Hindawi
Complexity
Volume 2021, Article ID 6647534, 16 pages
https://doi.org/10.1155/2021/6647534

Research Article

Forecasting Foreign Exchange Volatility Using Deep Learning Autoencoder-LSTM Techniques

Gunho Jung[1] and Sun-Yong Choi[2]

[1]*Department of Artificial Intelligence, Korea University, Seoul 02841, Republic of Korea*
[2]*Department of Financial Mathematics, Gachon University, Seongnam-si, Gyeoggi 13120, Republic of Korea*

Correspondence should be addressed to Sun-Yong Choi; sunyongchoi@gachon.ac.kr

Received 28 December 2020; Revised 30 January 2021; Accepted 20 March 2021; Published 31 March 2021

Academic Editor: Benjamin Miranda Tabak

Since the breakdown of the Bretton Woods system in the early 1970s, the foreign exchange (FX) market has become an important focus of both academic and practical research. There are many reasons why FX is important, but one of most important aspects is the determination of foreign investment values. Therefore, FX serves as the backbone of international investments and global trading. Additionally, because fluctuations in FX affect the value of imported and exported goods and services, such fluctuations have an important impact on the economic competitiveness of multinational corporations and countries. Therefore, the volatility of FX rates is a major concern for scholars and practitioners. Forecasting FX volatility is a crucial financial problem that is attracting significant attention based on its diverse implications. Recently, various deep learning models based on artificial neural networks (ANNs) have been widely employed in finance and economics, particularly for forecasting volatility. The main goal of this study was to predict FX volatility effectively using ANN models. To this end, we propose a hybrid model that combines the long short-term memory (LSTM) and autoencoder models. These deep learning models are known to perform well in time-series prediction for forecasting FX volatility. Therefore, we expect that our approach will be suitable for FX volatility prediction because it combines the merits of these two models. Methodologically, we employ the Foreign Exchange Volatility Index (FXVIX) as a measure of FX volatility. In particular, the three major FXVIX indices (EUVIX, BPVIX, and JYVIX) from 2010 to 2019 are considered, and we predict future prices using the proposed hybrid model. Our hybrid model utilizes an LSTM model as an encoder and decoder inside an autoencoder network. Additionally, we investigate FXVIX indices through subperiod analysis to examine how the proposed model's forecasting performance is influenced by data distributions and outliers. Based on the empirical results, we can conclude that the proposed hybrid method, which we call the autoencoder-LSTM model, outperforms the traditional LSTM method. Additionally, the ability to learn the magnitude of data spread and singularities determines the accuracy of predictions made using deep learning models. In summary, this study established that FX volatility can be accurately predicted using a combination of deep learning models. Our findings have important implications for practitioners. Because forecasting volatility is an essential task for financial decision-making, this study will enable traders and policymakers to hedge or invest efficiently and make policy decisions based on volatility forecasting.

1. Introduction

Among various financial asset markets, the foreign exchange (FX) market has become increasingly volatile and fluid over the past decade. According to data released by BIS (Bank for International Settlements) in April of 2019, the global trading volume of FX commodity markets was $6.6 trillion per day, representing a 30% increase compared to April of 2016 ($5.1 trillion). With the advent of globalization and increased demand for overseas investment, the number of FX transactions has increased rapidly based on investments in companies in various countries. Additionally, FX rates significantly affect the estimation of currency risks and profits for international trades. Governments and policymakers are keeping a close watch on FX fluctuations to perform risk management. Therefore, FX is considered to be the most important financial index for international monetary markets (Huang et al. [1]).

In addition to FX rates, FX volatility has also been a significant source of concern for practitioners. FX volatility is defined by fluctuations in FX rates, so it is also known as a measure of FX risk. Because FX risk is directly linked to transaction costs related to international trade, it is of great importance for multinational firms, financial institutions, and traders who wish to hedge currency risks. In this regard, FX volatility has affected the external sector competitiveness of international trade and the global economy.

In particular, financial asset price volatility is a crucial concern for scholars, investors, and policymakers. This is because volatility is important for derivative pricing, hedging, portfolio selection, and risk management (see Vasilellis and Meade [2], Knopf et al. [3], Brownlees and Gallo [4], Gallo and Otranto [5], and Bollerslev et al. [6]). Therefore, the forecasting and modeling of volatility have recently become the focus of many empirical studies and theoretical investigations in academia. Forecasting volatility accurately remains a crucial challenge for scholars.

Because many academics and practitioners are interested in volatility, many studies on volatility prediction have been reported. In these studies, many approaches have been utilized for forecasting. The autoregressive conditional heteroscedasticity (ARCH) and generalized ARCH (GARCH) models proposed by Bollerslev [7] are mainly used to predict volatility (Vee et al. [8], Dhamija and Bhalla [9], Bala and Asemota [10], Kambouroudis et al. [11], and Köchling et al. [12]). Various characteristics of volatility, such as leverage effects, volatility clustering, and persistence (Cont [13] and Cont [14]), are the main reasons for employing GARCH-based models. Based on the recent development of artificial neural network (ANN) models, the use of ANN methods for forecasting volatility has increased (Pradeepkumar and Ravi [15], Liu [16], Ramos-Pérez et al. [17], and Bucci [18]). Previous studies have employed various ANN models, such as the random forest (RF) (Breiman [19]), support vector machine (SVM) (Cortes and Vapnik [20]), and long short-term memory (LSTM) (Hochreiter and Schmidhuber [21]). Several studies have shown that ANN methods outperform GARCH-based models for forecasting time series (see Pradeepkumar and Ravi [15], Liu [16], and Bucci [18]). Additionally, hybrid models based on ANNs and GARCH-type models have been introduced (Hajizadeh et al. [22], Kristjanpoller et al. [23], Kristjanpoller and Minutolo [24], Kim and Won [25], Baffour et al. [26], and Hu et al. [27]). Such models are reported to have advantages compared to using ANNs or GARCH-based models alone. Additional literature on this topic will be covered in Section 2.

Based on the discussion above, we focus on volatility forecasting based on FX volatility. As measures of FX volatility, we adopt three FX volatility indexes (FXVIXs), namely, the FX euro volatility index (EUVIX), FX British pound volatility index (BPVIX), and FX yen volatility index (JYVIX), which are equally weighted indices of the Chicago Board Option Exchange's (CBOE's) 30 day implied volatility readings for the euro (EUR), pound sterling (GBP), and Japanese yen (JPY), respectively. Because the three currency pairs of EUR/USD, USD/JPY, and GBP/USD are the three most heavily traded currency pairs on the FX market, we selected the three corresponding FXVIX indices. Additionally, these indexes reflect global economic trends (see Ishfaq et al. [28], Dicle and Dicle [29], and Pilbeam [30]). As mentioned previously, the forecasting of volatility in the FX market is important for global firms, financial institutions, and traders who wish to hedge currency risks (see Guo et al. [31], Abdalla [32], and Menkhoff et al. [33]).

Practically, the FX market consists of three associated components: spot transactions, forward transactions, and derivative contracts (Baffour et al. [26]). Additionally, because FX was originally defined by two currencies, FX has more observable factors that affect changes compared to other financial indices. Furthermore, according to Liu et al. [34], the periodic characteristics of the FX market are some of the main reasons why it is difficult to predict changes in the FX market. Therefore, we utilize ANN models as data-driven methods, rather than model-driven methods such as GARCH-type models, to forecast the three aforementioned FXVIXs. In particular, we employ the LSTM and autoencoder (Rumelhart et al. [35]) models as ANN techniques. We propose a hybrid neural network model based on these two models. To combine an autoencoder with LSTM, we apply LSTM as an encoder and decoder for sequence data inside an autoencoder network. Therefore, the proposed hybrid model can leverage the advantages of both the autoencoder and LSTM. A detailed discussion of this topic is presented in Section 3.

Methodologically, we adopt a machine learning algorithm (LSTM) to implement an autoencoder-LSTM model for forecasting FXVIXs from 2010 to 2019. We optimize the adopted algorithms using a grid search procedure provided by Full-Stack Python. Testing is also performed using subperiod analysis to investigate whether data deviations and outliers affect model training. Such subperiod analysis has been commonly implemented in previous studies (Sharma et al. [36], García and Kristjanpoller [37], Ramos-Pérez et al. [17], and Choi and Hong [38]). Specifically, we split the entire sample period into three subperiods called Period 1 (January, 2010 to December, 2015), Period 2 (January, 2016 to December, 2016), and Period 3 (January, 2017 to December, 2019). Period 2 exhibits uncertainty in the European market based on the Brexit movement. In this manner, we investigate the accuracy of prediction and model performance according to different data states.

There are two major aspects of this study that differ from previous studies. First, we use FXVIXs, which play key roles in the FX market. Although previous empirical studies have predicted various types of financial asset price volatility using various models, research on forecasting FXVIXs is scarce. Additionally, research on FX price prediction and volatility prediction using various approaches is being conducted, but research on the prediction of the FXVIX is relatively rare. Therefore, it is necessary to predict FXVIX volatility. Second, we propose a hybrid model based on an autoencoder and LSTM to forecast the three FXVIXs. LSTM is known to be good at forecasting time series (Fischer and

Krauss [39], Kumar et al. [40], and Muzaffar and Afshari [41]), and one of the advantages of an autoencoder is that it can automatically extract features from input data (Phaisangittisagul and Chongprachawat [42], Zhang et al. [43], and Zeng et al. [44]). Therefore, the autoencoder technique has been widely used to predict time series data (Saha et al. [45], Lv et al. [46], Sagheer and Kotb [47], and Boquet et al. [48]). The proposed hybrid model has excellent potential as a novel method for forecasting the FXVIX and time series.

The main contributions of this paper can be summarized as follows:

> First, we expand upon previous studies by forecasting the FXVIX using ANN models. Our experiments were motivated by the observation that previous studies on the FX market have mainly focused on the FX rate, volatility of returns, or historical volatility. In particular, FXVIXs represent future FX risk measures for market participants. Therefore, our findings have important implications for practitioners managing FX risk exposure.
>
> Second, we propose a hybrid ANN model based on an autoencoder and LSTM. Forecasting performance results demonstrate that the proposed hybrid model outperforms traditional LSTM models. Consequently, this study contributes to the literature on developing ANN models by introducing a novel hybrid model.
>
> Our third major contribution is the optimization of model forecasting performance through subperiod analysis. Based on the empirical results of subperiod analysis, we can conclude that a wide distribution of input data and acceptable number of outliers improve forecasting performance.

The remainder of this paper is organized as follows. Section 2 presents a brief literature review on FX volatility and studies using machine learning in finance. Section 3 describes the data and methodology adopted in this study. Section 4 presents the results of empirical analysis for the full sample period and subperiod analysis. Finally, we provide concluding remarks in Section 5.

2. Literature Review

There is a vast body of literature on forecasting financial time series. In this section, we divide previous research into FX rate and FX volatility research according to the main focus of previous papers. Additionally, we also discuss literature on time-series forecasting using ANNs.

First, because the FX rate directly affects the income of multinational firms, many studies have focused on the forecasting FX rate and many studies have used ANN models to predict future FX rates. For example, Liu et al. [34] predicted EUR/USD, GBP/USD, and JPY/USD rates using a model based on a convolutional neural network (CNN). They demonstrated that such a model is suitable for processing 2D structural exchange rate data. Fu et al. [49] developed evolutionary support vector regression (SVR) models to forecast four Renminbi (RMB, Chinese yuan) exchange rates (CNY against USD, EUR, JPY, and GBP). They also demonstrated that the proposed model outperforms the multilayer perceptron (MLP) neural network, Elman neural network, and SVR models in terms of level forecasting accuracy measures. The authors of Sun et al. [50] introduced a novel ensemble deep learning approach based on LSTM and a bagging ensemble learning strategy to predict four major currencies (EUR/USD, GBP/USD, JPY/USD, and USD/CNY). According to their empirical results, their proposed model provided significantly improved forecasting accuracy compared to a traditional LSTM model.

As discussed in the previous section, FX volatility is also important for many academics and practitioners, so many studies have focused on FX volatility forecasting. In general, GARCH-based models have been used in many studies to predict FX volatility. Additionally, some studies have predicted FX volatility by incorporating different methodologies into GARCH models to improve forecasting power. For example, the authors of Vilasuso [51] predicted various FX rate volatilities (Canadian dollar, French franc, German mark, Italian lira, Japanese yen, and British pound) using a fractionally integrated GARCH (FIGARCH) model (Baillie et al. [52]). The empirical results of their study demonstrated that the FIGARCH model is better at capturing the features of FX volatility compared to the original GARCH model. The authors of Rapach and Strauss [53] demonstrated that structural breaks in the unconditional variance of FX rate returns can improve the forecasting performance of GARCH(1,1) models for FX volatility by incorporating the daily returns of the US dollar against the currencies of Canada, Denmark, Germany, Japan, Norway, Switzerland, and the UK. Pilbeam and Langeland [54] investigated whether various GARCH-based models can effectively forecast the FX volatility of the four currency pairs of the euro, pound, Swiss franc, and yen against the US dollar. In particular, their empirical results demonstrated that GARCH models perform better in periods of low volatility compared to periods of high volatility. You and Liu [55] employed the GARCH-MIDAS approach (Engle et al. [56]) to forecast the short-run volatility of six FX rates based on monetary fundamentals. They demonstrated that the forecasting power of daily FX volatility is significantly improved by including monthly monetary fundamental volatilities.

Various machine learning models have also been used to forecast time series originating from various fields, including engineering and finance. In finance, many studies have used machine learning to predict future stock prices. For example, Trafalis and Ince [57] compared SVR with backpropagation to a radial basis function network on the task of forecasting daily stock prices. Similarly, Henrique et al. [58] utilized SVR and a random walk (RW) method to predict daily stock prices in three different markets (Brazilian, American, and Chinese). Based on comparisons of the price prediction results of the SVR and RW models, they determined that SVR models may perform better than RW models in terms of predictive performance. Recently, various studies using machine learning methods and deep learning methodologies have been reported. For example, the authors of Selvin et al. [59] employed deep learning models, namely, a recurrent neural network (RNN), LSTM, and CNN to predict minute-wise stock prices. They determined that the CNN algorithm provided the best performance. Chong et al. [60] employed an autoencoder to extract features from stock data and constructed a deep neural network (DNN) to predict future stock returns. They determined that it is possible to extract features from a large set of raw data without relying on prior knowledge regarding predictors, which is one of the main advantages of DNNs. Pradeepkumar and Ravi [15] proposed a particle swarm optimization-trained quantile RNN to forecast FX volatility. Their model provides superior forecasting performance compared to the GARCH model. In [16] and [18], various ANN models were employed to predict the volatility of the S&P 500 stock index. According to the findings of these studies, ANN models are able to outperform traditional econometric methods, including GARCH and autoregressive moving average models. In particular, LSTM models seem to improve the accuracy of volatility forecasts. Additionally, Ramos-Pérez et al. [17] predicted S&P 500 index volatility using a stacked ANN model based on a set of various machine learning techniques, including gradient descent boosting, RF, and SVM. They demonstrated that volatility forecasts can be improved by stacking machine learning algorithms. Additionally, regardless of the volatility model adopted, high-volatility regimes lead to higher error rates.

Several studies have proposed hybrid models based on GARCH-based models and ANN models. For example, various GARCH-based models have been combined with ANNs based on MLPs and many hybrid models have been used to enhance the ability of GARCH models to forecast the volatility of stocks, gold, and FX rate returns (Hajizadeh et al. [22], Kristjanpoller et al. [23], Kristjanpoller and Minutolo [24], and Baffour et al. [26]). Additionally, some studies have proposed hybrids of LSTM and GARCH models and have used such models to predict the volatility of financial assets (Kim and Won [25] and Hu et al. [27]). According to empirical results, hybrid models based on GARCH and ANN techniques exhibit improved forecasting performance in terms of volatility accuracy.

In particular, we focus on studies using LSTM and autoencoder approaches for forecasting time series. LSTM, which was introduced by Hochreiter and Schmidhuber [21], has been widely used to forecast time series in many prediction studies. This method is mainly used to analyze time-series data because it can keep records of past data. Some studies have compared LSTM to traditional methods using neural networks or investigated such models by reconstructing both types of methods. As discussed by Siami-Namini et al. [61] and Ohanyan [62], as computing power improves, implementing deep learning models becomes more practical, and their performance exceeds that of traditional models. Additionally, Deorukhkar et al. [63] demonstrated that neural network models combined with autoregressive integrated moving average or LSTM models provide greater accuracy than either type of model individually. In [64], the method of applying preprocessed stock prices to an LSTM model using a wavelet transform was shown to be superior to traditional methods.

The autoencoder presented in [35] aims to generate a representation as close to an original input as possible from reduced encoding results. This method is a transformation of the basic model using stacked layers, denoising, and sparse representation and is used for financial time series prediction. Bao et al. [65] used LSTM and stacked autoencoders to forecast stock prices and demonstrated that this type of hybrid model is more powerful than an RNN or LSTM model alone. In [66], a stacked denoising autoencoder applied to gravitational searching was effective at predicting the direction of stock index movement, which is affected by underlying assets. Additionally, Sun et al. [67] explained that a stacked denoising autoencoder formed through the selection of training sets based on a K-nearest neighbors approach can improve the accuracy compared to traditional methods.

This study enhances the existing literature in two main aspects. We first propose a hybrid model that combines LSTM and an autoencoder to forecast FX volatility. There are other studies that have used hybrid models, but they have used models other than autoencoders and LSTM. Additionally, most studies have developed hybrid models based on GARCH models. However, as discussed above, LSTM and autoencoders perform well at time-series prediction, so we adopted these two types of models to forecast FX volatility. Second, as discussed in Section 1, FX volatility has great significance, but there is a significant lack of research on forecasting its changes. We contribute to the finance literature by forecasting FXVIXs using the proposed hybrid model.

3. Data Description and Methodologies

3.1. Data Description. The VIX was firstly implemented on the CBOE in 1993. This index is based on the real-time prices of options in the S&P 500 index. Because it is derived from the price inputs of S&P 500 index options, this index not only represents market expectations regarding 30 day forward-looking volatility but also provides a measure of market risk and investor sentiments. Subsequently, various VIXs with different basic assets were developed.

In this study, we investigated whether machine learning methods are suitable for forecasting FX volatility time-series data. Our data samples come from the CBOE. The CBOE is one of the world's largest exchange holding companies, and it provides several derivatives related to implied VIXs. We adopted three currency-related volatility indices, namely, the BPVIX, JYVIX, and EUVIX. Similar to a VIX, FX volatility is calculated using a formula that averages the weighted prices of out-of-the-money puts and calls.

We collected 2520 daily time series FXVIX data from January of 2010 to December of 2019. Based on fluctuations caused by the Brexit movement, the data were divided into subsets from 2010 to 2015, 2016, and 2017 to 2019 based on instabilities in 2016. The first period represents the period of recovery following the subprime mortgage crisis and contains the most data (1514 daily data). As shown in Figure 1, the variability of the entire section appears to be large. This observation is confirmed by Table 1. The standard deviations

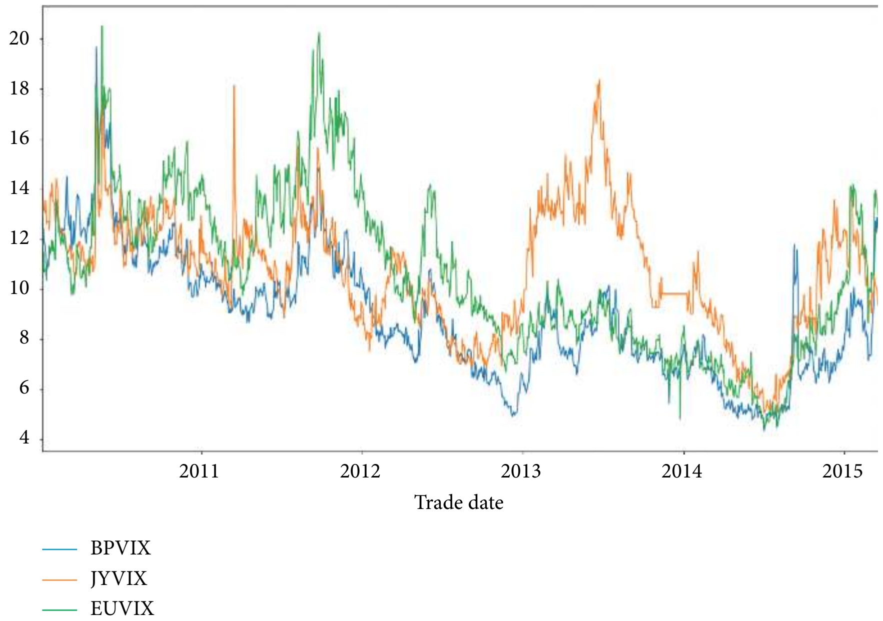


Figure 1: FXVIXs in period 1.

Table 1: Summary statistics for the three daily volatility indices.

	Max	Min	Mean	Std
Period 1				
BPVIX	19.67	4.33	8.93	2.46
JYVIX	19.17	5.03	10.58	2.36
EUVIX	20.51	4.43	10.78	3.04
Period 2				
BPVIX	29.10	5.31	12.02	4.21
JYVIX	17.54	9.60	12.80	1.70
EUVIX	19.41	4.62	9.67	1.75
Period 3				
BPVIX	15.68	5.16	9.24	2.03
JYVIX	14.66	4.29	8.18	1.81
EUVIX	12.74	3.99	7.11	1.42

of BPVIX, JYVIX, and EUVIX in this section are the largest among all periods, excluding BPVIX in 2016.

The second period represents the time around Brexit, which caused fluctuations in the global stock market, particularly in the European market. As shown in Figure 2, the UK index fluctuates the most, which affects the volatility of the European index. According to Table 1, BPVIX not only exhibits a high standard deviation but also has the largest difference between the maximum and minimum values.

The final period represents the time of uncertainty following the Brexit movement and recovery around the world. This period exhibits cyclic characteristics because the same problems arise repeatedly. Because intermediate trends between features of the first and second sections are visible, this section does not have any noteworthy features relative to the other sections. As shown in Figure 3, this period is longer than the second period, shorter than the first period, and less volatile than both periods, except JYVIX.

In this paper, for convenience, the three periods are referred to as Period 1, Period 2, and Period 3. Specifically, Period 1 ranges from 2010 to 2015, Period 2 covers 2016, and Period 3 ranges from 2017 to 2019. Similar subperiod analysis has been conducted in other studies (Gazioglu [68] and Grammatikos and Vermeulen [69]).

In machine learning, when constructing a model, performance evaluations are conducted. At this time, if a model trained on a particular training data set is evaluated on the same set, performance will be inflated by overfitting. Therefore, an original dataset should be divided into training and testing data, and a model should be trained on the training data. When evaluating performance, testing data, which were not used for training, are fed into the trained model. There is no ideal data allocation ratio for training and testing. With more training data, a model can see more examples and find better solutions, but overfitting may occur. Conversely, more testing data can lead to better generalization, but there underfitting may occur (Hastie et al. [70]).

According to Gu et al. [71], a simple data organization strategy generally uses 90% of the data for training and 10%

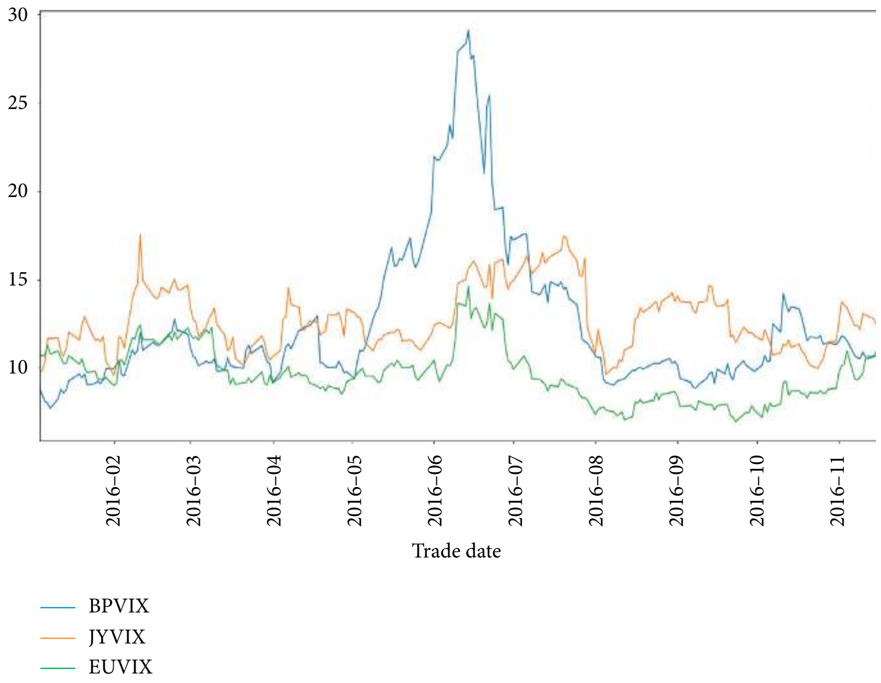


Figure 2: FXVIXs in period 2.

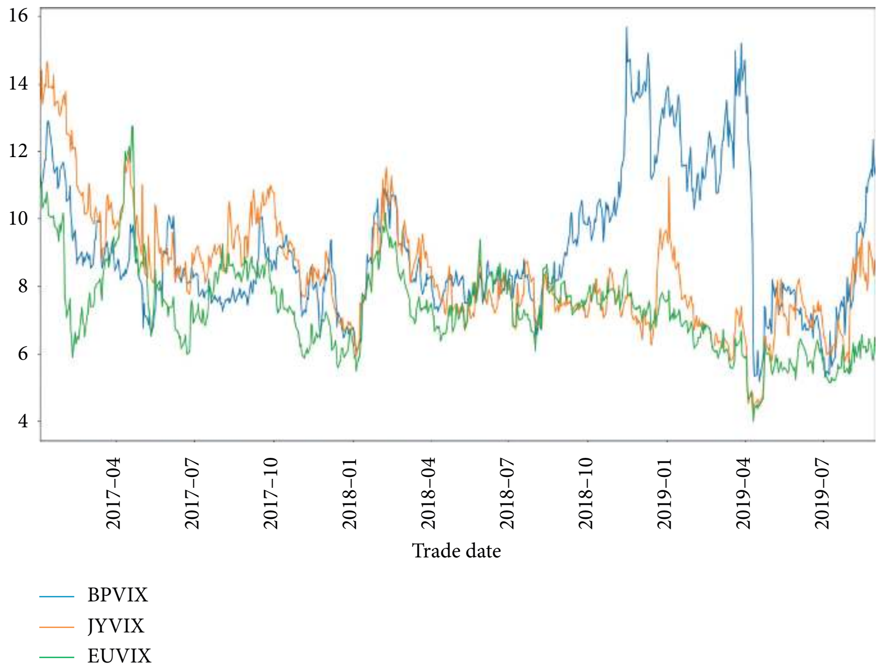


Figure 3: FXVIXs in period 3.

of the data for testing. This strategy was applied to the development of the Cubist regression tree model. We organized our data to use 85% of the data for training and 15% of the data for testing to avoid overfitting. Various data divisions are summarized in Table 2.

Cross-validation techniques were also applied to prevent overfitting. However, when a cross-validation method that selects random samples (e.g., K-fold cross-validation) is applied to time-series data, past values are predicted using future values. Therefore, in this study, time-series nested cross-validation was adopted to maintain the temporal order of the dataset for gradual overlapping and learning. The proposed model was trained and tuned on training and validation sets in each fold and then evaluated on a testing set. This allowed errors to be averaged to obtain an unbiased error estimate (Varma and Simon [72]).

Table 2: Descriptions of the training and testing datasets for each period.

Period	Training set	Test dataset
Period 1	(2010/01/04–2015/03/20)	(2015/03/23–2015/12/31)
Period 2	(2016/01/04–2016/11/15)	(2016/11/16–2016/12/30)
Period 3	(2017/01/03–2019/08/30)	(2019/09/03–2019/12/31)

3.2. LSTM. An RNN is a representative neural network with a recurrent hidden layer. Through this hidden layer, updates are backpropagated to train the model. By taking the results of previous hidden nodes as input data, it is possible to learn continuous forms. Therefore, RNNs are often used to analyze or predict time-series sequential data, such as stocks.

LSTM, which is a specific case of an RNN, was proposed by Hochreiter and Schmidhuber [21]. This model is designed to overcome the vanishing gradient problem of RNNs, where early layers are not trained properly when a network becomes deeper. Figure 4 presents the flow of RNN and LSTM progression. In contrast to the hidden units of the RNN, the LSTM structure consists of memory blocks. There are three steps (layers) in the LSTM model: the forget layer, which is the main advantage of LSTM, the input layer, and output layer. First, the forget gate f_t uses the sigmoid function, which is an activation function that converts current input data x_t and the previous hidden state h_{t-1} into numbers ranging from zero to one. Specifically, if an output is close to zero, it means that information cannot be passed to the next cell. In contrast, an output close to one means that information is passed to the next cell.

Second, the input gate i_t is a sigmoid function and decides which information in x_t and h_{t-1} is stored in the cell state C_t. At this step, there is also a tanh layer which creates a vector of new candidate values $(\check{C}_t)$ that could be added to the cell state C_t. The cell state C_t is updated by combining the outputs from the forget gate f_t and input gate i_t. By multiplying f_t and C_{t-1}, the amount of information from the previous time step cell that will be retained is determined. Furthermore, i_t times $\check{C}_t$ represents the update information from the input gate.

Finally, the output gate O_t applies the sigmoid function to the previous hidden state h_{t-1} and current input x_t to decide what the next hidden state should be. In addition, the current cell state C_t is passed through a tanh function. We multiply the tanh output with the sigmoid output to decide what information the hidden state should carry. In summary, the LSTM transition equations are defined as follows:

Gates:

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right),\\ i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right),\\ O_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right). \end{aligned} \tag{1}$$

Input transformation:

$$\check{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right). \tag{2}$$

Memory update:

$$\begin{aligned} C_t &= f_t \times C_{t-1} + i_t \times \check{C}_t,\\ h_t &= O_t \times \tanh\left(C_t\right), \end{aligned} \tag{3}$$

where W and b are the weights and biases, respectively. The × denotes elementwise multiplication.

In Figure 5, the input values x_t travel through three layers to overcome long-term dependencies using the following activation functions: σ (sigmoid) and tanh. The sigmoid function outputs a number between zero and one, which is a measure of how much information each component should convey. tanh helps keep the gradient as long as possible to prevent vanishing gradient problems.

3.3. Autoencoder

3.3.1. Basic Autoencoder. The autoencoder, which was first introduced in [35], utilizes a neural network consisting of an input layer, output layer, and hidden layers for self-supervised learning. Although this structure is similar to that of a typical neural network, the output and input layers have isomorphic vectors. The goal of this model is to derive a representation for an input dataset (e.g., dimensionality reduction) and make the reorganized data as close as possible to the input data. As shown in Figure 6, the encoder represents a stage at which the model can learn important characteristics of inputs and the decoder forms outputs similar to the inputs. The output represents a state in which the noise of the inputs is removed, resulting in more distinct characteristics. Based on these features, autoencoders are mainly used for image restoration or noise reduction.

$$Y = f\left(W_1 \cdot X + b\right), \tag{4}$$

$$\widetilde{X} = \widetilde{f}\left(W_2 \cdot Y + b\right), \tag{5}$$

where W_1 is the weight between input an X and hidden representation Y, W_2 is the weight between a hidden representation Y and output $\widetilde{X}$, and b is the bias, f and $\widetilde{f}$ represent the encoder and decoder, respectively, f accepts and compresses the input data (X) into a latent space (Y), and $\widetilde{f}$ is responsible for accepting latent space (Y) representations and reconstructing original inputs $(\widetilde{X})$.

This type of model is utilized in several methods to improve performance by manipulating hidden layers. A stacked autoencoder is used to solve the vanishing gradient problem by stacking hidden layers when a neural network is deep. Figure 7(a) presents a simple example of a stacked autoencoder. This structure increases the number of hidden nodes by stacking autoencoders hierarchically. A denoising autoencoder aims to extract stable structured data from dependent data by adding noise to input data and

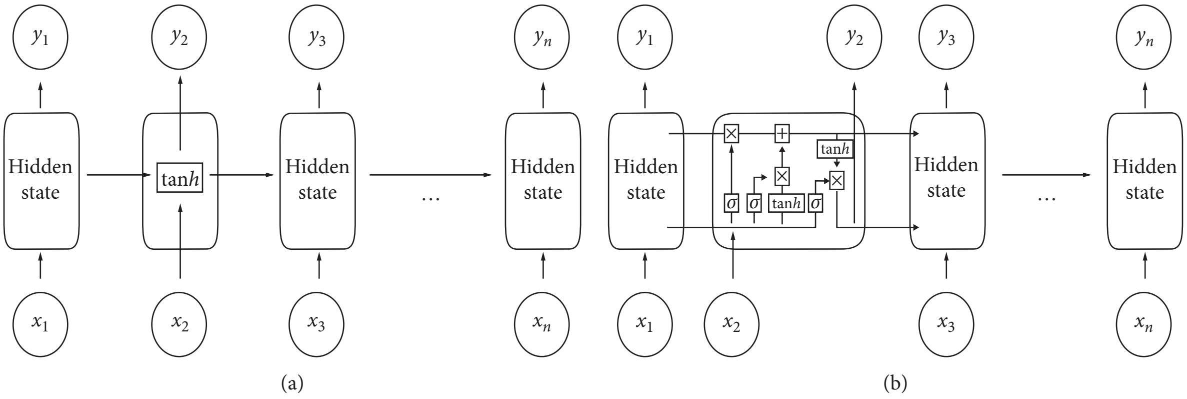


FIGURE 4: (a) RNN process. (b) LSTM process.

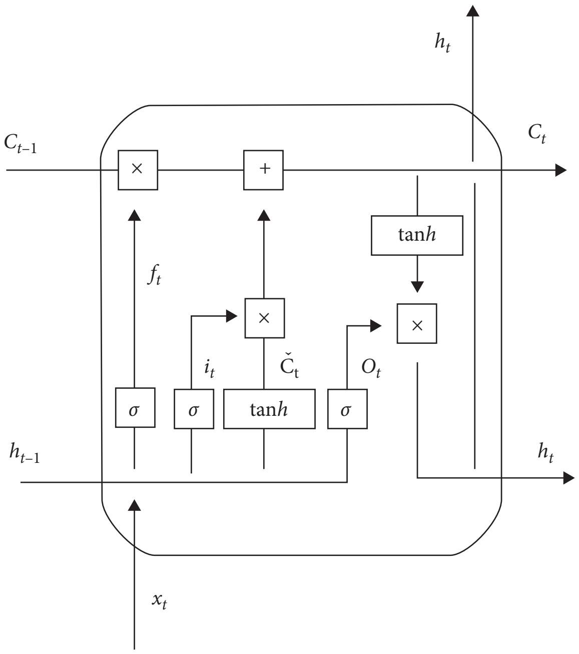


FIGURE 5: Structure of LSTM.

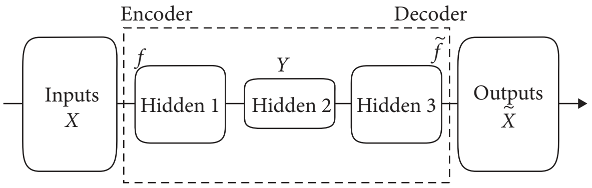


FIGURE 6: Structure of an autoencoder.

confirming that the output data correspond to pure input values. As shown in Figure 7(b), this model has a structure similar to that of a typical autoencoder, but it takes input data with added noise as new input data.

3.3.2. Autoencoder-LSTM. The autoencoder-LSTM model, which combines an autoencoder and advanced RNN, is implemented with an LSTM encoder and decoder for sequence data. This model has the same basic frame as an autoencoder, but is composed of LSTM layers, as shown in Figure 8(a). This model can learn complex and dynamic input sequence data from adjacent periods by using memory cells to remember long input sequence data.

The encoder and decoder components consist of two LSTM layers. To implement this structure, we adopted the "RepeatVector" tool provided by Keras, which is a deep learning API. Figure 8(b) presents the resulting structure.

3.4. Hyperparameter Optimization. A hyperparameter is a parameter that has a significant impact on the learning process. Maximizing model performance by finding optimal hyperparameter values to minimize a loss function is called hyperparameter optimization. This method is widely used in machine learning and deep learning. In this study, the well-known grid search method was adopted.

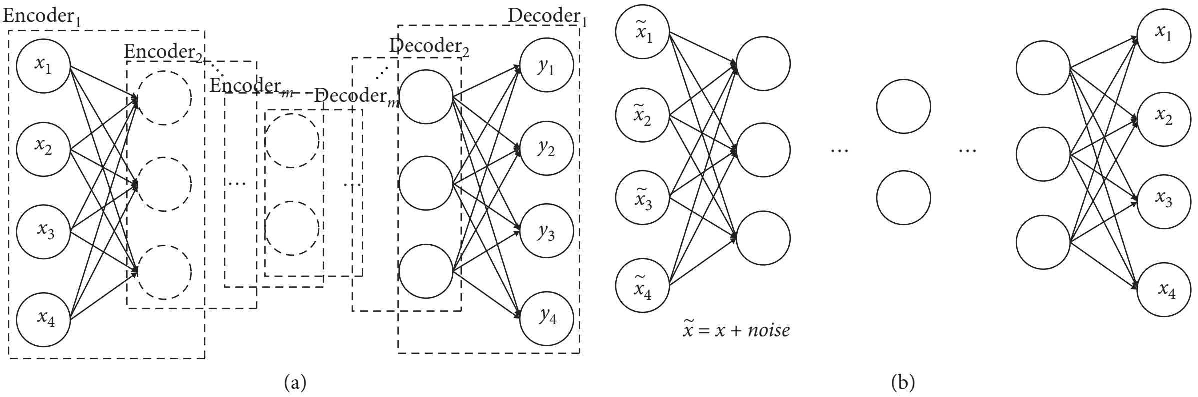


Figure 7: (a) Deep autoencoder. (b) Denoising and deep autoencoder (x_N denotes inputs with added noise).

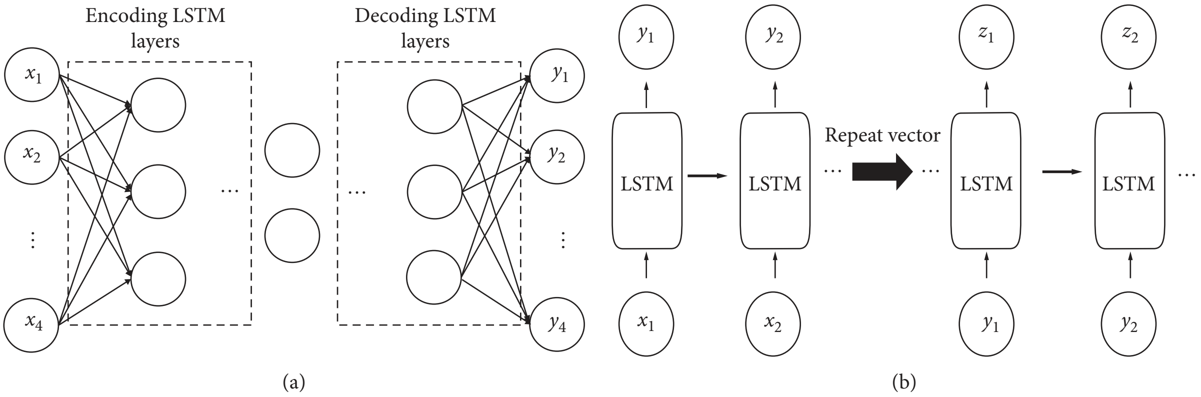


Figure 8: (a) Autoencoder-LSTM. (b) Structure of "RepeatVector."

A grid search finds the best parameters among a parameter set defined by a user and applies several parameter candidates to the model sequentially to identify the cases with the best performance. If there are few parameter candidates, optimal values can be obtained rapidly. However, if there are many candidates, optimization requires exponentially more time.

In this study, we adopted the grid search algorithm because it is the simplest and most widely used algorithm for obtaining optimal hyperparameters (Schilling et al. [73]). Although a random search can perform much better than a grid search on high-dimensional problems according to Hutter et al. [74], our data represent a simple time series and the candidate parameter set is limited. These are the main reasons why we adopted the grid search algorithm (Sun et al. [75] and Thornton et al. [76]). The Python technological stack was used for our experiments. We implemented the machine learning algorithms and grid search using the Scikit Learn, Keras, and TensorFlow packages.

We used a grid search to identify and apply optimal parameters for each section of our model. The optimized parameters are the batch size, activation function, and optimizer function. Two or three candidate groups were defined for each parameter.

More parameters and candidate groups could be defined, but it would increase training time significantly. We divided the data into three intervals and attempted to compare two models, thereby limiting the candidate groups to make the most of our limited resources.

Next, we optimized three parameters for stochastic gradient descent. The candidate batch sizes were 50 and 100, the activation functions were linear and ReLU, and the optimization functions were Adam, rmsprop, and nadam. The learning rates were default values built into each activation function (sprosprop: 0.001, Adam: 0.001, and nadam: 0.002).

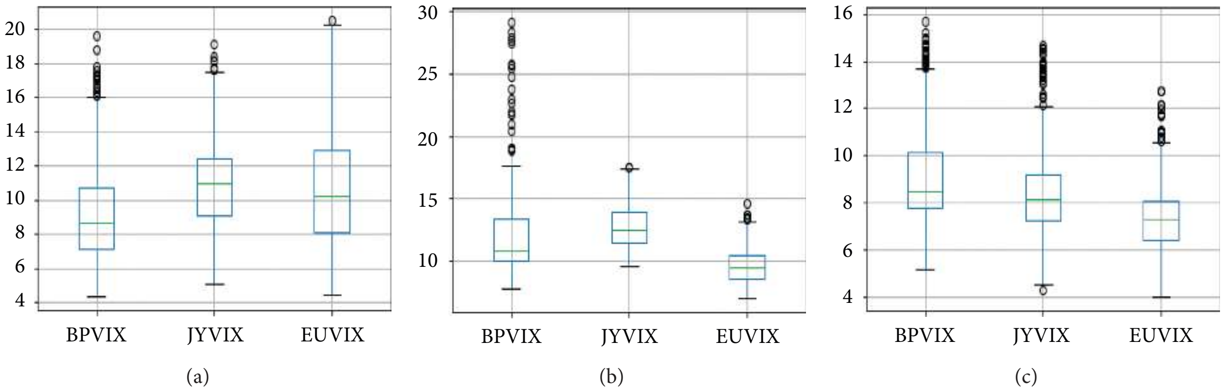
Finally, the autoencoder and autoencoder-LSTM models were unified into four layers: two encoding layers and two decoding layers. Based on the small amount of testing data, this small depth was determined to be sufficient.

4. Empirical Results

We used the aforementioned grid search to find optimal parameter combinations. Among a total of 12 parameter combinations, the best parameters were identified and six optimizations were performed for the two models (LSTM and autoencoder-LSTM) and three periods in the same

Table 3: Hyperparameter table.

	Activation	Optimizer	Batch size
Period 1			
LSTM	ReLU	RMSProp	50
AutoEncoder-LSTM	Linear	Nadam	100
Period 2			
LSTM	Linear	RMSProp	100
AutoEncoder-LSTM	Linear	RMSProp	100
Period 3			
LSTM	Linear	Adam	50
AutoEncoder-LSTM	Linear	Adam	100



(a)

(b)

(c)

Figure 9: (a) Outliers in period 1. (b) Outliers in period 2. (c) Outliers in Period 3.

Table 4: Five error measures for each model in the BPVIX.

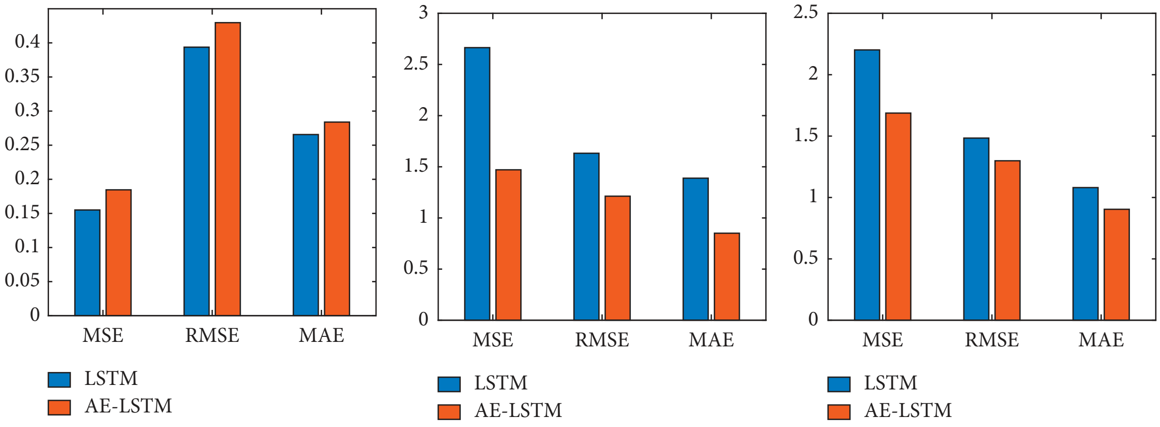
	MSE	RMSE	MAE	MPE (%)	MAPE (%)
Period 1					
LSTM	0.1550	0.3937	0.2656	−0.4904	3.0105
Autoencoder-LSTM	0.1846	0.4296	0.2839	−0.2830	3.2280
Period 2					
LSTM	2.6645	1.6323	1.3887	9.5562	19.0619
Autoencoder-LSTM	1.4706	1.2127	0.8508	−3.0532	10.1669
Period 3					
LSTM	2.2017	1.4838	1.0805	−6.1679	11.1552
Autoencoder-LSTM	1.6874	1.2990	0.9039	−0.5791	9.1082

Table 5: Five error measures for each model in the JYVIX.

	MSE	RMSE	MAE	MPE (%)	MAPE (%)
Period 1					
LSTM	0.6152	0.7843	0.4611	−0.4230	4.6438
Autoencoder-LSTM	0.4359	0.6602	0.4430	−1.3870	4.5153
Period 2					
LSTM	0.3321	0.5763	0.4525	−0.2644	3.5579
Autoencoder-LSTM	0.4261	0.6528	0.4933	0.79419	3.9160
Period 3					
LSTM	3.8273	1.9563	1.0422	−5.3252	15.0497
Autoencoder-LSTM	2.7176	1.6485	0.9447	2.6840	13.4985

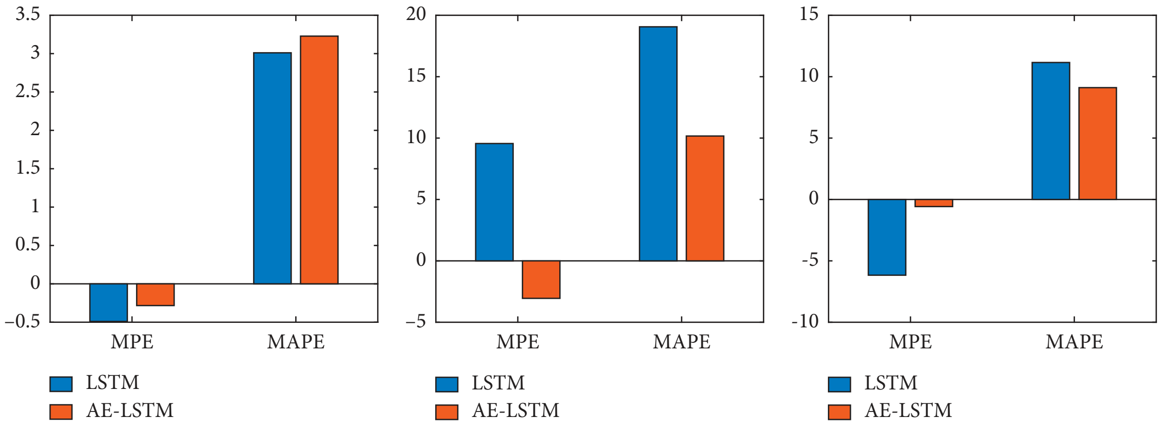
Table 6: Five error measures for each model in the EUVIX.

	MSE	RMSE	MAE	MPE (%)	MAPE (%)
Period 1					
LSTM	0.4104	0.6406	0.4893	2.5107	4.0954
Autoencoder-LSTM	0.2874	0.5361	0.3974	−0.2569	3.3006
Period 2					
LSTM	4.5074	2.1231	1.4746	3.8257	19.3755
Autoencoder-LSTM	3.8284	1.9566	1.4937	−6.4252	18.0702
Period 3					
LSTM	1.5205	1.2331	0.4530	0.5383	7.8089
Autoencoder-LSTM	1.3250	1.1511	0.5018	−1.9384	8.7721



(a) (b) (c)

Figure 10: MSE, RMSE, and MAE values of the BPVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.



(a) (b) (c)

Figure 11: MPE and MAPE values of the BPVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

manner. The results obtained via hyperparameter optimization are listed in Table 3.

The goal of this study was to obtain an accurate model for forecasting FXVIXs. We considered three FXVIXs with different distributions and outliers were different. We compare the forecasting performances of our models in terms of distributions and outliers. To this end, the forecasting results are split by period and separated by index. As methods for measuring error, the regression error metrics of mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) were adopted. Additionally, distributions were defined by variances and standard deviations. Outlier detection was applied using Tukey's box plot method, which defines outliers as samples that do not fall within the scope defined below:

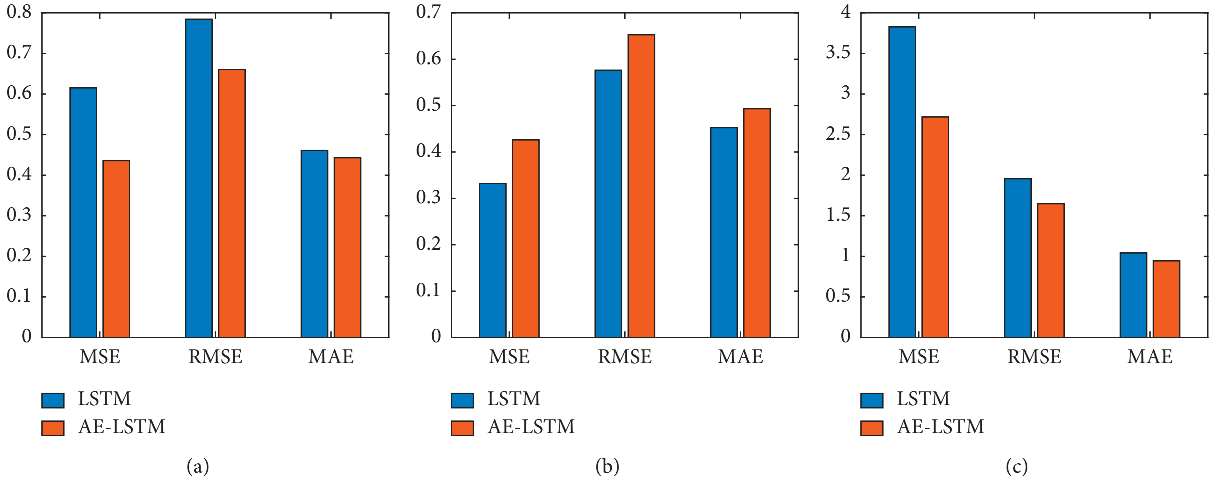


Figure 12: MSE, RMSE, and MAE values of the JYVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

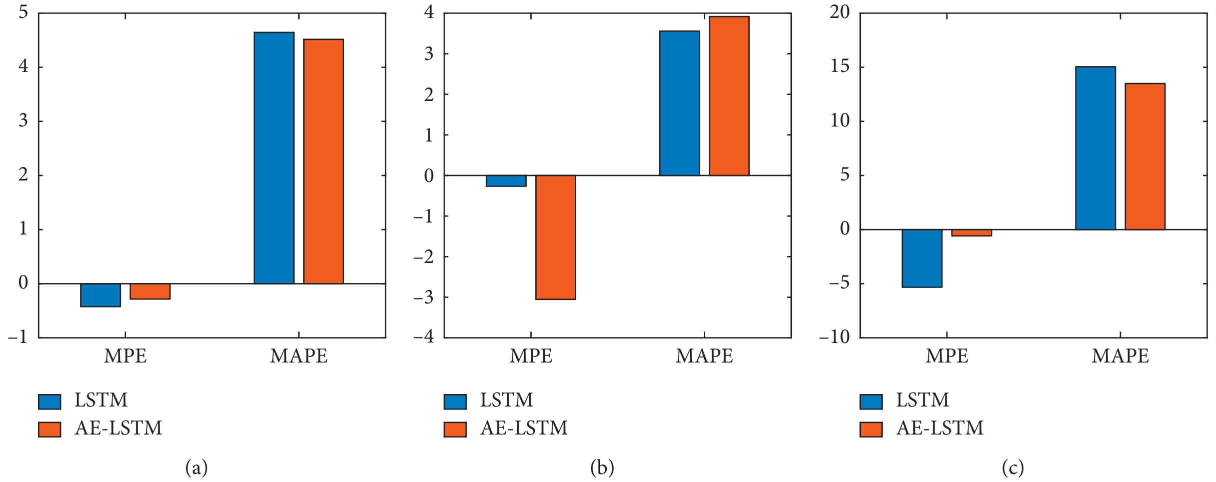


Figure 13: MPE and MAPE values of the JYVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

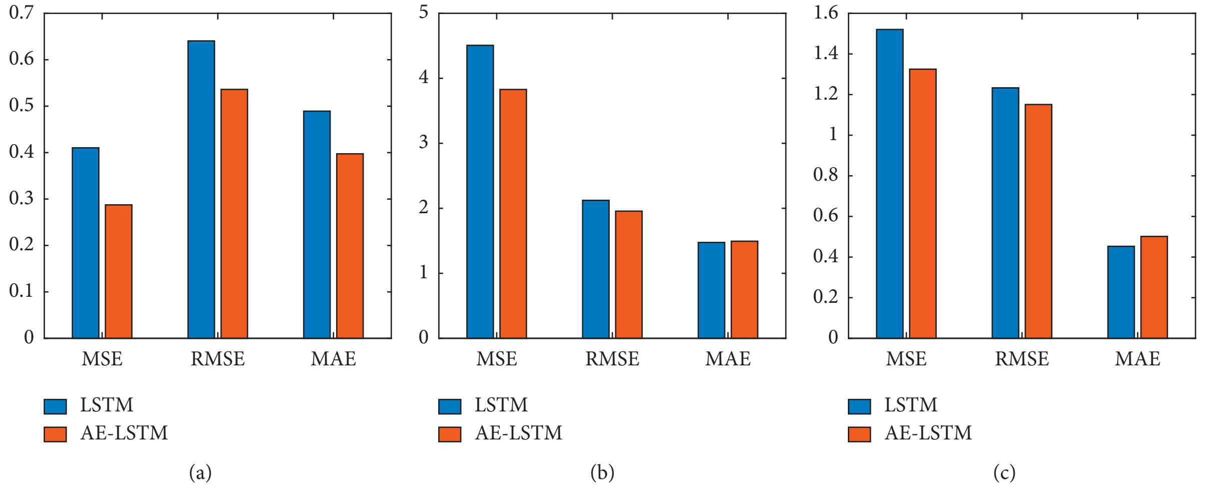


Figure 14: MSE, RMSE, and MAE values of the EUVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

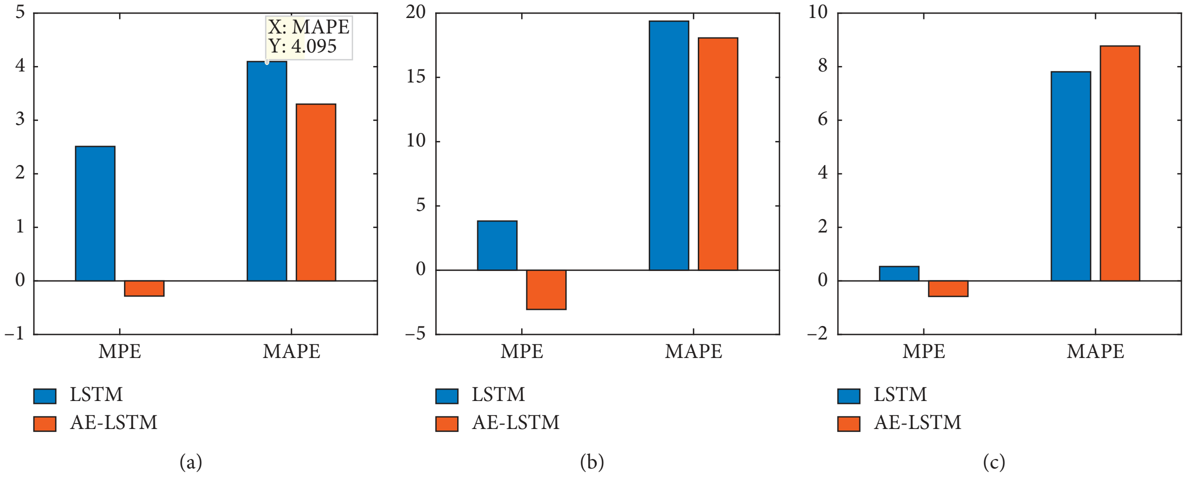


Figure 15: MPE and MAPE values of the EUVIX for the three periods. (a) Period 1. (b) Period 2. (c) Period 3.

$$[Q_1 - 1.5IQR, Q_3 + 1.5IQR], \tag{6}$$

where Q is the quantile and IQR is an interquantile range defined as follows: $IQR = Q_3 - Q_1$. To identify extreme outliers, multiplication by 1.5 was replaced with multiplication by 3.

Our main findings can be summarized as follows. First, the opportunities to learn volatility and forecast accuracy have a proportional relationship. In other words, there are many sections that rise and fall in the training data and learning these trends can improve prediction accuracy. As shown in Figure 9, the distribution is broad and there are many outliers in the order of (a) outliers in Period 1, (c) outliers in Period 3, and (b) outliers in Period 2. Second, as shown in Tables 4–6 and Figures 10–15, autoencoder-LSTM is affected more by variance and outliers than LSTM alone. In situations where variance and outliers exist in moderation, the LSTM model using an autoencoder, which can derive the features of inputs accurately, performs better than the model without an autoencoder. Third, among the deep learning methods, the autoencoder-LSTM exhibits the best prediction performance. In Tables 4–6, the results of the autoencoder-LSTM were analyzed to verify that the input data characteristics outperformed those of the general LSTM. Visual graphs of this trend are presented in Figures 10–15.

5. Summary and Concluding Remarks

The goal of this study was to develop a hybrid model based on deep learning models for forecasting FX volatility. In particular, we utilized the three FXVIXs as measures of FX volatility. An FXVIX represents the relationship between the currency of a country and the US dollar. Therefore, this study is meaningful because the FXVIX, which is related to the US and the global economy, sensitively reflects international economic trends.

Data-driven methods are more powerful than model-driven methods for forecasting asset price time-series data (see Kim et al. [77]). In this study, we investigated how event-driven data, which focus on events such as outliers in data-driven analysis, contribute to model performance. According to Shahid et al. [78], events and outliers are different, but outliers can be considered as a type of event. Because there is only one type of outlier in the data considered in this study, comparing differences in model performance accordingly is meaningful.

Our empirical results provide several interesting conclusions with useful practical implications. Our main findings can be summarized as follows. First, the spread of data and presence of outliers increase the accuracy of forecasting performance of the proposed model. Second, improvements in prediction accuracy are more pronounced with autoencoder-LSTM than with LSTM. Finally, for predicting FXVIXs, the autoencoder-LSTM model is superior to the LSTM.

Based on the empirical findings in Section 4, some implications can be observed. First, because the neural network model is a model created by mimicking the human brain, the data to be learned are important. As shown in this study, the forecasting accuracy of the hybrid model is affected by the number of cases for which variability and outliers can be learned. However, extreme outliers in Period 2 degraded the model's performance. Next, the use of an autoencoder, which can transform important properties of input data, similar to principal component analysis, is meaningful. Autoencoders are used for denoising images, watermark removal, dimensionality reduction, and feature variation among other tasks. In this study, we conceived the concept of feature variation. Additionally, several studies using autoencoders to predict time series have been recently published (Gensler et al. [79], Bao et al. [65], and Sagheer and Kotb [47]). Our study contributes to the literature by introducing a new approach called the autoencoder-LSTM for forecasting time series.

In practice, our findings can be helpful to researchers in economic research laboratories or policy managers who determine national economic policies because FXVIXs reveal important trends for FX that impact the global economy and volatility, meaning they can reveal market participant psychology. For example, Menkhoff et al. [33] demonstrated that global exchange volatility has a significant effect trading strategies based on financial data. Guo et al. [31] confirmed the effects of exchange rate volatility on the stock market. Similar experiments can be considered for future research on different financial indices, such as the S&P 500 and Dow Jones Industrial Average, which are important indices for understanding US and global markets (Ivanov et al. [80] and Liu et al. [81]). The US has the world's largest financial market and plays an important role in determining the trends of the international financial market. Therefore, we expect that predicting these indices will be as meaningful as predicting FXVIXs. Additionally, we expect that we can improve prediction accuracy by learning and incorporating data that can affect each index based on the results of this study, where we only considered FXVIXs. Additionally, hyperparameter optimization was performed using only a grid search, which is a commonly used machine learning algorithm, but we could increase the reliability of prediction by considering additional optimization algorithms.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors have declared that there are no conflicts of interest.

Acknowledgments

The work of S. Y. Choi was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (no. 2019R1G1A1010278).

References

[1] W. Huang, K. K. Lai, Y. Nakamori, and S. Wang, "Forecasting foreign exchange rates with artificial neural networks: a review," *International Journal of Information Technology & Decision Making*, vol. 3, no. 1, pp. 145–165, 2004.

[2] G. A. Vasilellis and N. Meade, "Forecasting volatility for portfolio selection," *Journal of Business Finance & Accounting*, vol. 23, no. 1, pp. 125–143, 1996.

[3] J. D. Knopf, J. Nam, and J. H. Thornton, "The volatility and price sensitivities of managerial stock option portfolios and corporate hedging," *The Journal of Finance*, vol. 57, no. 2, pp. 801–813, 2002.

[4] C. T. Brownlees and G. M. Gallo, "Comparison of volatility measures: a risk management perspective," *Journal of Financial Econometrics*, vol. 8, no. 1, pp. 29–56, 2010.

[5] G. M. Gallo and E. Otranto, "Forecasting realized volatility with changing average levels," *International Journal of Forecasting*, vol. 31, no. 3, pp. 620–634, 2015.

[6] T. Bollerslev, B. Hood, J. Huss, and L. H. Pedersen, "Risk everywhere: modeling and managing volatility," *The Review of Financial Studies*, vol. 31, no. 7, pp. 2729–2773, 2018.

[7] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[8] D. N. C. Vee, P. N. Gonpot, and S. Noor, "Forecasting volatility of usd/mur exchange rate using a garch (1, 1) model with ged and student'st errors," *University of Mauritius Research Journal*, vol. 17, no. 1, pp. 1–14, 2011.

[9] A. K. Dhamija and V. K. Bhalla, "Financial time series forecasting: comparison of various arch models," *Global Journal of Finance and Management*, vol. 2, no. 1, pp. 159–172, 2010.

[10] D. A. Bala and J. O. Asemota, "Exchange-rates volatility in Nigeria: application of garch models with exogenous break," *CBN Journal of Applied Statistics*, vol. 4, no. 1, pp. 89–116, 2013.

[11] D. S. Kambouroudis, D. G. McMillan, and K. Tsakou, "Forecasting stock return volatility: a comparison of garch, implied volatility, and realized volatility models," *Journal of Futures Markets*, vol. 36, no. 12, pp. 1127–1163, 2016.

[12] G. Köchling, P. Schmidtke, and P. N. Posch, "Volatility forecasting accuracy for bitcoin," *Economics Letters*, vol. 191, Article ID 108836, 2020.

[13] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," 2001.

[14] R. Cont, "Volatility clustering in financial markets: empirical facts and agent-based models," in *Long Memory in Economics*, Springer, Berlin, Germany, 2007.

[15] D. Pradeepkumar and V. Ravi, "Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network," *Applied Soft Computing*, vol. 58, pp. 35–52, 2017.

[16] Y. Liu, "Novel volatility forecasting using deep learning-long short term memory recurrent neural networks," *Expert Systems with Applications*, vol. 132, pp. 99–109, 2019.

[17] E. Ramos-Pérez, P. J. Alonso-González, and J. J. N. Velázquez, "Forecasting volatility with a stacked model based on a hybridized artificial neural network," *Expert Systems with Applications*, vol. 129, pp. 1–9, 2019.

[18] A. Bucci, "Realized volatility forecasting with neural networks," *Journal of Financial Econometrics*, vol. 18, no. 3, pp. 502–531, 2020.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

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

[22] E. Hajizadeh, A. Seifi, M. H. Fazel Zarandi, and I. B. Turksen, "A hybrid modeling approach for forecasting the volatility of S&P 500 index return," *Expert Systems with Applications*, vol. 39, no. 1, pp. 431–436, 2012.

[23] K. Werner, A. Fadic, and M. C. Minutolo, "Volatility forecast using hybrid neural network models," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2437–2442, 2014.

[24] K. Werner and M. C. Minutolo, "Gold price volatility: a forecasting approach using the artificial neural network–garch model," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7245–7251, 2015.

[25] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: a hybrid model integrating lstm with multiple garch-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[26] A. A. Baffour, J. Feng, and K. Evans, "A hybrid artificial neural network-gjr modeling approach to forecasting currency exchange rate volatility," *Neurocomputing*, vol. 365, pp. 285–301, 2019.

[27] Y. Hu, J. Ni, and L. Wen, "A hybrid deep learning approach by integrating lstm-ann networks with garch model for copper price volatility prediction," *Physica A: Statistical Mechanics and Its Applications*, vol. 557, Article ID 124907, 2020.

[28] M. Ishfaq, B. Q. Zhang, and S. M. R. Shah, "Global macroeconomic announcements and foreign exchange implied volatility," *International Journal of Economics and Financial Issues*, vol. 7, no. 5, p. 119, 2017.

[29] M. F. Dicle and B. Dicle, "Scottish independence referendum: risky or not?" 2017.

[30] P. Keith, "Brexit and its impact on the pound in the foreign exchange market," *The Economists' Voice*, vol. 16, no. 1, 2019.

[31] H. Guo, C. Neely, and J. Higbee, "Foreign exchange volatility is priced in equities," Technical report, Federal Reserve Bank of St. Louis, St. Louis, MO, USA, 2006.

[32] S. Z. S. Abdalla, "Modelling exchange rate volatility using garch models: empirical evidence from arab countries," *International Journal of Economics and Finance*, vol. 4, no. 3, pp. 216–229, 2012.

[33] L. Menkhoff, L. Sarno, M. Schmeling, and A. Schrimpf, "Carry trades and global foreign exchange volatility," *The Journal of Finance*, vol. 67, no. 2, pp. 681–718, 2012.

[34] C. Liu, W. Hou, and D. Liu, "Foreign exchange rates forecasting with convolutional neural network," *Neural Processing Letters*, vol. 46, no. 3, pp. 1095–1119, 2017.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, San Diego, CL, USA, 1985.

[36] P. Qaisar, "Forecasting stock index volatility with garch models: international evidence," *Studies in Economics and Finance*, vol. 32, no. 4, 2015.

[37] D. García and W. Kristjanpoller, "An adaptive forecasting approach for copper price volatility through hybrid and non-hybrid models," *Applied Soft Computing*, vol. 74, pp. 466–478, 2019.

[38] S.-Y. Choi and C. Hong, "Relationship between uncertainty in the oil and stock markets before and after the shale gas revolution: evidence from the ovx, vix, and vkospi volatility indices," *PLoS One*, vol. 15, no. 5, Article ID e0232508, 2020.

[39] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[40] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters," *Procedia Computer Science*, vol. 125, pp. 676–682, 2018.

[41] S. Muzaffar and A. Afshari, "Short-term load forecasts using lstm networks," *Energy Procedia*, vol. 158, pp. 2922–2927, 2019.

[42] E. Phaisangittisagul and R. Chongprachawat, "Receptive field resolution analysis in convolutional feature extraction," in *Proceedings of the 2013 13th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 485–489, Samui Island, Thailand, September 2013.

[43] M. Zhang, H. Wang, K. Zhou, and P. Cao, "Low probability of intercept radar signal recognition by staked autoencoder and svm," in *Proceedings of the 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, Zhejiang, China, October 2018.

[44] J. Zeng, J. Wang, L. Guo, G. Fan, K. Zhang, and G. Gui, "Cell scene division and visualization based on autoencoder and k-means algorithm," *IEEE Access*, vol. 7, pp. 165217–165225, 2019.

[45] M. Saha, P. Mitra, and R. S. Nanjundiah, "Autoencoder-based identification of predictors of indian monsoon," *Meteorology and Atmospheric Physics*, vol. 128, no. 5, pp. 613–628, 2016.

[46] S.-X. Lv, L. Peng, and L. Wang, "Stacked autoencoder with echo-state regression for tourism demand forecasting using search query data," *Applied Soft Computing*, vol. 73, pp. 119–133, 2018.

[47] A. Sagheer and M. Kotb, "Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems," *Scientific Reports*, vol. 9, no. 1, pp. 1–16, 2019.

[48] G. Boquet, A. Morell, J. Serrano, and J. L. Vicario, "A variational autoencoder solution for road traffic forecasting systems: missing data imputation, dimension reduction, model selection and anomaly detection," *Transportation Research Part C: Emerging Technologies*, vol. 115, Article ID 102622, 2020.

[49] S. Fu, Y. Li, S. Sun, and H. Li, "Evolutionary support vector machine for rmb exchange rate forecasting," *Physica A: Statistical Mechanics and Its Applications*, vol. 521, pp. 692–704, 2019.

[50] S. Sun, S. Wang, and Y. Wei, "A new ensemble deep learning approach for exchange rates forecasting and trading," *Advanced Engineering Informatics*, vol. 46, Article ID 101160, 2020.

[51] J. Vilasuso, "Forecasting exchange rate volatility," *Economics Letters*, vol. 76, no. 1, pp. 59–64, 2002.

[52] R. T. Baillie, T. Bollerslev, and H. O. Mikkelsen, "Fractionally integrated generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 74, no. 1, pp. 3–30, 1996.

[53] D. E. Rapach and J. J. K. Strauss, "Structural breaks and garch models of exchange rate volatility," *Journal of Applied Econometrics*, vol. 23, no. 1, pp. 65–90, 2008.

[54] P. Keith and K. N. Langeland, "Forecasting exchange rate volatility: garch models versus implied volatility forecasts," *International Economics and Economic Policy*, vol. 12, no. 1, pp. 127–142, 2015.

[55] Y. You and X. Liu, "Forecasting short-run exchange rate volatility with monetary fundamentals: a garch-midas approach," *Journal of Banking & Finance*, vol. 116, Article ID 105849, 2020.

[56] R. F. Engle, E. Ghysels, and B. Sohn, "Stock market volatility and macroeconomic fundamentals," *Review of Economics and Statistics*, vol. 95, no. 3, pp. 776–797, 2013.

[57] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, pp. 348–353, Como, Italy, July 2000.

[58] B. M. Henrique and V. A. Sobreiro, "Stock price prediction using support vector regression on daily and up to the minute prices," *The Journal of Finance and Data Science*, vol. 4, no. 3, pp. 183–201, 2018.

[59] R. V. Sreelekshmy Selvin, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (Icacci)*, pp. 1643–1647, Udupi, India, September 2017.

[60] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies," *Expert Systems with Applications*, vol. 83, pp. 187–205, 2017.

[61] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of arima and lstm in forecasting time series," in *Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1394–1401, Orlando, FL, USA, December 2018.

[62] H. Ohanyan, "Stock price forecast with deep learning lstm and econometric arima models," 2018.

[63] O. S. Deorukhkar, S. H. Lokhande, V. R. Nayak, and A. A. Chougule, "Stock price prediction using combination of lstm neural networks, arima and sentiment analysis," 2019.

[64] X. Liang, Z. Ge, L. Sun, M. He, and H. Chen, "Lstm with wavelet transform based data preprocessing for stock price prediction," *Mathematical Problems in Engineering*, vol. 2019, Article ID 1340174, 17 pages, 2019.

[65] 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, Article ID e0180944, 2017.

[66] J. Li, G. Liu, H. W. F. Yeung, J. Yin, Y. Y. Chung, and X. Chen, "A novel stacked denoising autoencoder with swarm intelligence optimization for stock index prediction," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, August 2017.

[67] H. Sun, W. Rong, J. Zhang, Q. Liang, and X. Zhang, "Stacked denoising autoencoder based stock market trend prediction via k-nearest neighbour data selection," in *Proceedings of the International Conference on Neural Information Processing*, pp. 882–892, Guangzhou, China, November 2017.

[68] S. Gazioglu, "Stock market returns in an emerging financial market: Turkish case study," *Applied Economics*, vol. 40, no. 11, pp. 1363–1372, 2008.

[69] T. Grammatikos and R. Vermeulen, "Transmission of the financial and sovereign debt crises to the emu: stock prices, cds spreads and exchange rates," *Journal of International Money and Finance*, vol. 31, no. 3, pp. 517–533, 2012.

[70] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, Berlin, Germany, 2009.

[71] Y. Gu, B. Wylie, S. Boyte et al., "An optimal sample data usage strategy to minimize overfitting and underfitting effects in regression tree models based on remotely-sensed data," *Remote Sensing*, vol. 8, no. 11, p. 943, 2016.

[72] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, no. 1, p. 91, 2006.

[73] N. Schilling, W. Martin, D. Lucas, and L. Schmidt-Thieme, "Hyperparameter optimization with factorized multilayer perceptrons," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 87–103, Porto, Portugal, September 2015.

[74] F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond manual tuning of hyperparameters," *KI - Künstliche Intelligenz*, vol. 29, no. 4, pp. 329–337, 2015.

[75] J. Sun, C. Zheng, X. Li, and Y. Zhou, "Analysis of the distance between two classes for tuning svm hyperparameters," *IEEE Transactions on Neural Networks*, vol. 21, no. 2, pp. 305–318, 2010.

[76] C. Thornton, F. Hutter, H. Holger, and K. Leyton-Brown, "Auto-weka: combined selection and hyperparameter optimization of classification algorithms," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 847–855, Chicago, IL, USA, August 2013.

[77] W. J. Kim, G. Jung, and S.-Y. Choi, "Forecasting Cds term structure based on nelson–siegel model and machine learning," *Complexity*, vol. 2020, Article ID 2518283, 15 pages, 2020.

[78] N. Shahid, I. H. Naqvi, and I. H. Naqvi, "Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey," *Artificial Intelligence Review*, vol. 43, no. 2, pp. 193–228, 2015.

[79] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 002858–002865, Budapest, Hungary, October 2016.

[80] S. I. Ivanov, F. J. Jones, and J. K. Zaima, "Analysis of DJIA, S&P 500, S&P 400, NASDAQ 100 and Russell 2000 ETFs and their influence on price discovery," *Global Finance Journal*, vol. 24, no. 3, pp. 171–187, 2013.

[81] C. Liu, J. Wang, D. Xiao, and Q. Liang, "Forecasting s&p 500 stock index using statistical learning models," *Open Journal of Statistics*, vol. 6, no. 6, pp. 1067–1075, 2016.