Hindawi
Complexity
Volume 2020, Article ID 6737951, 8 pages
https://doi.org/10.1155/2020/6737951

Research Article

Enhancing Stock Price Trend Prediction via a Time-Sensitive Data Augmentation Method

Xiao Teng,[1,2] **Tuo Wang**,[1,2] **Xiang Zhang**,[2,3] **Long Lan**,[2,3] **and Zhigang Luo**[1,2]

[1]*Science and Technology on Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China*
[2]*College of Computer, National University of Defense Technology, Changsha 410073, China*
[3]*Institute for Quantum Information & State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha 410073, China*

Correspondence should be addressed to Xiang Zhang; zhangxiang08@nudt.edu.cn, Long Lan; long.lan@nudt.edu.cn, and Zhigang Luo; zgluo@nudt.edu.cn

Received 14 October 2019; Revised 21 December 2019; Accepted 16 January 2020; Published 17 February 2020

Academic Editor: Rongqing Zhang

Stock trend prediction refers to predicting future price trend of stocks for seeking profit maximum of stock investment. Although it has aroused broad attention in stock markets, it is still a tough task not only because the stock markets are complex and easily volatile but also because real short-term stock data is so limited that existing stock prediction models could be far from perfect, especially for deep neural networks. As a kind of time-series data, the underlying patterns of stock data are easily influenced by any tiny noises. Thus, how to augment limited stock price data is an open problem in stock trend prediction, since most data augmentation schemes adopted in image processing cannot be brutally used here. To this end, we devise a simple yet effective time-sensitive data augmentation method for stock trend prediction. To be specific, we augment data by corrupting high-frequency patterns of original stock price data as well as preserving low-frequency ones in the frame of wavelet transformation. The proposed method is motivated by the fact that low-frequency patterns without noisy corruptions do not hurt the true patterns of stock price data. Besides, a transformation technique is proposed to recognize the importance of the patterns at varied time points, that is, the information is time-sensitive. A series of experiments carried out on a real stock price dataset including 50 corporation stocks verify the efficacy of our data augmentation method.

1. Introduction

In financial market, stock price trend is a type of important time series, which is closely relevant to the profits of the investment. Owing to short-term microstructure of the financial market, stock price trend data are highly volatile and uncertain. Though they provide the investors with decision messages for seeking profit maximum of stock investment, forecasting future stock price is still a tough task for decades.

Early methods utilize conventional statistical techniques to predict stock price trend. Among them, autoregression moving average (ARMA) and autoregression-integrated moving average (ARIMA) [1] are the most popular models, and in turn many variants have been explored [2–5]. For instance, Babu and Reddy [3] proposed a linear hybrid model which consists of ARIMA and GARCH models. Li and Chiang [5] proposed a forecasting model by integrating a neurofuzzy system and ARIMA models. Such statistical methods might be too limited to deal with such a dynamic and complex stock market because they fail to unveil the nonlinearity between stock prices at varied time points.

With the boom of deep learning, deep stock price prediction methods start to surge continually. Benefitting from powerful layer-wise representation, deep models have dominated the stock market prediction field [6, 7]. Nelson et al. [8] were the first to apply Vanilla LSTM [9] for stock price prediction and proved its effectiveness as its distinguished ability to capture long-term dependencies in input sequences. Combined with LSTM, some other frameworks [10–16] are also investigated to promote price prediction accuracy. In [10], to discover stock price patterns, the K-means algorithm is firstly used to cluster stock price subsequences, then a multibranch LSTM model is constructed which makes the final prediction based on the learned k clusters. In [13], both wavelet transform and attention mechanism are integrated into LSTM to make the price prediction. In addition, Zhang et al. [12] leveraged different underlying frequency patterns on the basis of LSTM and discrete Fourier transform (DFT) for stock price prediction. In detail, DFT serves to decompose the hidden states of memory cells into several frequency components, and then an inverse Fourier transform (IFT) process is used to combine such components to reconstruct the above hidden states.

As one knows, such deep models highly rely on large scale datasets, and thereby exhibit the capability of effective stock price prediction. In real life, only collecting around 2,520 samples could take ten years, which is far from the requirements for tuning a large collection of parameters in deep models. As a result, this might possibly induce the risk of model overfitting and thus limit the performance of prediction models on unseen data [17, 18]. To defeat this issue, a simple and effective scheme is data augmentation, which aims to augment data by coining new data similar to original data generative distribution. Bengio et al. [19] found that out-of-distribution examples are more beneficial to a deep learner than a traditional shallow one. However, it is nontrivial to exploit most existing data augmentation techniques from image processing regime for stock price data. This is because stock price data fed to the prediction models each time is extremely few, thus any tiny improper operations could hurt the underlying patterns of original data.

In this study, the focus is to address this issue. Here, we propose a simple yet effective data augmentation method for stock price trend prediction. Different from conventional augmentation schemes, which directly impose the transformations such as adding random noises to original time series, our data augmentation method considers how to perform the transformations over the unimportant patterns of original data as well as to preserve the underlying patterns within the dataset. This increases the data diversity. The insight behind the proposed augmentation method is that low-frequency patterns without noisy corruptions could not hurt the true patterns of original time-series data. As in Figure 1, low-frequency patterns are more relevant to the patterns of original data, as it can be viewed as the substitute of original data, while high-frequency ones are more irrelevant and random. According to this observation, amounts of new time-series samples are coined, and their data distribution resembles original time series. In specific, we first decompose the input time series into diverse frequency components and then adopt some transformations to change some components. In this work, the discrete wavelet transform (DWT) [20] is used, which provides detailed frequency and location information about original data. Besides, according to time-sensitive property of time series, we coin new data by reweighting stock price patterns of different time points in time series. This could avoid the impact of overdue historical data over the coined time series. Ablation studies and extensive experiments are carried out on a real stock price dataset including 50 corporation stocks to verify the efficacy of the proposed data augmentation method.

The main contributions of this work are two-fold:

(1) An effective data augmentation method is tailored for stock price data, which coins amounts of new time series by changing high-frequency components of original data while preserving low-frequency components.

(2) Based on the proposed data augmentation method, a decay factor is introduced to control the scale of noise over time series for further refining our method, which distinguishes the importance of the patterns at different time points. This might eliminate the interruption of overdue historical data over the coined time series.

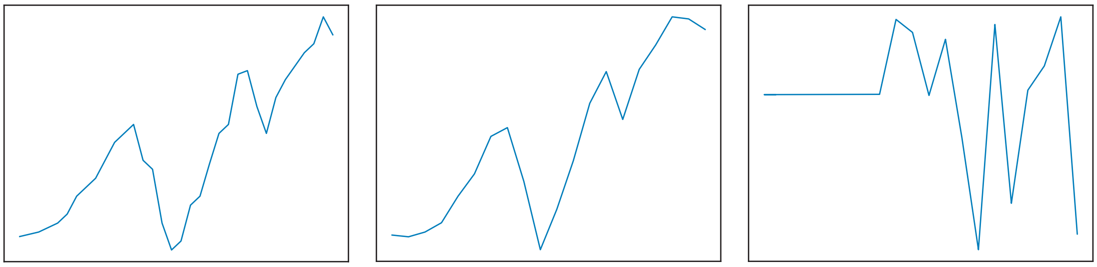
The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the proposed data augmentation method and transformation techniques in our work. In Section 4, a series of comparison experiments are conducted to evaluate the effectiveness of our proposed method. Finally, we conclude the paper in Section 5.

2. Related Work

Although numerous deep learning models have been widely used and showed the superiority over shallow learning methods in many areas, the overfitting problem often emerges because of insufficient data in many applications, including computer vision, natural language processing, and data mining. Data augmentation is one of the most practical ways to relieve such problem, and then numerous augmentation strategies have been applied in computer vision, such as translation, rotation, scaling, flipping, and shearing. Many convolutional neural networks work well in corporation with such augmentation techniques [21]. However, most of these data augmentation methods might not be brutally applied for other areas. Schlüter and Grill applied a series of data augmentation methods for singing voice detection. Results showed that very few methods induce performance gains in this task [18]. Devries and Taylor showed that performing transformations in the input space has limited effectiveness while operating in the feature space which can achieve a better result in many tasks [22].

When using the operations including translation, flipping, and scaling, the transformed image share the same information as the original image. However, these operations cannot keep the property for time series, as it is not obvious to obtain the discriminative information through a similar operation [23]. Thus, how to augment the time-series dataset for stock price prediction is still an open problem at present.

Up to now, several studies have tried to address such a problem. Le Guennec et al. [24] utilized window slicing, window warping, and dataset mixing to improve deep CNN



(a)

(b)

(c)

Figure 1: (a) A time-series instance, (b) the corresponding low-frequency components, and (c) high-frequency components.

models for time-series classification. Fawaz et al. [25] proposed a data augmentation method based on dynamic time warping distance to boost time-series classification. Obviously, such augmentation schemes are specially designed for classification task and might be far from optimal for regression task such as stock price prediction. To augment the dataset for stock price prediction, most efforts focus on utilizing similar stocks with the similar price tendency to expand the dataset. Zhang et al. [26] proposed a solution composed of two stages, i.e., similar stocks were collected according to their retracement probability density function (PDF) in the first stage and then the stocks in the same cluster act as the enlarged dataset to train the model. Besides, Yujin and Young proposed a ModAugNet model including an overfitting prevention LSTM and a prediction LSTM module [27], to relieve the overfitting problem. In the training process, ten companies' stocks highly correlated to the stock market index were collected and then randomnly combined five of them were fed to the prevention LSTM each time. In the end, the final prediction is made together with the features extracted from the target stock index.

Unlike such studies, which aims to augment the dataset by collecting similar stocks to enlarge the dataset [26, 27], we propose a more general data augmentation method for stock prediction analysis based on discrete wavelet transform (DWT), which requires no specific knowledge from external environments. Thus, our method is free from specific situations and can cooperate with other augmentation methods as above.

3. Method

This section firstly overviews the proposed data augmentation method, then details each procedure used in the data augmentation method.

3.1. The Overall Pipeline. As discussed above, the proposed data augmentation method augments dataset by changing high-frequency components with some transformations whilst keeping low-frequency ones unchanged. The idea behind our method is that low-frequency components are close to the original data; thus, low-frequency patterns without noisy corruptions will not hurt the true patterns of time series. Thus, it is more likely to generate new data following the same distribution as the original data. The overall process of the proposed method is shown in Figure 2. To decompose the original data in frequency domain, a series of techniques can be applied here, such as Fourier transform and discrete wavelet transform (DWT) [20]. In this paper, DWT is used as it can provide detailed frequency and location information with respect to the original data. By changing high-frequency components, we can coin amounts of time series. The usual operations to realize this goal contain data corruptions with random noises and interpolation [28]. Such methods are not optimal for time series. To this end, we design a novel transform operation for time series by introducing a decay factor to control the scale of noises over the original data during different time durations. As one knows, time series are time sensitive. When they are treated fairly and operated with the same operations, it might be harmful for keeping the underlying patterns of original data. This easily makes the ground truth uncertain. That is, the generated time series could be nothing but noises. The proposed transform operation introduces a decay factor to keep the underlying important information. In this way, the resultant synthetic time series are generated by combining the new transformed high-frequency components with original low-frequency ones.

To summarize, the proposed augmentation method is composed of three stages. A time series is decomposed into the corresponding high-frequency components and low-frequency ones at the first stage. Then, the proposed transform operation is performed over the high-frequency components, while preserving the low-frequency components. In the end, we compose the transformed high-frequency components and low-frequency ones into a brand time series.

3.2. Data Decomposition in Frequency Domain. In the proposed data augmentation method, the original time series need to be mapped into frequency domain. Many candidates can be used for this purpose. Among them, the discrete wavelet transform (DWT) is a typical continuous signal decomposition method. It can decompose time series into a set of diverse frequency subseries using a series of high-pass and low-pass filters in a level-by-level manner. This meets the requirements of the proposed data augmentation method.

For clarity, we review DWT for subsequent sections. Given a time series $x = \{x_1, x_2, \ldots, x_T\}$, the low- and

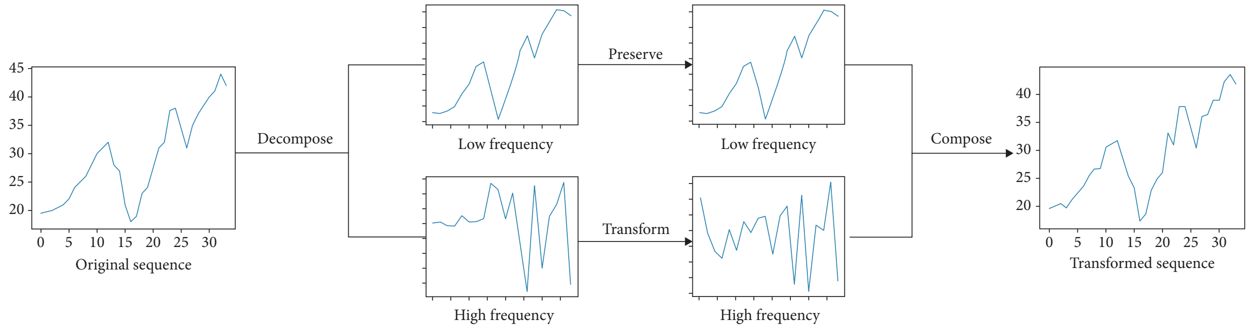


Figure 2: The overview of the proposed data augmentation method.

high-frequency subseries generated in the ith level are denoted as $x^l(i)$ and $x^h(i)$. Then, the corresponding low- and high-frequency subseries can be obtained using a low-pass filter $l = \{l_1, l_2, \ldots, l_K\}$ and a high-pass filter $h = \{h_1, h_2, \ldots, h_K\}$. The concrete functions are as follows:

$$\begin{aligned} a_n^l(i+1) &= \sum_{k=1}^{K} x_{n+k-1}^l(i) \cdot l_k, \\ a_n^h(i+1) &= \sum_{k=1}^{K} x_{n+k-1}^l(i) \cdot h_k, \end{aligned} \tag{1}$$

where $x_n^l(i)$ is the nth element of the low-frequency subseries in the ith level. As $x^l(0)$ is set to the input time series, low- and high-frequency subseries in the $(i+1)$th level $x^l(i+1)$ and $x^h(i+1)$ can be generated from the 1/2 down sampling of the intermediates $a^l(i+1)$ and $a^h(i+1)$, respectively. With the above transform, a set of diverse frequency subseries can be obtained from the original time series as $X(L) = \{x^h(1), x^h(2), \ldots, x^h(L), x^l(L)\}$, where L is the maximum level, and the frequency from $x^h(1)$ to $x^l(L)$ is from high to low.

3.3. Transform over High-Frequency Components. To augment dataset, each sampled time series fed to DWT is decomposed into diverse frequency subseries. Then, a series of transformations can be operated in the high-frequency components to generate new series. Among them, adding the noises following the Gaussian distribution to the high-frequency components is the most commonly used way. The operation can be formulated as follows:

$$s' = s + AX,\ X \sim N\left(0, \sigma_0^2\right), \tag{2}$$

where s is the original time series, which refers to the high-frequency subseries. A is the constant parameter which controls the scale of the noise. X is the noise matrix, of which distribution is with the zero mean and standard deviation $k\sigma_0$, wherein σ_0 is the standard deviation across the whole time series.

Unlike image or natural language data, time series such as stock price data are time sensitive. That is, the stock price at the current time point is closely related to that at short-time points rather than overdue time points. To tackle this problem, a decay factor $\lambda \in (0, 1)$ is introduced, which controls the scale of noises added to the data at different time points:

$$\begin{aligned} s_i' &= s_i + (1-\lambda)^{ik} AX,\ X \sim N\left(0, \sigma^2\right), \\ k &= \frac{L_0}{L_{\text{sub}}}, \end{aligned} \tag{3}$$

where s_i is the ith entry of the high-frequency subseries and i is the index. L_0 and L_{sub} are the length of original time series and subseries, respectively. In this way, it is more likely to generate data in the same distribution as original data whilst preserving the truly underlying patterns of the original series near the ground truth.

Except for simply adding noises to original data, interpolation [28] can be also used here, which is a data transformation commonly used in image processing. For each sample in the dataset, we find the near neighbors to generate new data with interpolation:

$$s' = (\tilde{s} - s)\beta + s, \tag{4}$$

where s refers to the high-frequency subseries of the input series and $\tilde{s}$ is the counterpart of the neighbor sequence. β is the coefficient in the range $\{0, 1\}$, which controls the freedom degree of interpolation. For example, when β is set to 0.5, both original time series and the neighbor ones are balanced. In our work, as the nearest neighbor is too similar to the original time series, we just choose one neighbor which is several time steps near the target one to perform interpolation.

To intuitively understand the above transformations, we illustrate different results of three transformations over the high-frequency components of two synthetic time series in Figure 3. And the details about how the augmented data were utilized are illustrated as Figure 4.

3.4. Stock Price Prediction. Given a time series of stock prices $\{p_t \mid t = 1, 2, \ldots, T\}$, where T is the length of the sequences fed to deep models, and deep models aim to predict the next price p_{T+1}. To get a higher accuracy in stock price, numerous models have been applied and made the progress to some extent. Among them, LSTM serves as the most effective one which captures long-and-short-term dependencies of the input sequences. In this work, we choose LSTM as the base model to conduct the stock price trend prediction. The

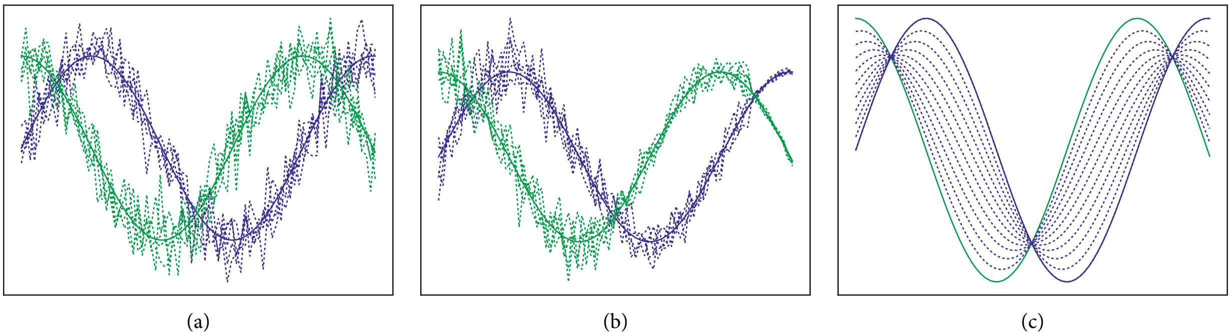


Figure 3: The transformed high-frequency components by (a) random noise, (b) our decay-scale noise operation, and (c) our interpolation operation.

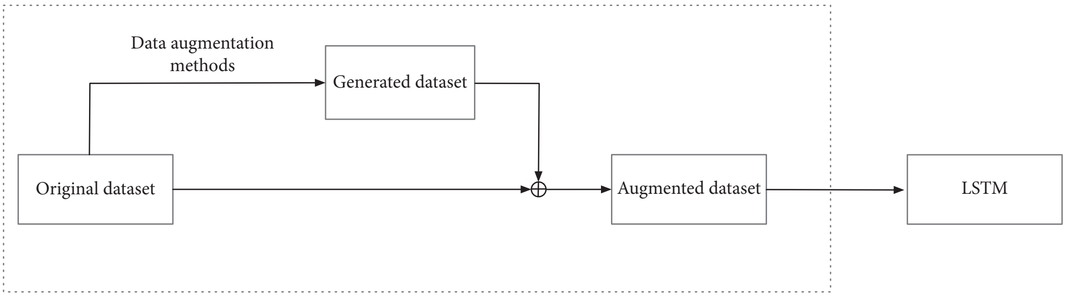


Figure 4: The diagram about how the augmented data are utilized in experiments.

structure of LSTM can be formulated in the following functions:

$$\begin{aligned} i_t &= \mathrm{sigmoid}\left(W_i\left[x_t; h_{t-1}\right] + b_i\right), \\ f_t &= \mathrm{sigmoid}\left(W_f\left[x_t; h_{t-1}\right] + b_f\right), \\ \tilde{c}_t &= \tanh\left(W_c\left[x_t; h_{t-1}\right] + b_c\right), \\ c_t &= i_t \circ \tilde{c}_t + f_t \circ c_{t-1}, \\ o_t &= \mathrm{sigmoid}\left(W_o\left[x_t; h_{t-1}\right] + b_o\right), \\ h_t &= o_t \circ \tanh\left(c_t\right), \end{aligned} \tag{5}$$

where x_t is the input value at each time t, h_t and h_{t-1} are hidden states of the LSTM, and c_t is the memory state. $\mathrm{sigmoid}(\cdot)$ and $\tanh(\cdot)$ are two types of the activation functions for three types of gating units: the input gate i_t, forget gate f_t, and output gate o_t. W_* and b_* denote weight matrices and bias vectors, respectively. "$\circ$" represents the operation of point-wise multiplication. Parameters of the model can be learned by standard back propagation with the mean squared error according to the MSE as the objective function:

$$O\left(\hat{y}_i, y_i\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2, \tag{6}$$

where N is the number of training samples. $\hat{y}_i$ and y_i are the predicted value and the ground truth of the ith sample in the training set, respectively. On the basis of the base model, we can evaluate the efficacy of our proposed data augmentation method.

4. Experiments

In this section, we evaluate the effectiveness of our method on a real-world dataset.

4.1. Dataset. The used dataset is a real-life stock price dataset. It includes the daily open prices of 50 stocks among 10 sectors from 2007 to 2016. The list of the stock symbols is given in Table 1. We treat the dataset from 2007 to 2014 as training set, while stock prices in both 2015 and 2016 are regarded as the validation set and test set, respectively. The LSTM model is trained on the training set of these 50 stocks, and then the average accuracy is evaluated on the test set to validate the performance of the trained model. To augment the dataset and enhance the performance of models, the proposed data augmentation methods were also applied, as shown in Figure 4. For each instance in the training set, a new training sample was also gemnerated to augment the dataset.

4.2. Results. To validate the proposed data augmentation method, a series of comparisons have been conducted by training LSTM with the new augmented dataset and original dataset, respectively. To be specific, to augment the dataset, several transformations have been operated on the high-frequency components of original time series. Then, the most suited transformation technique can be chosen for our method, in light of experimental results. Furthermore, the same operations are also imposed over original time series for

Table 1: Stock symbols of the dataset described in [12].

Sectors	Stock symbols				
Basic materials	BHP	DOW	RIO	SYT	VALE
Cyclicals	AMZN	CMCSA	DIS	HD	TM
Energy	CVX	PTR	RDS-B	TOT	XOM
Financials	BAC	BRK-B	JPM	SPY	WFC
Healthcare	JNJ	MRK	NVS	PFE	UNH
Industrials	BA	GE	MA	MMM	UPS
Noncyclicals	KO	MO	PEP	PG	WMT
Technology	AAPL	GOOGL	INTC	MSFT	ORCL
Telecommunications	CHL	DCM	NTT	T	VZ
Utilities	D	DUK	EXC	NGG	SO

fair comparison. Both MSE and MAE are used to evaluate the performance of the model, which are defined as follows:

$$\begin{aligned} \mathrm{MSE} &= \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2, \\ \mathrm{MAE} &= \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i - y_i\right|, \end{aligned} \tag{7}$$

where N is the number of training samples. $\hat{y}_i$ and y_i are the predicted value and the ground truth of the ith sample of the training set, respectively. In general, the hidden state dimension of LSTM is set to 50 and the length of time series fed to the model is $L = 24$. In addition, the batch size is set to 50. All the parameters are optimized in 2,000 epochs with the RMSProp optimizer and standard mean square error (MSE). In the procedure of the data preprocessing, a soft threshold is used to denoise high-frequency components of training samples induced by the wavelet transform [21]. Then, a set of transformation techniques are adopted, including random noise corruption, decay-scale noise corruption, and interpolation. Likewise, the same operations are also applied for original data. Experimental results are shown in Tables 2–4. To show the effectiveness of our method, two LSTM models, respectively, trained on augmented dataset and original dataset are tested on several individual stocks, and the resultant square error curves are shown in Figure 5.

Tables 2~4 show the results of comparison experiments. In these tables, the left records result from transformation techniques applied to the high-frequency patterns, while the right ones are from the identical transformations applied to original time series. From Table 2, the scale of the random noise has a direct influence on the performance of LSTM. When the scale of random noise is very low (which is set to 0.05), both LSTM models trained on the augmented dataset and original time series can achieve a relative sound results. With the rise in the scale of noise, LSTM on the right tends to be worse than before. The reason could be that the pattern of the input sequence has been damaged when the scale of the random noise is over a certain threshold. With the increase in the scale coefficient, LSTM trained on the augmented dataset from the high-frequency patterns performs better than that trained on the counterpart augmented from original time series. This could be attributed to the fact that adding random noise to high-frequency patterns as well as preserving low-frequency counterparts can capture the primary patterns of original time series, which confirms our previous claims.

Table 2: Comparison experiments by adding random noises with diverse scales over both the high-frequency subseries and original time series.

	Random noise		Random noise	
	High frequency (ours)		Original time series	
Scale coefficient (A)	MSE	MAE	MSE	MAE
0.05	5.147	0.934	5.142	0.939
0.1	**5.111**	**0.934**	5.418	0.986
0.15	5.205	0.931	5.313	0.942
No data augmentation			5.106	0.923

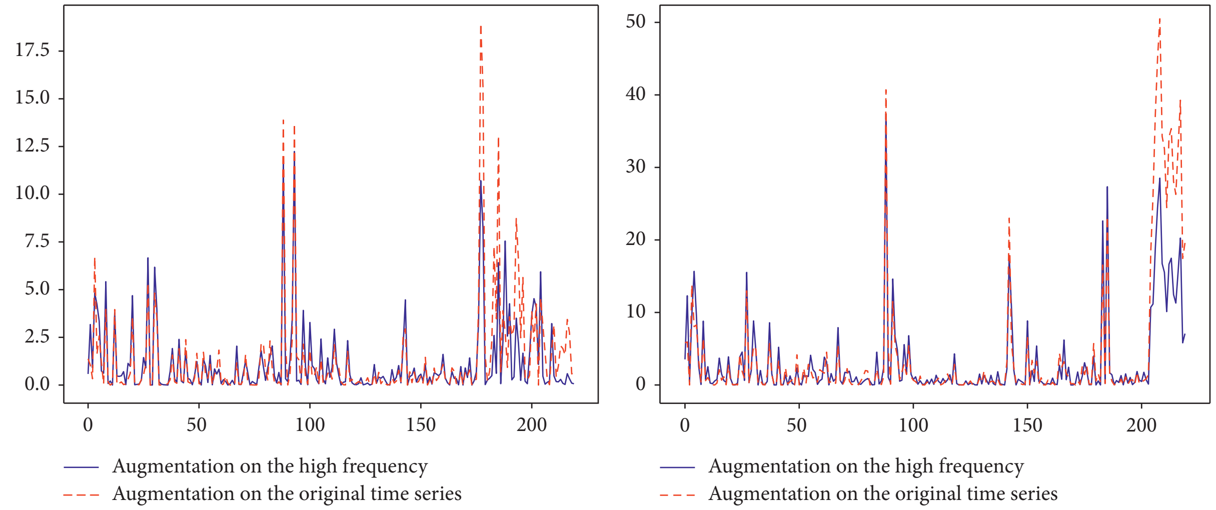
Table 3: Comparison experiments by adding decay-scale random noise with diverse scales to the high-frequency subseries and the original time series.

	Decay-scale random noise		Decay-scale random noise	
	High frequency (ours)		Original time series	
Scale coefficient (A)	MSE	MAE	MSE	MAE
0.05	5.105	0.929	5.119	0.926
0.1	**5.006**	**0.902**	5.130	0.921
0.15	5.117	0.927	5.115	0.925
No data augmentation			5.106	0.923

Table 4: Comparison experiments by applying interpolation with diverse balance coefficients to the high-frequency subseries and the original time series.

	Interpolation		Interpolation	
	High frequency (ours)		Original time series	
Balance coefficient (A)	MSE	MAE	MSE	MAE
0.2	4.921	0.903	5.218	0.951
0.4	5.050	0.915	5.458	0.976
0.6	**4.917**	**0.893**	5.461	0.977
0.8	5.192	0.932	5.5148	0.9647
No data augmentation			5.106	0.923

Another observation is that although different scales of noise have been applied to high-frequency patterns and original time series, respectively, LSTM models achieve limited performance gains. The reason is probably that the importance of data in different periods is not the same. If we simply treat them equally, this might damage the underlying patterns of original time series and make the ground truth confusing. To verify the viewpoint, a decay factor λ is introduced to control the scale of noises, and the results are shown in Table 3. In the experiments, as the trained LSTM models work well when λ is set to 0.1. It can be observed that using λ, LSTM models achieve performance gains, since the decay factor λ can maintain main patterns near the ground truth, to some degree. When adding the decay-scale noise to



(a)

(b)

Figure 5: The square error curves on (a) MA and (b) SPY stocks, respectively.

high-frequency patterns, the trained LSTM can yield the best result. This proves the previous claims.

From Table 4, the effectiveness of the interpolation transformation can be observed. When the parameter λ stays in a low level, LSTM trained on the augmented dataset performs better than that trained on the counterpart augmented dataset from the original time series. With the rise in the scale of noises, the first LSTM can achieve a relative sound result while the second does not work well. The reason could be that the interpolation over the high-frequency patterns can still keep the underlying patterns of original time series, which implies the efficacy of our method.

5. Conclusion

In this paper, we propose a general data augmentation method, which can be applied to the time series without any specific knowledge. It aims to preserve the main patterns of the original time series as it only operates on the high-frequency components. To keep most information near the real label, a decay factor is introduced to control the scale of noises added to time series. This ensures the coined data to be time sensitive. To evaluate the efficacy of the proposed data augmentation method, we conduct the experiments on the real stock price dataset based on the basic LSTM model. Experiment results show that the proposed data augmentation method can boost stock price prediction performance of the basic LSTM model.

Data Availability

The dataset used in our manuscript is widely used in many articles which can be downloaded from the Internet.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Science Foundation of China (61806213).

References

[1] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, CA, USA, 1976.

[2] J. H. Wang and J. Y. Leu, “Stock market trend prediction using ARIMA-based neural networks,” *Proceedings of International Conference on Neural Networks (ICNN)*, Washington, DC, USA, June 1996.

[3] C. N. Babu and B. E. Reddy, “Selected Indian stock predictions using a hybrid ARIMA-GARCH model,” in *Proceedings of the International Conference on Advances in Electronics*, Venice, Italy, August 2015.

[4] S. Wichaidit and S. Kittitornkun, “Predicting SET50 stock prices using CARIMA (Cross Correlation ARIMA),” in *Proceedings of the Computer Science & Engineering Conference*, Chiangmai, Thailand, December 2016.

[5] C. Li and T. W. Chiang, “Complex neurofuzzy ARIMA forecasting—a new approach using complex fuzzy sets,” *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 567–584, 2013.

[6] R. Singh and S. Srivastava, “Stock prediction using deep learning,” *Multimedia Tools and Application*, vol. 76, no. 18, pp. 18569–18584, 2017.

[7] K. Vaibhav and M. Garg, “Prediction of stock closing price by hybrid deep neural network,” *European Journal of Advances in Engineering and Technology*, vol. 5, no. 4, pp. 282–287, 2018.

[8] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, “Stock market's price movement prediction with LSTM neural networks,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, May 2017.

[9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] X. L. Shao, D. Ma, Y. Liu et al., “Short-term forecast of stock price of multi-branch LSTM based on K-means,” *International Conference on Systems & Informatics*, IEEE, 2018.

[11] L. C. Cheng, Y. H. Huang, and M. E. Wu, "Applied attention-based LSTM neural networks in stock prediction," in *Proceedings of the International Conference on Big Data (Big Data)*, Seattle, WA, USA, December 2018.

[12] L. Zhang, C. Aggarwal, and G. J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, Halifax, Canada, August 2017.

[13] Y. Zhao, Y. Shen, Y. Zhu et al., "Forecasting wavelet transformed time series with attentive neural networks," in *Proceedings of the International Conference on Data Mining (ICDM)*, Singapore, November 2018.

[14] Y. Zhao, Y. Shen, and Y. Huang, "DMDP: a dynamic multi-source default probability prediction framework," *Data Science and Engineering*, vol. 4, no. 1, pp. 3–13, 2019.

[15] J. Liu, Y. Chen, K. Liu et al., "Attention-based event relevance model for stock price movement prediction," in *Proceedings of the China Conference on Knowledge Graph and Semantic Computing*, Springer, Chengdu, China, August 2017.

[16] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction," *Expert System with Applications*, vol. 83, pp. 187–205, 2017.

[17] D. M. Hawkins, "The problem of overfitting," *Journal of Chemical Information and Computer Sciences*, vol. 44, no. 6, pp. 1912–1928, 2004.

[18] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, October 2015.

[19] Y. Bengio, F. Bastien, A. Bergeron et al., "Deep learners benefit more from out-of-distribution examples," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, April 2011.

[20] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[21] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[22] T. Devries and G. W. Taylor, "Dataset augmentation in feature space," 2017, http://arxiv.org/abs/1702.05538.

[23] G. Ramponi, P. Protopapas, M. Brambilla et al., "T-CGAN: conditional generative adversarial network for data augmentation in noisy time series with irregular sampling," 2018, http://arxiv.org/abs/1811.08295.

[24] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks," *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, 2016.

[25] H. I. Fawaz, G. Forestier, J. Weber et al., "Data augmentation using synthetic data for time series classification with deep residual networks," 2018, http://arxiv.org/abs/1808.02455.

[26] J. Zhang, W. Rong, Q. Liang, H. Sun, and Z. Xiong, "Data augmentation based stock trend prediction using self-organising map," in *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, Guangzhou, China, November 2017.

[27] B. Yujin and K. H. Young, "ModAugNet: a new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Systems with Applications*, vol. 113, pp. 457–480, 2018.

[28] N. V. Chawla, K. W. Bowyer, L. O. Hall et al., "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.