% from the normal distribution). For the AR(P) process generated from the ϵ-Laplace distribution, the insensitivity parameter ϵ is set as 0.2 (the proportion of support vector 84.15%), 0.6 (the proportion of support vector 54.85%), and 1.0 (the proportion of support vector 31.73%), while the scale parameter is fixed at 1. In the second innovation generation approach, 70% of innovations follow the ϵ-Laplace distribution with the same parameter settings as mentioned before while the remaining 30% of innovations v_t are produced from the normal distribution $N(0, \delta^2)$ with

Table 2
Linear simulations with ϵ-Laplace innovations: comparison of the MAE, the RMSE, and the MRE.

Panel A: Linear simulations with AR(1) process								
Parameter setting		Estimate	Statistical SVR			TemporalSVR		
ϕ_1	ϵ	$\hat{\epsilon}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$
0.4	0.2	0.22	0.21	0.19	0.20	1.49	1.50	1.58
0.4	0.6	0.58	0.19	0.18	0.17	1.24	1.23	1.29
0.4	1	0.98	0.19	0.18	0.18	1.11	1.11	1.12
0.6	0.2	0.26	0.14	0.13	0.14	1.66	1.66	1.70
0.6	0.6	0.58	0.13	0.12	0.14	1.42	1.43	1.50
0.6	1	0.99	0.12	0.12	0.13	1.34	1.33	1.26
0.8	0.2	0.25	0.09	0.09	0.12	1.77	1.79	1.99
0.8	0.6	0.60	0.09	0.09	0.11	1.66	1.69	1.75
0.8	1	0.99	0.09	0.09	0.11	1.74	1.75	1.70
Panel B: Linear simulations with AR(2) process: $\phi_2 = 0.1$								
Parameter setting		Estimate	Statistical SVR			TemporalSVR		
ϕ_1	ϵ	$\hat{\epsilon}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$
0.4	0.2	0.21	0.20	0.19	0.23	1.41	1.42	1.51
0.4	0.6	0.56	0.20	0.20	0.22	1.19	1.19	1.22
0.4	1	0.97	0.20	0.20	0.20	1.26	1.26	1.43
0.6	0.2	0.23	0.12	0.12	0.13	1.52	1.52	1.81
0.6	0.6	0.55	0.12	0.12	0.14	1.33	1.33	1.37
0.6	1	0.97	0.12	0.12	0.13	1.35	1.36	1.44
0.8	0.2	0.28	0.09	0.09	0.13	2.09	2.14	2.05
0.8	0.6	0.56	0.07	0.07	0.11	1.64	1.65	1.88
0.8	1	1.00	0.08	0.09	0.13	1.76	1.76	1.98

Table 3
Linear simulations with mixture innovations: comparison of the MAE, the RMSE, and the MRE.

Panel A: Linear simulations with AR(1) process								
Parameter setting		Estimate	Statistical SVR			TemporalSVR		
ϕ_1	ϵ	$\hat{\epsilon}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$
0.4	0.2	0.38	0.21	0.20	0.20	1.40	1.39	1.43
0.4	0.6	0.74	0.20	0.19	0.20	1.20	1.19	1.28
0.4	1	1.11	0.21	0.20	0.23	1.08	1.08	1.18
0.6	0.2	0.38	0.12	0.11	0.13	1.20	1.21	1.37
0.6	0.6	0.75	0.13	0.12	0.14	1.25	1.26	1.21
0.6	1	1.07	0.13	0.12	0.14	1.25	1.26	1.25
0.8	0.2	0.44	0.09	0.09	0.10	1.58	1.60	1.62
0.8	0.6	0.71	0.09	0.09	0.11	1.60	1.61	1.83
0.8	1	1.06	0.09	0.09	0.12	1.75	1.79	2.01
Panel B: Linear simulations with AR(2) process: $\phi_2 = 0.1$								
Parameter setting		Estimate	Statistical SVR			TemporalSVR		
ϕ_1	ϵ	$\hat{\epsilon}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$
0.4	0.2	0.38	0.20	0.19	0.22	1.17	1.17	1.27
0.4	0.6	0.72	0.20	0.19	0.19	1.12	1.11	1.14
0.4	1	1.10	0.20	0.19	0.21	1.11	1.12	1.13
0.6	0.2	0.38	0.12	0.12	0.14	1.34	1.33	1.37
0.6	0.6	0.77	0.12	0.12	0.14	1.18	1.18	1.21
0.6	1	1.01	0.13	0.13	0.15	1.25	1.25	1.30
0.8	0.2	0.42	0.08	0.08	0.10	1.64	1.65	1.70
0.8	0.6	0.75	0.08	0.08	0.12	1.51	1.51	1.59
0.8	1	1.10	0.08	0.08	0.11	1.57	1.59	1.69

δ^2 set at 2. The sample number (nobs.) is set as 500 for the linear simulations, while the first 400 samples are used as training sets. Moreover, in our linear simulation setting, according to the variance analysis on data generation, the contribution of the temporal component is from 35.06% to 76.23%. In addition, some statistical implications for the parameter settings are listed in Appendix A. To eliminate the inference from the random factor, all the simulations are implemented 100 times to illustrate their capacity for comparison. For the linear simulation, the kernel is selected as the linear function, and the regularization coefficient is set as $|y_I/\sigma|_{(0.95)}$ as recommended by Cherkassky and Ma [35]. The linear regression simulations based on the ϵ-Laplace distribution and the mixture distribution are displayed in Tables 2 and 3, respectively.

In Table 2, based on MAE, RMSE, and MRE ratios, the proposed TemporalSVR framework with the AR(P) process can perform the linear relationship with the temporal correlation (AR(1) process) modeling more efficiently. For example, when there is a temporal correlation ($\phi_1 = 0.4$), compared with AR(p)-X, the RMSE ratio from all the simulations is larger than 1.10. This means the proposed TemporalSVR framework is more effective for the innovations following the ϵ-Laplace distribution, while the AR(1)-X focuses on the linear relationship and temporal process with innovations from normal distributions. In addition, we find the same trend that when the temporal correlation increases, the TemporalSVR framework can achieve better performance. For example, when ϵ is set as 1, the RMSE ratio increases from 1.11 ($\phi_1 = 0.4$) to 1.75 ($\phi_1 = 0.8$). A similar point can be obtained from the linear system with the AR(2) process: The proposed TemporalSVR framework also achieves a perfect performance. Especially, at $\phi_1 = 0.8$ and $\phi_2 = 0.1$, compared with AR(2)-X, the RMSE, RMSE, and MRE ratios are more than 1.75 with $\epsilon = 1$ and $\sigma = 1$. On the other

hand, we also demonstrate the efficiency of our working likelihood method. Specifically, with the true innovation distribution, the insensitivity parameter ϵ can be accurately estimated during our iterative procedure, which can further improve the performance of the proposed TemporalSVR framework.

In addition to the linear simulation with the AR(P) process from the ϵ-Laplace distribution, the proposed TemporalSVR framework is validated by using complex mixture innovations from the normal distribution and the ϵ-Laplace distribution, and the performance is still superior to that of the AR(P)-X framework (as shown in Table 3). Interestingly, although the true innovation is from the mixture distribution, the working likelihood method still can search for a proper hyper-parameter ϵ to provide good estimators. The proposed TemporalSVR framework with the working likelihood method achieved a better forecasting performance in the test set.

Finally, according to Panels A and B in Tables 2 and 3, compared with statistical SVR [41], the AR(P)-X model can provide more accurate predictions with three ratios. This means considering temporal patterns can effectively improve forecasting performance.

To sum up, these two simulations illustrate that the proposed framework can auto-recognize the temporal dependency and use this information to model the temporal data.

5.2. Nonlinear regression with AR(P) process

For nonlinear system simulations that use the AR(P) process, a popular test function *sinc* from the SVR Refs. [45] is considered as

$$y_t = \beta_1 \cdot \frac{\sin\left(x_{t-1}\right)}{x_{t-1}} + u_t, t = 1, 2, \ldots, T,$$

where x_{t-1} is generated from the uniform distribution $u[-4\pi, 4\pi]$; and u_t follows the AR(1) process and the AR(2) process, respectively. Specifically, the scale coefficient of the nonlinear system β_1 is set as 2. Here, the temporal component u_t is generated as the same as those in the linear simulations, and we generate 500 samples for the nonlinear simulation in each experiment. We use the first 400 observations as the training set and the remaining observations as the test samples. All the simulations are repeated 100 times to compare their average capacity for temporal data modeling. In the statistical SVR and TemporalSVR framework, the kernel function in SVR is set as a radial basic function with width 1 [33]; the regularization coefficient is calculated as $|y_t/\sigma|_{(0.95)}$. The simulation results for the AR(P) process with innovations from the insensitive Laplace distribution and the mixture distribution are reported in Tables 4 and 5, respectively.

As illustrated in Table 4, the proposed TemporalSVR framework can effectively extract the nonlinear relationship and the temporal correlation to model the time series. Two findings can be observed from the nonlinear simulation with the innovation from the insensitive Laplace distribution. The main finding is that compared with AR(P)-X, the MAE, RMSE, and MRE ratios are all larger than 1 in all simulations; this means that our proposed TemporalSVR framework is more efficient than the AR(P)-X model for time series generated with the ϵ-Laplace distribution. One of the most obvious cases is that the simulation for the AR(2) process with $\beta_1 = 2$, $\phi_1 = 0.6$, $\epsilon = 0.2$, $\sigma = 1$ has a high RMSE ratio of 1.81. The second finding is that with the parameter ϕ_1 increasing, from 0.4 to 0.8, all ratios reduce compared with AR(P) model. In other words, if the contribution of the AR(P) process component increases, i.e., the variance of the temporal component increases, the temporal framework performs better.

For the nonlinear simulation with the AR(P) process from the mixture distribution in Table 5, the proposed TemporalSVR framework is superior to the AR(P)-X. For example, for the simulation setting as $\phi_1 : 0.8, \epsilon : 1, P : 2$, compared with the AR(2)-X model, the MAE, RMSE, and MRE ratios are 1.27, 1.33, and 1.31, respectively. When further exploring the underlying mechanism, the loss function for our proposed TemporalSVR framework is the ϵ-Laplace loss that is robust to outliers, hence more reliable estimates during model training.

To summarize, based on these numeral simulation results, we can conclude that the proposed TemporalSVR framework with the AR(p) process not only can extract the pattern between the inputs and the output but also can introduce the temporal dependency for time-series regression modeling. Furthermore, according to the three ratio indexes, the TemporalSVR framework can provide more accurate predictions than the statistical SVR framework and the AR(P)-X model.

6. A case study

Crude oil is an indispensable energy source, which influences the production processes of nearly all manufactured goods for energy and transportation [46]. However, the oil price is often very volatile [47]. A highly accurate crude oil price forecast is crucial for energy security for governments [48].

Therefore, considering this concern, in this section, we employ a crude oil price forecasting project to validate the forecasting capacity of the proposed TemporalSVR framework. Specifically, we use the investigated data (11/01/2002-25/09/2015) in Miao et al. [49] to forecast the weekly nominal West Texas intermediate spot price (WTI). Furthermore, we use 10 predictors from variables that Miao et al. [49] selected for WTI forecasting, including lagWTI, world steel production (Steel World), CRB raw materials index (CRB Rind), ISM manufacturing index (ISM), US dollar index (DXY), monthly measure of log actual global economic activity calculated from monthly change (Kilian Index) reported by Kilian [50], China steel production (Steel China), MSCI world index (MSCI), global crude oil export (Global Export), and global crude oil closing stock (Global Stock). Moreover, we developed an extended window strategy for our TemporalSVR training and forecasting. The initial training set is from 11/01/2002 to 12/08/2011 to forecast one-week ahead WTI; then, along the time axis, we add the new observation data in the training set for the next WTI prediction, which is shown in Fig. 2. We use two error indexes based on the observed crude oil price (WTI) to measure the forecasting performance, including the MAE and the RMSE. We select three basic benchmark models (statistical SVR) [6], nonlinear autoregressive neural network with external input (NARX-NN) [51], SVR-ARIMA, i.e., in the hybrid framework, the statistical SVR is first used to model the pattern between the predictors and the responses and the generated residuals then are fitted by ARIMA, to demonstrate the effectiveness of the proposed TemporalSVR framework in the forecasting project. In addition, we have investigated four recent DL techniques, i.e., Transformer [8], Informer [9], Reformer [10], and Autoformer [11], to illustrate the effectiveness of the proposed TemporalSVR. All predictors are normalized before the modeling. As for the parameter setting of the TemporalSVR framework, we select the radial basic function with $\gamma = 0.01$ as the kernel and set the regularization parameter C at 1. In the NARX-NN, the hidden node is fixed at 8. The NARX-NN is executed by using the R Package 'nnfor' [52], while SVR-ARIMA is executed by using R Packages 'forecast' [53] and 'e1071' [33]. The details of the experimental setting for each DL technique are provided in Appendix B. All methods are performed on an Intel i7-8700 CPU with 16.0 GB of RAM.

In the proposed TemporalSVR, the lag parameter P determines the training of the temporal pattern. To select the parameter, some alternative values are set at 1, 2, 3, 4, 5, and 6. According to the AIC change results (shown in Table 6) in the initial training set, lag parameter P is fixed at 4 with AIC Change = −5.38 for the crude oil price forecasting. The lag in the NARX-NN model is set at the same value.

We report the RMSE index, the MAE index, and the execution time of all forecasting models for the WTI forecasting in Table 7. Based on the comparison results, the proposed TemporalSVR achieves the most accurate predictions with high efficiency among the four

Table 4
Nonlinear simulations with c-Laplace innovations: comparison of the MAE, the RMSE, and the MRE.

Panel A: Nonlinear simulations with AR(1) process							
Parameter setting		Statistical SVR			TemporalSVR		
ϕ_1	c	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$
0.4	0.2	0.92	0.96	0.97	1.82	1.92	1.47
0.4	0.6	0.85	0.89	0.81	1.51	1.62	1.08
0.4	1	0.78	0.83	0.81	1.33	1.42	1.05
0.6	0.2	0.62	0.65	0.84	1.76	1.89	1.51
0.6	0.6	0.57	0.61	0.69	1.52	1.65	1.22
0.6	1	0.55	0.59	0.71	1.32	1.43	1.09
0.8	0.2	0.40	0.43	0.74	1.73	1.85	1.57
0.8	0.6	0.38	0.41	0.59	1.47	1.56	1.49
0.8	1	0.35	0.37	0.58	1.30	1.40	1.25
Panel B: Nonlinear simulations with AR(2) process							
Parameter setting		Statistical SVR			TemporalSVR		
ϕ_1	c	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$
0.4	0.2	0.82	0.86	0.96	1.80	1.89	1.47
0.4	0.6	0.76	0.80	0.81	1.49	1.60	1.19
0.4	1	0.70	0.74	0.77	1.29	1.38	1.03
0.6	0.2	0.53	0.56	0.73	1.69	1.81	1.37
0.6	0.6	0.50	0.53	0.69	1.56	1.67	1.55
0.6	1	0.45	0.48	0.68	1.27	1.37	1.16
0.8	0.2	0.28	0.30	0.49	1.60	1.69	1.62
0.8	0.6	0.27	0.29	0.49	1.41	1.49	1.53
0.8	1	0.25	0.27	0.47	1.30	1.36	1.33

Table 5
Nonlinear simulations with mixture innovations: comparison of the MAE, the RMSE, and the MRE.

Panel A: Nonlinear simulations with AR(1) process							
Parameter setting		Statistical SVR			TemporalSVR		
ϕ_1	c	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$
0.4	0.2	0.88	0.95	0.84	1.64	1.76	1.20
0.4	0.6	0.85	0.90	0.85	1.48	1.59	1.05
0.4	1	0.79	0.84	0.78	1.36	1.46	1.04
0.6	0.2	0.63	0.67	0.84	1.65	1.76	1.50
0.6	0.6	0.58	0.62	0.81	1.43	1.55	1.23
0.6	1	0.55	0.58	0.81	1.32	1.42	1.16
0.8	0.2	0.39	0.41	0.69	1.55	1.67	1.52
0.8	0.6	0.38	0.41	0.64	1.38	1.48	1.39
0.8	1	0.34	0.37	0.58	1.30	1.40	1.59
Panel B: Nonlinear simulations with AR(2) process							
Parameter setting		Statistical SVR			TemporalSVR		
ϕ_1	c	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$	$\text{ratio}_{\text{MAE}}$	$\text{ratio}_{\text{RMSE}}$	$\text{ratio}_{\text{MRE}}$
0.4	0.2	0.81	0.85	0.91	1.67	1.77	1.24
0.4	0.6	0.76	0.81	0.78	1.49	1.59	1.07
0.4	1	0.74	0.78	0.80	1.35	1.44	1.05
0.6	0.2	0.52	0.55	0.79	1.60	1.70	1.42
0.6	0.6	0.50	0.53	0.72	1.43	1.52	1.31
0.6	1	0.47	0.50	0.63	1.30	1.38	1.14
0.8	0.2	0.29	0.31	0.52	1.46	1.55	1.51
0.8	0.6	0.27	0.30	0.49	1.40	1.49	1.55
0.8	1	0.26	0.28	0.53	1.27	1.33	1.31

Table 6
The AIC change with different lag P values.

P	1	2	3	4	5	6
Change	0	−0.84	−1.90	**−5.38**	−2.27	8.44

investigated benchmark models. There are three points. The first one is that incorporating the temporal pattern in SVR can remarkably reduce forecasting errors. For example, the RMSE and MAE indexes decrease from 2.55 (SVR) to 2.22 (TemporalSVR), and from 1.93 (SVR) to 1.73 (TemporalSVR), respectively. Next, we compare the TemporalSVR with the simple combined SVR-ARIMA model which assembles the prediction from the two combined models. As explained, the combined method is not very elegant where two loss functions are not consistent. As a result, the two error indexes of the SVR-ARIMA model (the RMSE 2.33 and the MAE 1.78) are worse than those of our proposed TemporalSVR. Finally, in comparison to the popular NARX-NN model, the proposed TemporalSVR is slightly superior with a smaller RMSE index. The proposed TemporalSVR is more efficient than the NARX-NN model. The execution time of the proposed framework is around 635.42 s, while that of the NARX-NN model is nearly 1,731.35 s even with 8 hidden nodes.

In addition to two error indexes (MAE and RMSE), the success ratio, i.e., the WTI movement direction accuracy indicator, is used to measure the forecasting performance for WTI. Here, the movement direction prediction results are visualized in Fig. 3 where 'purple' means 'success' and 'yellow' means 'failure'. Our proposed TemporalSVR can provide more accurate WTI movement direction predictions compared with the other three benchmarks. Moreover, according to the success ratios reported in Table 7, the success ratio by using the TemporalSVR is

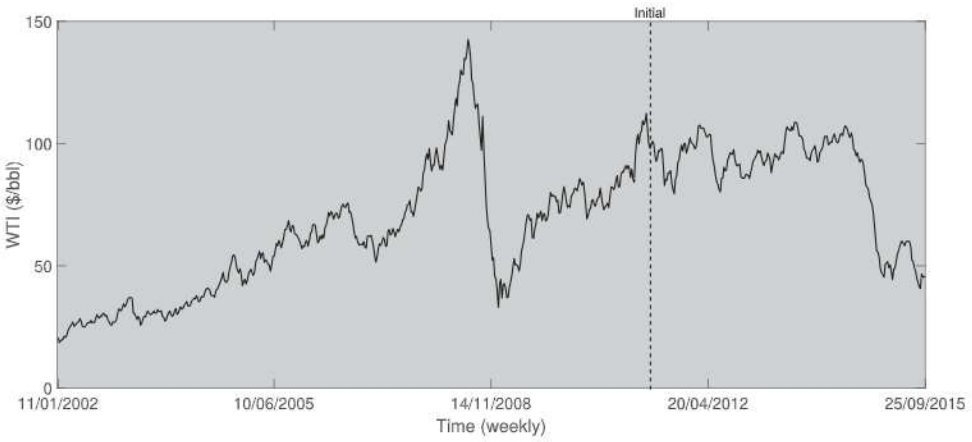

Fig. 2. The investigated weekly crude oil price (WTI).

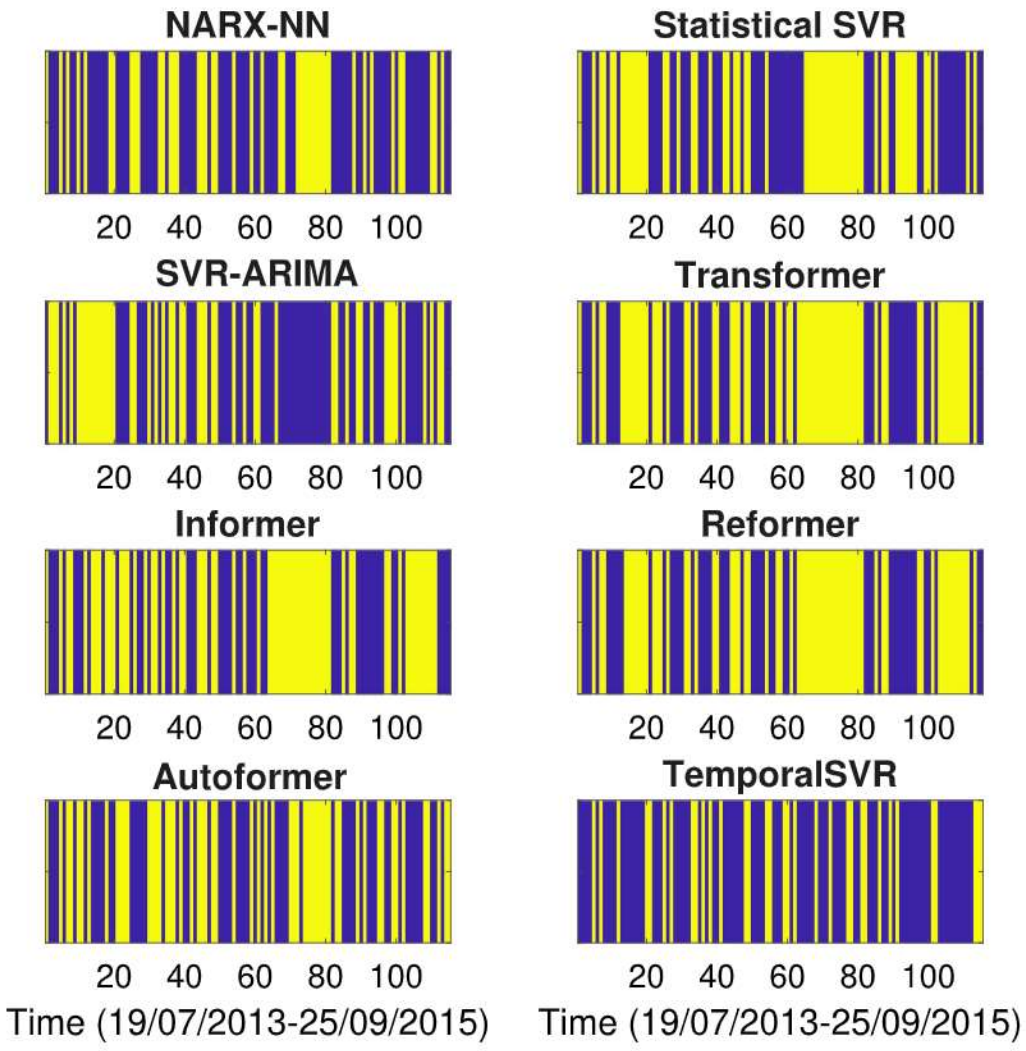

Fig. 3. The WTI movement direction predictions from all considered models in the test set: success (purple) and failure (yellow).

71.30% is much larger than that by using other benchmarks (NARX-NN: 61.74%, SVR: 49.57%, and SVR-ARIMA: 56.52%). Particularly, compared with NARX-NN, our proposed model can provide a higher success ratio with less execution time. Furthermore, we also conduct lasso regression as a benchmark which is the best model in Miao et al. [49], and obtain its corresponding success ratio is 53.04%. Additionally, we explore four advanced deep-learning methods for predicting WTI price. The Root Mean Squared Error (RMSE) values and success ratios for Transformer, Informer, Reformer, and Autoformer are (4.74, 42.61%), (2.81, 45.22%), (5.11, 44.35%), and (2.26, 51.30%) respectively. However, all these metrics are inferior to the corresponding values of 2.22 (RMSE) and 71.30% (success ratio) achieved by our TemporalSVR. This

Table 7
The comparison of eight investigated forecasting models.

Model	MAE	RMSE	Success ratio	Execution time (s)
NARX-NN	1.78	2.33	61.74%	1731.35
Statistical SVR	1.93	2.55	49.57%	3.54
SVR-ARIMA	1.73	2.25	56.52%	12.83
Transformer	3.76	4.74	42.61%	191.50
Informer	2.39	2.81	45.22%	233.95
Reformer	4.07	5.11	44.35%	133.44
Autoformer	1.83	2.26	51.30%	165.63
TemporalSVR	1.73	2.22	71.30%	635.42

observation underscores the superiority of our proposed TemporalSVR, especially when dealing with smaller training datasets, owing to its fewer parameters.

To sum up, according to the crude oil price forecasting project, the proposed TemporalSVR framework can provide more accurate predictions that are highly efficient for temporal data modeling. This implicitly means the proposed iterative procedure for training the model is very efficient, and the working likelihood method is very reliable for estimating the parameter in complex data analysis.

7. Conclusion

SVR, as a popular statistical learning method, can effectively approximate nonlinear systems by mapping features in reproducing a kernel Hilbert space. The temporal pattern often is not sufficiently extracted from the data provided. Thus, we have proposed the TemporalSVR framework to model temporal data where two kernel components are developed to learn the pattern between the predictors and the responses and the temporal pattern simultaneously. In addition, we have designed an iterative procedure for the proposed TemporalSVR training and presented a working likelihood method for estimating the insensitivity parameter c during the iterative procedure.

Furthermore, we have executed a variety of numerical simulations, encompassing both linear and nonlinear scenarios, incorporating autoregressive processes (AR(1) and AR(2)). These simulations aim to showcase the efficacy of the TemporalSVR framework in modeling temporal data. The simulation results reveal two crucial findings:

- Reliability of the training procedure: In the case of linear simulations featuring autoregressive processes generated from insensitive Laplace errors, the working likelihood method demonstrates the capability to offer a dependable estimate of the insensitivity parameter c. In other words, the estimate of c closely aligns with the true value.
- Superior forecasting performance: Firstly, in comparison with the AR-X model, as indicated by the RMSE ratio, our proposed framework exhibits improvements ranging from 8% to 114%. This improvement is attributed to the robustness of the proposed optimization objective and kernels in approximating temporal patterns. Secondly, when compared to statistical SVR without accounting for temporal correlations, we emphasize the significance of an additional linear kernel for capturing temporal information. Specifically, as the temporal contribution increases, the enhancement from the additional kernel becomes more pronounced.

Additionally, we applied the TemporalSVR framework alongside seven benchmark models in a project focused on forecasting crude oil prices. Our proposed framework exhibited the most favorable forecasting performance, achieving an RMSE of 2.22 and a success ratio of 71.30%. The case study further underscored the superiority of our proposed framework in modeling temporal data.

The current study lays the groundwork for several potential avenues of future research. To begin, more efficient optimization approaches shall be explored to find optimal solutions of (4). Next, recognizing the autoregressive process as a specific pattern in time series, the proposed TemporalSVR framework, akin to statistical time-series modeling, holds promise for broader application in handling models exhibiting time-varying "volatility", such as ARIMA and autoregressive conditional heteroscedasticity (ARCH). Furthermore, categorizing the proposed procedure as M-estimators, whose asymptotics (consistency, rate of convergence, and limited distribution) are well-established in the works of Newey and McFadden [54],van der Vaart and Wellner [55] and der Vaart [56], invites exploration into the asymptotic properties of the developed procedures. Additionally, certain methods proposed involve estimators defined by mathematical programming, and a follow-up investigation, inspired by Hsieh et al. [57], could provide further insights into the inference of these estimators. Moreover, there is an opportunity to extend our proposed method into a decomposition-ensemble model, aligning with the approach outlined by Aamir [58]. Finally, the effectiveness of the proposed framework makes it applicable to various time-series forecasting tasks, including those in power systems, business management, and environmental monitoring.

CRediT authorship contribution statement

Jinran Wu: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **You-Gan Wang:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition. **Hao Zhang:** Methodology, Conceptualization.

Declaration of competing interest

All authors declare no conflict of Interest.

Data availability

Data will be made available on request.

Acknowledgments

The initial draft of the paper was a chapter in the Ph.D. degree thesis [59] done by Jinran Wu at the Queensland University of Technology. The source code of the proposed TemporalSVR framework for crude oil price forecasting will be available online when the work is published. This work was supported in part by the Australian Research Council project DP160104292.

Appendix A. The implication for simulation settings

In our numerical simulation, the parameter settings follow some requirements as below. First, the intercept is chosen as 0 because it is a shifting parameter, and it does not affect the simulation results. Second, the parameter σ is fixed as 1. This is because using the parameter set $(\beta_1, \sigma = 1)$ will provide the same simulation results as using $(\sigma\beta_1, \sigma)$ and all the MSE for any methods are scaled by σ^2 and the MAE for all the methods is scaled by σ. This means the relative performance is not affected when we fix σ at 1. We choose different β_1 values to reflect the different extent of variations explained by the regression model. Third, for the AR(p) process to be stationary, we must have $|\phi_1| < 1$ for an AR(1) process $u_t = \phi_1 u_{t-1} + v_t$, and $|\phi_2| < 1$, $\phi_2 + \phi_1 < 1$, and $\phi_2 - \phi_1 < 1$ for an AR(2) process $u_t = \phi_1 u_{t-1} + \phi_2 u_{t-2} + v_t$. Last, the contribution of the temporal component is measured by the variance of the u_t as $D(u_t) = D(v_t)/(1 - \phi_1^2)$ for the AR(1) process and $D(u_t) = (1-\phi_2)D(v_t)/\left((1+\phi_2)(1-\phi_1-\phi_2)(1+\phi_1-\phi_2)\right)$ for the AR(2) process,

Table A.8
The variance values of u_t in numerical simulations.

Panel A: v_t from c-Laplace distribution							
		AR(1)			AR(2): $\phi_2 = 0.1$		
ϕ_1		0.4	0.6	0.8	0.4	0.6	0.8
	0.2	2.42	3.18	5.65	2.16	3.70	9.80
c	0.6	2.70	3.55	2.41	2.58	4.13	10.93
	1.0	3.17	4.17	7.41	2.83	4.85	12.83
Panel B: v_t from mixture distribution							
		AR(1)			AR(2): $\phi_2 = 0.1$		
ϕ_1		0.4	0.6	0.8	0.4	0.6	0.8
	0.2	2.41	3.17	5.62	2.15	3.68	9.75
c	0.6	2.60	2.32	6.09	2.33	3.98	10.54
	1.0	2.93	3.86	6.86	2.62	4.48	11.87

Table B.9
Experimental settings for four DL techniques.

	Transformer	Informer	Reformer	Autoformer
forecasting task		multivariate predict univariate		
input sequence length	3	7	7	7
start token length	1	1	1	1
prediction sequence length	1	1	1	1
encoder input size	11	11	11	11
decoder input size	11	11	5	11
output size	1	1 1	1	
dimension of model	512	512	512	512
number of heads	8	8	8	8
number of encoder layers	2	7	2	1
number of decoder layers	1	1	1	1
dimension of fully convolutional networks	2048	2048	2048	2048
attention factor	3	3	3	3
dropout	0.05	0.05	0.05	0.05
activation	gelu	gelu	gelu	gelu
train epochs	10	10	10	10
batch size of train input data	12	12	12	12
early stopping patience	3	3	3	3
optimizer learning rate	0.0001	0.0001	0.0001	0.0001
loss function	MSE	MSE	MSE	MSE
experiments times	1	1	1	1

where $D(v_t)$ is the variance of the innovation v_t. In addition, for innovations generated c-Laplace distribution, its corresponding variance is calculated as:

$$\begin{aligned} D(v_t) &= E(v_t^2) \\ &= 2\int_0^{+\infty} \frac{1}{2(1+\epsilon)} \exp(-|v_t|_\epsilon) v_t^2 dv_t = \frac{\epsilon^3+3\epsilon^2+6\epsilon+6}{3(1+\epsilon)}. \end{aligned}$$

Moreover, we report the variance of u_t with parameter settings used in the numerical simulation in Table A.8.

Appendix B. The parameter settings for four DL techniques

The details of experimental settings are reported in Table B.9.

References

[1] Gilles Blanchard, Olivier Bousquet, Pascal Massart, et al., Statistical performance of support vector machines, Ann. Statist. 36 (2) (2008) 489–531.

[2] Alex J. Smola, Bernhard Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.

[3] Min Zhu, Jinran Wu, You-Gan Wang, Multi-horizon accommodation demand forecasting: A New Zealand case study, Int. J. Tour. Res. 23 (3) (2021) 442–453.

[4] Jinran Wu, You-Gan Wang, Yu-Chu Tian, Kevin Burrage, Taoyun Cao, Support vector regression with asymmetric loss for optimal electric load forecasting, Energy 223 (2021) 119969a.

[5] Peide Shi, Chih-Ling Tsai, A joint regression variable and autoregressive order selection criterion, J. Time Series Anal. 25 (6) (2004) 923–941.

[6] Jinran Wu, You-Gan Wang, A working likelihood approach to support vector regression with a data-driven insensitivity parameter, Int. J. Mach. Learn. Cybern. 14 (3) (2023) 929–945.

[7] Liya Fu, You-Gan Wang, Fengjing Cai, A working likelihood approach for robust regression, Stat. Methods Med. Res. 29 (12) (2020) 3641–3652.

[8] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, Yunhe Wang, Transformer in transformer, in: Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 15908–15919.

[9] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, Wancai Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 11106–11115.

[10] Nikita Kitaev, Lukasz Kaiser, Anselm Levskaya, Reformer: The efficient transformer, in: International Conference on Learning Representations, 2019.

[11] Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Adv. Neural Inf. Process. Syst. 34 (2021) 22419–22430.

[12] Ervin Ceperic, Vladimir Ceperic, Adrijan Baric, A strategy for short-term load forecasting by support vector regression machines, IEEE Trans. Power Syst. 28 (4) (2013) 4356–4364.

[13] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J. Hill, Yan Xu, Yuan Zhang, Short-term residential load forecasting based on LSTM recurrent neural network, IEEE Trans. Smart Grid 10 (1) (2017) 841–851.

[14] Xuejiao Li, Zhiwei Cheng, Qibing Yu, Yun Bai, Chuan Li, Water-quality prediction using multimodal support vector regression: Case study of Jialing River, China, J. Environ. Eng. 143 (10) (2017) 04017070.

[15] Chi-Jie Lu, Tian-Shyug Lee, Chih-Chou Chiu, Financial time series forecasting using independent component analysis and support vector regression, Decis. Support Syst. 47 (2) (2009) 115–125.

[16] Wei Gao, Muhammad Aamir, Ani Bin Shabri, Raimi Dewan, Adnan Aslam, Forecasting crude oil price using Kalman filter based on the reconstruction of modes of decomposition ensemble model, IEEE Access 7 (2019) 149908–149925.

[17] Laiba Sultan Dar, Muhammad Aamir, Zardad Khan, Muhammad Bilal, Nattakan Boonsatit, Anuwat Jirawattanapanit, Forecasting crude oil prices volatility by reconstructing eemd components using ARIMA and FFNN models, Front. Energy Res. 10 (2022) 991602.

[18] Vladimir Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2013.

[19] Rob J. Hyndman, George Athanasopoulos, Forecasting: Principles and Practice, OTexts, 2018.
[20] Li-Juan Cao, Francis Eng Hock Tay, Support vector machine with adaptive parameters in financial time series forecasting, IEEE Trans. Neural Netw. 14 (6) (2003) 1506–1518.
[21] Luca Ghelardoni, Alessandro Ghio, Davide Anguita, Energy load forecasting using empirical mode decomposition and support vector regression, IEEE Trans. Smart Grid 4 (1) (2013) 549–556.
[22] Qinghua Hu, Shiguang Zhang, Man Yu, Zongxia Xie, Short-term wind speed or power forecasting with heteroscedastic support vector regression, IEEE Trans. Sustain. Energy 7 (1) (2015) 241–249.
[23] Jinran Wu, You-Gan Wang, Iterative learning in support vector regression with heterogeneous variances, IEEE Trans. Emerg. Top. Comput. Intell. 7 (2) (2022) 513–522.
[24] Bo-Juen Chen, Ming-Wei Chang, et al., Load forecasting using support vector machines: A study on EUNITE competition 2001, IEEE Trans. Power Syst. 19 (4) (2004) 1821–1830.
[25] Jie Shi, Wei-Jen Lee, Yongqian Liu, Yongping Yang, Peng Wang, Forecasting power output of photovoltaic systems based on weather classification and support vector machines, IEEE Trans. Ind. Appl. 48 (3) (2012) 1064–1069.
[26] Mujahed Al-Dhaifllah, David T. Westwick, Identification of auto-regressive exogenous Hammerstein models based on support vector machine regression, IEEE Trans. Control Syst. Technol. 21 (6) (2012) 2083–2090.
[27] Bangzhu Zhu, Yiming Wei, Carbon price forecasting with a novel hybrid ARIMA and least squares support vector machines methodology, Omega 41 (3) (2013) 517–524.
[28] Larry R. Medsker, L.C. Jain, Recurrent neural networks, Des. Appl. 5 (64–67) (2001) 2.
[29] Hansika Hewamalage, Christoph Bergmeir, Kasun Bandara, Recurrent neural networks for time series forecasting: Current status and future directions, Int. J. Forecast. 37 (1) (2021) 388–427.
[30] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[32] Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola, Vladimir Vapnik, et al., Support vector regression machines, Adv. Neural Inf. Process. Syst. 9 (1997) 155–161.
[33] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, Friedrich Leisch, Chih-Chung Chang, Chih-Chen Lin, Maintainer David Meyer, Package 'e1071', R J. (2019) 1–66.
[34] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–27.
[35] Vladimir Cherkassky, Yunqian Ma, Practical selection of SVM parameters and noise estimation for SVM regression, Neural Netw. 17 (1) (2004) 113–126.
[36] Yunan Wu, Lan Wang, A survey of tuning parameter selection for high-dimensional regression, Annu. Rev. Stat. Appl. 7 (2020) 209–226.
[37] Bernhard Scholkopf, Peter L. Bartlett, Alex J. Smola, Robert Williamson, Shrinking the tube: A new support vector regression algorithm, Adv. Neural Inf. Process. Syst. 11 (1999) 330–336.
[38] Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Science & Business Media, 2009.
[39] Zeyi Wen, Bin Li, Ramamohanarao Kotagiri, Jian Chen, Yawen Chen, Rui Zhang, Improving efficiency of SVM k-fold cross-validation by alpha seeding, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 2768–2774.
[40] N. Eugene Savin, Conflict among testing procedures in a linear regression model with autoregressive disturbances, Econometrica (1976) 1303–1315.
[41] You-Gan Wang, Jinran Wu, Zhi-Hua Hu, Geoffrey J. McLachlan, A new algorithm for support vector regression with automatic selection of hyperparameters, Pattern Recognit. 133 (2023) 108989.
[42] Miguel Carrasco, Julio López, Sebastián Maldonado, Epsilon-nonparallel support vector regression, Appl. Intell. 49 (2019) 4223–4236.
[43] Bruce Maxwell Brown, You-Gan Wang, Standard errors and covariance matrices for smoothed rank estimators, Biometrika 92 (1) (2005) 149–158.
[44] Ji Zhu, Saharon Rosset, Robert Tibshirani, Trevor Hastie, 1-norm support vector machines, Adv. Neural Inf. Process. Syst. 16 (2003) 49–56.
[45] Wei Chu, S. Sathiya Keerthi, Chong Jin Ong, Bayesian support vector regression using a unified loss function, IEEE Trans. Neural Netw. 15 (1) (2004) 29–44.
[46] Sunil Butler, Piotr Kokoszka, Hong Miao, Han Lin Shang, Neural network prediction of crude oil futures using B-splines, Energy Econ. 94 (2021) 105080.
[47] Ana María Herrera, Liang Hu, Daniel Pastor, Forecasting crude oil price volatility, Int. J. Forecast. 34 (4) (2018) 622–635.
[48] Mengxi He, Yaojie Zhang, Danyan Wen, Yudong Wang, Forecasting crude oil prices: A scaled PCA approach, Energy Econ. 97 (2021) 105189.
[49] Hong Miao, Sanjay Ramchander, Tianyang Wang, Dongxiao Yang, Influential factors in crude oil price forecasting, Energy Econ. 68 (2017) 77–88.
[50] Lutz Kilian, Not all oil price shocks are alike: Disentangling demand and supply shocks in the crude oil market, Amer. Econ. Rev. 99 (3) (2009) 1053–1069.
[51] Tsungnan Lin, Bill G. Horne, Peter Tino, C. Lee Giles, Learning long-term dependencies in NARX recurrent neural networks, IEEE Trans. Neural Netw. 7 (6) (1996) 1329–1338.
[52] Nikolaos Kourentzes, nnfor: Time series forecasting with neural networks, 2019, URL https://CRAN.R-project.org/package=nnfor, R package version 0.9, 6.
[53] Rob J. Hyndman, Yeasmin Khandakar, et al., Automatic time series forecasting: The forecast package for R, J. Stat. Softw. 27 (3) (2008) 1–22.
[54] Whitney K. Newey, Daniel McFadden, Large sample estimation and hypothesis testing, in: Handbook of Econometrics, vol. 4, 1994, pp. 2111–2245.
[55] A.W. van der Vaart, Jon A. Wellner, Weak Convergence and Empirical Processes: With Applications to Statistics, Springer Science & Business Media, 2013.
[56] Aad W. Van der Vaart, Asymptotic Statistics, vol. 3, Cambridge University Press, 2000.
[57] Yu-Wei Hsieh, Xiaoxia Shi, Matthew Shum, Inference on estimators defined by mathematical programming, J. Econometrics 226 (2) (2022) 248–268.
[58] Muhammad Aamir, Crude Oil Price Forecasting Based on the Reconstruction of IMFs of Decomposition Ensemble Model with ARIMA and FFNN Models, (Ph.D. thesis), Universiti Teknologi Malaysia, 2018.
[59] Jinran Wu, Statistical Support Vector Machines with Optimizations, (Ph.D. thesis), Queensland University of Technology, 2022.