GARCH-LSTM for Stock Price Prediction Using Sentiment Analysis

Guanghui Chen, Li Wang

Abstract—**This research introduces a new hybrid model called the SF-GARCH-LSTM model, designed for stock price forecasting. It combines sentiment analysis, long short-term memory networks (LSTM), and generalized autoregressive conditional heteroskedasticity (GARCH) models. The model first utilizes Bidirectional Encoder Representations from Transformers (BERT) to classify stock review titles into positive and negative sentiments, generating a sentiment factor (SF). Then, GARCH parameters are calculated using multiple GARCH models based on historical stock prices. Finally, the LSTM model combines stock price data, sentiment factors, and GARCH parameters to predict future stock prices. Experiment results demonstrate that the new proposed SF-GARCH-LSTM model significantly improves prediction accuracy compared to LSTM, SF-LSTM, and GARCH-LSTM models, highlighting the importance of incorporating sentiment information into financial forecasting.**

Index Terms—**Sentiment analysis, BERT, Hybrid model, GARCH, LSTM, SF-GARCH-LSTM.**

I. INTRODUCTION

Predicting stock prices has long been a critical focus for both investors and researchers. Accurate stock price prediction is essential for making informed investment decisions and managing risks. Prior studies has highlighted the influence of various factors on stock prices, including fundamental data, technical indicators, and market sentiment. Emotional factors have proven to be significant in this context. With the rise of social media,, investors now have greater opportunities for interaction and communication, leading to more frequent exchanges of opinions and emotional expressions. These sentiments, shared on social networks, can influence individual investor behavior and subsequently impact the broader stock market. For example, [1] found that daily fluctuations in public mood significantly correlate with daily movements in the closing prices of the Dow Jones Industrial Average. [2] integrated news analysis with a GARCH-jump model for stock price prediction. [3] demonstrated that sentiment expressed on StockTwits has predictive power for short-term stock market movements. [4] analyzed the correlations between Bitcoin market metrics and Twitter posts that express emotional signals regarding Bitcoin. [5] and [6] constructed a sentiment index that reflects changes in investor sentiment, revealing that fluctuations in this index affect not only individual stocks but also the overall stock market. Additionally, [7] showed that incorporating positive and negative emotions related to stock prices as input factors can improve the accuracy of stock price predictions.

Relying solely on investor sentiment for stock price prediction presents significant limitations. To address this, it is crucial to integrate sentiment analysis with stock price data. Considering the various factors that affect stock prices, a comprehensive understanding of volatility trends is essential for making informed predictions.

GARCH models are widely recognized for their effectiveness in modeling financial time series exhibiting fluctuating volatility and clustering effects. These models not only allow for the estimation of conditional volatilities but also adeptly capture the inherent volatility patterns and underlying financial dynamics within the data. Numerous studies have highlighted the efficacy of GARCH models in this domain (e.g., [8], [9], [10], [11]).

Traditional econometric models, such as autoregressive integrated moving average (ARIMA) model [12], are effective for short-term predictions but often struggle to capture nonlinear relationships. However, advancements in deep learning have increasingly addressed these limitations. Notable architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been explored in [13]. While RNNs are widely used for time series forecasting due to their effectiveness, they face challenges such as vanishing or exploding gradients, particularly with long data sequences. To overcome these issues, variants such as long short-term memory (LSTM) networks [14] and gated recurrent units (GRU) [15] have gained prominence, offering robust solutions for managing long-term dependencies and handling the nonlinear characteristics of data. In [16], the performance of three methods - RNN, LSTM, and GRU - was evaluated for stock price prediction, while [17] highlighted the use of search economics methods to optimize LSTM parameters.

Furthermore, extensive research suggests that a single model often fails to capture both linear and nonlinear patterns in time series effectively. As a result, hybrid models have been developed as an effective solution, merging the advantages of various forecasting techniques to enhance both interpretability and accuracy. In deep learning, such hybrid approaches are increasingly prevalent in price forecasting. For instance, [18] combined LSTM with GARCH models to predict garlic prices, while [19] integrated LSTM with ARIMA models for stock price forecasting.

In this study, we seek to improve the accuracy of stock price predictions by integrating LSTM, sentiment analysis, and GARCH models into a unified predictive framework. Motivated by prior studies, our approach harnesses the strengths of deep learning and traditional econometric models to provide more reliable forecasts. Our methodology begins with the collection of stock-related titles and comments,

Manuscript received August 5, 2024; revised May 27, 2025.

Guanghui Chen is a postgraduate student at the School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Macau 999078, China. (e-mail: 1040701123@qq.com)

Li Wang is an assistant professor at the School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Macau 999078, China. (corresponding author, e-mail: liwang-fi@must.edu.mo)

which we process using BERT for fine-tuning classification to assign positive and negative sentiment labels. This step enables us to calculate a sentiment factor (SF) that reflects investor sentiment. Next, we utilize historical stock price data as the primary dataset for our analysis. Recognizing the common heteroscedastic nature of stock prices and other financial instruments, we employ GARCH-type models to estimate conditional volatilities and residuals for each trading day. By combining the stock price data, GARCH parameters, and sentiment factors, we construct the input features for the LSTM model. The output of this hybrid approach, which we refer to as the SF-GARCH-LSTM model, is a prediction of stock prices. This integrated framework aims to offer a more thorough understanding of the factors affecting stock prices while improving prediction accuracy. By combining sentiment analysis, econometric modeling, and deep learning, our approach seeks to connect traditional financial models and modern machine learning techniques.

The organization of the paper is as follows: Section II reviews related literature. Section III outlines the relevant methods and introduces the proposed SF-GARCH-LSTM model. Section IV examines the experimental results, while Section V concludes the study.

II. Related Work

Forecasting stock prices in time series analysis remains a pivotal area of research. To improve prediction accuracy and address the limitations of single forecasting models, researchers have developed a variety of advanced techniques, resulting in an increasing use of hybrid models. Recently, the incorporation of machine learning methods into financial time series analysis has emerged as a beneficial addition to traditional models like ARIMA and GARCH. These machine learning approaches excel at handling the complexities of high-dimensional and nonlinear data, making them particularly effective in the dynamic and complex stock market environment.

The analysis of emotions in financial contexts has garnered significant attention from researchers in recent years. For example, [20] demonstrated that the BERT model significantly outperforms both LSTM and support vector machine methods in accurately analyzing investor sentiment. Similarly, [21] developed a sentiment analysis model tailored to Chinese stock reviews using BERT, showcasing its effectiveness in handling complex linguistic data. Building on this, [22] combined BERT, bidirectional LSTM, and multi-head attention mechanisms for sentiment analysis, while [23] utilized financial BERT to calculate sentiment scores from summary text data for predicting the S&P 500 index. In addition, [24] integrated historical stock data with Twitter sentiment classification to improve the accuracy of stock price forecasts. Meanwhile, [25] employed a CNN-based sentiment analysis model to classify text data from online social networks, combining daily sentiment scores with stock prices using LSTM for more precise predictions.

In our work, we leverage BERT to analyze sentiment by categorizing investors' comment headlines as either positive or negative, facilitating the computation of the sentiment factor (SF). This approach improves the incorporation of sentiment analysis into stock price prediction frameworks, offering a deeper insight into market dynamics.

In related studies, [27] introduced the use of variational mode decomposition to separate carbon prices into high-frequency and low-frequency components. Similarly, [28] combined LSTM with GARCH-type models to predict stock price index volatility. Additionally, [29] proposed using GARCH model outputs as inputs for neural networks to enhance predictive performance. [30] developed a hybrid model combining GARCH with a distribution manipulation strategy based on LSTM for predicting stock market volatility, while [31] applied GARCH and LSTM to forecast commodity market returns volatility. Moreover, [18] utilized a combination of GARCH and LSTM to predict garlic prices.

We propose a hybrid model that integrates sentiment analysis, GARCH type models, and LSTM to enhance stock price prediction. The improved predictive capability of this hybrid model stems from three key factors. First, sentiment factors, derived from sentiment analysis, reflect the emotions and psychological tendencies of market participants, providing valuable insights into market sentiment and further enhancing the accuracy of stock price predictions. Second, econometric models like GARCH are highly effective at capturing the volatile features of price series, specifically addressing the heteroskedasticity inherent in financial time series and extracting critical volatility characteristics of stock prices. Third, machine learning models excel at identifying nonlinear and unpredictable patterns within price sequences, as highlighted by [26].

To date, no research has combined sentiment factors, GARCH models, and LSTM in the manner proposed here. Given the well-established correlation between investor sentiment and stock price movements, our approach begins by collecting sentiment-related text comments on stocks. We apply BERT for text classification, enabling the computation of the sentiment factor (SF). Next, considering the heteroskedastic nature of stock price series, we use the GARCH model to estimate conditional volatilities and residuals. Finally, we integrate stock price data, sentiment factors, and GARCH parameters as inputs to the LSTM model, effectively addressing the non-smooth and nonlinear characteristics of the data.

We call this comprehensive hybrid model the SF-GARCH-LSTM model, which aims to enhance the accuracy of stock price forecasting by utilizing the complementary strengths of sentiment analysis, econometric modeling, and deep learning.

III. Methods

This section outlines the proposed method. First, the BERT model is employed to classify headline information into positive and negative categories, allowing for the calculation of the sentiment factor (SF). Next, the GARCH model is utilized to extract volatility characteristics from stock price data. Following this, the LSTM model is applied to identify essential dependencies in time series data, facilitating more accurate stock price predictions. Finally, we introduce the SF-GARCH-LSTM algorithm, which integrates these components. The key technologies and algorithms underpinning this approach are discussed in detail below.

A. Bidirectional Encoder Representations from Transformers (BERT)

1) BERT: The BERT model, introduced by [32], is a pre-trained language model built upon the encoder architecture of the Transformer framework. As depicted in Fig. 1 from [33], the Transformer comprises an encoder (left) and a decoder (right), with BERT exclusively utilizing the encoder component. A key feature of BERT is its implementation of multi-head attention, which enables the model to capture contextual relationships within text. The calculation of multi-head attention is defined in Eq. (1) ([33]):

$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (1)$$

Queries are consolidated into matrix Q, keys are grouped into matrix K and values are grouped into matrix V. The dot-product attention mechanism enhances both speed and memory efficiency by leveraging optimized matrix multiplication code for implementation. To prevent excessively large values during the computation, the dot product is scaled by $\frac{1}{\sqrt{d_k}}$.

However, the encoding methods for inputs differ between Transformer and BERT. The Transformer uses positional encoding, specifically employing sine and cosine functions as defined in Eq. (2) and Eq. (3):

$$PE_{(pos,2i)}=\sin\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right), \quad (2)$$

$$PE_{(pos,2i+1)}=\cos\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right), \quad (3)$$

where pos represents the position, and i denotes the dimension. In contrast, BERT incorporates segment embeddings to address issues related to sentence pairs. As shown in Fig. 2 ([32]), each input sequence begins with a special classification token, [CLS]. For sentence pairs, both sentences are merged into a single sequence, with the [SEP] token acting as a delimiter between them. The [SEP] token not only distinguishes whether a segment belongs to sentence A or B but also indicates its specific position within the sequence. Additionally, [SEP] signifies the end of a sentence.

2) Pre-training for BERT: Mask Language Model (MLM) + Next Sentence Prediction (NSP): BERT utilizes an unsupervised objective function, specifically autoencoding (AE), which involves predicting and reconstructing the original data from corrupted input data using contextual information. A key component of BERT's pre-training is the Masked Language Model (MLM) task, a novel prediction objective designed to learn contextual word representations that align with natural language patterns. In this task, certain words in the input text are fully masked, allowing the model to leverage surrounding context to predict the masked words effectively. During training, 15% of the words in the input are randomly selected as masked regions to optimize learning. Of these selected words, 80% are replaced with a mask token, 10% are swapped with random alternative words, and the remaining 10% remain unchanged.

The Next Sentence Prediction (NSP) task, on the other hand, involves analyzing pairs of sentences referred to as Sentence A and Sentence B. In some cases, Sentence B follows Sentence A sequentially and is labeled as "IsNext." In other cases, Sentence B is randomly selected from the corpus and labeled as "NotNext." The model's objective is to determine whether Sentence B logically and sequentially follows Sentence A.

These two tasks, MLM and NSP, work together to enable BERT to capture both contextual and relational information during pre-training.

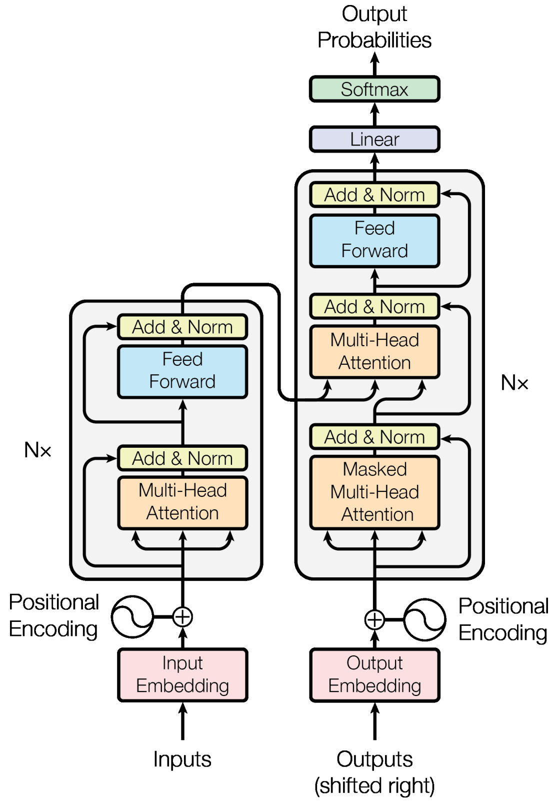

Fig. 1: Transformer model architecture: the left side forms the basis of BERT.

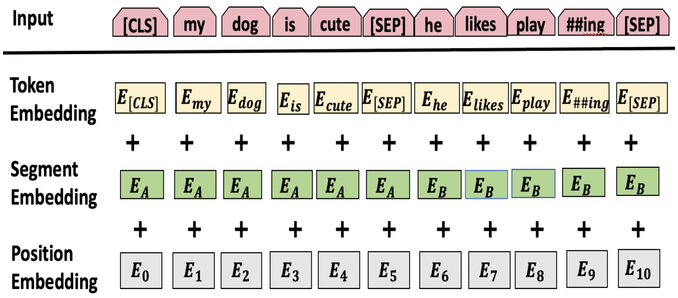

Fig. 2: BERT input representation: combination of token, segment, and positional embeddings.

Comment Title	Label
哈哈，稳了，买入一点点。	1
Haha, steady, buy a little bit.	positive
波段回升行情已启动，不要犹豫。	1
The band rebound has started, don't hesitate.	positive
今天又跳水，唉！	0
Diving again today, alas!	negative
易跌难涨。	0
It is easy to fall but difficult to rise.	negative

Fig. 3: Examples of stock review titles with sentiment labels

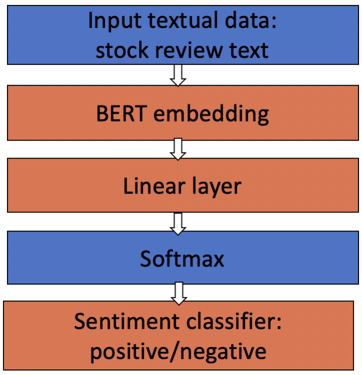

Fig. 4: BERT model training process flowchart

3) Sentiment classification using BERT model: BERT is a pre-trained language model created by Google, offering a collection of nine pre-trained models: six English models, two multilingual models, and one Chinese model. Since this study focuses on Chinese text, the specific model used is `Bert_Base_Chinese`. This model supports both simplified and traditional Chinese and features a configuration of 12 layers, 768 hidden units, 12 attention heads, and 110 million parameters. While the pre-trained BERT model can be directly applied to classification tasks, its initial accuracy is relatively low, as demonstrated in subsequent test results. To improve performance, additional pre-training on task-specific datasets is required.

In this study, the labeled dataset is categorized into positive and negative instances, as illustrated in Fig. 3. The model is then retrained using the original pre-trained target data, such as the masked language model, to adapt its parameters to the target dataset. Finally, the classification performance is fine-tuned by optimizing both the training process and classification output, aiming to derive the most effective model for the data. Once fine-tuned, the enhanced model is applied to the complete dataset used in this study.

Fig. 4 illustrates the BERT model's classification process. The input consists of emotionally labeled sentences (positive and negative). First, the text is processed by BERT to generate vector representations. These vectors are subsequently passed through a linear layer, followed by a softmax activation function, which produces the classification result as either positive or negative sentiment. By following the BERT model training process outlined in Fig. 4, a more refined model is obtained, which is then applied to the entire dataset. The daily sentiment factor score is calculated using the following equation:

$$\mathrm{SF}^{t}=\frac{\mathrm{num}_{+}^{t}-\mathrm{num}_{-}^{t}}{\mathrm{num}^{t}}, \tag{4}$$

where t represents the date, num^t is the total number of posts on that day, and num_+^t and num_-^t denote the number of posts classified as positive and negative, respectively, on date t.

4) Text classification evaluation: The BERT model is employed to classify sentiments in stock review texts, a common task in classification. Precision, recall, and F1-score metrics are used for evaluating classification performance. The binary classification problem is categorized into four scenarios based on changes in both the model's predictions and the actual outcomes. In the context of classification metrics, TP (True Positive) refers to instances accurately identified as belonging to the positive class. Conversely, FN (False Negative) refers to instances that are incorrectly classified as negative even though they belong to the positive class. FP (False Positive) occurs when an instance is wrongly labeled as positive, despite being part of the negative class. Lastly, TN (True Negative) indicates instances correctly identified as belonging to the negative class. These terms are crucial for evaluating the accuracy of classification models.

Precision Rate: Also referred to as the accuracy rate, measures the ratio of true positive predictions to the total number of instances classified as positive. This is expressed in Eq. (5):

$$\mathrm{P}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{5}$$

Recall Rate: This metric represents the proportion of correctly predicted positive instances to all actual positive instances, as shown in Eq.(6):

$$\mathrm{R}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \tag{6}$$

F1-Score: The F1-score balances recall and precision, as expressed in Eq. (7):

$$\mathrm{F1}=2\frac{\mathrm{P}\cdot\mathrm{R}}{\mathrm{P}+\mathrm{R}}. \tag{7}$$

Macro-Averaging: This approach first computes the precision (P), recall (R), and F1-score for each class. It then determines the arithmetic average across all classes, as illustrated in Eqs. (8), (9), and (10):

$$\mathrm{Macro\_P}=\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P_i}, i=1,2,\cdots,n. \tag{8}$$

$$\mathrm{Macro\_R}=\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{R_i}, i=1,2,\cdots,n. \tag{9}$$

$$\mathrm{Macro\_F1}=\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{F1_i}, i=1,2,\cdots,n. \tag{10}$$

The higher these metrics, the better the model's performance.

B. GARCH models

In the study of time series data, initial regression models typically assumed that the variance within the series remained constant, implying uniform fluctuations over time. However, this assumption proves inadequate for time series exhibiting variable volatility. Parameter estimations made under the assumption of constant variance often lack precision, and significance tests for these parameters cannot be reliably conducted. To address these issues, GARCH-type models, including GARCH, exponential generalized autoregressive conditional heteroskedasticity (EGARCH), and threshold generalized autoregressive conditional heteroskedasticity (TGARCH), were developed, primarily for financial time series analysis. These models effectively tackle variable volatility by incorporating past volatility data as a conditional element and using an autoregressive framework to account for fluctuations. GARCH-type models are particularly adept at managing aspects like heteroskedasticity, volatility clustering, leverage effects, and asymmetrical responses to market movements. Overall, these models offer a strong framework for analyzing and modeling the dynamic nature of volatility in time series data, significantly improving upon earlier models that assumed static variance.

1) GARCH model: The GARCH model, introduced by [8], extends the ARCH model initially proposed by [9]. This extension is especially effective when fitting time series data with significantly high p-values. The GARCH (p, q) model can be expressed in the following way:

$$y_t = \mu_t + \sigma_t \eta_t, \quad \eta_t \sim N(0,1), \tag{11}$$

$$\varepsilon_t = \sigma_t \eta_t \quad \varepsilon_t | \chi_{t-1} \sim N\left(0, \sigma_t^2\right), \tag{12}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \tag{13}$$

where y_t represents the time series value, μ_t denotes the mean predicted by the average equation, ε_t refers to the error term, which is unpredictable and follows a normal distribution based on all available information up to time $t-1$, σ_t^2 indicates the conditional variance, α_i is the coefficient for the ARCH term; and β_j is the coefficient for the GARCH term.

2) EGARCH model: The limitation of GARCH models in effectively capturing the different impacts of positive and negative shocks on price series has prompted the creation of asymmetric GARCH models. The EGARCH model, proposed by [10], represents an advancement over the traditional GARCH model. This model is unique because it allows for negative coefficients in the variance equation, unlike the standard GARCH model, which requires all coefficients to be nonnegative.

EGARCH exhibits a leverage effect, and its standard equation for conditional variance is:

$$\ln \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \left(\left| \frac{\varepsilon_{t-i}}{\sigma_{t-i}} \right| + \gamma_i \frac{\varepsilon_{t-i}}{\sigma_{t-i}} \right) + \sum_{j=1}^{q} \beta_j \ln \sigma_{t-j}^2, \tag{14}$$

where γ_i denotes the leverage coefficient. The contribution of positive ε_{t-i} to the log volatility is $\alpha_i(1+\gamma_i)\left|\frac{\varepsilon_{t-i}}{\sigma_{t-i}}\right|$, while the contribution of negative ε_{t-i} to the log volatility is $\alpha_i(1-\gamma_i)\left|\frac{\varepsilon_{t-i}}{\sigma_{t-i}}\right|$. In practical applications, we generally expect γ_i to be negative.

3) TGARCH model: The primary limitation of the GARCH model is its inability to model asymmetric responses to positive and negative shocks. This limitation stems from the conditional variance in the GARCH equation being influenced by the size of the residuals, rather than their sign. In contrast, the threshold GARCH (TGARCH) variant, introduced by [11], is capable of modeling this asymmetry in responses to different types of news. The TGARCH model characterizes the conditional variance as follows:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} (\alpha_i + 1(\varepsilon_{t-i} < 0)\gamma_i) \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \tag{15}$$

where the characteristic function $1(\varepsilon_{t-i} < 0)$ takes the value of 1 when ε_{t-i} is negative and 0 otherwise. The parameters γ_i are used to capture the effect of negative returns on volatility.

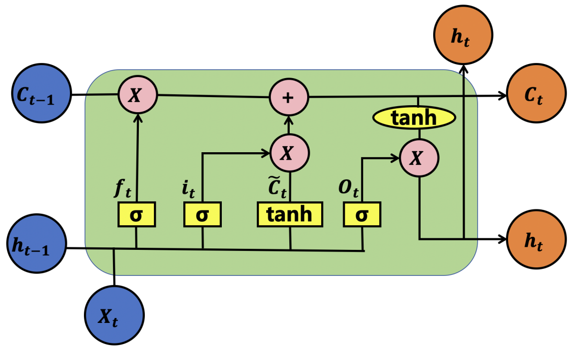

Fig. 5: The LSTM model architecture.

In summary, GARCH type models are employed to analyze the volatility characteristics of stock prices. We select best GARCH type models to fit the stock data, and output the corresponding conditional volatilities and residuals.

C. LSTM

The LSTM model is a specialized form of RNN developed to tackle the problems of gradient vanishing and gradient explosion that commonly occur in standard RNNs. It is particularly effective for predicting nonlinear variable time series. The core of LSTM is its gating logic. Compared to RNNs, LSTM incorporates an additional component known as a “memory cell” along with three gate structures: the input gate, forget gate, and output gate. This architecture enables LSTM to handle longer time series data more effectively. The structural design of LSTM, which underpins its advanced capabilities, is illustrated in Fig. 5 ([14]).

Forget gate:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f), \tag{16}$$

where f_t indicates the weight that determines which information to retain, W_f represents the weight matrix associated with the forget gate, and b_f is the bias term for this gate.

Input gate:

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i), \tag{17}$$

$$\widetilde{C}_t = tanh(W_c \times [h_{t-1}, x_t] + b_c), \tag{18}$$

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t, \tag{19}$$

where W_i and W_c are the weight matrices for the input gates, while b_i and b_c represent the bias terms associated with input gates.

Output gate:

$$o_t = \sigma(W_O \times [h_{t-1}, x_t] + b_O), \tag{20}$$

$$h_t = o_t \times \tanh(C_t), \tag{21}$$

where o_t represents the output from the cell state, derived through the sigmoid function, and h_t indicates the predicted stock price.

In this work, the structure of the LSTM component is illustrated in Fig. 6. The input layer receives the data and forwards it to the LSTM layer. To optimize performance while maintaining simplicity and effectiveness, we employ two LSTM layers. Additionally, we incorporate two fully connected layers, given the relatively low complexity of the

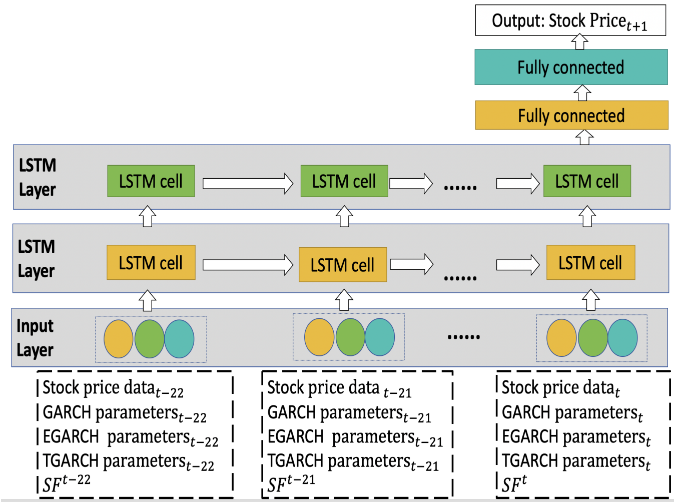

Fig. 6: Flowchart of the LSTM component

data. We utilize a sliding window approach in the LSTM to fit and predict stock prices, selecting a window size of 22, which corresponds to the number of trading days in a month. This model exemplifies both simplicity and computational efficiency.

D. SF-GARCH-LSTM models

The flowchart of the SF-GARCH-LSTM model is presented in Fig. 7. This model consists of three main components.

The first component focuses on sentiment analysis, where we utilize the BERT pre-training model to fine-tune the classification of emotions in text data (positive or negative). This process enables us to derive daily sentiment factors, as outlined in Eq. (4).

The second component involves fitting GARCH models to extract fluctuation characteristics from historical stock price data. In these models, we employ conditional volatilities and model residuals as parameters. The conditional volatilities reflect fluctuations in the stock market, while the model residuals capture price changes not accounted for by the model. These parameters are estimated using GARCH, EGARCH, and TGARCH models.

The third component consists of stock price data, which includes the opening price, closing price, highest price, lowest price, volume, the simple moving average over 22 days ($SMA22_t$), and the exponential moving average (EMA_t). For a detailed overview, refer to Table I.

We will use the sentiment factors, GARCH parameters, and stock price data as inputs for the LSTM model. As shown in Fig. 6, the sentiment factors, GARCH parameters, and stock prices from the 22 days preceding the forecasted time point are utilized as inputs for the LSTM. The LSTM parameters are then trained accordingly. Once training is complete, the model will be employed to predict stock prices in the test set.

IV. Experiment and Results

A. Data Sources

For stock prices, we have selected the CSI300 index, which serves as a significant indicator of the trends in the Chinese market. The time frame for our analysis spans from August 6, 2015, to September 28, 2023. We downloaded the historical CSI300 prices from the Yingwei Financial website. Additionally, we sourced relevant stock comment titles from the Oriental Fortune Stock Bar, an online community that provides real-time market commentary on individual stocks. These comment titles will be utilized for sentiment analysis.

TABLE I: Features utilized in stock price data.

Features	The source
opening price	from the data
closing price	from the data
highest price	from the data
lowest price	from the data
volume	from the data
$SMA22_t$	$SMA22_t = \frac{p_{t-1}+\ldots+p_{t-22}}{22}$
EMA_t	$EMA_t = p_{t-1} \cdot 0.3 + EMA_{t-1} \cdot 0.7$

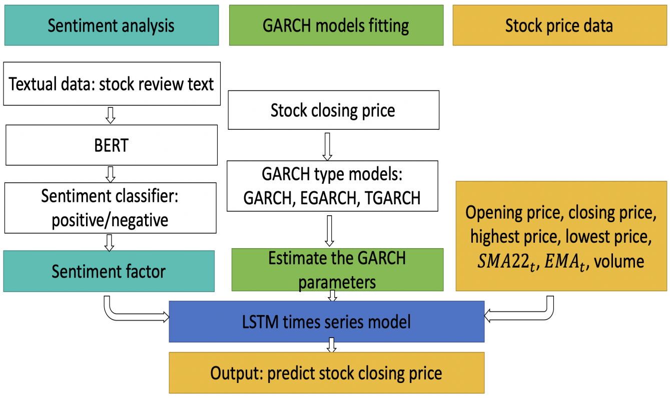

Fig. 7: Flowchart of the SF-GARCH-LSTM model

B. Textual data and sentiment analysis

We collected relevant comment titles from the Oriental Fortune network for the CSI300, resulting in a total of 42,177 titles. Figure 8 displays some example titles. The BERT model training process, illustrated in Figure 4, is employed to conduct sentiment analysis. First, we downloaded the Chinese language model pre-trained by `Bert_Base_Chinese` from Google, and then selected 10,000 entries from the 42,177 datasets for fine-tuning. This subset was manually labeled with positive and negative emotions, as shown in Fig. 3. The specific hand-labeled dataset is utilized to train the model, leading to a refined model that is ultimately applied to the entire textual dataset.

Out of the 10,000 manually labeled data points, there were 6,818 instances of negative emotions and 3,182 instances of positive emotions. The corresponding weights for these emotions are 1.0 and 2.1426, respectively. Given the average text message length of 21.4849, we set the maximum length to 30. Fine-tuning is performed using a masked language model (MLM). We randomly choose 80% of the data for the training set and 20% for the test set, with a batch size of 64.

We input the text data to obtain the BERT embeddings, after which a linear layer is applied to map the output features (dimension 768) of the BERT model to the desired output (dimension 2) for the binary classification problem. Finally, we utilize the softmax activation function to convert the input into probability outputs, applying weights to calculate the

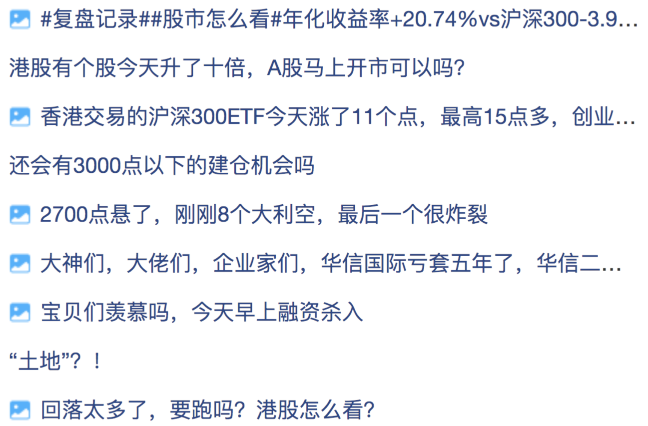
#复盘记录##股市怎么看#年化收益率+20.74%vs沪深300-3.9...
港股有个股今天升了十倍，A股马上开市可以吗?
香港交易的沪深300ETF今天涨了11个点，最高15点多，创业...
还会有3000点以下的建仓机会吗
2700点悬了，刚刚8个大利空，最后一个很炸裂
大神们，大佬们，企业家们，华信国际亏套五年了，华信二...
宝贝们羡慕吗，今天早上融资杀入
“土地”？！
回落太多了，要跑吗？港股怎么看？

Fig. 8: Titles of comments from the Oriental Fortune Network CSI 300 Bar on October 2, 2024

cross-entropy loss. We instantiate a classification model and utilize the fl_cls function, employing the Adam optimizer with a learning rate of 1e-6. Additionally, we configure the learning rate scheduler to adopt the StepLR strategy, reducing the learning rate by multiplying it by 0.9 after each epoch.

An early stopping mechanism is implemented with a maximum patience value of 10. If the validation performance does not improve for 10 consecutive epochs, early stopping is triggered, terminating the training cycle. We set a total of 3 epochs and initialize the best accuracy to 0. After fine-tuning and training BERT, the best model is saved to a file. The text classification accuracy rates before and after fine-tuning are shown in Table II and Table III. After fine-tuning BERT, we observed a remarkable increase in text classification accuracy of 39%, underscoring the significance of the fine-tuning process. Example results are illustrated in Fig. 9. In the ‘Pre’ column, 0 indicates a negative classification, while 1 indicates a positive classification. The left endpoint of ‘Probability’ represents the likelihood of a negative sentiment, and the right endpoint represents the likelihood of a positive sentiment. From the classification outcomes, we can determine the daily counts of 0 and 1 tags, enabling us to compute the daily sentiment factors according to Eq. (4). Figure 10 visualizes the daily distribution of positive and negative comments.

TABLE II: Text classification accuracy metrics before fine-tuning

Accuracy: 0.614				
text	precision	recall	f1-score	support
0 negative	0.78	0.60	0.67	1342
1 position	0.44	0.65	0.53	658
macro avg	0.61	0.62	0.60	2000

TABLE III: Text classification accuracy metrics after fine-tuning

Accuracy: 0.998				
text	precision	recall	f1-score	support
0 negative	1.00	1.00	1.00	1342
1 position	0.99	1.00	1.00	658
macro avg	1.00	1.00	1.00	2000

Comment Title	Pre	Probability
主力资金咋沪深三百，节后不乐观。 The main funds are not optimistic after the holiday.	0	[0.94603, 0.05396]
仿佛一直在黑暗中看不到未来的光明。 It seems I can’t see the light of the future in the darkness.	0	[0.92397, 0.07602]
交银指数星球回升向好。 Bocom Index Planet rebounds and improves.	1	[0.02312, 0.97687]
熊霸天下【兴奋】 The bear dominates the world [Excited]	1	[0.36241, 0.63759]

Fig. 9: Example of BERT sentiment classification results.

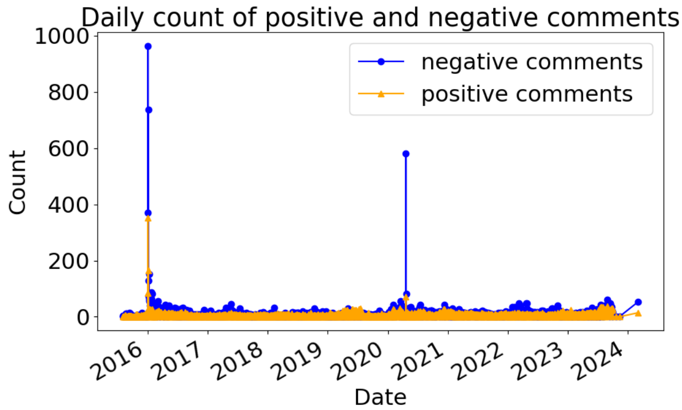

Fig. 10: Daily distribution of positive and negative sentiment classifications based on comment titles.

C. Analysis of the results of the GARCH models

To quantify the volatility of the stock price sequence and enhance its stability, we construct the first-order difference of the logarithm of the stock price, represented as

$$P_t = log(price_t) - log(price_{t-1}). \qquad (22)$$

GARCH type models can be employed to analyze the volatility characteristics of the log return rate series. When selecting a model, the smaller AIC and BIC indicates a better fit. We choose GARCH (1,2), EGARCH (2,1), and TGARCH (2,1) with a t-distribution to fit the stock closing price. And we utilized the conditional volatilities and residuals estimated from these three models. Some examples of the conditional volatilities and residuals estimated from the GARCH-type models are shown in Table IV.

TABLE IV: GARCH type models: volatilities and residuals

date	garch_volatility	garch_residual	egarch_volatility
2015/9/9	0.0599	0.0231	0.0314
2015/9/10	0.0589	-0.0085	0.0304
2015/9/11	0.0563	0.0007	0.0296

date	egarch_residual	tgarch_volatility	tgarch_residual
2015/9/9	0.0190	0.0428	0.0167
2015/9/10	-0.0126	0.0424	-0.0149
2015/9/11	-0.0003	0.0386	-0.0057

D. Loss function

The model's predictive ability is assessed using three evaluation metrics: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i^{'} - y_i\right|, \tag{23}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i^{'} - y_i)^2, \tag{24}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i^{'} - y_i\right)^2}, \tag{25}$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\frac{|y_i^{'} - y_i|}{y_i}. \tag{26}$$

In these equations, the predicted value of the stock price is denoted as $y_i^{'}$, the actual price is denoted as y_i, and the total number of predicted samples is denoted as N. Smaller values of these metrics indicate more accurate stock forecasts.

E. SF-GARCH-LSTM experiments and results

To evaluate the effectiveness of combining sentiment factors and GARCH model parameters with LSTM for stock price prediction, we conducted extensive experiments. These included individual models such as LSTM, two-method combinations (e.g., SF-LSTM, G-LSTM, E-LSTM, T-LSTM, GE-LSTM, GT-LSTM, ET-LSTM, and GET-LSTM), as well as three-method combinations (e.g., SF-G-LSTM, SF-E-LSTM, SF-T-LSTM, SF-GE-LSTM, SF-GT-LSTM, SF-ET-LSTM, and SF-GET-LSTM). Here G, E, T respectively denote GARCH, EGARCH and TGARCH. Table V summarizes the input variables for both individual and hybrid models. In Table V, 'Stock price data' represents the features from the stock market, the detailed stock price features are shown in Table I, 'SF' represents sentiment factors, while 'GARCH', 'EGARCH', and 'TGARCH' refer to the conditional volatilities and residuals estimated by these three models. These parameters are used as inputs to the hybrid SF-GARCH-LSTM models.

TABLE V: Input variables for each model.

Model	price	SF	GARCH	EGARCH	TGARCH
LSTM	✓				
SF-LSTM	✓	✓			
G-LSTM	✓		✓		
E-LSTM	✓			✓	
T-LSTM	✓				✓
SF-G-LSTM	✓	✓	✓		
SF-E-LSTM	✓	✓		✓	
SF-T-LSTM	✓	✓			✓
GE-LSTM	✓		✓	✓	
GT-LSTM	✓		✓		✓
ET-LSTM	✓			✓	✓
SF-GE-LSTM	✓	✓	✓	✓	
SF-GT-LSTM	✓	✓	✓		✓
SF-ET-LSTM	✓	✓		✓	✓
GET-LSTM	✓		✓	✓	✓
SF-GET-LSTM	✓	✓	✓	✓	✓

We selected the stock price of CSI300 from August 6, 2015, to September 28, 2023. The dataset was split into a

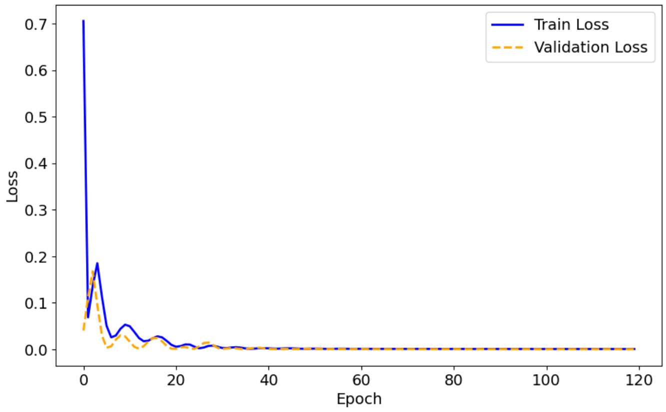

Fig. 11: Visualization of loss improvement across epochs.

training set and a test set in an 80:20 ratio. The training set includes data for 1560 days, while the test set comprises data for 391 dates. Next, the data is normalized, and a sliding window was constructed with a window size set to 22. Subsequently, an nn.LSTM was created to build the LSTM layer. The input layer dimension is specified as input_size, while the hidden layer dimension is set to three times the input_size. Two LSTM layers are utilized in the architecture. Following this, two fully connected layers are created to map the output of the LSTM hidden layer to the same dimension and then to an output dimension of 1. As shown in Fig. 11, the loss converges after 40 epochs, and we chose to train for a total of 60 epochs with a learning rate of 0.01. The normalized stock price was predicted, and MAE, MSE, RMSE, and MAPE were calculated on the test set.

To establish a baseline for comparison, we employed CNN and GRU architectures alongside LSTM for data processing. This study utilizes opening price, closing price, highest price, lowest price, $SMA22_t$, EMA_t and volume data to evaluate the performance of these models. To ensure methodological fairness, all models are configured with parameters consistent with those of the LSTM. The results presented in Table VI demonstrate that the LSTM model is more effective for analyzing the dataset used in this research.

TABLE VI: Evaluation metrics results for CNN, GRU, and LSTM Models

Model	MAE	MSE	RMSE	MAPE
CNN	0.0925	0.0138	0.1175	2.5668%
GRU	0.0712	0.0075	0.0869	1.9818%
LSTM	0.0189	0.0006	0.0246	1.4044%

Table VII presents the performance and accuracy of the GARCH-LSTM models. From this table, it is evident that the GARCH-LSTM hybrid model outperforms the LSTM model in predictions when new explanatory variables, such as conditional volatilities, residuals, are included, especially when two or three GARCH models are incorporated. This improvement is attributable to the GARCH model's ability to reflect future market volatility, while the residuals capture information not accounted for by the GARCH model. Among the GARCH-LSTM models, GET-LSTM demonstrates the best performance. Table VIII displays the accuracy of SF-GARCH-LSTM models. Comparing this table with the previous one reveals that incorporating the sentiment factor has

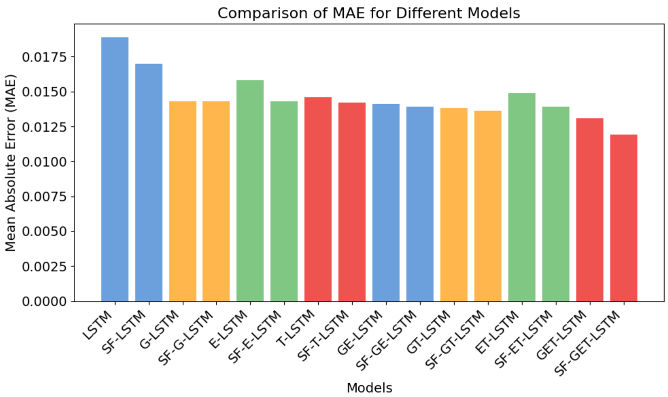

Fig. 12: Comparison of MAE across different models

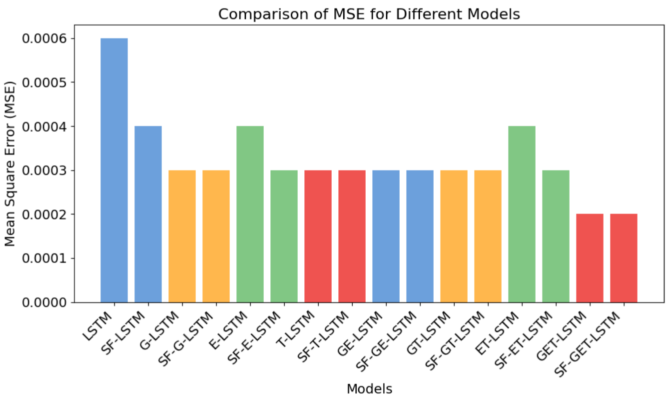

Fig. 13: Comparison of MSE across different models

decreased the errors for all models. This indicates that adding the sentiment factor as explanatory variables can enhance the models' ability to predict stock prices. Models that include sentiment factors perform better because these factors reflect public confidence in the future development of the market, influencing investors' willingness to buy stocks, which in turn affects stock market fluctuations.

TABLE VII: Evaluation metrics results for GARCH-LSTM model

Model	MAE	MSE	RMSE	MAPE
LSTM	0.0189	0.0006	0.0246	1.4044%
G-LSTM	0.0143	0.0003	0.0184	1.0680%
E-LSTM	0.0158	0.0004	0.0216	1.1916%
T-LSTM	0.0146	0.0003	0.0188	1.0799%
GE-LSTM	0.0141	0.0003	0.0179	1.0556%
GT-LSTM	0.0138	0.0003	0.0175	1.0272%
ET-LSTM	0.0149	0.0004	0.0210	1.0946%
GET-LSTM	0.0131	0.0002	0.0168	0.9727%

TABLE VIII: Evaluation metrics results for SF-GARCH-LSTM model

Model	MAE	MSE	RMSE	MAPE
SF-LSTM	0.0170	0.0004	0.0219	1.2556%
SF-G-LSTM	0.0143	0.0003	0.0189	1.0604%
SF-E-LSTM	0.0143	0.0003	0.0180	1.0599%
SF-T-LSTM	0.0142	0.0003	0.0184	1.0472%
SF-GE-LSTM	0.0139	0.0003	0.0175	1.0393%
SF-GT-LSTM	0.0136	0.0003	0.0174	1.0102%
SF-ET-LSTM	0.0139	0.0003	0.0175	1.0366%
SF-GET-LSTM	0.0119	0.0002	0.0157	0.8849%

The SF-GARCH-LSTM models outperform both the LSTM and GARCH-LSTM models. The hybrid model that integrates sentiment factors shows better performance compared to models without sentiment factors. The experimental results using actual data align with our expectations. By providing the model with additional information on market volatility and public sentiment, we can achieve more accurate stock price predictions. Among all the models, the SF-GET-LSTM model demonstrates the best predictive performance, featuring the lowest MAE, MSE, RMSE, and MAPE. Fig. 12 and Fig. 13 illustrate these results more intuitively.

Fig. 14 shows that the predictions of the SF-LSTM model are more accurate compared to those of the LSTM model. Similarly, Fig. 15 shows that the SF-G-LSTM model offers predictions that are closer to the actual values than the G-LSTM model. Therefore, we can conclude that incorporating the sentiment factor enhances the performance of the hybrid models. These figures clearly show that the predicted stock prices closely align with the actual stock prices, indicating that these hybrid models are effective in forecasting.

To closely examine the role of the sentiment factor, we selected January 2019 as the test set, using the period from August 6, 2015, to December 2018 as the training set. January 2019 was chosen for testing because the Chinese stock market reached a low during this time, influenced by the pressures of US-China trade friction, which left the public uncertain about the market's future direction.

We will analyze whether incorporating sentiment factors into model training affects the prediction of the current month's data compared to models without sentiment factors. Performance comparisons were made between LSTM and SF-LSTM, as well as between G-LSTM and SF-G-LSTM. The results show that models incorporating the SF module significantly outperform their counterparts without it, as evidenced in Figs. 16 and 17.

V. Conclusion

This paper introduces the SF-GARCH-LSTM model, a novel hybrid approach for stock price prediction that integrates sentiment analysis, GARCH models, and LSTM. Sentiment analysis is performed using a pre-trained BERT model, achieving a fine-tuned classification accuracy of 99.8%, which is utilized to derive daily sentiment factors (SF). GARCH parameters are estimated using GARCH, EGARCH, and TGARCH models to capture the heteroskedastic characteristics of financial time series. These features, combined with stock price data, are fed into the LSTM model to extract temporal dependencies. The proposed SF-GARCH-LSTM model demonstrates superior performance, highlighting its potential for accurate stock price forecasting.

References

[1] J. Bollen, H. Mao and X. Zeng, "Twitter mood predicts the stock market," *Journal of computational science* , vol. 2, no. 1, pp1-8, 2011.

[2] S. P. Sidorov, A. Revutskiy, A. Faizliev, and V. Balash. "Stock volatility modelling with augmented GARCH model with jumps," *IAENG International Journal of Applied Mathematics*, vol. 44, no. 4, pp212-220, 2014.

[3] H. Chen, p. De, Y. Hu and B. Hwang, "Wisdom of crowds: The value of stock opinions transmitted through social media," *The review of financial studies*, vol. 27, no. 5, pp1367-1403, 2014.

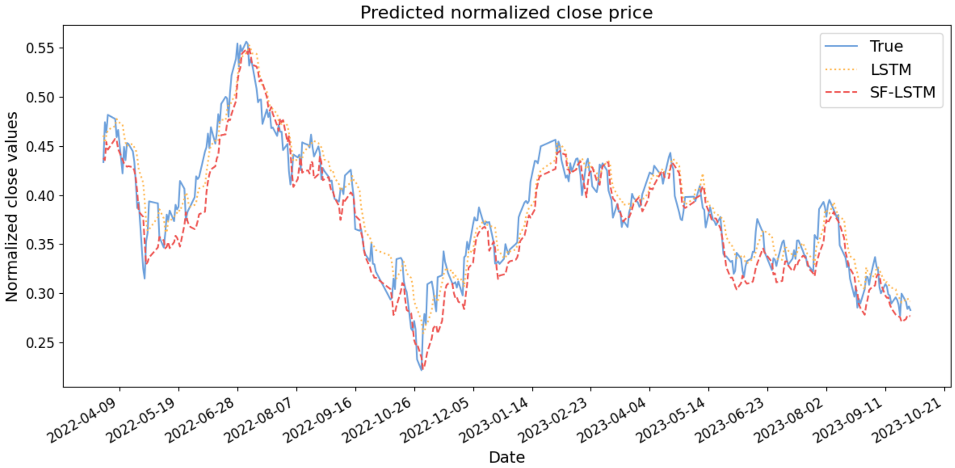

Fig. 14: Comparison of stock price predictions: LSTM vs. SF-LSTM

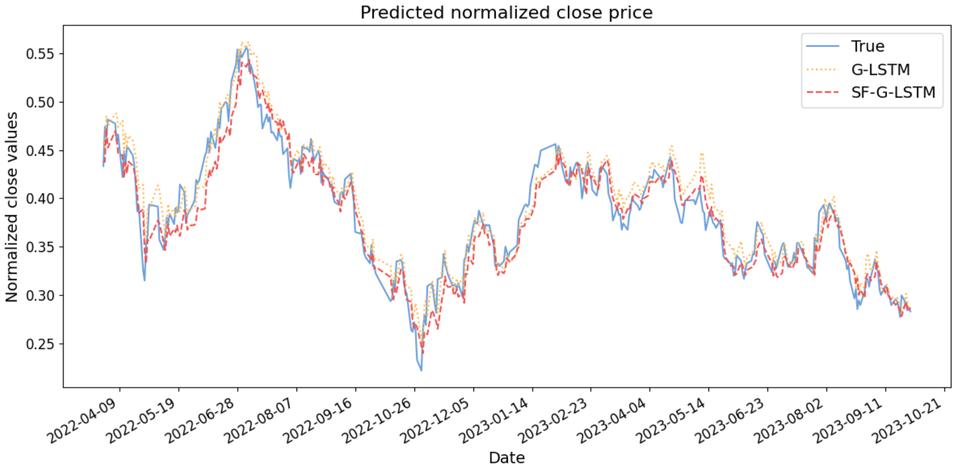

Fig. 15: Comparison of stock price predictions: G-LSTM vs. SF-G-LSTM

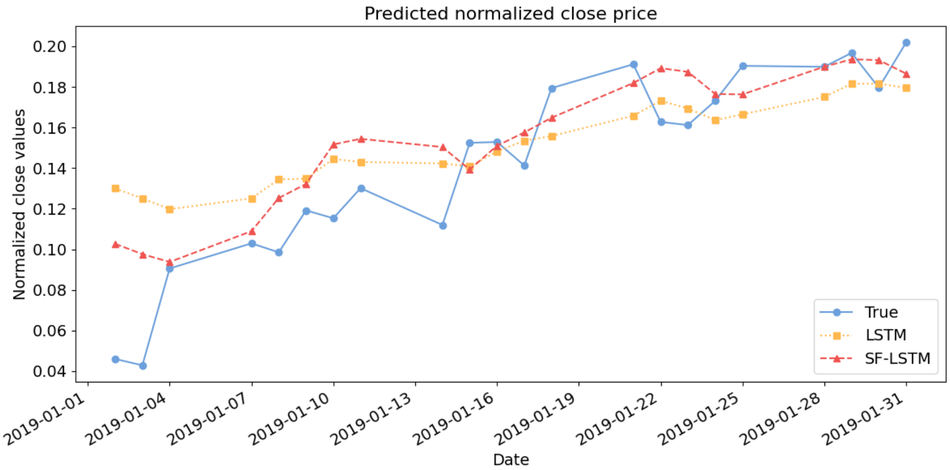

Fig. 16: Comparison of stock price predictions: LSTM vs. SF-LSTM in January 2019

[4] J. Kaminski, "Nowcasting the bitcoin market with twitter signals," *arXiv preprint arXiv:1406.7577*, 2014.

[5] M. Baker and J. Wurgler, "Investor sentiment and the cross-section of stock returns," *The journal of Finance*, vol. 61, no. 4, pp1645-1680, 2006.

[6] M. Baker and J. Wurgler, "Investor sentiment in the stock market," *Journal of economic perspectives*, vol. 21, no. 2, pp129-151, 2007.

[7] T. Swathi, N. Kasiviswanath and A. Rao, "An optimal deep learning-based LSTM for stock price prediction using twitter sentiment analysis," *Applied Intelligence*, vol. 52, pp13675-13688, 2022.

[8] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of econometrics*, vol. 31, no. 3, pp307-327, 1986.

[9] R. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica: Journal of the econometric society*, pp987-1007, 1982.

[10] D. Nelson, "Conditional heteroskedasticity in asset returns: A new approach," *Econometrica: Journal of the econometric society*, pp347-370, 1991.

[11] J. Zakoian, "Threshold heteroskedastic models," *Journal of Economic Dynamics and control*, vol. 18, no. 5, pp931-955, 1994.

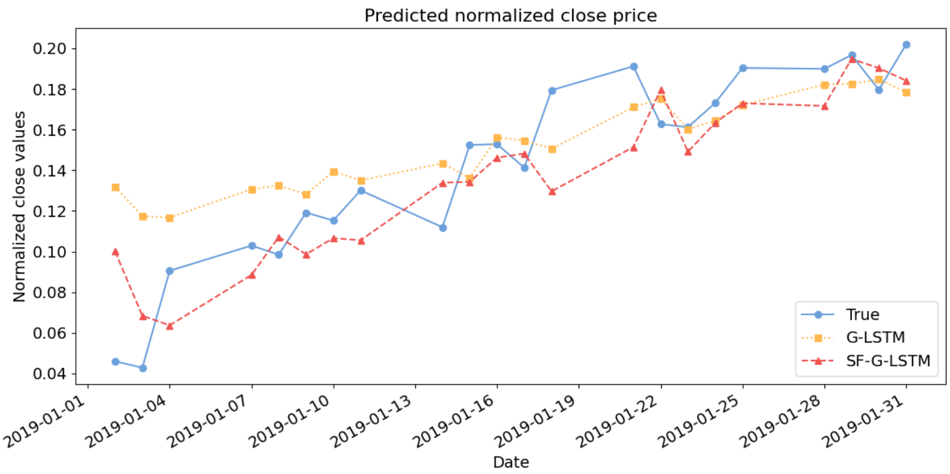

Fig. 17: Comparison of stock price predictions: G-LSTM vs. SF-G-LSTM in January 2019

[12] D.J. Bartholomew, "*Time series analysis: forecasting and control,*" John Wiley & Sons, 1971.

[13] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. Menon and K. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *2017 international conference on advances in computing, communications and informatics (icacci)*, pp1643-1647, 2017..

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp1735-1780, 1997.

[15] K. Cho, M. Van, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 1997.

[16] Z. Berradi, M. Lazaar, H. Omara, and O. Mahboub, "Effect of architecture in recurrent neural network applied on the prediction of stock price," *IAENG International Journal of Computer Science*, vol. 47, no. 3, pp436-441, 2020.

[17] A. S. Girsang, F. Lioexander, and D. Tanjung," Stock price prediction using lstm and search economics optimization," in *IAENG International Journal of Computer Science*, vol. 47, no. 4, pp758-764, 2020.

[18] Y. Wang, P. Liu, K. Zhu, L. Liu, Y. Zhang and G. Xu, "A Garlic-Price-Prediction Approach Based on Combined LSTM and GARCH-Family Model," *Applied Sciences*, vol. 12, no. 22, pp11366, 2022.

[19] C. Ma, J. Wu, H. Hu, Y.N. Chen and J.Y. Li, "Predicting stock prices using hybrid LSTM and ARIMA model," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 3, pp424-432, 2024.

[20] M. Li, W. Li, F. Wang, X. Jia and G. Riu, "Applying BERT to analyze investor sentiment in stock market," *Neural Computing and Applications*, vol. 33, pp4663-4676, 2021.

[21] M. Li, L. Chen, J. Zhao and Q. Li, "Sentiment analysis of Chinese stock reviews based on BERT model," *Applied Intelligence*, vol. 51, pp5016-5024, 2021.

[22] Y. Wang, X. Cheng, and X. Meng, "Sentiment analysis with an integrated model of BERT and bi-LSTM based on multi-head attention mechanism," *IAENG International Journal of Computer Science*, vol. 50, no. 1, pp255-262, 2023.

[23] J. Kim, H. Kim and S. Choi, "Forecasting the S&P 500 index using mathematical-based sentiment analysis and deep learning models: a FinBERT transformer model and LSTM," *Axioms*, vol. 12, no. 9, pp835, 2023.

[24] Y. Ayyappa, B. Kumar, S. Priya, S. Akhila, T. Reddy and S. Goush, "Forecasting Equity Prices using LSTM and BERT with Sentiment Analysis," in *2023 International Conference on Inventive Computation Technologies (ICICT)*, IEEE, pp643-648, 2023.

[25] N. Jing, Z. Wu and H. Wang, "A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction," *Expert Systems with Applications*, vol. 178, pp115019, 2021.

[26] J. Liu and X. Huang, "Forecasting crude oil price using event extraction," *IEEE Access*, vol. 9, pp149067-149076, 2021.

[27] Y. Huang, X. Dai, Q. Wang and D. Zhou, "A hybrid model for carbon price forecasting using GARCH and long short-term memory network," *Applied Energy*, vol. 285, , pp116485, 2021.

[28] H. Kim and C. Won, "Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103, pp25-37, 2018.

[29] M. Seo and G. Kim, "Hybrid forecasting models based on the neural networks for the volatility of bitcoin," *Applied Sciences*, vol. 10, no. 14, pp4768, 2020.

[30] E. Koo and G. Kim, "A hybrid prediction model integrating garch models with a distribution manipulation strategy based on lstm networks for stock market volatility," *IEEE Access*, vol. 10, pp34743-34754, 2022.

[31] K. Kakade, A. Mishra, K. Ghate and S. Gupta, "Forecasting Commodity Market Returns Volatility: A Hybrid Ensemble Learning GARCH-LSTM based Approach," *Intelligent Systems in Accounting, Finance and Management*, vol. 29, no. 2, pp103-117, 2022.

[32] J. Devlin, M. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, vol. 1, pp4171-4186, 2019.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez,U. Kaiser and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.