

Received March 18, 2022, accepted March 23, 2022, date of publication March 31, 2022, date of current version April 5, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3163723

# A Hybrid Prediction Model Integrating GARCH Models With a Distribution Manipulation Strategy Based on LSTM Networks for Stock Market Volatility

EUNHO KOO<sup>1</sup> AND GEONWOO KIM<sup>2</sup>

<sup>1</sup>Center for AI and Natural Sciences, Korea Institute for Advanced Study, Seoul 02455, Republic of Korea

<sup>2</sup>School of Liberal Arts, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea

Corresponding author: Geonwoo Kim (geonwoo@seoultech.ac.kr)

The work of Eunho Koo was supported by the Korea Institute for Advanced Study (KIAS) under Individual Grant AP086801.


The work of Geonwoo Kim was supported by the National Research Foundation of Korea Grant through the Korea Government under Grant NRF-2017R1E1A1A03070886.

**ABSTRACT** Accurate prediction of volatility is one of the most important tasks in financial decision making. Recently, the hybrid models integrating artificial neural networks with GARCH-type models have been developed, and performance gains from the models have been found to be outstanding. However, there have been few studies of hybrid models considering the nature of the distribution of financial data. Distribution of volatility time-series is highly concentrated near zero, and such aspect can cause low prediction performance on the whole domain of probability density function because weights in the networks can be trained to obtain accurate prediction only for the high frequency region, that is, near zero. To overcome the challenge, we propose a new hybrid model with GARCH-type models based on a novel non-linear filtering method to mitigate concentration property of volatility. For the filtering, we utilize root-type functions that transform extremely left-biased and pointed distribution of original volatility to a volume-upped (VU) distribution shifted to the right. Long short-term memory (LSTM) is employed as the basic implementation model, and the realized volatility of S&P 500 is predicted using the proposed models. It is found that the proposed hybrid model (VU-GARCH-LSTM) obtains 21.03% performance gain with respect to the root mean square error (RMSE) against the mean performances of the existing hybrid models integrating LSTM with GARCH-type models. Furthermore, the proposed model improves prediction performance in the right domain region of label probability density by making the prediction distribution comparable to the label distribution.

**INDEX TERMS** Stock market volatility, long short-term memory, GARCH models, distribution manipulation.

## I. INTRODUCTION

In the financial market, one of the most important issues is an exact prediction of stock market volatility. Volatility in finance presents generally the amount of fluctuation of financial time-series data and is used as risk measure since investment risk depends on volatility of underlying asset. Therefore, an accurate prediction of volatility is clearly an important issue for financial practitioners, such as investors, traders and risk managers, who want to gain stable profits.

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

However, the accurate prediction of volatility has been still a challenging task because the volatility is an unobserved latent variable and has complex features as heavy tail and non-stationary behavior. Motivated by these, we study a new novel model to improve the accuracy of prediction of stock market volatility.

There have been statistical models for predicting volatility that has the time-varying characteristics. As the parametric model to describe the conditional heteroscedasticity of financial volatility, the ARCH (Autoregressive Conditional Heteroskedasticity) model was suggested by Engle [1]. Bollerslev [2] generalized the ARCH model with the

moving average terms and introduced the GARCH (Generalized ARCH) model. Since GARCH model was introduced, GARCH-type models have been improved by many researchers. GARCH model of Bollerslev has the limitation that can not capture the leverage effect. To overcome this limitation and to provide more accurate prediction of volatility, Nelson [3] and Zakoian [4] proposed the EGARCH (Exponential GARCH) and the TGARCH (Threshold GARCH), respectively. In addition, there have been various extensions of the GARCH model in modeling stochastic volatility, and various GARCH-type models have been developed by many researchers [4]–[8]. However, GARCH-type models are difficult to capture complex fluctuation and nonlinear correlation of financial time-series data. In this study, we overcome the shortcomings using long short-term memory (LSTM) approach which is one of non-parametric models and improve in accuracy of volatility forecasts.

ANN is one of non-parametric models and can capture complex non-linear relationships between predictors and responses. Since ANN does not need the strict assumptions to predict time-series data, ANN has been used widely for the prediction of financial time-series. Hamid and Iqbal [9] predicted implied volatility from S&P 500 Index futures options using ANN approach and found that forecasts from ANN outperform implied volatility forecasts. Sermpinis *et al.* [10] used higher order neural network (HONN) approach to improve a forecasting performance of the realized volatility of the FTSE 100 futures. Xu *et al.* [11] suggested a novel UMIDAS-SVQR model combining unrestricted mixed data sampling (UMIDAS) and support vector quantile regression (SVQR) and showed that the model is superior to several models in terms of accuracy. Ramos-Pérez *et al.* [12] introduced a new model with the stacked algorithms based on a set of machine learning techniques and predicted S&P 500 volatility. In addition, Idrees *et al.* [13] predicted efficiently the Indian stock market volatility to construct an appropriate ARIMA model. Wang *et al.* [14] proposed a hybrid time-series predictive neural network (HTPNN) by learning the fusion feature of news and time-series to promote the prediction accuracy. Furthermore, there have been various hybrid models based on machine learning approach for predicting of financial time-series [15]–[17].

In recent years, the hybrid models that incorporate GARCH-type models into the ANN models have been proposed for better prediction of financial time-series. In general, outputs of GARCH-type models have been used as input of the ANN models, and the hybrid models have been called ‘ANN-GARCH’ models. More specifically, ANN-GARCH hybrid models have shown the improvements of prediction accuracy of financial time-series such as stock market indices, gold price, oil price, metals, exchange rate and cryptocurrency, etc [18]–[27]. That is, the results with ANN-GARCH hybrid models have indicated that the models improve the traditional GARCH-type models and ANN models in predictive ability.

Various combinations of GARCH and ANN models have been used to improve the prediction performance of stock market volatility. However, the performance on the right region of the target distribution (i.e. abnormal events) is still lacking, and the improvement of a breakthrough methodology for the challenge is still meager. The volatility has an extremely biased distribution to the left, which is a major impediment to improve the prediction performance. To address this, we propose a methodology for reducing the extremeness by manipulating the volatility distribution artificially. More specifically, for the prediction of realized volatility, we propose a hybrid model that combines GARCH-types models and LSTM with a novel data filtering methodology that manipulates distributions of input data. The followings are our contributions:

- A novel data filtering methodology is proposed to alleviate the extreme bias of volatility distribution.
- The proposed method outperforms the benchmarked strategies (various combinations of GARCH-type models and LSTM) in whole domain region, with exceptional performance in the right domain region of the target distribution.
- The proposed method makes the distribution of prediction similar to the target distribution.

The remainder of this paper is structured as follows. In Section II, we review the basic models used in this paper. In Section III, we introduce the data for the prediction and propose new hybrid models by integrating LSTM with the GARCH-type models and the proposed strategy based on the distribution manipulation. In Section IV, we present the experimental results of the proposed hybrid models with some meaningful figures and tables. Finally, the conclusion is given in Section V.

## II. MODELS

### A. GARCH-TYPE MODELS

#### 1) GARCH

The ARCH model [1] is a method that describes explicitly the change in variance over time in stochastic time-series. It estimates current volatility of a time-series using a linear combination of error terms which involve the Gaussian random noise and past information of volatilities. ARCH ( $q$ ) with order  $q$  is defined by

$$y_t = \mu_t + \sigma_t z_t, z_t \sim N(0, 1), \quad (1)$$

$$\varepsilon_t = \sigma_t z_t, \varepsilon_t | \mathcal{F}_{t-1} \sim N(0, \sigma_t^2), \quad (2)$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^q \alpha_i \varepsilon_{t-i}^2, \quad (3)$$

where  $y_t$  is a given stochastic time-series, and  $\mu_t$  is its drift.  $\sigma_t$ ,  $\mathcal{F}_t$  and  $N(0, 1)$  denote the volatility at time  $t$ , information up to time  $t$  and the standard Gaussian distribution, respectively, and all coefficients are set to be non-negative. GARCH [2] is an extension of the ARCH model that supplements a moving average component to autoregressive

component. GARCH ( $p, q$ ) is defined by

$$y_t = \mu_t + \sigma_t z_t, z_t \sim N(0, 1), \tag{4}$$

$$\varepsilon_t = \sigma_t z_t, \varepsilon_t | \mathcal{F}_{t-1} \sim N(0, \sigma_t^2), \tag{5}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^q \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^p \beta_i \sigma_{t-i}^2, \tag{6}$$

where all coefficients are positive. In (6), the volatility is predicted by taking the weighted sum of the predicted variance (second term) and observed volatilities (third term).

### 2) EGARCH

Unlike GARCH model, EGARCH allows the coefficients to be negative [3]. Consequently, the model can reflect the leverage effect that is an asymmetric information effects. Namely, a negative shock has greater effect on the volatility rather than a positive shock with the same magnitude. EGARCH ( $p, q$ ) is defined by

$$\varepsilon_t = \sigma_t z_t, \varepsilon_t | \mathcal{F}_{t-1} \sim N(0, \sigma_t^2), \tag{7}$$

$$\ln(\sigma_t^2) = \alpha_0 + \sum_{i=1}^q \alpha_i g(z_{t-i}) + \sum_{i=1}^p \beta_i \ln(\sigma_{t-i}^2), \tag{8}$$

$$g(z_t) = \theta z_t + \omega \left( |z_t| - \sqrt{\frac{2}{\pi}} \right), \tag{9}$$

where  $\alpha_i, \beta_i, \theta$  and  $\omega$  are parameters to be estimated. The function  $g(z_t)$  is linear in  $z_t$  with slope  $\theta + \omega$  (or  $\theta - \omega$ ) if  $z_t$  is positive (or negative).

### 3) TGARCH

TGARCH has utilized flexible lag structure in the volatility keeping the GARCH tractability, and improved the complexity of EGARCH [4]. TGARCH ( $p, q$ ) is defined by

$$\varepsilon_t = \sigma_t z_t, z_t \sim N(0, 1), \varepsilon_t | \mathcal{F}_{t-1} \sim N(0, \sigma_t^2), \tag{10}$$

$$\varepsilon^+ = \max(\varepsilon, 0), \varepsilon^- = \max(-\varepsilon, 0),$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^q [\alpha_{i1} (\varepsilon_{t-i}^+)^2 + \alpha_{i2} (\varepsilon_{t-i}^-)^2] + \sum_{i=1}^p \beta_i \sigma_{t-i}^2, \tag{11}$$

where  $\alpha_0 > 0, \alpha_{i1} \geq 0, \alpha_{i2} \geq 0, \beta_i \geq 0$  for all  $i$ . EGARCH and TGARCH have some similarities since the models can detect asymmetries in volatility. However, EGARCH imposes a constant structure at all lags while different lags may yield opposite contributions in TGARCH.

### B. LSTM

Recurrent neural network (RNN) is a class of artificial neural networks, which is designed for the prediction of sequential data including time-series data [28]. RNN consists of input, hidden, and output layers, and can unfold with a certain depth suitable to input dataset. Classical RNN has a defect called the vanishing gradient problem [29], and LSTM overcame the problem by modifying and improving its structure [30].

The feed-forwarding process of LSTM for the input data  $x_t$  and hidden state  $h_t$  at time-step  $t$  can be formulated as

$$i_t = \sigma(W_1 X + b_1), \tag{12}$$

$$f_t = \sigma(W_2 X + b_2), \tag{13}$$

$$o_t = \sigma(W_3 X + b_3), \tag{14}$$

$$g_t = \tanh(W_4 X + b_4), \tag{15}$$

$$c_t = c_{t-1} * f_t + g_t * i_t, \tag{16}$$

$$h_t = \tanh(c_t) * o_t, \tag{17}$$

where  $W_i$  and  $b_i$  are weights and bias terms, respectively, and  $X = \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$ . Functions  $\sigma$  and  $\tanh$  are defined by  $\sigma = 1/(1+e^{-x})$  and  $\tanh = \frac{e^x + e^{-x}}{e^x - e^{-x}}$ , respectively. The symbol '\*' means the elementwise product of two vectors. Net flow in a LSTM block is illustrated in Fig. 1. It can be shown that the total number of parameters in LSTM is  $4(K_I K_H + K_H^2)$  where  $K_I$  and  $K_H$  indicate the dimension of  $x_t$  and  $h_t$ , respectively.

## III. DATA AND PROPOSED STRATEGY

### A. EXPLANATORY VARIABLES

In this study, 16 financial indices are employed as the input explanatory variables. The variables are listed in Table 1. For each variable, the common daily closed values from 1st-Jan-2004 to 30th-Nov-2020 (the total size of data is  $3179 \times 19$ ) are used. The variables are selected based on the works of [31]–[33], and downloaded from Yahoo Finance website. Except for VIX, we change each original price (closed price) into the square of log return. That is, we use  $y_t$  defined by  $y_t = (\ln(R_t/R_{t-1}))^2$ , where  $R_t$  is closed price at time  $t$ .

TABLE 1. 16 explanatory input variables employed in this study.

Input variables	Description
JPY/USD	Exchange rate of Japan Yen to United States Dollar
Nikkei 225	Nikkei index
CNY/USD	Exchange rate of China Yuan to United States Dollar
SSE	Shanghai Composite Index
SEHK	Hang Seng Composite Index
GBP/USD	Exchange rate of British Pound to United States Dollar
EUR/USD	Exchange rate of Euro to United States Dollar
CAC 40	France Stock Market Index
DAX	German Stock Market Index
VIX	Chicago Board Options Exchange Volatility Index
US Bond Yields	10-year/long term yields
Crude Oil	Crude Oil Price
Gold	Gold price
Nasdaq	Nasdaq Composite Index in the United States
Dow Jones	Dow Jones Index in the United States
S&P 500	S&P 500 Stock Market Index in the United States

We employ realized volatility of S&P 500 as the label (target value). The realized volatility represents how much the stock price has changed during a certain period, and defined by

$$RV_t = \frac{1}{P} \sum_{i=1}^P (L_{t+i} - \bar{L})^2,$$

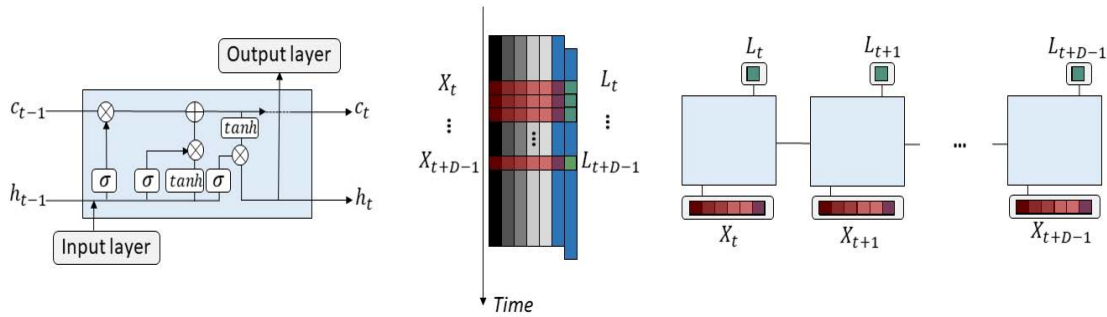


FIGURE 1. Structure of the LSTM cell (left) and a schematic diagram (right) for the process of the unfolded LSTM.

where  $p$ ,  $L_t$ , and  $\bar{L}$  are the period, the log return rate of S&P 500 at time  $t$ , and the averaged value of the log return rate during  $p$  days, respectively. In this study, we set  $p = 5$ . This means that  $RV_t$  is estimated from the data of one-week trading days. S&P 500 price is used not only as an input variable but also as the target value, to take full use of its historical information. However, the realized volatility is only used in the label.

**B. VOLUME-UP STRATEGY**

In this subsection, we propose a novel method termed Volume-Up (VU) as a non-linear filtering of input data. The distribution of the volatility time-series is extremely concentrated close to zero, but the low-frequency events distributed to the right domain of the label probability density function (PDF) have a practical significance. However, because the general loss function in a neural network only measures the average of total errors and the network is learned to decrease the average, such network properties can degrade the prediction performance of the (low-frequency) events. VU has been developed to mitigate the nature of biased input distribution, and to shift the distribution to the right in the label PDF domain. In this study, we evaluate the relationship between the artificial mitigation of the biased aspect of volatility distribution using VU strategy and the improvement in prediction performances in terms of various error metrics.

Let  $X$  and  $Y$  be two random variables, and let their corresponding PDFs and cumulative distribution functions (CDFs) be  $f_X, f_Y$  and  $F_X, F_Y$ , respectively. Let  $g(x)$  be an increasing function such that  $Y = g(X)$ . Then, we have

$$F_Y(y) = P(Y < y) = P(g(X) < y) = P(X < g^{-1}(y)) = F_X(g^{-1}(y)). \tag{18}$$

where  $g(x)$  is a monotonic function. Differentiation on both side of (18) yields

$$f_Y(y) = F'_Y(y) = \frac{d}{dy} F_X(g^{-1}(y)) = f_X(g^{-1}(y)) \frac{d}{dy} g^{-1}(y) \tag{19}$$

where  $F'$  and  $g^{-1}$  denote derivative of  $F$  and inverse of  $g$ , respectively. The equation (19) implies that PDF of  $Y$  can be

expressed explicitly in terms of PDF of  $X$  and inverse of  $g$ . In this study, we apply a root type function to the function  $g$  in above calculations. That is, we define the function  $g$  as  $g(x) = x^\alpha, 0 < \alpha < 1$ , where the number  $\alpha$  is the hyperparameter of VU to be analyzed. In this study, the hyperparameter is tested with nine values  $\alpha = i/10, i = 1, 2, \dots, 9$ . By direct calculation using (19), we can obtain

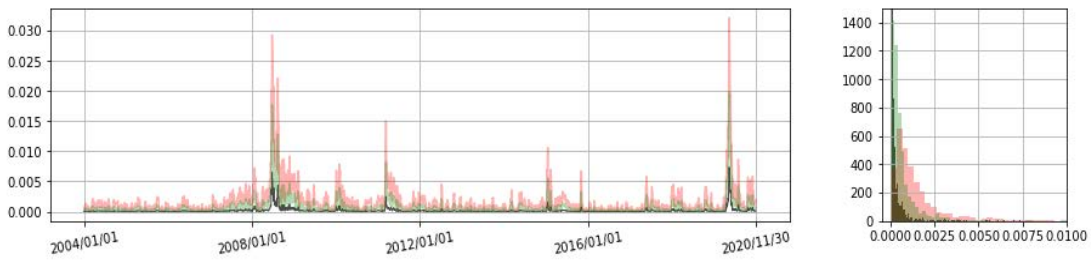
$$f_Y(y) = f_X(y^{1/\alpha}) \frac{1}{\alpha} y^{1-1/\alpha}. \tag{20}$$

In (20), for example, if  $\alpha = 1/2$ , then  $f_Y(y) = f_X(y^2) \frac{2}{y}$ . We state some remarks on VU. First, the function root-type function is intentionally chosen to make it similar trait of  $F_X(x)$ . It can be shown from (18) and (19) that if  $g(x) = F_X(x)$ , then the random variable  $Y$  has an uniform distribution. That is, applying CDF of  $X$  to  $g(x)$  makes PDF of  $Y$  to be flattened. Functions with CDF-like shaped would alleviate the extremely biased property of input-data distribution. Second, the function  $g(x)$  maps a narrower interval near zero into a wider interval near zero, and a wider interval far from zero into a narrower interval far from zero. Third, the distribution of  $Y = g(X)$  is pulled to the right direction. Fourth, as the hyperparameter  $\alpha$  approaches to 1 (from the left side),  $g(x)$  approaches to the identity filtering. Fifth, the decoding process is simple. The final prediction is obtainable by applying  $g^{-1}(x) = x^{1/\alpha}$  to the output. Distribution changes of S&P 500 volatility by  $g(x)$  are illustrated in Fig. 2.

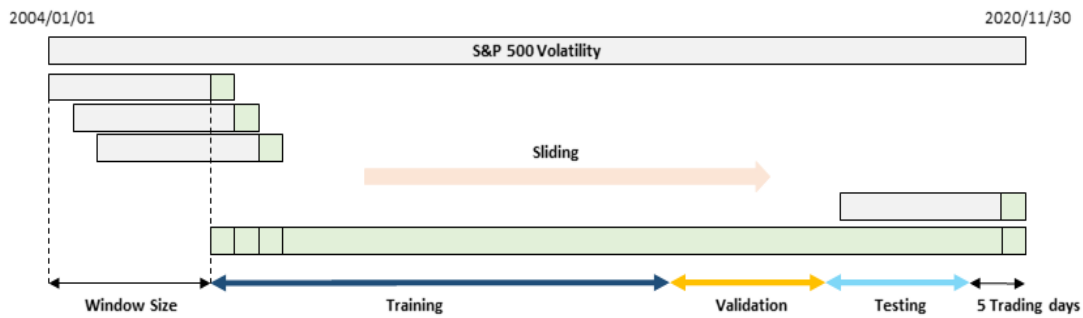
**C. PROPOSED HYBRID MODELS: INTEGRATION OF GARCH MODELS, LSTM WITH VU STRATEGY**

Numerous studies have shown that the incorporation of neural networks (NN) and GARCH-type models improves prediction performances of stock market volatility compared to applying GARCH-type models only [25], [31], [34]. Also, Kim and Won [35] reported that employing information extracted by multiple GARCH-type models as inputs yields better performance than by a single GARCH model. In this study, we propose a new hybrid model combining LSTM, GARCH-type models and VU strategy. In particular, to clarify the performance gains from the VU, we divide the experiments into the VU-applied cases and the native (non-VU) cases.





**FIGURE 2.** Squared volatility time-series of S&P 500 used in this study from 1st January 2004 to 30th November 2020. Black corresponds to the original data. Red and green correspond to VU applied data with hyperparameter  $\alpha=0.7$  and  $\alpha=0.8$ , respectively. Left and right indicate three time-series and their distributions, respectively.



**FIGURE 3.** Rolling window method. Green squares are obtained from each of GARCH models using previous information (fixed-length gray rectangles), and used as part of the inputs of LSTM.

The rolling window method is employed for the experiments. The method continuously incorporates relatively recent observations to the model fitting and prediction, which can help the networks adapt well to the current economic situation. Fig. 3 shows the method in detail. We apply the method to outputs of GARCH-type models (as well as other explanatory variables) to make them the input of the proposed model. Based on works of [31], [36], we employ 9 models GARCH (1,1), GARCH (2,2), GARCH (3,3), EGARCH (1,1), EGARCH (2,2), EGARCH (3,3), TGARCH (1,1), TGARCH (2,2), and TGARCH (3,3) as the GARCH-type models to be tested. For the simplicity, we denote the predictions obtained from the models as G1, G2, G3, E1, E2, E3, T1, T2, and T3, respectively. We also present the residuals of all invited GARCH-type models for S&P 500 in Fig. 4.

To evaluate the prediction performances of the proposed hybrid models, total 10 comparative experiments have been conducted (Table 2). The experiments consist of the following: 1) Pure-LSTM (P-LSTM) that only use explanatory variables as input data, 2) GARCH applied LSTM (G-LSTM) that uses G1, G2, and G3 as input data, 3) EGARCH applied LSTM (E-LSTM) that uses E1, E2, and E3 as input data, 4) TGARCH applied LSTM (T-LSTM) that uses T1, T2, and T3 as input data, 5) Mixed GARCH models applied LSTM (GET-LSTM), that uses predictions obtained from all GARCH-type models as input data, 6) VU applied LSTM for each of fore-mentioned 5 experiments (denote them as VU-P-LSTM, VU-G-LSTM, VU-E-LSTM, VU-T-LSTM, and VU-GET-LSTM, respectively).

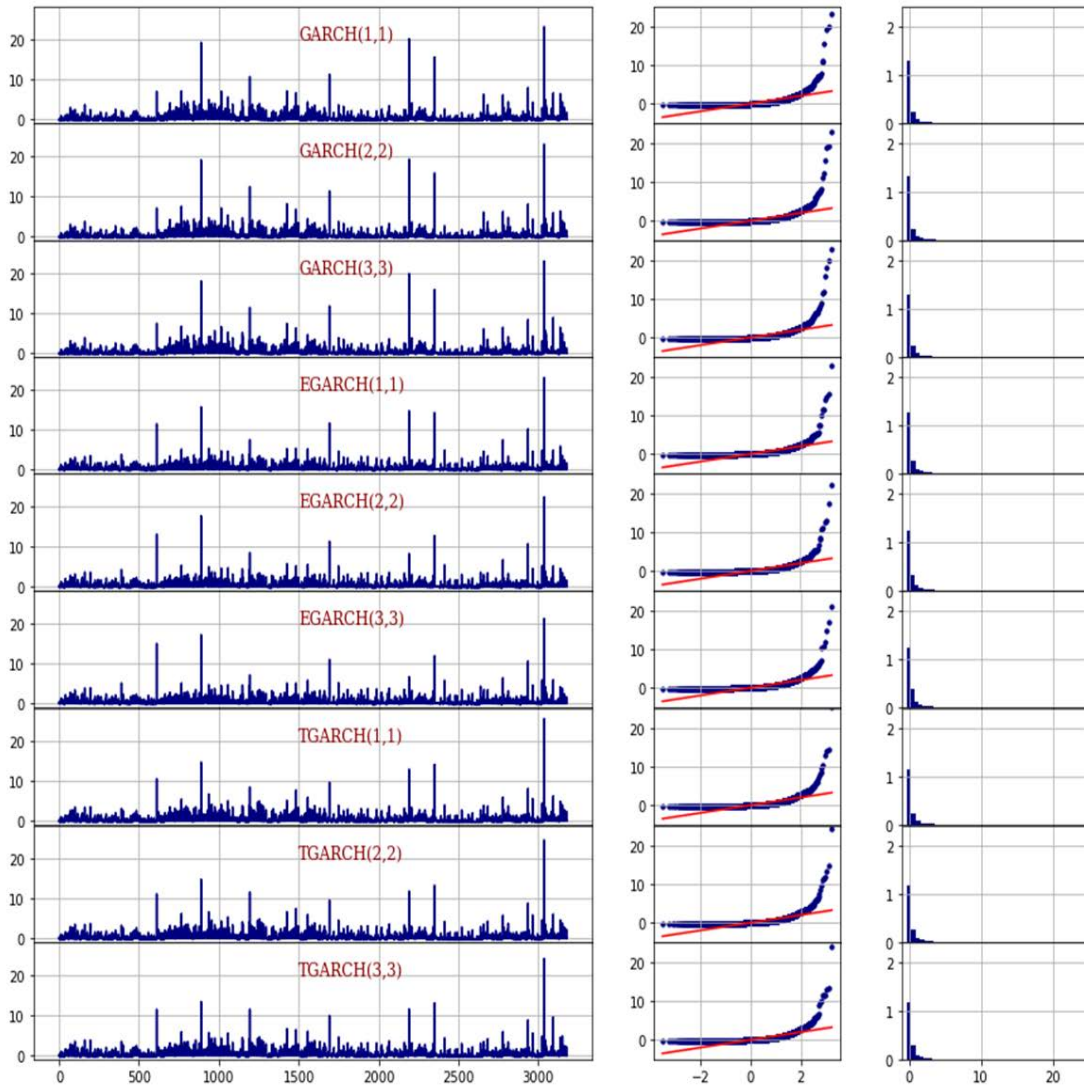
In GET-LSTM, if we employ all predictions obtained by three GARCH models as input data, the input dimension would be different compared to other single GARCH based models. Also, the experiments have concerns about overfitting caused by a large input dimension. In order to avoid the problem, we use Bayesian information criterion (BIC) to select only three of the nine inputs. BIC is an estimator of prediction error for statistical model selection, and defined by  $BIC = k \ln(n) - 2 \ln(\hat{L})$  where  $k$  is the number of estimated parameters in the model,  $\hat{L}$  is the maximum value of the likelihood function for the model, and  $n$  indicates the number of sample size. For a fixed window size  $W$ , we select three of the nine input candidates with the lowest BIC values, and use them as the input data of GET-LSTM.

## IV. EXPERIMENTS AND RESULTS

### A. EXPERIMENTS

#### 1) SETUP

As described in Section II-B, the total number of parameters in LSTM is  $4(K_I K_H + K_H^2)$  where  $K_I$  and  $K_H$  denote input dimension and hidden dimension (dimension of each gate), respectively. The input dimension  $K_I$  for each experiment is presented in Table 2. The hidden dimension is set to be  $K_H = 4$  which has been found as the best number in between over-fitting and under-fitting conditions. The RELU function [37] defined by  $RELU(x) = \max(x, 0)$  is set to be located in the output layer as an activation function. Batch size and epochs are set to be 512 and 200, respectively. Linear decreasing learning rate (from 0.005 to 0.0005 as



**FIGURE 4.** Residuals analysis for GARCH models. The first, second, and third columns represent residuals, the quantile-quantile plots (Q-Q plots), and the residual distributions for all invited GARCH models.

epochs increase) is employed. The ratio of train, validation, test of dataset is chosen to 6:2:2. The Adam optimizer [38] is applied as an optimization tool, which combines the gradient descent with momentum [39], moving average, and RMSprop [40]. Mean square error loss function defined by  $\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$  is used as the loss function where  $N$ ,  $y$ ,  $\hat{y}$  denote number of samples, label, predictions, respectively. We used the uniform normalization method, which assigns maximum and minimum of data to be 1 and 0. Also, the uniform parameter initialization on  $[-0.2, 0.2]$  is chosen. For the hyperparameters (hidden dimension for each gate, depth of LSTM, learning rate, epochs, and batch size) selection, we use a mixture of random search method and grid search method. The learning rate, batch size, and epochs are firstly extracted using the random search method. And then, using the grid search method, we extract the other hyperparameters whose candidates are selected near values that performed well

in the random search step. All hyperparameters are selected so that the error induced from the validation dataset can be minimized.

## 2) ERROR METRICS

The performances are measured with four error metrics. Root mean square error (RMSE) (defined by  $\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$ ) where  $N$ ,  $y$ ,  $\hat{y}$  denote number of samples, label, predictions, respectively) and correlation coefficient (CC) (defined by  $\text{cov}(y, \hat{y}) / (\sigma_y \sigma_{\hat{y}})$  where  $\text{cov}(x, y)$  indicates the covariance between two variables and  $\sigma_X$  is a standard deviation of  $X$ ) are employed as basic tools for error measurement. In addition, we adopt the area of the overlapping section of two distributions (those of the label and predicted value) as an error metric. Although a larger common section is not a necessary and sufficient condition for a better prediction, we assume that the overlapping distribution likely leads to performance

**TABLE 2. Configuration of input data in 10 experiments.**

Notation	VU	Input Data					Input Dimension
		Exp. Var.	G1, G2, G3	E1, E2, E3	T1, T2, T3		
P-LSTM	N	Y	N	N	N	16	
VU-P-LSTM	Y	Y	N	N	N	19	
G-LSTM	N	Y	Y	N	N	19	
VU-G-LSTM	Y	Y	Y	N	N	19	
E-LSTM	N	Y	N	Y	N	19	
VU-E-LSTM	Y	Y	N	Y	N	19	
T-LSTM	N	Y	N	N	Y	19	
VU-T-LSTM	Y	Y	N	N	Y	19	
GET-LSTM	N	Y	Best three with respect to BIC			19	
VU-GET-LSTM	Y	Y	Best three with respect to BIC			19	

**TABLE 3. BIC among all window size and GARCH models. Lowest three values for each window size are indicated in boldface.**

Window size	30	60	90	120	150	180	210	240	270	300	330	360	390	420	450	480	
B I C	G1	108	200	290	379	466	553	640	726	812	898	984	1070	1155	1241	1326	1411
	G2	102	193	282	371	458	544	631	717	803	888	974	1060	1145	1231	1316	1401
	G3	<b>96</b>	<b>186</b>	<b>275</b>	<b>363</b>	<b>450</b>	<b>536</b>	622	<b>708</b>	794	880	<b>964</b>	1051	1137	1222	1307	1392
	E1	99	212	303	396	478	574	634	719	808	895	985	1063	1146	1225	1309	1390
	E2	<b>94</b>	210	285	375	463	546	<b>618</b>	717	<b>787</b>	<b>874</b>	965	<b>1048</b>	<b>1129</b>	<b>1214</b>	<b>1294</b>	<b>1377</b>
	E3	<b>87</b>	<b>180</b>	<b>264</b>	<b>351</b>	<b>432</b>	<b>520</b>	<b>601</b>	<b>687</b>	<b>769</b>	<b>853</b>	<b>938</b>	<b>1019</b>	<b>1102</b>	<b>1187</b>	<b>1268</b>	<b>1354</b>
	T1	114	207	296	384	471	556	641	727	812	897	982	1067	1151	1236	1320	1404
	T2	106	197	286	373	458	543	628	713	798	882	967	1052	1136	1220	1304	1388
	T3	98	<b>187</b>	<b>274</b>	<b>361</b>	<b>446</b>	<b>531</b>	<b>615</b>	<b>700</b>	<b>784</b>	<b>869</b>	<b>953</b>	<b>1038</b>	<b>1122</b>	<b>1206</b>	<b>1290</b>	<b>1374</b>

gains in this study. RMSE on the extreme region is employed as another error metric to evaluate the prediction performance of rapid changes in financial indices. The extreme region is defined by  $[l, \infty)$  where  $P(l < x) = 0.2$ . In other words, the extreme region corresponds to the rightmost 20% of the probability.

## B. RESULTS

### 1) DETERMINATION OF WINDOW SIZE

In this study, the performances are tested using 16 values of window size  $W$  ( $W = 30 \times i, i = 1, 2, \dots, 16$ ) and evaluated using four error metrics. Also, to determine input data in GET-LSTM model, we investigate BIC values for each of 9 GARCH models (Table 3), and choose lowest three of them for each window size.

As mentioned in Section III-B, nine values of the hyperparameter  $\alpha$  ( $\alpha = i/10, i = 1, 2, \dots, 9$ ) are tested to evaluate the VU strategy for each experiment. We firstly define a 'positive section' as the set of values of  $\alpha$  at which the performance of VU is better than the native (non-VU) strategy. According to the results in the next section, the positive section accounts for about 70% of the possible  $\alpha$ -region, that is, the interval (0,1) for each implementation model and error metric. To determine the window size, we present the prediction performances by comparing the proposed model and native experiments for each implementation model and error metric in Fig. 5.

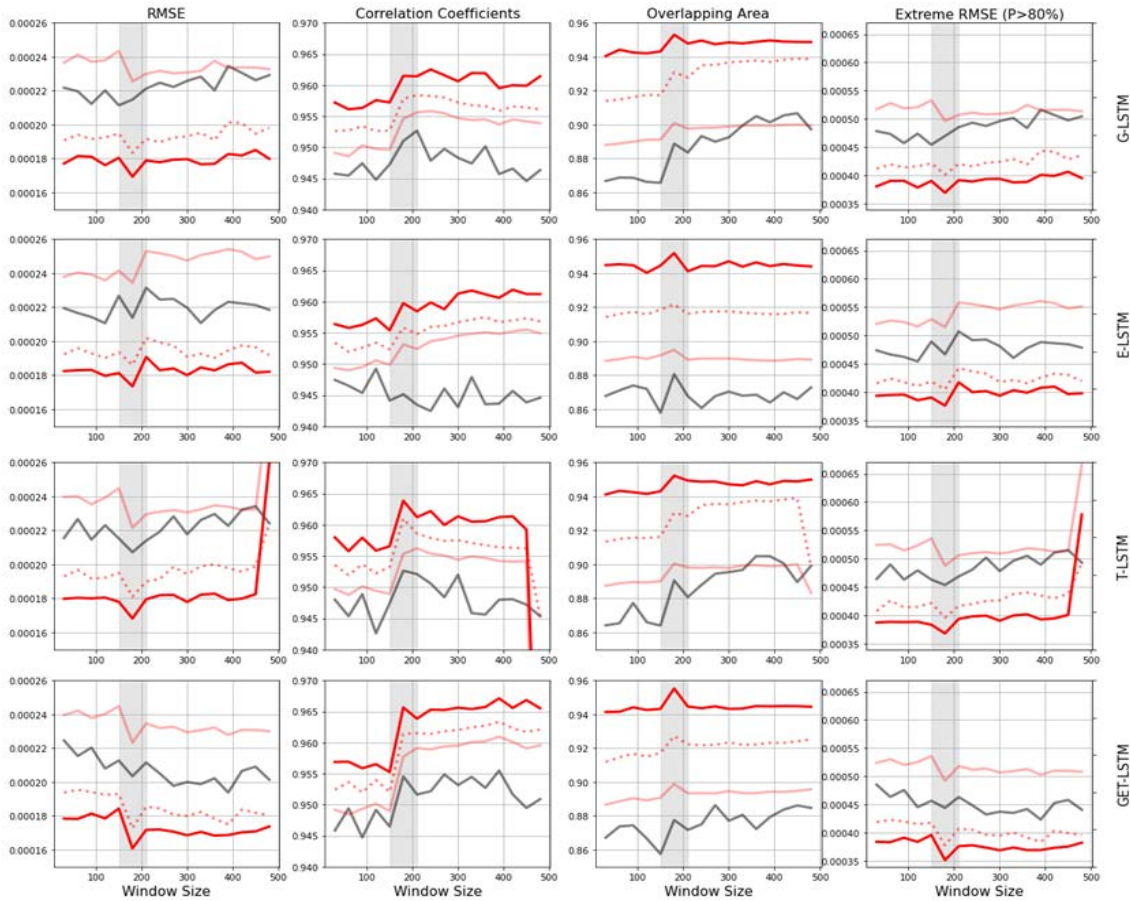
According to Fig. 5, the performance at  $W = 180$  (trading days) outperforms the other cases in general. This aspect means that variations of log-return for the S&P 500 dataset

are more influenced by the relatively recent observation than by historical information longer than about 180 trading days. We present the best performance (for each model and error metric) and corresponding window size in Table 4. For a more detailed evaluation of the proposed model, we focus on the case of  $W = 180$  in the next subsection.

### 2) ANALYSIS FOR WINDOW SIZE OF 180 TRADING DAYS

In this section, fixing the window size to be 180 trading days, we evaluate the proposed hybrid model against the native model in terms of three measurements: 1) Mean performance over all values of hyperparameter  $\alpha$ , 2) Mean performance for  $\alpha$  in the positive section, 3) Best performance corresponding to a specific  $\alpha$ .

In Fig. 6, we compare the performances according to GARCH models, error metrics and hyperparameter  $\alpha$ ,  $0 < \alpha < 1$ . For RMSE, the performance gains of G-LSTM, E-LSTM and T-LSTM against P-LSTM are 6.9%, 7.3% and 10.8%, respectively. Furthermore, the performance gain of GET-LSTM against P-LSTM is 12.5%. This result shows that the addition of informations obtained from multiple GARCH models to the explanatory input data improves performances even more compared to the case of adding single GARCH model information. In the VU-applied case, we find that the performance gain is outstanding compared to the existing model (non-VU), the corresponding results are presented in Table 5 and Table 6. In general, VU-GARCH models based LSTM (VU-G-LSTM, VU-E-LSTM, VU-T-LSTM and VU-GET-LSTM) show better performance in the positive section and lower performance outside the positive section compared



**FIGURE 5.** Intercomparison of performances according to GARCH models, error metrics, and window size ( $x$ -axis in each panel). Black indicates the performance obtained by the native strategy (Non-VU). Solid pink, dotted pink, and red denote VU-performances on the whole  $\alpha$ -regions, on the positive section, and for a specific  $\alpha$  value (best case), respectively. Gray shaded regions indicate the vicinity of window size 180.

**TABLE 4.** Window size corresponding to the best performance for each error metric.

	RMSE	CC	Overlapping Area	Extreme RMSE
	G-LSTM / E-LSTM / T-LSTM / GET-LSTM			
Native case	150 / 120 / 180 / 390	210 / 120 / 180 / 390	450 / 180 / 360 / 270	150 / 120 / 180 / 390
Mean performance for all $\alpha$	180 / 180 / 180 / 180	240 / 450 / 210 / 390	180 / 180 / 180 / 180	180 / 180 / 180 / 180
Mean performance in positive section	180 / 180 / 180 / 180	210 / 360 / 180 / 390	450 / 180 / 450 / 180	180 / 180 / 180 / 180
Best performance among all $\alpha$	180 / 180 / 180 / 180	240 / 420 / 180 / 390	180 / 180 / 180 / 180	180 / 180 / 180 / 180

**TABLE 5.** RMSE percentage gains against P-LSTM in various  $\alpha$ -regions.

Model	Mean over all $\alpha$	On Positive section	The best case
VU-P-LSTM	2.8	20.0	25.6
VU-G-LSTM	3.1	21.1	27.2
VU-E-LSTM	-0.7	20.0	25.4
VU-T-LSTM	4.8	22.1	27.7
VU-GET-LSTM	4.0	25.7	30.9

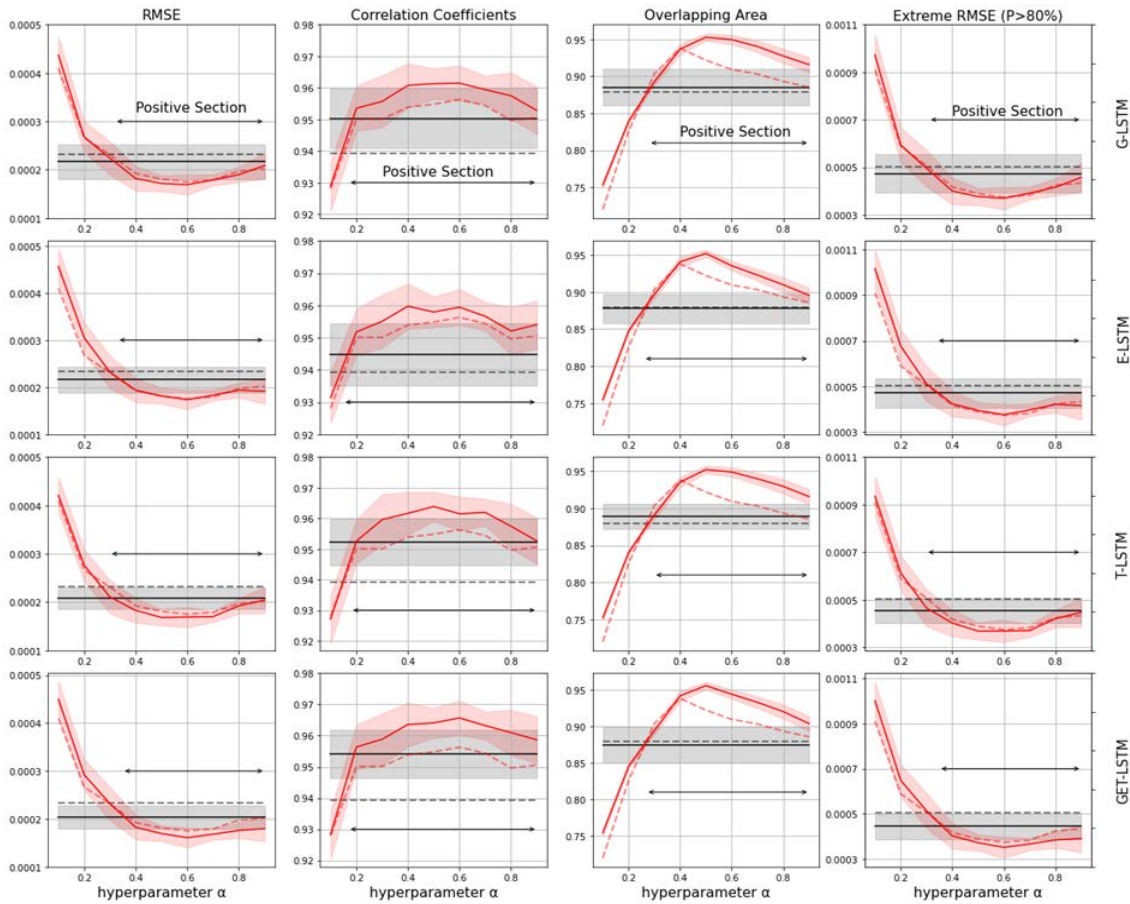
**TABLE 6.** RMSE percentage gains against VU-P-LSTM in various  $\alpha$ -regions.

Model	Mean over all $\alpha$	On Positive section	The best case
VU-G-LSTM	0.2	1.7	2.1
VU-E-LSTM	-3.7	0.3	-0.3
VU-T-LSTM	2.0	3.0	2.7
VU-GET-LSTM	1.2	7.5	7.0

to VU-P-LSTM (Fig. 6 and Table 6). The performances in terms of other error metrics (CC, overlapping area, and RMSE on extreme region) show similar trends to RMSE. The performance gains of VU-GET-LSTM against other models are presented in Table 7.

The comparative performances of all experiments are presented in Fig. 7. It can be found that if  $\alpha$  is restricted in the positive section then the proposed model outperforms the existing model. It is noteworthy that VU-P-LSTM outperforms G-LSTM, E-LSTM, T-LSTM and GET-LSTM, which





**FIGURE 6.** Intercomparison of performances according to GARCH models, error metrics, and the hyperparameter  $\alpha$  (x-axis in each panel) for the fixed window size 180. Dotted black, solid black, dotted red, and solid red represent P-LSTM, GARCH (types)-LSTM, VU-P-LSTM, and VU-GARCH (types)-LSTM models, respectively. Bidirectional arrows indicate the positive section of  $\alpha$ .

**TABLE 7.** Performance percentage gains of VU-GET-LSTM against other models in various  $\alpha$ -regions.

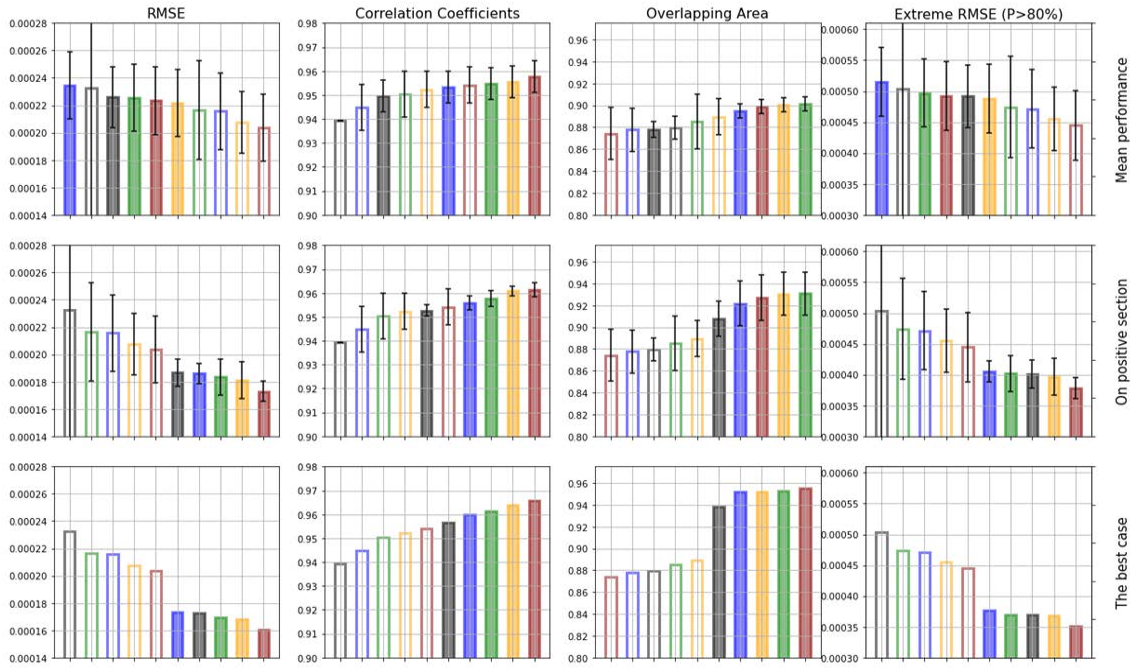
Error metric	Model	Mean over all $\alpha$	On Positive section	The best case
RMSE	P-LSTM	4.0	25.7	30.9
	VU-P-LSTM	1.2	7.5	7.0
	GET-LSTM	-9.7	15.2	21.0
CC	P-LSTM	2.0	2.4	2.8
	VU-P-LSTM	0.8	0.9	1.0
	GET-LSTM	0.4	0.8	1.2
Area	P-LSTM	2.2	5.4	8.6
	VU-P-LSTM	2.4	2.2	1.9
	GET-LSTM	2.8	6.1	9.3
Extreme	P-LSTM	2.2	24.9	30.1
	VU-P-LSTM	-0.1	5.7	4.8
	GET-LSTM	-10.7	15.0	20.9

implies performance gain from VU strategy is greater than the gain from GARCH model for all error metrics. In addition, it is not able to find that VU-G-LSTM, VU-E-LSTM, and VU-T-LSTM have the model's superiority. However, the proposed VU-GET-LSTM model outperforms the other models.

Besides the evaluation with respect to the overlapping area (over all region of label PDF domain), it is necessary to investigate the overlapping region near the region because most prediction and label are concentrated near zero. Fig. 8

shows comparative distribution (near zero) of P-LSTM, GET-LSTM, and VU-GET-LSTM. It can be confirmed that the distribution of prediction obtained from VU-GET-LSTM model become closer to that of label (realized volatility) compared to other native models.

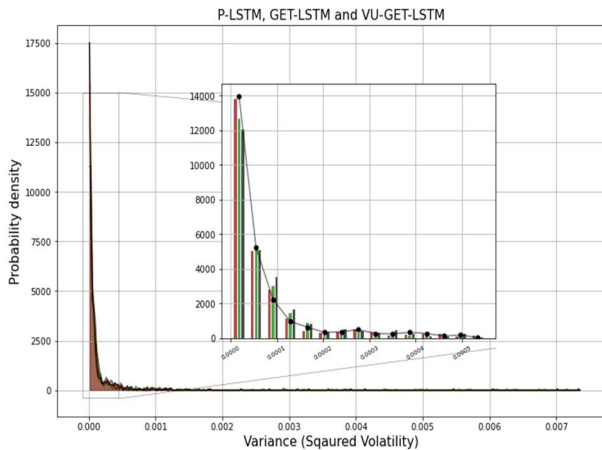
The proposed hybrid models do not show better prediction performance for the hyperparameter  $\alpha$  near zero (about  $0 < \alpha < 0.3$ ) compared to models with native strategies. However, based on the facts that the positive sections occupy



**FIGURE 7.** Performance comparisons among 10 models on various  $\alpha$ -regions. Black, green, blue, orange, and darkred represent P-LSTM, G-LSTM, E-LSTM, T-LSTM, and GET-LSTM, respectively. Edge colored and full colored bars indicate non-VU and VU-applied experiments, respectively.

**TABLE 8.** Sum of all absolute valued entries in  $19 \times 19$  correlation coefficients matrix for each  $\alpha$  in unit interval.

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
CC sum	125.0	128.7	131.2	132.1	131.1	128.6	124.7	120.1	115.3	110.4



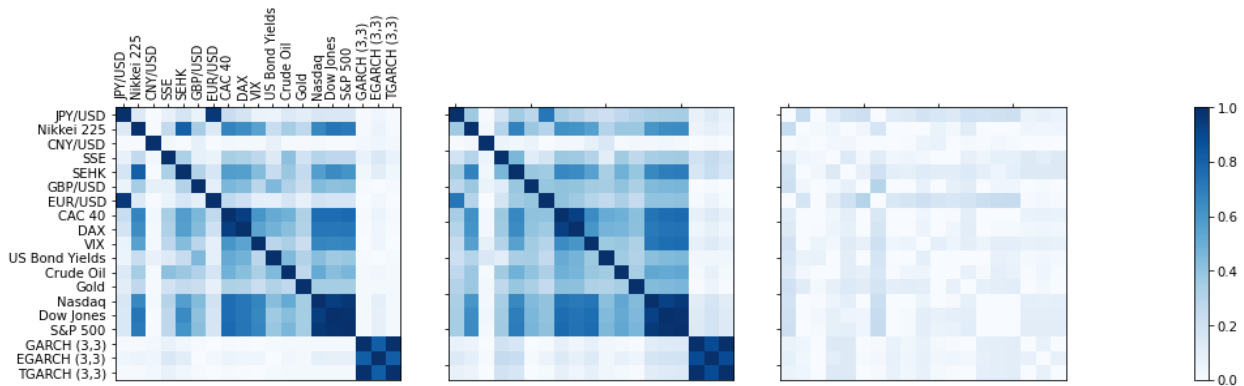
**FIGURE 8.** Performance comparison in the distribution (PDF) near zero between P-LSTM (black), GET-LSTM (green), and VU-GET-LSTM (darkred). Solid curve in zoom-in figure represents the label.

wider domain than the non-positive sections for all error metrics, we conclude that the models with VU outperform the native models. Based on these results, we now summarize and discuss our experimental results. First, the positive section occupies more than 70% of interval (0, 1) for each error metric and GARCH integrated model. Also the intersection of

**TABLE 9.** Lengths of the confidence interval (C-I) for all experiments. The values represent the mean (over all prediction) of  $2K$  such that  $P(\overline{\text{pred}} - K \leq \text{label} \leq \overline{\text{pred}} + K) = 0.95$  with respect to the standard normal distribution where  $\overline{\text{pred}}$  denotes the sample mean of predictions.

Model	C-I	Model	C-I (C-I on positive section)
P-LSTM	0.208	VU-P-LSTM	0.129 (0.075)
G-LSTM	0.170	VU-G-LSTM	0.125 (0.068)
E-LSTM	0.197	VU-E-LSTM	0.130 (0.071)
T-LSTM	0.176	VU-T-LSTM	0.124 (0.069)
GET-LSTM	0.181	VU-GET-LSTM	0.125 (0.079)

16 positive sections in Fig. 6 contains (0.3, 1). That is, excluding root type functions that increase rapidly near zero (the functions correspond to  $g(x) = x^\alpha$  in (18) for  $0 < \alpha < 0.3$ ), the hybrid models with VU outperform the native models for most root type functions. Second, improvements in RMSE on extreme region show that VU is an appropriate strategy for predicting time-series that involve large volatilities. It is expected that VU can be well applied in regression analysis to detect abnormal events occurred in flood, financial crisis prediction, and so on. Third, the positive section in error metric CC ( $0.1 < \alpha < 1$  in Fig. 6) occupies the widest region compared to other error metrics. It can be estimated that VU enhances prediction performance by making similar moving trend of label and prediction. Meanwhile, it is found that VU increases the sum of all correlation coefficients between all



**FIGURE 9.** Absolute valued correlation coefficients inter-comparison for 19 inputs of GET-LSTM (or VU-GET-LSTM) in case of window size 180. Left and middle correspond to correlation coefficients (between indices) of the original and VU applied volatility, respectively, and right represents their difference (subtract middle from left).

input indices (Fig. 9 and Table 8). However, higher correlation coefficients of input do not guarantee better performances. For example, correlation coefficients correspond to  $\alpha=0.2$  and  $\alpha=0.6$  are similar (128.7 and 128.6, respectively in Table 8), however, according to the Fig. 6, the performance at  $\alpha=0.6$  is better than at  $\alpha=0.2$  for all error metrics. It is noteworthy that  $\alpha=1$  in Table 8 corresponds to the native (Non-VU) case. Fourth, as  $\alpha$  approaches to 1 from the left side, the performance of proposed algorithm is reduced to that of the native strategy. It is speculated that a filtering approach using concave-up functions (instead of convex functions) is an appropriate filtering strategy. Fifth, Table 9 shows that when the VU strategy is applied, the length of the confidence interval of the prediction is shorter (rather than non-VU case), and it is also shorter on the positive section than on all  $\alpha$ -regions. This aspect demonstrates that the results obtained on the positive section are more reliable. Sixth, in Fig. 4, it is found that input data induced by various GARCH models have outliers which usually affect model performance. However, the LSTM is used as the implementation model in this study, and the sigmoid function and the tangent hyperbolic function are located in the LSTM cell (from (12) to (17)). The functions have the squashing effect, and prevent outliers of input data from having an unduly large and disturbing effect on the learning [41].

## V. CONCLUSION

The distributions of various financial data including the volatility of S&P 500 index are extremely biased to the left and are concentrated near zero. This aspect can cause low prediction performance on the right part of distribution (abnormal events) as well as on whole probability density domain. Meanwhile, incorporation of various explanatory variables with GARCH-type models has been reported to improve performance by adding reliable information to input data in neural networks. To overcome the challenge of the former with the advantage of the latter, we develop the hybrid model using VU strategy to alleviate the biased property of

financial data and to obtain better performance gain. Specifically, we use the outputs of GARCH-type models as input as well as the explanatory variables and combine VU strategy. The concave function  $x^\alpha$ ,  $0 < \alpha < 1$  is employed in VU strategy, and it plays a crucial role in mitigating biased property of volatility.

In general, it is known that the current S&P500 index is highly affected by relatively recent observations rather than historical data far from the present time. Thus, we adopt the rolling window method to obtain input of sequential structure. From the proposed model, we obtain several meaningful results. Firstly, performances at a specific window size (180 trading days in this experiment) are superior compared to performances at other window size for each error metric. Secondly, performance gains obtained from the proposed model are conspicuous compared to the gains obtained from GARCH based models as well as the native strategy. Especially, performance gains are noteworthy when the hyperparameter  $\alpha$  is restricted on the positive section that occupies more than 70% of the interval (0, 1). Thirdly, the proposed model enhances performance on the right region as well as other region of probability density. This shows that the proposed algorithm is an appropriate approach to predict abnormal events in financial market. Finally, we expect that modifications of the concave function will improve prediction performance further, and this will be considered in the future work.

## REFERENCES

- [1] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica: J. Econ. Soc.*, vol. 50, no. 4, pp. 987–1007, 1982.
- [2] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econ.*, vol. 31, no. 3, pp. 307–327, 1986.
- [3] D. B. Nelson, "Conditional heteroskedasticity in asset returns: A new approach," *Econometrica: J. Econ. Soc.*, vol. 59, no. 2, pp. 347–370, 1991.
- [4] J.-M. Zakoian, "Threshold heteroskedastic models," *J. Econ. Dyn. Control*, vol. 18, no. 5, pp. 931–955, Sep. 1994.
- [5] C. Klüppelberg, A. Lindner, and R. Maller, "A continuous-time GARCH process driven by a Lévy process: Stationarity and second-order behaviour," *J. Appl. Probab.*, vol. 41, no. 3, pp. 601–622, 2004.



- [6] D. Li, X. Zhang, K. Zhu, and S. Ling, "The ZD-GARCH model: A new way to study heteroscedasticity," *J. Econ.*, vol. 202, no. 1, pp. 1–17, Jan. 2018.
- [7] P. Otto, W. Schmid, and R. Garthoff, "Generalised spatial and spatiotemporal autoregressive conditional heteroscedasticity," *Spatial Statist.*, vol. 26, pp. 125–145, Aug. 2018.
- [8] F. Zhu, W. Quan, Z. Zheng, and S. Wan, "A Bayesian learning method for financial time-series analysis," *IEEE Access*, vol. 6, pp. 38959–38966, 2018.
- [9] S. A. Hamid and Z. Iqbal, "Using neural networks for forecasting volatility of S&P 500 Index futures prices," *J. Bus. Res.*, vol. 57, no. 10, pp. 1116–1125, 2004.
- [10] G. Sermpinis, J. Laws, and C. L. Dunis, "Modelling and trading the realised volatility of the FTSE100 futures with higher order neural networks," *Eur. J. Finance*, vol. 19, no. 3, pp. 165–179, Mar. 2013.
- [11] Q. Xu, L. Wang, C. Jiang, and X. Zhang, "A novel UMIDAS-SVQR model with mixed frequency investor sentiment for predicting stock market volatility," *Expert Syst. Appl.*, vol. 132, pp. 12–27, Oct. 2019.
- [12] E. Ramos-Pérez, P. J. Alonso-González, and J. J. Núñez-Velázquez, "Forecasting volatility with a stacked model based on a hybridized artificial neural network," *Expert Syst. Appl.*, vol. 129, pp. 1–9, Sep. 2019.
- [13] 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.
- [14] Y. Wang, H. Liu, Q. Guo, S. Xie, and X. Zhang, "Stock volatility prediction by hybrid neural network," *IEEE Access*, vol. 7, pp. 154524–154534, 2019.
- [15] H. Muzhou, L. Taohua, Y. Yunlei, Z. Hao, L. Hongjuan, Y. Xiugui, and L. Xinge, "A new hybrid constructive neural network method for impacting and its application on tungsten price prediction," *Int. J. Speech Technol.*, vol. 47, no. 1, pp. 28–43, Jul. 2017.
- [16] H. Niu, K. Xu, and W. Wang, "A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network," *Appl. Intell.*, vol. 50, no. 12, pp. 4296–4309, Jul. 2020.
- [17] Y. Xu, C. Yang, S. Peng, and Y. Nojima, "A hybrid two-stage financial stock forecasting algorithm based on clustering and ensemble learning," *Int. J. Speech Technol.*, vol. 50, no. 11, pp. 3852–3867, Nov. 2020.
- [18] A. Episcopos and J. Davis, "Predicting returns on Canadian exchange rates with artificial neural networks and EGARCH-M models," *Neural Comput. Appl.*, vol. 4, no. 3, pp. 168–174, Sep. 1996.
- [19] M. A. de Oliveira, "The influence of ARIMA-GARCH parameters in feed forward neural networks prediction," *Neural Comput. Appl.*, vol. 20, no. 5, pp. 687–701, Jul. 2011.
- [20] W. Kristjanpoller and M. C. Minutolo, "Gold price volatility: A forecasting approach using the artificial neural network-GARCH model," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 7245–7251, 2015.
- [21] W. Kristjanpoller and M. C. Minutolo, "Forecasting volatility of oil price using an artificial neural network-GARCH model," *Expert Syst. Appl.*, vol. 65, pp. 233–241, Dec. 2016.
- [22] S. Lahmiri, "Modeling and predicting historical volatility in exchange rate markets," *Phys. A, Stat. Mech. Appl.*, vol. 471, pp. 387–395, Apr. 2017.
- [23] W. Kristjanpoller and E. Hernández, "Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors," *Expert Syst. Appl.*, vol. 84, pp. 290–300, Oct. 2017.
- [24] E. Hajizadeh, M. Mahootchi, A. Esfahanipour, and M. M. Kh, "A new NN-PSO hybrid model for forecasting Euro/Dollar exchange rate volatility," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 2063–2071, 2019.
- [25] M. Seo, S. Lee, and G. Kim, "Forecasting the volatility of stock market index using the hybrid models with Google domestic trends," *Fluctuation Noise Lett.*, vol. 18, no. 1, Mar. 2019, Art. no. 1950006.
- [26] M. Seo and G. Kim, "Hybrid forecasting models based on the neural networks for the volatility of bitcoin," *Appl. Sci.*, vol. 10, no. 14, p. 4768, Jul. 2020.
- [27] R. Liao, W. Yamaka, and S. Sriboonchitta, "Exchange rate volatility forecasting by hybrid neural network Markov switching Beta-t-EGARCH," *IEEE Access*, vol. 8, pp. 207563–207574, 2020.
- [28] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 240–254, Mar. 1994.
- [29] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.
- [30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [31] E. Hajizadeh, A. Seifi, M. F. Zarandi, and I. Turksen, "A hybrid modeling approach for forecasting the volatility of S&P 500 index return," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 431–436, 2012.
- [32] D. Pradeepkumar and V. Ravi, "Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network," *Appl. Soft Comput.*, vol. 58, pp. 35–52, Sep. 2017.
- [33] G. W. Schwert, "Why does stock market volatility change over time?" *J. Finance*, vol. 44, no. 5, pp. 1115–1153, Dec. 1989.
- [34] W. Kristjanpoller, A. Fadic, and M. C. Minutolo, "Volatility forecast using hybrid neural network models," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2437–2442, Apr. 2014.
- [35] 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 Syst. Appl.*, vol. 103, pp. 25–37, Aug. 2018.
- [36] L. Luo, S. Pairote, and R. Chatpatanasiri, "GARCH-type forecasting models for volatility of stock market and MCS test," *Commun. Statist.-Simul. Comput.*, vol. 46, no. 7, pp. 5303–5312, Aug. 2017.
- [37] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.
- [38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [39] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, 1999.
- [40] T. Tieleman and G. Hinton, "Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.
- [41] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *Int. J. Forecasting*, vol. 36, no. 1, pp. 75–85, Jan. 2020.



**EUNHO KOO** received the Ph.D. degree in applied mathematics from Yonsei University, South Korea, in 2017. He was a Researcher at the Institute of Industrial Science, University of Tokyo, Japan, from 2017 to 2019, and a Research Professor at the Center for Mathematical Analysis and Computation, Yonsei University, from 2019 to 2022. He is currently working as a Researcher with the Center for AI and Natural Sciences, Korea Institute for Advanced Study, South Korea. His research interests include statistical machine learning and prediction of time series based on neural networks.



**GEONWOO KIM** received the M.S. degree in statistics and the Ph.D. degree in mathematical finance from Yonsei University, South Korea, in 2008 and 2014, respectively. He is currently an Assistant Professor with the School of Liberal Arts, Seoul National University of Science and Technology, South Korea. His research interests include computational finance, derivatives pricing, applications of derivatives for finance, and forecasting of financial time series using machine learning approach.

• • •