IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 3, 2021, accepted May 25, 2021, date of publication June 4, 2021, date of current version June 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3086537

Forecasting Stock Market Indices Using Padding-Based Fourier Transform Denoising and Time Series Deep Learning Models

DONGHWAN SONG[1], **ADRIAN MATIAS CHUNG BAEK**[2], **AND NAMHUN KIM**[2]

[1]Department of System Design and Control Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, South Korea
[2]Department of Mechanical Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, South Korea

Corresponding author: Namhun Kim (nhkim@unist.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant through the Korea Government [Ministry of Science and ICT (MSIT)] under Grant 2020-0-01336, and in part by the Artificial Intelligence Graduate School Program [Ulsan National Institute of Science and Technology (UNIST)].

ABSTRACT Approaches for predicting financial markets, including conventional statistical methods and recent deep learning methods, have been investigated in many studies. However, financial time series data (e.g., daily stock market index) contain noises that prevent stable predictive model learning. Using these noised data in predictions results in performance deterioration and time lag. This study proposes padding-based Fourier transform denoising (P-FTD) that eliminates the noise waveform in the frequency domain of financial time series data and solves the problem of data divergence at both ends when restoring to the original time series. Experiments were conducted to predict the closing prices of S&P500, SSE, and KOSPI by applying data, from which noise was removed by P-FTD, to different deep learning models based on time series. Results show that the combination of the deep learning models and the proposed denoising technique not only outperforms the basic models in terms of predictive performance but also mitigates the time lag problem.

INDEX TERMS Deep learning, denoising framework, Fourier transform, stock index prediction, time series.

I. INTRODUCTION

For decades, stock market prediction has been receiving steady attention from researchers and investors as an attractive field. Nonetheless, accurately predicting the future of the stock market has remained an open question because stock markets are dynamic and possess several unpredictable factors. According to the Efficient Market Theory proposed by Fama [1], financial markets are unpredictable because all new information is already reflected on the price. Contrary to this view, numerous studies have predicted stock markets by taking diverse approaches, starting from conventional statistical models to machine learning and deep learning models in accordance with advancements in computational performance [2]–[7].

Banerjee [8] forecasted the stock index for a day using an autoregressive integrated moving average, whereas Liu and Hung [9] studied the stock index volatility through a generalized autoregressive conditional heteroskedasticity model. These econometric models are employed under the assumption that the time series data are linear. Because of linearity assumption, conventional approaches demonstrate limited performances when predicting the financial time series, which are mostly nonlinear and nonstationary. Accordingly, machine learning models, such as the support vector machine and artificial neural network (ANN) models, have been applied to forecast the value or direction of the stock market index to overcome the shortcomings of linear models [10]–[12]. Recently, deep learning models, including long short-term memory (LSTM) [13]–[18], and their variants [19]–[25] have been popularly proposed for stock prediction.

A major problem in stock prediction using deep learning methods is that the financial time series contains considerable noise [26]. When predicting using noise-included data, learning becomes unstable because of the unwanted fitting data generated despite using machine learning and deep learning models [27]. This could result in overfitting or underfitting problems [28], [29]. Moreover, a time lag occurs, wherein the

The associate editor coordinating the review of this manuscript and approving it for publication was Lorenzo Mucchi.

predicted value lags in the direction and size of the actual data values in the past.

In view of these limitations, studies to eliminate noises in financial data have emerged. Raudys *et al.* [30] compared different moving averages and verified the exponential moving average to achieve the best smoothing performance. Babu and Reddy [31] proposed moving average (MA) to smoothen time series data and explore their trend component before applying the ANN model. However, noise removal using MA techniques is accompanied by the time lag problem caused by the reflection of the historical data. The smoothness of the MA approach is inversely proportional to the time lag [30]. Several studies using denoising techniques other than MA have been conducted to reduce the effect of noise in the financial time series data and enhance the capability of prediction models. Lu [32] proposed a denoising technique based on independent component analysis integrated with a back-propagation neural network for stock price prediction. Awajan *et al.* [33] used empirical mode decomposition (EMD) along with a bagging method to predict the daily stock market prices of six countries.

In recent years, denoising algorithms based on transform methods have gained preference for outperforming many traditional methods such as the MA filter and simple nonlinear noise reduction [34]. For example, Yu *et al.* [35] proposed a hybrid model comprising of empirical wavelet transform (EWT) and optimized extreme learning machine (ELM) to present a stable and precise prediction of financial time series. Chan Phooi M'ng and Mehralizadeh [34] proposed Wavelet-PCA denoising (WPCA), a hybrid model using wavelet transform (WT) and principal component analysis. They applied their denoising method with ANN to analyze and forecast the financial future markets. Meanwhile, Li and Tam [36] combined the real-time wavelet denoising (RTWD) with LSTM, a time series-based learning model, to predict East Asian stock market indexes. Bao *et al.* [37] combined wavelet transforms with stacked autoencoders (WSAEs) and LSTM to predict the closing price of six different market indices with corresponding index futures. Their model outperformed other models in the predictive performances. However, these wavelet denoising methods have limitations in terms of retrieving weak signals with magnitudes close to the noise [38]. Fourier transform (FT) is another method that can be applied to various fields for denoising different types of discrete and continuous data including image data [39]–[41]. FT has also been shown to be effective in denoising time series data. Chen and Chen [42] proposed a fuzzy time series forecasting model by combining the entropy discretization technique and fast FT (FFT) algorithm. Despite FT methods being strong denoising methods, they have diverging issues when applied to financial time series due to information lost in the removal process.

This study proposes a denoising technique using FFT with padding to remove the noises of financial time series data without leading to divergence and time lag. We verify the performance by conducting experiments to predict the stock indices of the next day using different time series-based deep learning models and comparing to models from previous research. The remainder of this paper is organized as follows: Section 2 explains the methods used herein, including the proposed methodology; Section 3 presents the details of the experimental procedures; Section 4 summarizes the experimental results and discussion; and Section 5 provides the conclusions of this study.

II. RELATED WORK

A. FOURIER TRANSFORM

FT is a mathematical tool used to convert a finite sequence of waveform data in the time domain into equally spaced data in the frequency domain [43]. The original data are restored through an additional Fourier analysis using FT samples as the coefficients of complex sinusoids at the corresponding FT frequencies. This process is known as the inverse FT. Therefore, a classical FT and its inverse are said to be in a one-to-one relationship between the time $[x(t)]$ and frequency $[X(\omega)]$ domains.

Discrete FT (DFT) is the most common type of Fourier analysis applied to a discrete complex-valued series. DFT breaks down a waveform in a time domain into a series of sinusoidal terms, each with a unique magnitude, frequency, and phase. The DFT process converts the time-based waveform expressed in complex functions into clearer sinusoidal functions, which when combined, can exactly replicate the original waveform. DFT transforming a sequence of N complex numbers $\{x_n\}$ into $\{X_k\}$ is presented below:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N} \tag{1}$$

Its inverse is given by

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k \cdot e^{i2\pi kn/N} \tag{2}$$

where log expression $e^{\pm i2\pi kn/N}$ can be expressed as a combination of sines and cosines according to Euler's formula

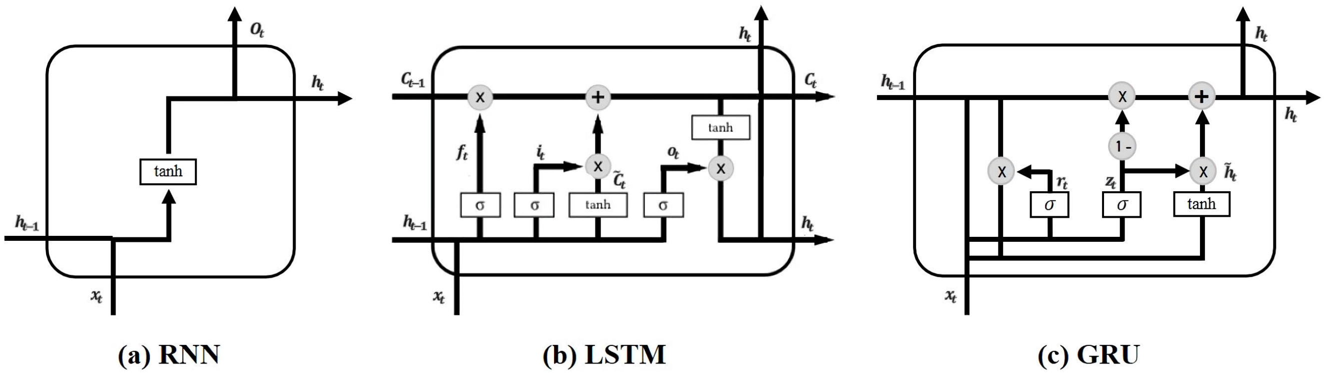
$$e^{i\omega} = \cos\omega + i\sin\omega \tag{3}$$

Hence, equations (1) and (2) can be expressed as equations (4) and (5), respectively:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot [\cos\frac{2\pi kn}{N} - i \cdot \sin\frac{2\pi kn}{N}] \tag{4}$$

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k \cdot [\cos\frac{2\pi kn}{N} + i \cdot \sin\frac{2\pi kn}{N}] \tag{5}$$

DFT shows advantages in many fields, but computing it directly is often computationally too expensive to be practical. FFT was introduced by Cooley and Tukey [44] in 1965. It is an optimized approach for implementing FT. The computation complexity of FT can be reduced from $O(N^2)$ to $O(N\log N)$. FFT is widely used in many practical



(a) RNN (b) LSTM (c) GRU

FIGURE 1. Cell diagram of each time series-based deep learning model.

applications because it is believed to be an effective method for disturbed signal denoising.

B. RECURRENT NEURAL NETWORK

A recurrent neural network (RNN) is a type of ANN with a "memory" that captures past information, making it suitable for arbitrarily long sequence data [45]. This "memory" is known as the hidden state, which is the main and most important feature of an RNN. An RNN comprises an input layer, a hidden layer, and an output layer. The hidden states h_t and the output of a single hidden layer o_t given an input sequence at time step t x_t can be derived as follows:

$$h_t = H(W_{xh}x_t + W_{hh}h_{h-1} + b_h) \tag{6}$$
$$o_t = O(W_{ho}h_t + b_o) \tag{7}$$

where W denotes the connection weights between the layers shared across all steps and b represents the bias vectors. Equations (6) and (7) and Fig. 1(a) show that the hidden state h_t is calculated based on the previous hidden state h_{t-1} and the input at the current step x_t. $H(\cdot)$ and $O(\cdot)$ are the activation functions, which, in most cases, are expressed as tanh.

C. LONG SHORT-TERM MEMORY

Hochreiter and Schmidhuber [46] proposed an RNN variation, called LSTM, which was designed to circumvent the long-term dependency problem. A memory block comprising multiple gates and a memory cell is the most important feature of an LSTM system. Fig. 1(b) represents a memory block of the LSTM model.

LSTM has three main gates: input gate(i_t), forget gate(f_t), and output gate(o_t). The input gate controls the input signal that alters the memory cell state. The forget gate regulates the amount of the previous cell state (h_{t-1}) that can pass through. The output gate decides whether to allow the state of the memory cell to influence the other units. The calculations for each gate and cell state are expressed as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{8}$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{9}$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{10}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{11}$$
$$\tilde{c}_t = \tanh(W_{co}x_t + W_{co}h_{t-1} + b_c) \tag{12}$$
$$h_t = o_t \odot \tanh(c_t) \tag{13}$$

where c_t is the memory cell; $\tilde{c}_t$ is the internal hidden state; $\sigma(\cdot)$ is a sigmoid function; $\tanh(\cdot)$ is a hyperbolic tangent function; and $\odot$ is the elementwise vector product.

D. GATED RECURRENT UNIT

The gated recurrent unit (GRU) was introduced by Cho *et al.* [47] to deal with the vanishing gradient problem of an RNN. GRU is a variation of LSTM because it shows a similar structure to that of a long short-term memory with the forget gate. In GRU, the memory cell and the hidden state are combined into a vector $\tilde{h}_t$, while the input and forget gates are combined into a gate controller known as the reset gate z_t. Despite the simplified parameters, GRU has shown a performance comparable with that of LSTM [48], [49].

The gated unit has several variations; the most general form is depicted in Fig. 1(c). The cell states and the output of each layer can be calculated as follows:

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \tag{14}$$
$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \tag{15}$$
$$\tilde{h}_t = \tanh(W_{xg}x_t + W_{hg}(r_t \odot h_{t-1} + b_z) \tag{16}$$
$$h_t = (z_t \odot h_t + (1 - z_t)\tilde{h}_t) \tag{17}$$

Equations (14) to (17) and Fig. 1(c) show that the update gate r_t, which is also the gate controller, controls both the forget and input gates. Although the reset gate z_t seems like the update gate, its weights and usage are different.

III. PROPOSED PADDING-BASED FT DENOISING METHOD

In the financial market, it is commonly accepted that the volatility resulting from short-term traders, who tend to execute relatively high-frequency trades throughout the day with small assets, can affect the daily price of the stock but has

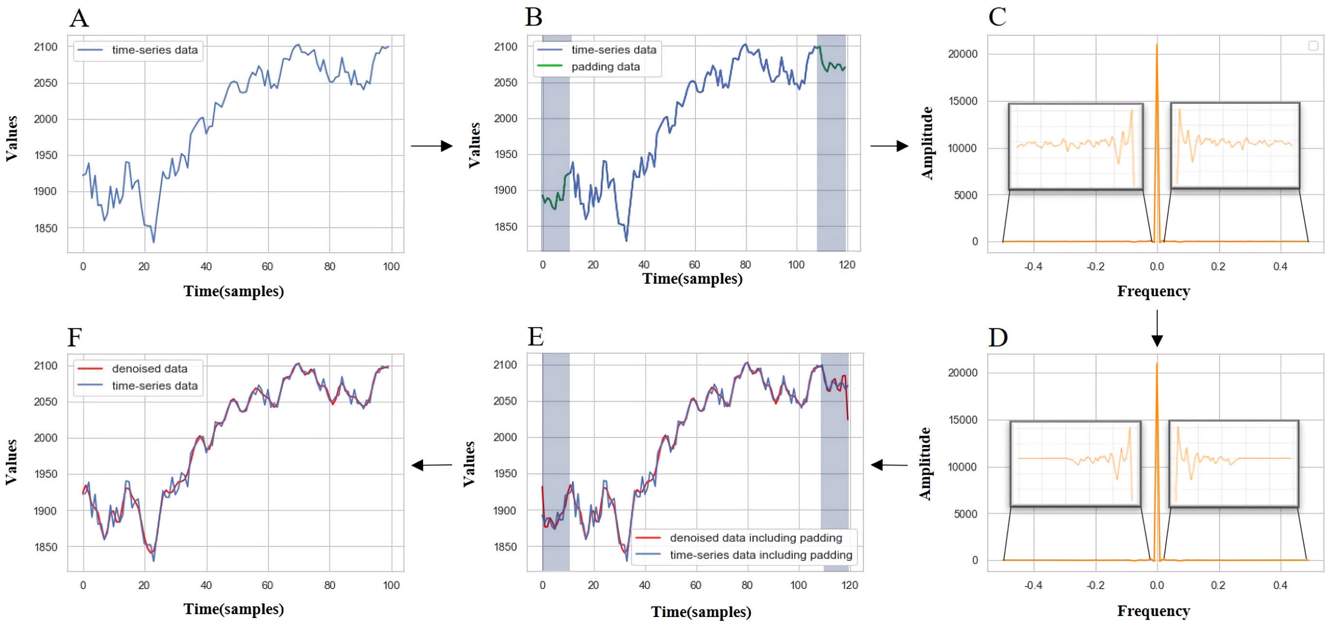


FIGURE 2. Illustration of the steps involved in padding-based Fourier transform denoising.

minimal effects on its momentum or major trend. In this study, such volatility is considered noise. Because of the frequent buying and selling of the short-term traders, this noise corresponds to a waveform with low amplitude and high frequency in the frequency domain and must be removed for stable learning of predictive models. Unfortunately, it is difficult to effectively remove such noise in the time domain. Hence, using filtering approaches, such as MA, only leads to issues such as time lagging and unstable learning. FFT was used in this study to remove the noise in financial time series data by separating the data into different frequency domain waveforms and eliminating the low-amplitude and high-frequency waveform. Both ends of the time series diverge when the original time series is restored with the removed noise waveform, thereby resulting in a large error from the original data.

In this study, we proposed a method for removing noise from a frequency domain and restoring it into smoothed data without significant loss of original information by applying the padding technique to FT. Fig. 2 presents the overall process of the proposed method.

Fig. 2(a) shows the original time series data X(t) with small and highly volatile noises. Here, σ_1 and σ_2, which represents the recent *n* sample volatility at both ends of X(t), are derived as follows:

$$\sigma_1 = \sqrt{\frac{\sum_1^n (X_k - \overline{X_1})}{n}}, \text{ where } \overline{X_1} = \sqrt{\frac{\sum_1^n X_k}{n}} \tag{18}$$

$$\sigma_2 = \sqrt{\frac{\sum_{N-n}^N (X_k - \overline{X_2})}{n}}, \text{ where } \overline{X_2} = \sqrt{\frac{\sum_{N-n}^N X_k}{n}} \tag{19}$$

$$X_{t-1} = X_t + N_1, t = 1, \ldots, 1 - m \tag{20}$$

$$X_{t+1} = X_t + N_2, t = N, \ldots, N + m \tag{21}$$

where, N_1 and N_2 are samples from the normal distribution of $(0, \sigma_1)$ and that of $(0, \sigma_2)$, respectively. These samples, which reflected the recent volatility of the original time series, were attached to both ends of the data. This process was repeated as many as m times (i.e., size of the padding area). Fig. 2(b) displays the resulting time series with padding regions.

Fig. 2(c) illustrates the result of decomposing the padded time series data (padded as above in the time domain) into different $N + 2m$ frequencies using FFT. The decomposed frequencies displayed a bilateral symmetry based on the center. The x-axis in Fig. 2(c) represents the decomposed frequencies, in which the absolute value increased from the center to both ends. The y-axis represents the amplitude of each signal waveform, which also corresponded to the Fourier coefficient. A large value means a waveform that significantly affects the original data. In other words, waveforms with low amplitude and high frequency are considered noise and should be removed to smoothen the original data because they generate considerable variation in a short period.

Therefore, the Fourier coefficient for a waveform, in which the frequency does not exceed ε, is set to 0 to remove the noise (Fig. 2(d) and (22)).

$$A_k = \begin{cases} A_k & \text{if } f_k < |\varepsilon| \\ 0 & \text{if } f_k \geq |\varepsilon| \end{cases}, k = 1 - m, \ldots, N + m, \tag{22}$$

where A_k is the frequency amplitude at k and ε is the frequency threshold value. After removing the noise waveform of less than ε, $N + 2m$ waveforms with different frequencies were multiplied with the corresponding Fourier coefficient and restored to the original time series through inverse FFT.

In this process, each frequency has a different periodic function form. Moreover, divergence occurs at both ends of the time series as the amplitude of the frequency waveform below the threshold is converted into zero to eliminate the noise and then merged into the original data (Fig. 2(e)). The padded $2m$ data added to prevent the formation of the diverging region are removed from the restored time series. Fig. 2(f) shows the smoothed time series graph obtained from the P-FTD process on the original time series data.

The noise-removal process in time series through the padding-based FT denoising is summarized as follows:

Step 1 (A → B): The padding area is created by adding randomly sampled values from the normal distributions $(0,\sigma_1)$ and $(0,\sigma_2)$ to the data at both ends and repeating this m times.

Step 2 (B → C): FFT is used to transform the padded time series data from the time domain to the frequency domain and split them into $N + 2m$ waveforms.

Step 3 (C → D): Among the decomposed waveforms, the amplitude values of the waveforms with a frequency value higher than the threshold (ε) are converted into 0 to remove noise.

Step 4 (D → E): The remaining waveforms are recombined with the data from the frequency domain to the time domain using the inverse FFT.

Step 5 (E → F): The padding areas containing the diverging part are removed to restore the denoised time series data with the same length as that of the original time series.

IV. EXPERIMENT

This section presents the details of the experiment used to verify the performance of the stock index prediction through P-FTD. Fig. 3 shows the overall research process to which the proposed model was applied. The specific data, preprocessing, models, hyperparameters, and evaluation metrics involved in the experiment are explained herein.

A. DATASET

The research datasets used in this study comprised representative stock market indices from the United States, China, and Korea. The data comprised of the daily prices and volume from January 2, 2001 to April 17, 2020. The input features comprised the daily open (stock price at the start of each trading day), high (highest price of each trading day), low (lowest price of each trading day), close (stock price at the end of each trading day), and volume (number of shares traded each day) of S&P500, SSE, and KOSPI. An output feature was set to the closing price of the next day. All data can be obtained from Yahoo Finance and investing.com.

Fig. 4 and Table 1 summarize how the extracted data were divided into three subsets (i.e., training, validation, and testing sets with proportions of 70% training, 10% validation, and 20% testing, respectively). The blue color in Fig. 4 indicates the training period, the yellow color indicates the validation period, and the red color indicates the testing period.

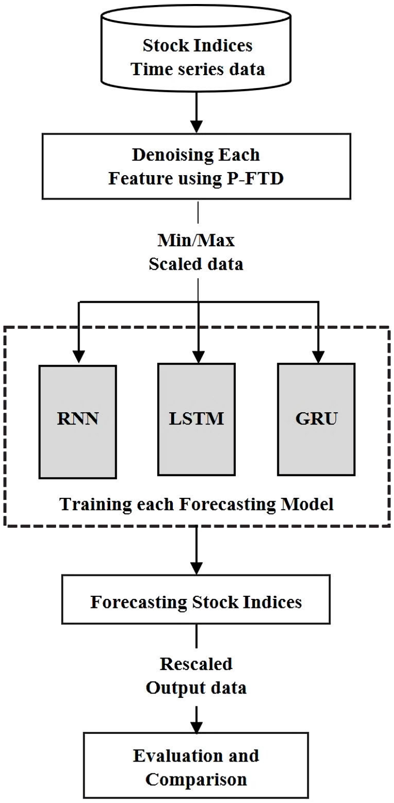


FIGURE 3. Experimental framework of the proposed model.

TABLE 1. Summary of the training and testing dataset after batch preprocessing.

Dataset	Training period (sample)	Validation period (sample)	Testing period (sample)
S&P500	2001/01/02- 2014/07/15 (3383)	2014/07/16- 2016/06/14 (483)	2016/06/15- 2020/04/17 (967)
SSE	2001/01/02- 2014/07/23 (3261)	2014/07/24- 2016/06/20 (465)	2016/06/21- 2020/04/17 (933)
KOSPI	2001/01/02- 2014/06/30 (3345)	2014/07/01- 2016/06/03 (475)	2016/06/07- 2020/04/17 (950)

B. PREPROCESSING

Min-max normalization, also known as min-max scaling, was performed on the input data before being feeding them into the model for more stable learning of the prediction model. Large values of the input overwhelm the weight adjustment during network training even if errors occur owing to other factors [50]. Moreover, input features can be dominated by a particular input feature when the magnitudes of each variable are different. This scaling aims to ensure that larger input features do not overwhelm smaller input features by rescaling

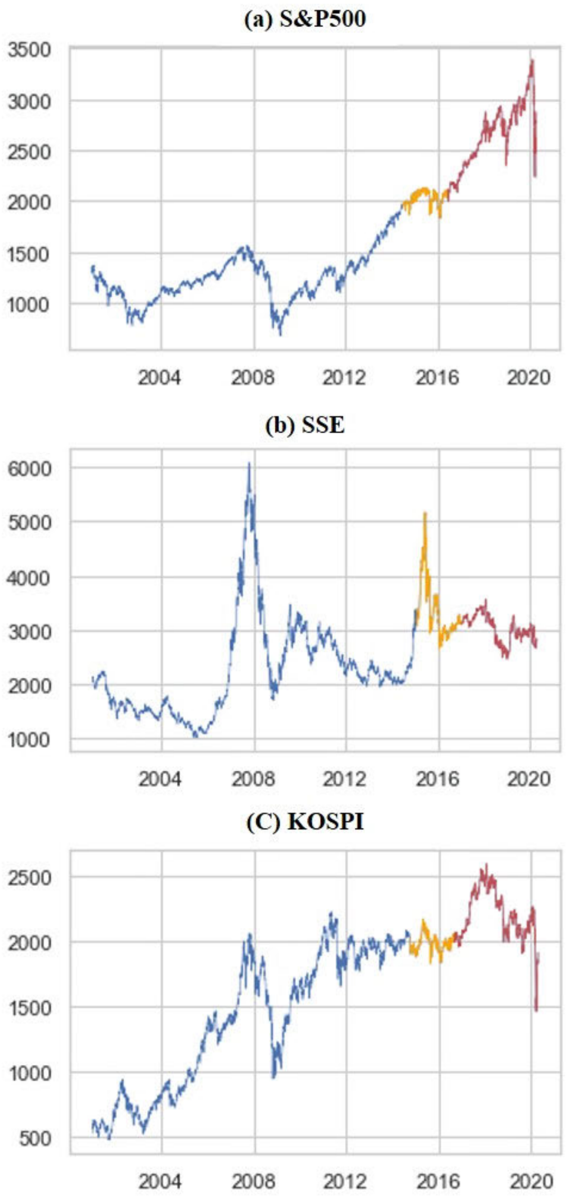


FIGURE 4. Daily closing prices from January 02, 2001 to April 17, 2020 for each stock market index.

the size of each variable between 0 and 1. The min–max normalization is calculated as follows:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{23}$$

where x is the original value and x_{max} and x_{min} are the maximum and minimum values of each feature, respectively.

C. MODELS AND HYPERPARAMETERS

In this study, we conducted experiments by integrating the P-FTD process into three different time series-based deep learning models: RNN, LSTM, and GRU. The P-FTD-integrated models (i.e., P-FTD_RNN, P-FTD_LSTM, and P-FTD_GRU) are categorized as Group 1, and the basic models (i.e., RNN, LSTM, and GRU) as Group 2.

TABLE 2. Hyperparameters of P-FTD.

Hyperparameter	Value	Search space
Volatility period(N)	40	10, 20, 30, 40
Padding area size(m)	40	10, 20, 30, 40
Threshold(ε)	0.2	0.1, 0.2, 0.3

TABLE 3. Hyperparameters of the time series-based deep learning models.

Hyperparameter	Value	Search space
Timestep	20	10, 20, 60, 120, 240
Learning rate	0.001	0.01, 0.001, 0.0001
Input dimension	5	-
Hidden dimension	10	10, 20, 30
Output dimension	1	-
Loss function	MSE	-
Optimizer	Adam	-
Early stopping	20	10, 20, 30

Hyperparameters of P-FTD (i.e., volatility period, padding area size, and threshold) and the predictive models (i.e., time step, learning rate, hidden dimensions, and early stop) were optimized using a grid search algorithm. The predictive models were generated using the hyperparameters obtained through this process and their performances were then evaluated by feeding the testing data. The value and search space of the hyperparameters are summarized in Tables 2 and 3.

Volatility period N, padding area size m, and threshold value ε for the noise removal were set to 40, 40, and 0.2, respectively. These values can be adjusted by the experimenter considering each P-FTD hyperparameter characteristic. As the volatility period N increases, a padding area reflecting the recent volatility trend of a longer period is obtained, whereas as N decreases, a padding area that is significantly affected by the recent volatility is obtained. The divergence region can be more stably removed as the padding area size m increases; however, a random sample has a greater effect on the original time series. The smaller the threshold value ε, the more numbers of high-frequency waveforms are removed in the P-FTD process and a smoother time series is restored.

Timestep, learning rate, input dimension, hidden dimension, and output dimension were set to 20 (trading days in a month), 0.001, 5 (open, high, low, close, volume), 10, and 1, respectively. The other parameters, i.e., the optimization algorithm and the loss function, were represented by the Adam optimizer and the mean squared error, respectively. Early stopping was employed to prevent the model from overfitting and terminate model learning when the loss of the validation data did not decrease by more than 20 epochs. These values of the hyperparameters were fixed in every model to validate the denoising effects of P-FTD on each model.

D. EVALUATION METRICS

The forecasting performance of each model was evaluated based on the following metrics: MAE, RMSE, MAPE, and hit ratio. Equations (24) to (28) define these metrics as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - f_i| \tag{24}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f_i)^2} \tag{25}$$

$$MAPE(\%) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - f_i}{y_i}\right| \times 100 \tag{26}$$

Here, y_i is the actual value, while f_i is the forecast value. Variable n represents the number of test samples. A lower value of these indicators denotes a smaller difference between the actual and forecasted values, showing better network performance.

The performance of predicting the next-day direction is evaluated using the hit ratio equation:

$$Hit\ ratio = \frac{1}{n}\sum_{i=1}^{n}D_i(i = 1, 2, \ldots, n) \tag{27}$$

where D_i is the directional match result for the i-th trading day.

$$D_i = \begin{cases} 1, & (y_{t+1} - y_t)(f_{t+1} - f_t) > 0, \\ 0, & Otherwise. \end{cases} \tag{28}$$

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we describe experiments, in which P-FTD was applied on different deep learning models, conducted to verify the denoising effects in stock market index prediction. Fig. 5 shows April 17, 2019 through April 17, 2020 S&P500 data, P-FTD-denoised data and noise data. The denoised data exhibited a smoothed and stable time series even in the volatile sections of the original data, where the trend was unstable. Furthermore, the index values at both ends of the time series stabilized without divergence when the data were denoised using padding techniques and restored to their time domain form.

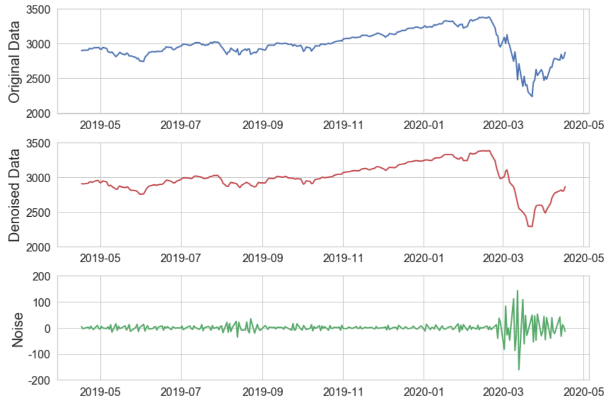


FIGURE 5. S&P500 P-FTD-denoised data and their noise from April 17, 2019 to April 17, 2020.

Figs. 6 to 8 show the actual and forecasted values of the S&P500, SSE, and KOSPI indices in the last two months from February 21, 2020 to April 17, 2020. The forecasting performances of the Group 2 models were compared with those of the Group 1 models, which are P-FTD-integrated models.

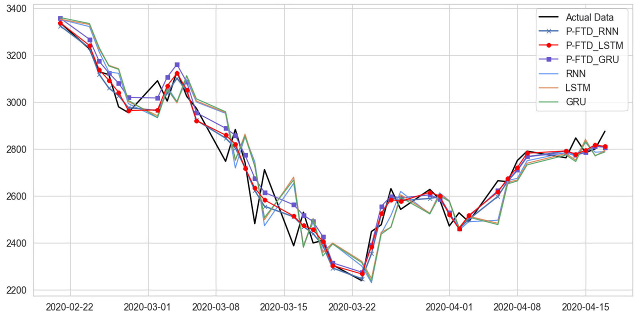


FIGURE 6. Comparison of the predicted values before and after applying P-FTD for each S&P500 model.

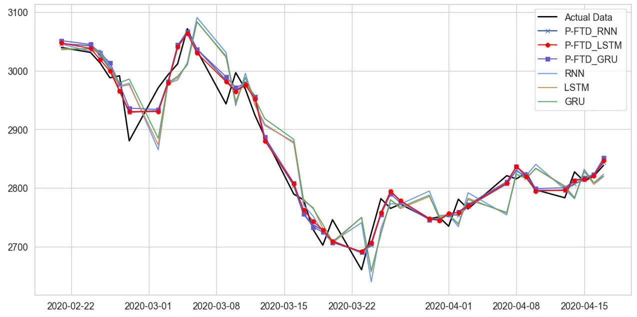


FIGURE 7. Comparison of the predicted values before and after applying P-FTD for each SSE model.

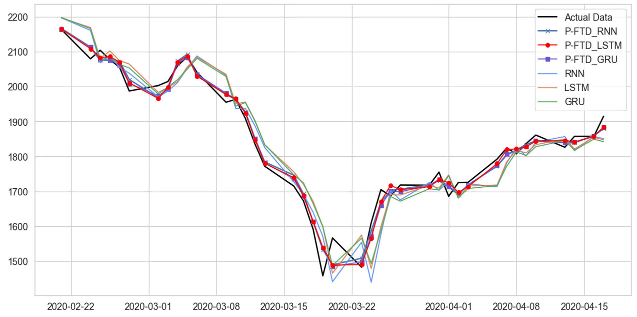


FIGURE 8. Comparison of the predicted values before and after applying P-FTD for each KOSPI model.

Table 4 presents the evaluation results for each model. Among the basic models belonging to Group 2, GRU showed the best performance in predicting S&P500 and SSE, and

TABLE 4. Summary of the forecasting evaluation results for each model.

	S&P500			SSE			KOSPI		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
P-FTD_RNN (Group 1)	12.922	19.577	0.469	10.031	13.684	0.338	9.915	12.857	0.449
P-FTD_LSTM (Group 1)	**9.252**	**15.855**	**0.344**	**9.667**	**13.091**	**0.325**	**7.795**	**10.669**	**0.362**
P-FTD_GRU (Group 1)	11.359	18.313	0.424	10.903	14.395	0.365	8.594	11.572	0.394
RNN (Group 2)	19.423	32.874	0.726	22.084	31.047	0.741	15.589	21.732	0.728
LSTM (Group 2)	18.157	32.508	0.694	22.055	31.158	0.739	**14.752**	**21.159**	**0.693**
GRU (Group 2)	**18.156**	**32.378**	**0.682**	**21.636**	**30.882**	**0.725**	15.482	21.519	0.725

TABLE 5. Comparison of RMSE according to time lag.

	S&P500			SSE			KOSPI		
	Lag 0	Lag 1	Lag 2	Lag 0	Lag 1	Lag 2	Lag 0	Lag 1	Lag 2
P-FTD_RNN (Group 1)	**19.577**	25.241	39.319	**13.684**	25.289	41.959	**12.857**	16.849	27.598
P-FTD_LSTM (Group 1)	**15.855**	23.355	37.128	**13.091**	25.343	41.238	**10.669**	12.986	23.990
P-FTD_GRU (Group 1)	**18.313**	23.812	34.548	**14.395**	22.591	37.322	**11.572**	16.426	29.209
RNN (Group 2)	32.874	**8.524**	31.288	31.047	**11.234**	34.124	21.732	**7.493**	22.792
LSTM (Group 2)	32.508	**10.130**	30.976	31.158	**6.922**	33.074	21.159	**4.042**	20.561
GRU (Group 2)	32.378	**8.248**	30.279	30.880	**7.822**	32.355	21.519	**6.559**	20.650

LSTM in predicting KOSPI. P-FTD_LSTM exhibited the best performance among the denoised models in Group 1. A comparison of the best performing models in each group illustrated that P-FTD_LSTM outperformed GRU. S&P500 had 49.04% (MAE), 51.03% (RMSE), and 49.56% (MAPE); SSE had 55.31% (MAE), 57.60% (RMSE), and 55.17% (MAPE); and KOSPI had 47.16% (MAE), 49.58% (RMSE), and 47.76% (MAPE).

Table 4 and Figs. 6 to 8 show a slight difference in the predicting performances within each group but a large difference in the performances between the groups.

In addition to the abovementioned results, two interesting aspects were also confirmed from the experimental results. First, the time lag improved. In Figs. 6 to 8, a time lag is observed in Group 2 when predicting the next day's index with the noised data. This result can be attributed to the characteristics of the predictive models that learn weights by minimizing errors and the fact that following the value of the previous day tends to give a lower prediction error more than actually predicting the value of the next day. By contrast, predictive models integrated with the P-FTD showed an improved time lag even at the transition points.

To confirm the time lag improvement, a comparison of the RSME results of each model for the three indexes according to the time delay was performed (Table 5). The actual index value was compared with the predicted value of the same day in Lag 0, one day after in Lag 1, and two days after in Lag 2. In the table, Group 1 models showed the lowest RMSE values in Lag 0, whereas Group 2 models showed the lowest RMSE values in Lag 1. The result indicates that P-FTD models, unlike basic models optimized to the value of the day before prediction and predict the following day, make predictions independent of the values of the previous day.

The second aspect is the formation of a robust model as a response to minor changes. Models without the P-FTD process respond significantly to minor changes in the original data and reduce the prediction performances. The P-FTD models responded more stably to small volatilities and made predictions based on the major trends of the financial time series data because the noises differing from the major trends were eliminated.

In addition to the next-day stock index prediction, the coincidence of the directionality of the stock index was further verified using the hit ratio, the trials percentage when the predicted direction is correct. The average hit ratios for Groups 1 and 2 were 72.15% and 49.43%, respectively, and the best hit ratios were 73.58% and 50.70%, respectively. The average hit ratio for Group 1 increased by 45.96% compared with that of Group 2. Fig. 9 also shows that all models in Group 1 were more efficient in predicting the direction of the daily closing price of all stock indices than the models in Group 2. The accurate prediction performance of the models with the

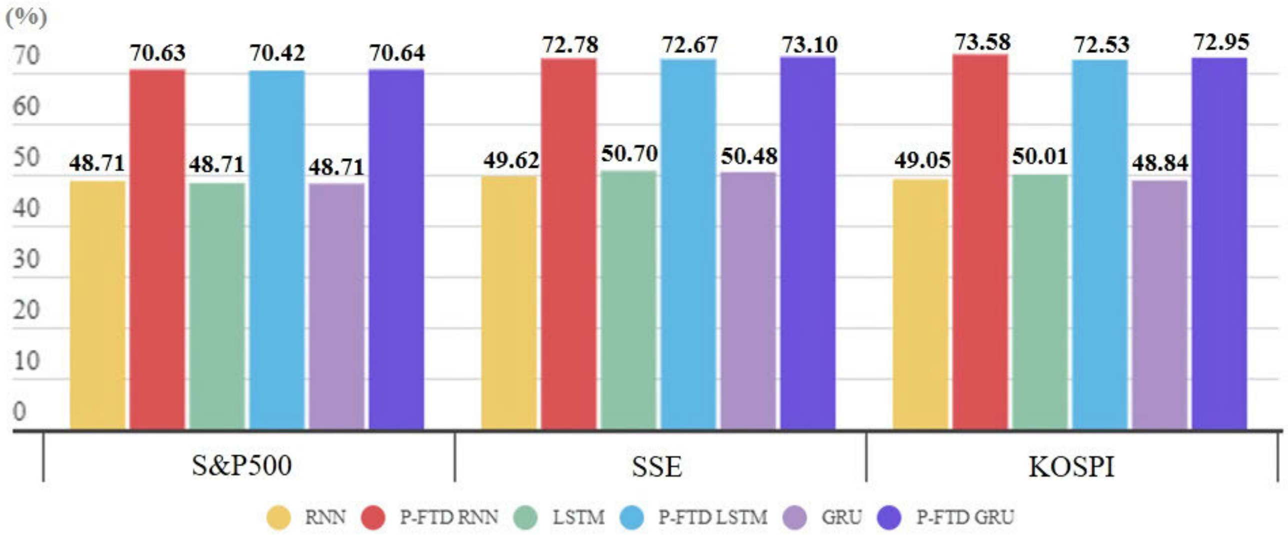


FIGURE 9. Comparison of the hit ratios before and after denoising for each model.

TABLE 6. Performance comparison of previous research works.

Index data	Denoising model	Prediction model	MAPE(%)	Ref.
S&P500	WT	LSTM	1.5	[37]
	WSAEs	LSTM	1.1	
	EWT	LSTM	0.7583	[35]
	EWT	ELM	0.6830	
	Proposed models		**0.344 - 0.469**	-
SSE	RTWD	LSTM	1.3824	[36]
	EWT	LSTM	1.2860	[35]
	EWT	ELM	1.4086	
	Proposed models		**0.325 - 0.338**	-
KOSPI	RTWD	LSTM	0.6346	[36]
	WT	NN	2.805	[34]
	WPCA	NN	2.243	
	Proposed models		**0.362 - 0.449**	-

P-FTD-denoised data is beneficial to investors and can be a favorable candidate for predicting the direction of next day's closing price.

To further illustrate the denoising effects of P-FTD on the time-based deep learning models, the performance of the proposed models was compared with that of the aforementioned studies (Table 6). These studies present different denoising methods combined with deep learning models to predict the next day's stock index using S&P500, SSE, and KOSPI data. As the evaluation metric to compare the prediction performances of the different studies, the error ratio between predicted and actual value MAPE was selected because other evaluation metrics mentioned in Section IV.D are significantly affected by the period and scale of data observed. The table reveals that time series based deep learning models integrated with P-FTD outperform other deep learning models combined with different denoising methods. The prediction performance of P-FTD_LSTM was higher by 75.3% than RTWD-LSTM, which was the best-performing model among those in the literature for predicting the KOSPI index.

VI. CONCLUSION

In this study, we proposed a denoising method based on FFT with a padding technique to remove noises, which result in unstable learning of predictive models and time lag, in the stock market index. The proposed method also presents a solution to the divergence problem of the time series data when transformed from the frequency domain to the time domain using FFT. The divergence, which occurs at both ends of the time series data owing to the original information being removed along with the noise waveforms, was prevented by first adding the padding areas containing the diverging components and then removing them from the restored time series. The performance of the proposed denoising technique was verified through experiments that predicted the next day's stock index by applying it to the major indices of different countries and diverse time series-based deep learning models.

The daily price and the trading volume data of S&P500, SSE, and KOSPI, which are the representative indices of the United States, China, and Korea, respectively, were used in this experiment. As predictive models, the RNN, LSTM, and GRU deep learning models were used, compared, and analyzed to investigate the noise-reduction performance through P-FTD. According to the results, GRU and LSTM showed the best performance among the basic models, whereas P-FTD_LSTM showed the best performance among the denoised models. Comparing the best-performing models, the MAE, RMSE, and MAPE values of P-FTD_LSTM decreased for all indexes. The performances of the basic and denoised models for predicting the directionality of the stock index were also compared using the hit ratio. The average hit ratio for the denoised models was greater than that for the basic models by 45.96%. These results demonstrate that the prediction after removing the noise from the input data using the P-FTD shows better performance in all indices and models compared with that performed without noise removal. The results also show an improvement in the time lag, in which the predicted values follow preceding values when predicted using noised data, and a robust model that responds more stably to noises. In addition, the performance of the proposed models with that of other previous studies were compared and verified their superiority.

Although only the financial time series data were treated in this study, the proposed denoising technique could be used to filter other types of time series data. Moreover, it has strength in that users can adjust the range of noise values.

This study has a limitation. The padding technique was used to prevent only the divergence that occurs when restoring the original data after removing the waveforms with low amplitude and high frequency. The padding values were randomly sampled from a normal distribution considering the volatility of the original data. The number of padded values that is smaller than that of the original data has minor effects on the restored data after the P-FTD processing. However, if the number of padded values increases compared with the number of original data, the restored data can be affected by another type of noise generated by the padded values. Therefore, additional studies on methods of finding padding values that minimize the impact when the original data are small are needed. In addition, the correlation among individual parameters and the denoising performance in the P-FTD process must be investigated.

REFERENCES

[1] E. F. Fama, "Efficient capital markets: II," *J. Finance*, vol. 46, no. 5, pp. 1575–1617, Dec. 1991.

[2] P. R. Junior, F. L. R. Salomon, and E. E. de Oliveira Pamplona, "ARIMA: An applied time series forecasting model for the Bovespa stock index," *Appl. Math.*, vol. 5, no. 21, p. 3383, 2014.

[3] S. M. Idrees, M. A. Alam, and P. Agarwal, "A prediction approach for stock market volatility based on time series data," *IEEE Access*, vol. 7, pp. 17287–17298, 2019.

[4] B. Krollner, B. J. Vanstone, and G. R. Finnie, "Financial time series forecasting with machine learning techniques: A survey," in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Mach. Learn.*, Apr. 2010, pp. 1–8.

[5] X. Zhang, S. Qu, J. Huang, B. Fang, and P. Yu, "Stock market prediction via multi-source multiple instance learning," *IEEE Access*, vol. 6, pp. 50720–50728, 2018.

[6] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Syst. Appl.*, vol. 83, pp. 187–205, Oct. 2017.

[7] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150199–150212, 2020.

[8] D. Banerjee, "Forecasting of indian stock market using time-series ARIMA model," in *Proc. 2nd Int. Conf. Bus. Inf. Manage. (ICBIM)*, Jan. 2014, pp. 131–135.

[9] H.-C. Liu and J.-C. Hung, "Forecasting S&P-100 stock index volatility: The role of volatility asymmetry and distributional assumption in GARCH models," *Expert Syst. Appl.*, vol. 37, no. 7, pp. 4928–4934, Jul. 2010.

[10] S. Pyo, J. Lee, M. Cha, and H. Jang, "Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the Korean stock markets," *PLoS ONE*, vol. 12, no. 11, Nov. 2017, Art. no. e0188107.

[11] A. Dingli and K. S. Fournier, "Financial time series forecasting—A deep learning approach," *Int. J. Mach. Learn. Comput.*, vol. 7, no. 5, pp. 118–122, Oct. 2017.

[12] V. K. S. Reddy, "Stock market prediction using machine learning," *Int. Res. J. Eng. Technol.*, vol. 4, no. 10, pp. 1033–1035, Oct. 2018.

[13] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1419–1426.

[14] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018.

[15] Y. Baek and H. Y. Kim, "ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Syst. Appl.*, vol. 113, pp. 457–480, Dec. 2018.

[16] M. Nikou, G. Mansourfar, and J. Bagherzadeh, "Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms," *Intell. Syst. Accounting, Finance Manage.*, vol. 26, no. 4, pp. 164–174, Oct. 2019.

[17] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. Salwana, "Deep learning for stock market prediction," *Entropy*, vol. 22, no. 8, p. 840, Jul. 2020.

[18] S. Mehtab and J. Sen, "A time series analysis-based stock price prediction using machine learning and deep learning models," 2020, *arXiv:2004.11697*. [Online]. Available: http://arxiv.org/abs/2004.11697

[19] M. Qiu and Y. Song, "Predicting the direction of stock market index movement using an optimized artificial neural network model," *PLoS ONE*, vol. 11, no. 5, May 2016, Art. no. e0155133.

[20] D. L. Minh, A. Sadeghi-Niaraki, H. D. Huy, K. Min, and H. Moon, "Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network," *IEEE Access*, vol. 6, pp. 55392–55404, 2018.

[21] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PLoS ONE*, vol. 14, no. 2, Feb. 2019, Art. no. e0212320.

[22] M. Wen, P. Li, L. Zhang, and Y. Chen, "Stock market trend prediction using high-order information of time series," *IEEE Access*, vol. 7, pp. 28299–28308, 2019.

[23] S. Mehtab, J. Sen, and A. Dutta, "Stock price prediction using machine learning and LSTM-based deep learning models," 2020, *arXiv:2009.10819*. [Online]. Available: http://arxiv.org/abs/2009.10819

[24] H. Liu and Z. Long, "An improved deep learning model for predicting stock market price time series," *Digit. Signal Process.*, vol. 102, Jul. 2020, Art. no. 102741.

[25] Y. Zhang, B. Yan, and M. Aasma, "A novel deep learning framework: Prediction and analysis of financial time series using CEEMD and LSTM," *Expert Syst. Appl.*, vol. 159, Nov. 2020, Art. no. 113609.

[26] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decis. Support Syst.*, vol. 47, no. 2, pp. 115–125, May 2009.

[27] H. Hassani, A. Dionisio, and M. Ghodsi, "The effect of noise reduction in measuring the linear and nonlinear dependency of financial markets," *Nonlinear Anal., Real World Appl.*, vol. 11, no. 1, pp. 492–502, Feb. 2010.

[28] J.-H. Wang, J.-H. Jiang, and R.-Q. Yu, "Robust back propagation algorithm as a chemometric tool to prevent the overfitting to outliers," *Chemometric Intell. Lab. Syst.*, vol. 34, no. 1, pp. 109–115, Aug. 1996.

[29] W. Zhao, D. Chen, and S. Hu, "Detection of outlier and a robust BP algorithm against outlier," *Comput. Chem. Eng.*, vol. 28, no. 8, pp. 1403–1408, Jul. 2004.

[30] A. Raudys, V. Lenčiauskas, and E. Malčius, "Moving averages for financial data smoothing," in *Proc. Int. Conf. Inf. Softw. Technol.*, Oct. 2013, pp. 34–45.

[31] C. N. Babu and B. E. Reddy, "A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data," *Appl. Soft Comput.*, vol. 23, pp. 27–38, Oct. 2014.

[32] C.-J. Lu, "Integrating independent component analysis-based denoising scheme with neural network for stock price prediction," *Expert Syst. Appl.*, vol. 37, no. 10, pp. 7056–7064, Oct. 2010.

[33] A. M. Awajan, M. T. Ismail, and S. AL Wadi, "Improving forecasting accuracy for stock market data using EMD-HW bagging," *PLoS ONE*, vol. 13, no. 7, Jul. 2018, Art. no. e0199582.

[34] J. C. P. M'ng and M. Mehralizadeh, "Forecasting east asian indices futures via a novel hybrid of wavelet-PCA denoising and artificial neural network models," *PLoS ONE*, vol. 11, no. 6, Jun. 2016, Art. no. e0156338.

[35] H. Yu, L. J. Ming, R. Sumei, and Z. Shuping, "A hybrid model for financial time series forecasting—Integration of EWT, ARIMA with the improved ABC optimized ELM," *IEEE Access*, vol. 8, pp. 84501–84518, 2020.

[36] Z. Li and V. Tam, "Combining the real-time wavelet denoising and long-short-term-memory neural network for predicting stock indexes," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–8.

[37] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0180944.

[38] M. Srivastava, C. L. Anderson, and J. H. Freed, "A new wavelet denoising method for selecting decomposition levels and noise thresholds," *IEEE Access*, vol. 4, pp. 3862–3877, 2016.

[39] L. Chiron, M. A. van Agthoven, B. Kieffer, C. Rolando, and M.-A. Delsuc, "Efficient denoising algorithms for large experimental datasets and their applications in Fourier transform ion cyclotron resonance mass spectrometry," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 4, pp. 1385–1390, Jan. 2014.

[40] J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng, "Fast non-local algorithm for image denoising," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 1429–1432.

[41] A. Mustafi and S. K. Ghorai, "A novel blind source separation technique using fractional Fourier transform for denoising medical images," *Optik*, vol. 124, no. 3, pp. 265–271, Feb. 2013.

[42] M.-Y. Chen and B.-T. Chen, "Online fuzzy time series analysis based on entropy discretization and a fast Fourier transform," *Appl. Soft Comput.*, vol. 14, pp. 156–166, Jan. 2014.

[43] R. N. Bracewell, *The Fourier Transform and Its Applications*. New York, NY, USA: McGraw-Hill, 1986.

[44] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 9, pp. 297–301, Apr. 1965.

[45] A. M. Logar, E. M. Corwin, and W. J. B. Oldham, "A comparison of recurrent neural network learning algorithms," in *Proc. IEEE Int. Conf. Neural Netw.*, Mar. 1993, pp. 1129–1134.

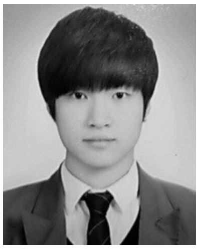
[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[47] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: http://arxiv.org/abs/1409.1259

[48] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[49] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2342–2350.

[50] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2012.



DONGHWAN SONG received the B.S. degree from the Department of Mechanical and Advanced Materials Engineering, Ulsan National Institute of Science and Technology, Ulsan, South Korea, in 2015, where he is currently pursuing the Ph.D. degree with the Department of System Design and Control Engineering. He has been involved in several research projects, such as SVM-based automatic product quality inspection system using thermal image data and modeling of financial behavior of agent using deep learning. His current research interests include time series analysis based on machine learning and deep learning in the field of finance and economic.

ADRIAN MATIAS CHUNG BAEK received the M.S. degree in mechanical engineering from the Ulsan National Institute of Science and Technology, South Korea, in 2020, where he is currently pursuing the Ph.D. degree with the Department of Mechanical Engineering. His research interests include artificial intelligence, in particular, deep learning and reinforcement learning and their application in 3D printing.

NAMHUN KIM received the B.S. and M.S. degrees from KAIST, in 1998 and 2000, respectively, and the Ph.D. degree in industrial and manufacturing engineering from Penn State University, University Park, PA, USA, in 2010. He worked as a Senior Researcher with Samsung Corning Company Ltd., for five years. He worked as a Research Associate with Penn State University, from January to June 2010. He joined with UNIST, South Korea, in 2010, where he is currently working as an Associate Professor with the Department of System Design and Control Engineering, acting as the Director of the 3D Printing Research Center. His research interests include manufacturing technologies with emphasis on additive manufacturing 3D printing, manufacturing system modeling, and agent-based simulation.