



# Towards an efficient machine learning model for financial time series forecasting

Arun Kumar<sup>1</sup> · Tanya Chauhan<sup>1</sup> · Srinivasan Natesan<sup>1</sup>  · Nhat Truong Pham<sup>2</sup> · Ngoc Duy Nguyen<sup>3</sup> · Chee Peng Lim<sup>3</sup>

Accepted: 25 May 2023 / Published online: 10 June 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

Financial time series forecasting is a challenging problem owing to the high degree of randomness and absence of residuals in time series data. Existing machine learning solutions normally do not perform well on such data. In this study, we propose an efficient machine learning model for financial time series forecasting through carefully designed feature extraction, elimination, and selection strategies. We leverage a binary particle swarm optimization algorithm to select the appropriate features and propose new evaluation metrics, i.e. mean weighted square error and mean weighted square ratio, for better performance assessment in handling financial time series data. Both indicators ascertain that our proposed model is effective, which outperforms several existing methods in benchmark studies.

**Keywords** Time series data · Financial forecasting · Machine learning · Particle swarm optimization

## 1 Introduction

Time series forecasting is important in a multitude of domains such as finance, weather, traffic monitoring, budget analysis, military planning, advanced manufacturing, supply chain, and disease pandemic (Lee et al. 2022; Alghamdi et al. 2022;

Kelany et al. 2020; Ma et al. 2022; Volkova et al. 2017; Sezer et al. 2020). Forecasting is crucial for individuals and organizations in proactive decision making and policy planning. Nowadays, there is a wide array of machine learning and deep learning models for forecasting with time series data, which vary from one domain to another (Henrique et al. 2019; Zhong and Enke 2019; Aguilar-Rivera et al. 2015; Yu et al. 2017, 2018; Jiang and Zhang 2018; Karevan and Suykens 2020; Hewage et al. 2021; Johnstone and Sulungu 2021).

Generally, time series data samples have repeating long-term or short-term patterns, or both. In such cases, one can attempt to learn these repeating patterns by using convolutional and/or new recurrent neural networks or other attention-based learning mechanisms (Lai et al. 2018). In financial domains, on the other hand, data samples normally have a few recurring patterns along with a high level of randomness (Tang and Shi 2021). As a result, it is challenging to design and train a deep neural network (DNN) to produce accurate predictions. Existing state-of-the-art DNNs do not perform well on these data samples, e.g. in daily trading data with volatile information (Lai et al. 2018).

To evaluate a forecasting algorithm, three widely used performance metrics are mean square error (MSE), mean absolute error (MAE), and correlation (CORR) (Lai et al. 2018). However, in financial analysis, a low value of MSE or a high value of CORR may not be useful, especially if

✉ Srinivasan Natesan  
natesan@iitg.ac.in

Arun Kumar  
arun170104020@iitg.ac.in

Tanya Chauhan  
tanya170123052@iitg.ac.in

Nhat Truong Pham  
truongpham96@skku.edu

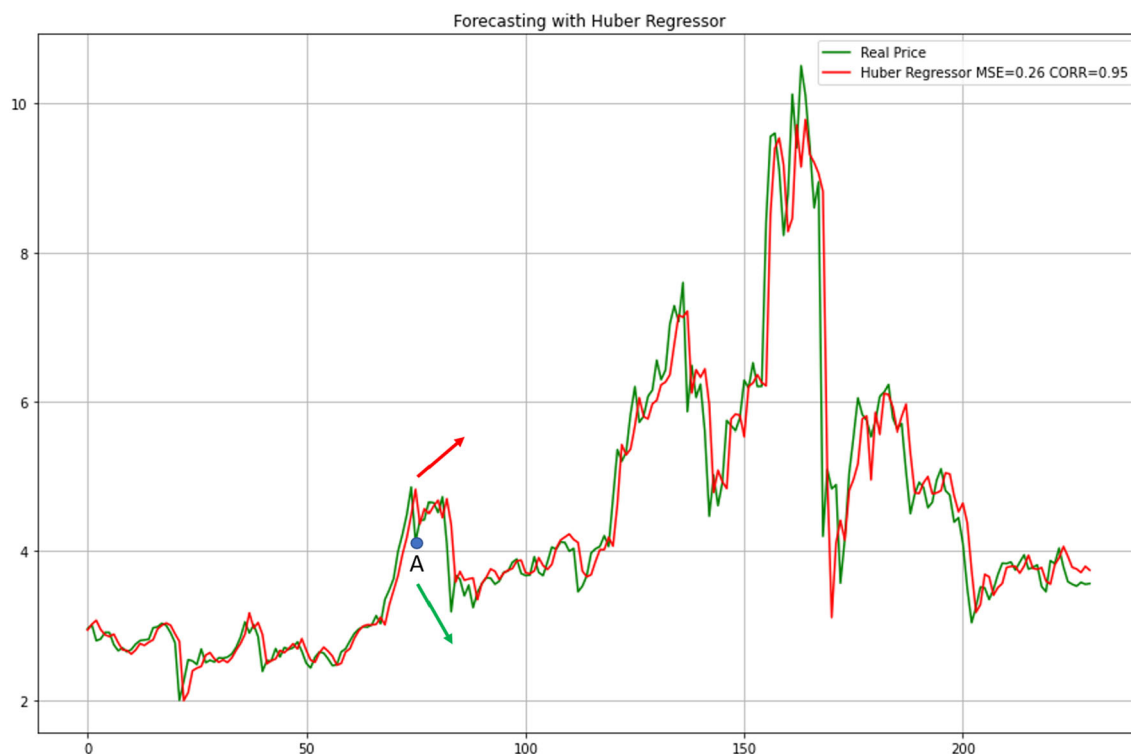
Ngoc Duy Nguyen  
garlicdevs@gmail.com

Chee Peng Lim  
chee.lim@deakin.edu.au

<sup>1</sup> Department of Mathematics, Indian Institute of Technology Guwahati, Guwahati 781039, India

<sup>2</sup> Computational Biology and Bioinformatics Laboratory, Department of Integrative Biotechnology, College of Biotechnology and Bioengineering, Sungkyunkwan University, Suwon, Gyeonggi-do 16419, Republic of Korea

<sup>3</sup> Institute for Intelligent Systems Research and Innovation, Deakin University, Waurn Ponds, VIC 3216, Australia



**Fig. 1** A “shifting” issue of MSE and CORR in financial time series data forecasting. At point A, the actual price decreases, while the forecasted price increases

the interest is in the tendency of the market, e.g. a higher or lower price on the next day. Figure 1 depicts a limitation of MSE and CORR in analysing financial data. Specifically, we use Huber regression (Sun et al. 2020) to forecast a daily time series. We achieve an exceptionally low MSE and high CORR, and the forecasting estimates become a “shifting” version of the ground truth. This shifting issue presents a crucial obstacle, which essentially makes the algorithm unusable. Therefore, in this paper, we propose new performance metrics to evaluate a time series forecasting algorithm, i.e. mean weighted square error (MWSE) and mean weighted square ratio (MWSR), as explained in the next section.

Our proposed approach leverages machine learning to establish a pipeline consisting of data pre-processing tasks, including feature extraction, feature elimination, and feature selection. During feature extraction, an algorithm computes the correlations among multiple time series data and extracts three types of time-domain features: long-term, medium-term, and short-term features. These features capture irregularities and fluctuations throughout the historical data. As shown in Fig. 2, the correlation coefficients act as cross-domain features, while time-domain features are extracted independently for each time series data. In addition, we use XGBoost (Chen et al. 2015) for feature elimination. It is an ensemble method based on decision trees in a gradient boosting framework. We rank the features based on their scores

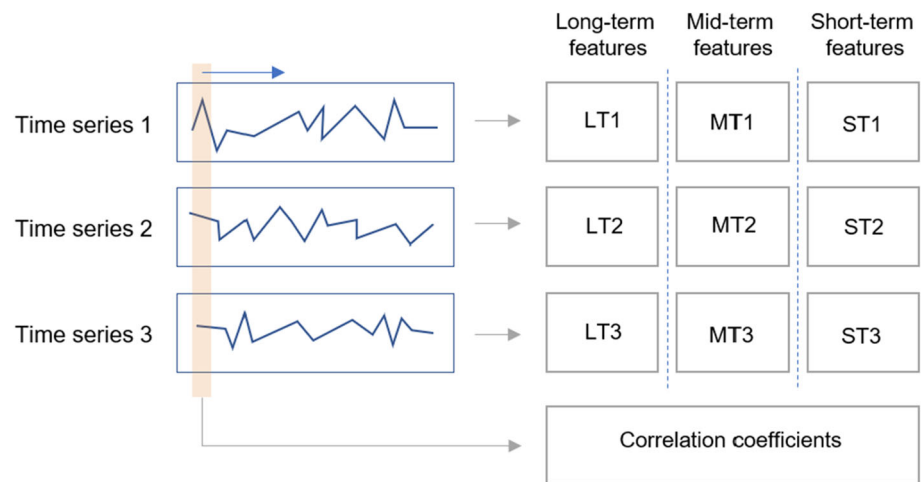
and eliminate those having a score lower than a specified threshold. Then, we perform feature selection using a binary particle swarm optimization (PSO) algorithm (Assareh et al. 2010).

In summary, the contributions of this paper are:

1. We propose two new metrics, namely MWSE and MWSR, for performance evaluation pertaining to financial time series forecasting;
2. We propose a method that can operate with univariate and multivariate time series data forecasting. In multivariate time series data, the correlation coefficients are computed to facilitate forecasting.
3. We leverage machine learning (e.g. linear regression) as the basis of our proposed method. The underlying algorithms can be applied to a wide array of domains, especially where computational resources are limited (e.g. without GPUs).
4. We evaluate our proposed method with benchmark problems, which outperforms state-of-the-art algorithms with respect to MWSE and MWSR, especially in short-term time series forecasting.

The rest of the article is organized as follows. In Sect. 2, we present existing studies related to financial time series forecasting. In Sect. 3, our proposed method is explained in

**Fig. 2** The process of extracting features from multivariate time series data



detail. In Sect. 4, we present performance evaluation and comparison of our proposed method against several state-of-the-art methods. Concluding remarks are given in Sect. 5.

## 2 Related work

Many statistical methods are available in the literature for time series forecasting. Auto-regressive integrated moving average (ARIMA) is widely used in many applications (Singh and Mohapatra 2019; Büyüksahin and Ertekin 2019; Liu et al. 2016; Amini et al. 2016; Barak and Sadegh 2016; Xu et al. 2022). ARIMA is a generalization of auto-regressive moving average (ARMA), which is a combination of auto-regressive (AR) and moving average (MA) techniques. ARMA models operate under the assumption that the time series is stationary. In contrast, ARIMA models handle nonstationary time series by incorporating a parameter to represent the number of non-seasonal differences with respect to stationarity. To build an ARIMA model, the Box–Jenkins method (Box et al. 2015) can be used. It consists of an iterative three-stage procedure, i.e. model identification, parameter estimation, and model checking. Owing to the high computational cost of ARIMA models, they are seldom used in dealing with higher dimensional multivariate time series data. Our developed method, which is based on machine learning, offers an alternative to handle multivariate time series data in a computationally efficient manner.

Vector auto-regression (VAR) achieves a high performance among statistical models in multivariate time series forecasting. VAR generalizes univariate auto-regressive models by enabling processing and analysis of multivariate time series. Owing to its simplicity and effectiveness, it has been widely used in various areas (Bashir and Wei 2018; Vankadara et al. 2022; Wang et al. 2022; Ouma et al. 2022; Safikhani and Shojaie 2022; Deshmukh and Paramasivam

2016; Taveeapiradeecharoen et al. 2019; Munkhdalai et al. 2020; Ngueyep and Serban 2015; Maleki et al. 2020), especially for handling a variety of time series data such as financial data and gene expression series. Several variants of VAR have also been developed, e.g. Gaussian VAR and Elliptical VAR. Gaussian VAR model assumes that the latent innovations are independent and identically distributed (Qiu et al. 2015). Gaussian VAR is restrictive to processing of light-tailed time series. In Qiu et al. (2015), Gaussian VAR was generalized to Elliptical VAR, in order to accommodate heavy-tailed time series. Nevertheless, similar to ARIMA, VAR is also computationally inefficient.

Considering time series forecasting as a regression problem allows us to devise efficient methods such as support vector regression (SVR), ridge regression, and Lasso regression (Lu et al. 2009; Henrique et al. 2018; Ristanoski et al. 2013; Liu et al. 2021). Many regression models with different loss functions and regularization terms have been employed for time series forecasting (Alhnaity and Abbod 2020; Liu and Li 2020; Gupta et al. 2019). These linear methods are efficient for multivariate time series forecasting, owing to the availability of high-quality off-the-shelf solvers in the machine learning community. However, such models fail to capture complex nonlinear relationships in multivariate time series data. In contrary, our proposed method leverages time-domain features to capture the nonlinear relationships in multivariate time series forecasting.

It is a challenging task to develop highly accurate forecasting models for financial data, because they are highly nonlinear, irregular, and volatile in nature. While many traditional statistical models have been used for exchange rate forecasting, such as ARIMA, VAR, linear regression, generalized auto-regressive conditional heteroscedasticity (GARCH) and co-integration models (Chortareas et al. 2011; Tseng et al. 2001; McCrae et al. 2002; Moosa and Vaz 2016; West and Cho 1995; Carriero et al. 2009; Joseph 2001), they

perform poorly and become unusable when handling complex financial time series data. A multiscale decomposition ensemble learning method (Sun et al. 2019) was developed for exchange rate forecasting using a combination of variational mode decomposition (VMD) and a support vector neural network (SVNN). Firstly, the exchange rate time series was decomposed into several components with VMD. Then, the SVNN was used to predict the obtained component series and produce the ensemble results. The VMD/SVNN model outperformed single and ensemble-based forecasting models.

Prophet is a popular method for time series forecasting (Taylor and Letham 2018). It is robust against outliers, missing values, and trend shifts in analysing time series, leading to reliable and high-quality forecasts. It is currently used in real-world environments such as by Facebook to generate reliable forecasts in planning and goal setting. Prophet provides functions to adjust predictions based on human-interpretable parameters. Depending on the problem domain, an expert in the field can effectively adjust the relevant parameters to produce an accurate prediction. Prophet uses an additive regression model that normally works well with default parameters, while users can make adjustments to the relevant components to optimize the quality of forecasting.

Long-term dependencies in time series data can be handled by using recurrent neural networks (RNNs), which is equipped with an optional attention mechanism. However, RNNs fail to capture temporal patterns across multiple time steps. A novel attention mechanism was proposed in (Shih et al. 2019). It uses a set of filters and works in the frequency domain to capture temporal patterns. The mechanism enables the attention component to learn interdependencies among multiple variables, leading to discovery of patterns across multiple time steps.

Long short-term memory (LSTM) is a type of RNNs that has a “larger” memory bank to handle time series efficiently. An improved version of LSTM, namely long short-term memory network-enhanced forget-gate network (LSTM-EFG), was introduced in (Devi et al. 2020). Specifically, LSTM-EFG uses cuckoo search optimization (CSO) (Gandomi et al. 2013) to overcome the limitations of traditional forecasting models. It has shown its usefulness as an operational tool for wind power plant management.

A long- and short-term time-series network (LSTNet) (Lai et al. 2018) was proposed to learn the inherent involvement of long-term and short-term patterns in multivariate time series. LSTNet outperforms state-of-the-art methods on different data sets in terms of MSE and CORR. However, LSTNet does not perform well on financial data, where residuals are uncommon. Different types of time-domain features and an objective function of PSO are used in our proposed method to tackle the problems.

## 3 Proposed methods

Our method leverages PSO, XGBoost and ridge regression, as explained in the following subsections.

### 3.1 Background

#### 3.1.1 PSO

PSO is a population-based meta-heuristic algorithm. The motivation of PSO originates from the social activity of a flock of fishes, where each individual (i.e. a particle) acts as a candidate solution. Particles move around in the search space to find the optimal position. Their movements are motivated by a cognitive desire to perform an efficient exploration. Because PSO does not use gradient information, its objective function has to be differentiable. Although PSO is originally used for tackling search problems in a continuous search-space problems, it is possible to utilize a binary version of PSO (Kennedy and Eberhart 1997) for optimization in the discrete space. In particular, we use binary PSO for feature selection in this study. Specifically, a particle is treated as a vector of  $n$  variables  $p_i = [x_{i1}, x_{i2}, \dots, x_{in}]$  where  $i > 0$  and  $x_{ij} \in \{0, 1\}$ . If  $x_{ij} = 0$ , the corresponding feature is “off” and vice versa. By using an objective function,  $F$ , we can select a set of features that optimizes  $F$  by movements of all particles  $p_i$ . Figure 3 depicts the use of binary PSO in feature selection.

#### 3.1.2 XGBoost

XGBoost, developed in (Chen and Guestrin 2016), uses gradient boosting to ensemble a set of weak decision-tree-based regressors. Decision trees are generated by computing the gradient descent information of an objective function. XGBoost generates a decision score for each feature, which is then used to rank features and remove irrelevant features with respect to a threshold. Figure 4 shows a general structure of XGBoost.

#### 3.1.3 Ridge regression

Linear regression treats the relationship between inputs and outputs as a linear function. Due to its simplicity, the method is sensitive to a small change in inputs, and is especially unstable against outliers. In such scenarios, the model weights of the model can be arbitrarily large. To tackle this problem, we modify the loss function to incorporate a regularized term that applies penalties with respect to large weights. Using this mechanism in ridge regression, we can restrict the absolute values of weights corresponding to less important inputs. The following functions represent the loss

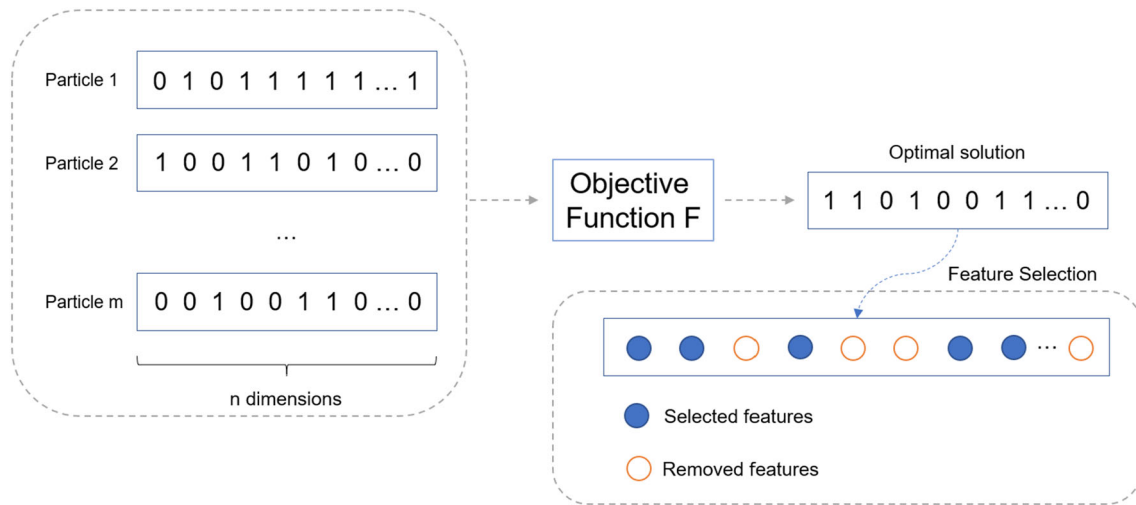


Fig. 3 The relationship between binary PSO and feature selection

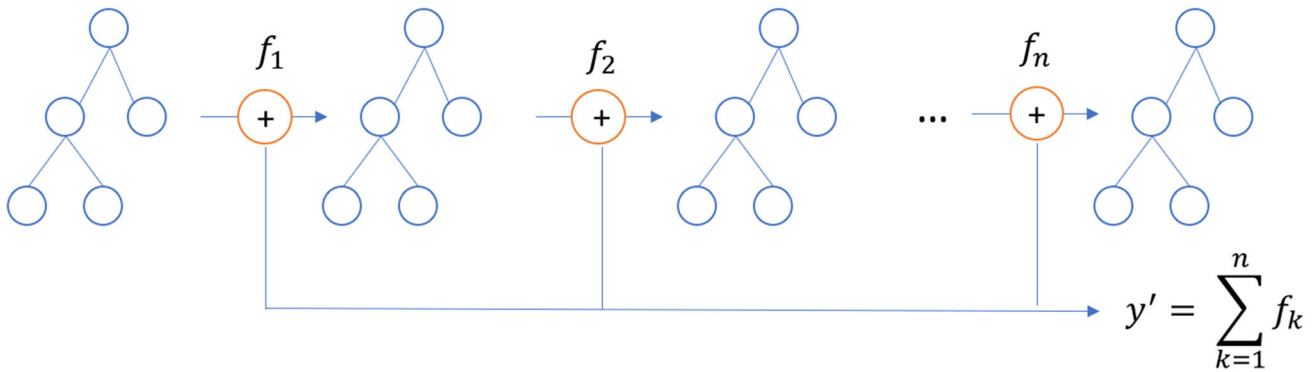


Fig. 4 Architecture of XGBoost

function of ridge regression:

$$F(\theta) = \sum_{i=1}^l \left( y_i - \sum_{j=1}^k \theta_j x_{ij} \right)^2 + \alpha \sum_{j=1}^k \theta_j^2,$$

where  $\theta$  is the weights (parameters) while  $\alpha \geq 0$  controls the amplitude of regularization.

### 3.1.4 Evaluation metrics

We use the following metrics to evaluate the forecasting performance: empirical correlations (CORR), coefficient of determination ( $R^2$ ), mean square error (MSE), mean absolute error (MAE) and our proposed metrics mean weighted square error (MWSE) and mean weighted square ratio (MWSR). Specifically,

$$\text{CORR} = \frac{1}{n} \sum_{i=1}^n \frac{\sum_j (y_{ij} - M(y_i))(\hat{y}_{ij} - M(\hat{y}_i))}{\sqrt{\sum_i (y_{ij} - M(y_i))^2 (\hat{y}_{ij} - M(\hat{y}_i))^2}}, \quad (1)$$

where  $y$  and  $\hat{y} \in \mathbf{R}^{n \times J}$  are the true and predicted values, respectively.  $M(\cdot)$  denotes the mean function. By using the same notation, we have the following definitions:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - M(y))^2}, \quad (2)$$

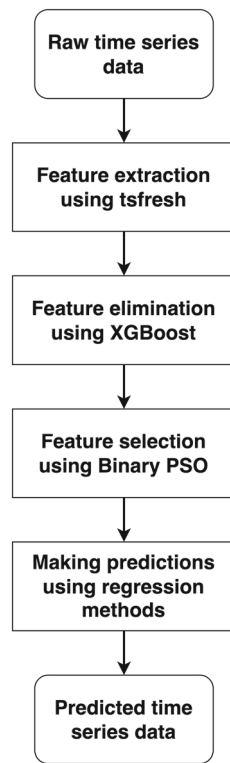
$$\text{MSE} = \sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}, \quad (3)$$

$$\text{MAE} = \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{n}. \quad (4)$$

In finance, users normally are interested in the tendency of an item, i.e. increase or decrease in the stock price the next day. Therefore, we define MWSE to amplify terms that provide an incorrect tendency and vice versa.

$$\text{MWSE} = \sum_{i=2}^n \frac{\Phi((y_i - y_{i-1})(\hat{y}_i - \hat{y}_{i-1}))(y_i - \hat{y}_i)^2}{n - 1}, \quad (5)$$

**Fig. 5** Flowchart of our proposed scheme



and

$$\Phi(x) = \begin{cases} \alpha, & \text{if } x < 0. \\ 1 - \alpha, & \text{otherwise.} \end{cases}$$

where  $\alpha \in [0.5, 1]$ .

In addition, we formulate another metrics, i.e. MWSR, to reflect the dominance of incorrect prediction terms within MSE, specifically where a low MWSR score is desirable.

$$\text{MWSR} = \frac{\text{MWSE}}{\text{MSE}}. \quad (6)$$

### 3.2 Proposed scheme

Our proposed model consists of four stages, as shown in Fig. 5. In feature extraction, we use three windows to capture different parts of the available time series data. These windows represent long-term, medium-term, and short-term characteristics of time-series data. Specifically, the short-term window provides knowledge on the most recent trends or fluctuations in the time series. The medium-term window extends a little further back in time and captures medium-term characteristics (e.g. in weeks). The long-term window aims to focus on long-term characteristics in the time series (e.g. months). In each window, we employ the TSFresh library (Christ et al. 2018) to extract time-domain features. To limit the number of features generated by TSFresh, we select the top 20 relevant features, as follows.

1. Absolute energy

$$\sum_{i=1}^n x_i^2. \quad (7)$$

2. Mean absolute change

$$\frac{1}{n} \sum_{i=1}^{n-1} |x_{i+1} - x_i|. \quad (8)$$

3. Mean

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i. \quad (9)$$

4. Standard deviation

$$\delta = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}. \quad (10)$$

5. Variation coefficient

$$\frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}}{\bar{x}}. \quad (11)$$

6. Skewness

$$s = \frac{\sum_{i=1}^n (x_i - \bar{x})^3}{\delta^3}. \quad (12)$$

7. Kurtosis

$$\frac{\sum_{i=1}^n (x_i - \bar{x})^4}{(n-1)\delta^4}. \quad (13)$$

8. Maximum

$$\mathbf{Max}_{i=1}^n \{x_i\}. \quad (14)$$

9. Minimum

$$\mathbf{Min}_{i=1}^n \{x_i\}. \quad (15)$$

10. Benford correlation

$$\log_{10} \left( 1 + \frac{1}{d} \right), \quad (16)$$

where  $d$  is a leading digit number from 1 to 9.

11. Root mean square

$$\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}. \quad (17)$$

## 12. Absolute sum of changes

$$\sum_{i=1}^{n-1} |x_{i+1} - x_i|. \quad (18)$$

## 13. C3 (lag = 1 and lag = 2)

$$\frac{1}{n-2\text{lag}} \sum_{i=1}^{n-2\text{lag}} x_{i+2\text{lag}} \cdot x_{i+\text{lag}} \cdot x_i. \quad (19)$$

14. Quantile ( $q = 0.1$  and  $q = 0.9$ ).15. Autocorrelation ( $l = 1, l = 2, l = 3$ ).

$$\frac{1}{(n-l)\sigma^2} \sum_{t=1}^{n-l} (x_t - \bar{x})(x_{t+l} - \bar{x}) \quad (20)$$

The last five raw data samples are used as features in our model, resulting in a total of 65 features. In the case of multivariate data forecasting, we calculate the Pearson correlation coefficient for every pair of time series signals. Figure 6 shows our feature extraction procedure with multiple windows.

Subsequently, we eliminate the less important or irrelevant features using XGBoost with a loss function on MWSE. The features with an importance factor lower than  $1e-6$  are removed. Then, binary PSO is employed to select the best features with respect to an objective function based on MWSE. Finally, we use ridge regression for forecasting. Algorithm 1 presents our proposed method in detail. In the case of multivariate time series, the algorithm is repeatedly applied to each signal.

**Algorithm 1** Forecasting with ridge regression and PSO

---

```

1:  $X_{train}, Y_{train}, X_{valid}, Y_{valid} = \text{get\_data}()$            ▷ Prepare data
2:  $Reg = \text{Ridge}()$                                            ▷ Create Ridge regressor
3:
4: Function particle(mask):   ▷ Objective function for each particle
5:    $X_{tsub} = X_{train}[\text{mask} = 1]$ 
6:    $X_{vsub} = X_{valid}[\text{mask} = 1]$ 
7:    $Reg.\text{fit}(X_{tsub}, Y_{train})$ 
8:    $Y_{vpred} = Reg.\text{predict}(X_{vsub})$ 
9:   return MWSE( $Y_{vpred}, Y_{valid}$ )
10:
11:  $Pso = \text{PSO}(\text{objective}=\text{particle})$ 
12:  $F_{best} = Pso.\text{optimise}()$                                ▷ Get best features
13:
14:  $X_{train} = X_{train}[F_{best}]$ 
15:  $Reg.\text{fit}(X_{train}, Y_{train})$                              ▷ Train the regressor with filtered data
16:
17:  $X_{test}, Y_{test} = \text{get\_test\_data}()$                    ▷ Get test data
18:  $X_{test} = X_{test}[F_{best}]$ 
19:  $Y_{tpred} = Reg.\text{predict}(X_{test})$ 
20: return  $Y_{tpred}$ 

```

---

## 4 Performance evaluation

### 4.1 Simulation study

To evaluate our proposed method, a well-known challenging problem is used, i.e. the crypto-currency signal (EOS) forecasting problem (EOSIO 2022). We collect trading data of EOS from 1 January, 2019 to 1 August, 2021. The data set has 5 different time series, i.e. opening price, maximum price, minimum price, closing price, and volume. We take the logarithm of the data samples and scale them within the range of  $[0, 1]$ . Our aim is to predict the EOS data sample in the next 1, 2, and 3 days. We conduct two case studies: univariate and multivariate time series forecasting. In the univariate case, we predict the daily closing price of EOS. In the multivariate case, we predict 5 time series signals concurrently. We use the most recent 120 (4 months) data samples for performance evaluation. The remaining data samples are divided for training and validation with a ratio of 4:1.

The following algorithms are used for performance comparison:

1. We use ARIMA and VAR for univariate and multivariate cases, respectively. In each case, the algorithm is trained only once. To avoid re-training of the model on a daily basis, we derive a Kalman filter retrieved from the trained model and establish the new model. By observing partial autocorrelation, autocorrelation, and augmented Dickey–Fuller statistics, we adopt the following parameters for ARIMA and VAR: {lag = 2, difference = 0, moving average window = 10}.
2. We use Prophet from Facebook to predict EOS data in both univariate and multivariate cases. Both settings of daily training (one-step prediction) or one-time training (multi-step prediction) are evaluated to select the best model. The default parameters of Prophet are utilized.
3. In LSTNet, we employ a grid search to obtain the optimal parameters: {window = 72, skip = 6, batch size = 16, epochs = 100, learning rate = 0.01, and highway window = 12}.
4. In our proposed algorithm, we use a heuristic search to yield the following parameters: {long-term window = 90, medium-term window = 45, short-term window = 7, PSO acceleration factor = 2.05, weight = 0.7, distance = L2, number of iterations = 3000, and number of particles = number of features}.

### 4.2 Results and discussion

Tables 1 and 2 present the performance of one-day ahead prediction from four algorithms in univariate and multivariate settings, respectively. Our model outperforms other algorithms, and achieves the lowest MWSE and MWSR scores.

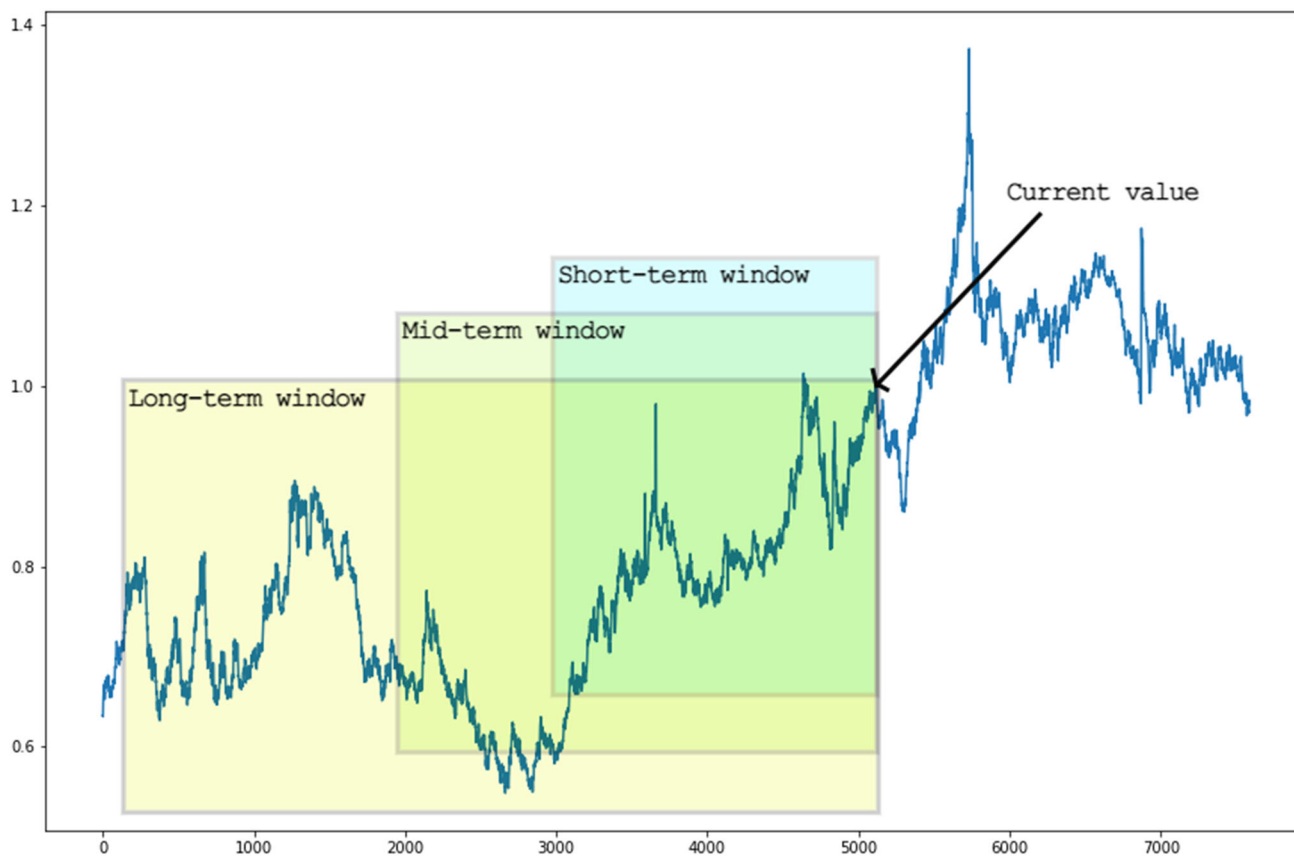


Fig. 6 Multiple windows for time domain features

Table 1 Results of four algorithms in univariate prediction (horizon = 1)

Algorithm	ARIMA	Prophet	LSTNet	Our scheme
MSE	<b>0.00088</b>	0.00378	0.00097	0.00094
MAE	0.02	0.0538	<b>0.0198</b>	0.0199
CORR	<b>0.8427</b>	0.3444	0.8205	0.8252
MWSE	0.00033	0.0021	0.00025	<b>0.00017</b>
MWSR	0.375	0.55	0.257	<b>0.18</b>

The significance of [bold] denotes the best one

In other words, our model is capable of predicting the price trend well, especially in short-term and univariate setting. ARIMA provides stable results in terms of MSE, MAE and CORR, while LSTNet depicts good MWSR values in the multivariate case. Prophet performs the worst in this study.

Similarly, Tables 3 and 4 summarize the performance of four algorithms with a horizon of 2 with respect to univariate and multivariate time series forecasting, respectively. Tables 5 and 6 show the performance of four algorithms with a horizon of 3 in univariate and multivariate settings, respectively. Since Prophet uses the multi-day ahead prediction strategy, its results are the same with different horizons. As such, Prophet is useful for long-term prediction. ARIMA and LST-

Table 2 Results of four algorithms in multivariate prediction (horizon = 1)

Algorithm	VAR	Prophet	LSTNet	Our scheme
MSE	<b>0.002678</b>	0.007836	0.003912	0.002883
MAE	<b>0.0267</b>	0.06912	0.03906	0.03050
CORR	<b>0.8725</b>	0.3603	0.7698	0.8469
MWSE	0.00154	0.003994	0.001894	<b>0.001434</b>
MWSR	0.575	0.509	<b>0.484</b>	0.497

The significance of [bold] denotes the best one

Table 3 Results of four algorithms in univariate prediction (horizon = 2)

Algorithm	ARIMA	Prophet	LSTNet	Our scheme
MSE	<b>0.001386</b>	0.00378	0.001602	0.001562
MAE	<b>0.02466</b>	0.0538	0.02648	0.0277
CORR	<b>0.7474</b>	0.3444	0.6965	0.7059
MWSE	0.0007	0.0021	<b>0.0006341</b>	0.0007417
MWSR	0.505	0.555	<b>0.395</b>	0.47

The significance of [bold] denotes the best one

Net produce good results with horizons of 2 and 3, indicating their usefulness in medium-term forecasting.



**Table 4** Results of four algorithms in multivariate prediction (horizon = 2)

Algorithm	VAR	Prophet	LSTNet	Our scheme
MSE	<b>0.004433</b>	0.007836	0.005376	0.005354
MAE	<b>0.03881</b>	0.06912	0.04539	0.04288
CORR	<b>0.7313</b>	0.3603	0.6418	0.6754
MWSE	0.002526	0.003994	<b>0.002433</b>	0.002497
MWSR	0.569	0.509	<b>0.452</b>	0.466

The significance of [bold] denotes the best one

**Table 5** Results of four algorithms in univariate prediction (horizon = 3)

Algorithm	ARIMA	Prophet	LSTNet	Our scheme
MSE	<b>0.001926</b>	0.00378	0.002204	0.002114
MAE	<b>0.02992</b>	0.0538	0.03238	0.03346
CORR	<b>0.6443</b>	0.3444	0.5762	0.6148
MWSE	<b>0.0010</b>	0.0021	0.001256	0.001179
MWSR	<b>0.519</b>	0.55	0.569	0.557

The significance of [bold] denotes the best

**Table 6** Results of four algorithms in multivariate prediction (horizon = 3)

Algorithm	VAR	Prophet	LSTNet	Our scheme
MSE	<b>0.005517</b>	0.007836	0.006666	0.006693
MAE	<b>0.04532</b>	0.06912	0.05341	0.05211
CORR	<b>0.638</b>	0.3603	0.3901	0.5295
MWSE	<b>0.002899</b>	0.003994	0.003362	0.003843
MWSR	0.525	0.509	<b>0.504</b>	0.57

The significance of [bold] denotes the best one

**Table 7** Results of four algorithms in multivariate prediction with horizon (H) of 1, 2, 3, and 4, using ETH symbol from 1 January, 2019 to 1 January, 2023

H	Model	MSE	MAE	CORR	MWSE	MWSR
1	Ours	0.004226	0.028748	0.878809	0.00137	0.32418363
1	LSTNet	0.007936	0.075008	0.53818	0.003898	0.49117944
1	Prophet	0.015484	0.11014	0.640938	0.007373	0.47616895
1	VAR	0.094557	0.202957	0.32204	0.0519	0.54887528
2	Ours	0.004147	0.030604	0.798275	0.001542	0.37183506
2	LSTNet	0.007747	0.072393	0.473732	0.003565	0.46017813
2	Prophet	0.033693	0.16245	0.410768	0.017384	0.51595287
2	VAR	0.300446	0.292688	0.198496	0.1741	0.57947185
3	Ours	0.004444	0.033241	0.743162	0.002341	0.52677768
3	LSTNet	0.009252	0.082148	0.501601	0.00446	0.48205793
3	Prophet	0.062508	0.222286	0.454044	0.02883	0.46122096
3	VAR	2.226296	0.698038	0.161172	1.314238	0.59032492
4	Ours	0.004665	0.035501	0.699747	0.001684	0.36098607
4	LSTNet	0.012096	0.095033	0.501512	0.006363	0.52604167
4	Prophet	0.102088	0.282429	0.48539	0.050511	0.49477901
4	VAR	3.261225	0.815466	0.097872	1.636897	0.501929

Furthermore, we conducted an extended experiment to evaluate the effectiveness and robustness of our proposed method. Based on trading data of the Ethereum platform (ETH) from January 1, 2019, to January 1, 2023, we apply multivariate prediction with horizons of 1, 2, 3, and 4 for four algorithms. The results of MSE, MAE, CORR, MWSE, and MWSR for each horizon are recorded, as summarized in Table 7. Clearly, our proposed method outperforms all other compared algorithms in terms of expanding the time series data set and changing the symbol (namely ETH).

## 5 Conclusion

In this paper, we have proposed two new metrics, namely MWSE and MWSR, for reliable performance evaluation of financial time series forecasting, particularly for financial data. Instead of relying on raw values, MWSE and MWSR are designed to measure the trend characteristics of financial time series data, which is a critical aspect in finance when the focus is on identifying trends such as increases or decreases in prices over a period of time. We have conducted a performance comparison study with different state-of-the-art algorithms on a challenging problem pertaining to cryptocurrency time series data, i.e. EOS. The results have indicated that our proposed method is efficient for short-term forecasting. ARIMA and LSTNet are good for medium-term forecasting, while Prophet is useful for long-term forecasting. For further work, we will develop a software tool for short-term forecasting daily trading price forecasting for fintech applications.

**Acknowledgements** The authors express their sincere thanks to the referees whose valuable comments helped to improve the presentation.

**Funding** Not applicable.

**Data availability** Data are publicly available.

## Declarations

**Conflict of interest** There is no conflict of interest.

## References

- Aguilar-Rivera R, Valenzuela-Rendón M, Rodríguez-Ortiz J (2015) Genetic algorithms and Darwinian approaches in financial applications: a survey. *Expert Syst Appl* 42(21):7684–7697
- Alghamdi T, Mostafi S, Abdelkader G et al (2022) A comparative study on traffic modeling techniques for predicting and simulating traffic behavior. *Future Internet* 14(10):294
- Alhnaity B, Abbod M (2020) A new hybrid financial time series prediction model. *Eng Appl Artif Intell* 95(103):873
- Amini MH, Kargarian A, Karabasoglu O (2016) Arima-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation. *Electr Power Syst Res* 140:378–390
- Assareh E, Behrang M, Assari M et al (2010) Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran. *Energy* 35(12):5223–5229
- Barak S, Sadeh SS (2016) Forecasting energy consumption using ensemble ARIMA–ANFIS hybrid algorithm. *Int J Electr Power Energy Syst* 82:92–104
- Bashir F, Wei HL (2018) Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm. *Neurocomputing* 276:23–30
- Box GE, Jenkins GM, Reinsel GC et al (2015) *Time series analysis: forecasting and control*. Wiley
- Büyüksahin ÜÇ, Ertekin Ş (2019) Improving forecasting accuracy of time series data using a new ARIMA–ANN hybrid method and empirical mode decomposition. *Neurocomputing* 361:151–163
- Carriero A, Kapetanios G, Marcellino M (2009) Forecasting exchange rates with a large Bayesian VAR. *Int J Forecast* 25(2):400–417
- Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp 785–794
- Chen T, He T, Benesty M et al (2015) Xgboost: extreme gradient boosting. *R package version 04-2* 1(4):1–4
- Chortareas G, Jiang Y, Nankervis JC (2011) Forecasting exchange rate volatility using high-frequency data: is the euro different? *Int J Forecast* 27(4):1089–1107
- Christ M, Braun N, Neuffer J et al (2018) Time series feature extraction on basis of scalable hypothesis tests (tsfresh-a python package). *Neurocomputing* 307:72–77
- Deshmukh SS, Paramasivam R et al (2016) Forecasting of milk production in India with ARIMA and VAR time series models. *Asian J Dairy Food Res* 35(1):17–22
- Devi AS, Maragatham G, Boopathi K et al (2020) Hourly day-ahead wind power forecasting with the EEMD–CSO–LSTM–EFG deep learning technique. *Soft Comput* 24(16):12391–12411
- EOSIO (2022) Home–eosio blockchain software & services. <https://eos.io/>
- Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29(1):17–35
- Gupta D, Pratama M, Ma Z et al (2019) Financial time series forecasting using twin support vector regression. *PLoS ONE* 14(3):e0211–402
- Henrique BM, Sobreiro VA, Kimura H (2018) Stock price prediction using support vector regression on daily and up to the minute prices. *J Financ Data Sci* 4(3):183–201
- Henrique BM, Sobreiro VA, Kimura H (2019) Literature review: machine learning techniques applied to financial market prediction. *Expert Syst Appl* 124:226–251
- Hewage P, Trovati M, Pereira E et al (2021) Deep learning-based effective fine-grained weather forecasting model. *Pattern Anal Appl* 24(1):343–366
- Jiang W, Zhang L (2018) Geospatial data to images: a deep-learning framework for traffic forecasting. *Tsinghua Sci Technol* 24(1):52–64
- Johnstone C, Sulungu ED (2021) Application of neural network in prediction of temperature: a review. *Neural Comput Appl* 33(18):11487–11498
- Joseph NL (2001) Model specification and forecasting foreign exchange rates with vector autoregressions. *J Forecast* 20(7):451–484
- Karevan Z, Suykens JA (2020) Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Netw* 125:1–9
- Kelany O, Aly S, Ismail MA (2020) Deep learning model for financial time series prediction. In: *2020 14th International conference on innovations in information technology (IIT)*. IEEE, pp 120–125
- Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In: *1997 IEEE international conference on systems, man, and cybernetics. Computational cybernetics and simulation*, IEEE, pp 4104–4108
- Lai G, Chang W, Yang Y et al (2018) Modeling long- and short-term temporal patterns with deep neural networks. In: *Collins-Thompson K, Mei Q, Davison BD et al (eds) The 41st international ACM SIGIR conference on research & development in information retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08–12, 2018*. ACM, pp 95–104. <https://doi.org/10.1145/3209978.3210006>
- Lee K, Jeong Y, Joo S et al (2022) Outliers in financial time series data: outliers, margin debt, and economic recession. *Mach Learn Appl* 10(100):420
- Liu C, Hoi SCH, Zhao P et al (2016) Online ARIMA algorithms for time series prediction. In: *Schuermans D, Wellman MP (eds) Proceedings of the thirtieth AAAI conference on artificial intelligence, February 12–17, 2016, Phoenix, Arizona, USA*. AAAI Press, pp 1867–1873. <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12135>
- Liu J, Li Y (2020) Study on environment-concerned short-term load forecasting model for wind power based on feature extraction and tree regression. *J Clean Prod* 264(121):505
- Liu M, Luo K, Zhang J et al (2021) A stock selection algorithm hybridizing grey wolf optimizer and support vector regression. *Expert Syst Appl* 179(115):078
- Lu CJ, Lee TS, Chiu CC (2009) Financial time series forecasting using independent component analysis and support vector regression. *Dec Support Syst* 47(2):115–125
- Ma K, Pang X, Pang H et al (2022) A novel method for favorable zone prediction of conventional hydrocarbon accumulations based on rusboosted tree machine learning algorithm. *Appl Energy* 326(119):983
- Maleki M, Wraith D, Mahmoudi MR et al (2020) Asymmetric heavy-tailed vector auto-regressive processes with application to financial data. *J Stat Comput Simul* 90(2):324–340
- McCrae M, Lin YX, Pavlik D et al (2002) Can cointegration-based forecasting outperform univariate models? an application to Asian exchange rates. *J Forecast* 21(5):355–380
- Moosa IA, Vaz JJ (2016) Cointegration, error correction and exchange rate forecasting. *J Int Financ Mark Inst Money* 44:21–34

- Munkhdalai L, Li M, Theera-Umpon N et al (2020) Var-gru: a hybrid model for multivariate financial time series prediction. In: Asian conference on intelligent information and database systems. Springer, pp 322–332
- Ngueyep R, Serban N (2015) Large-vector autoregression for multilayer spatially correlated time series. *Technometrics* 57(2):207–216
- Ouma YO, Moalafhi DB, Anderson G et al (2022) Dam water level prediction using vector autoregression, random forest regression and MLP-ANN models based on land-use and climate factors. *Sustainability* 14(22):14,934
- Qiu H, Xu S, Han F et al (2015) Robust estimation of transition matrices in high dimensional heavy-tailed vector autoregressive processes. In: International conference on machine learning. PMLR, pp 1843–1851
- Ristanoski G, Liu W, Bailey J (2013) Time series forecasting using distribution enhanced linear regression. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 484–495
- Safikhani A, Shojaie A (2022) Joint structural break detection and parameter estimation in high-dimensional nonstationary VAR models. *J Am Stat Assoc* 117(537):251–264
- Sezer OB, Gudelek MU, Ozbayoglu AM (2020) Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. *Appl Soft Comput* 90(106):181
- Shih SY, Sun FK, Hy Lee (2019) Temporal pattern attention for multivariate time series forecasting. *Mach Learn* 108(8):1421–1441
- Singh S, Mohapatra A et al (2019) Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. *Renew Energy* 136:758–768
- Sun Q, Zhou WX, Fan J (2020) Adaptive Huber regression. *J Am Stat Assoc* 115(529):254–265
- Sun S, Wang S, Wei Y (2019) A new multiscale decomposition ensemble approach for forecasting exchange rates. *Econ Model* 81:49–58
- Tang C, Shi Y (2021) Forecasting high-dimensional financial functional time series: An application to constituent stocks in Dow Jones index. *J Risk Financ Manag* 14(8):343
- Taveeapiradeecharoen P, Chamnongthai K, Aunsri N (2019) Bayesian compressed vector autoregression for financial time-series analysis and forecasting. *IEEE Access* 7(16):777–16,786
- Taylor SJ, Letham B (2018) Forecasting at scale. *Am Stat* 72(1):37–45
- Tseng FM, Tzeng GH, Yu HC et al (2001) Fuzzy ARIMA model for forecasting the foreign exchange market. *Fuzzy Sets Syst* 118(1):9–19
- Vankadara LC, Faller PM, Hardt M et al (2022) Causal forecasting: generalization bounds for autoregressive models. In: Uncertainty in artificial intelligence. PMLR, pp 2002–2012
- Volkova S, Aytou E, Porterfield K et al (2017) Forecasting influenza-like illness dynamics for military populations using neural networks and social media. *PLoS ONE* 12(12):e0188-941
- Wang H, Liu Y, Wang D et al (2022) Multi-layer echo state network with nonlinear vector autoregression reservoir for time series prediction. In: International conference on neural computing for advanced applications. Springer, pp 489–502
- West KD, Cho D (1995) The predictive ability of several models of exchange rate volatility. *J Econom* 69(2):367–391
- Xu D, Zhang Q, Ding Y et al (2022) Application of a hybrid ARIMA-LSTM model based on the SPEI for drought forecasting. *Environ Sci Pollut Res* 29(3):4128–4144
- Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Lang J (ed) Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden. [ijcai.org](https://doi.org/10.24963/ijcai.2018/505), pp 3634–3640. <https://doi.org/10.24963/ijcai.2018/505>
- Yu R, Li Y, Shahabi C et al (2017) Deep learning: a generic approach for extreme condition traffic forecasting. In: Chawla NV, Wang W (eds) Proceedings of the 2017 SIAM international conference on data mining, Houston, Texas, USA, April 27–29, 2017. SIAM, pp 777–785. <https://doi.org/10.1137/1.9781611974973.87>
- Zhong X, Enke D (2019) Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financ Innov* 5(1):1–20

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.