The International Society of Data Scientists
Journal of Data Science and Artificial Intelligence
Vol. 1, No. 1. Online ISSN: 2831-4794
Available at https://www.isods.org/publication/

STL Decomposition-based Stock Price Forecasting using LSTM model

Tieu Khoi Mai[a], Pham Hoang Vuong[a], Vu Ngoc Thanh Sang[a],
Nguyen Phan Thu Hang[b], Pham The Bao*[a]
[a]Information Science Faculty, Sai Gon University, Ho Chi Minh City, Vietnam 70000
[b] Faculty of Business Administration, Sai Gon University, Ho Chi Minh City, Vietnam 70000

ABSTRACT

Stock price forecasting is one of the challenging research applications in time series data analysis because it is affected by numerous factors such as political news, weather, and pandemics. In addition, stock prices have several strong statistical univariate properties like unit-root property, the fat tail phenomenon, and volatility clustering. Equally important, time series in the financial market expresses strong seasonality and unpredicted components. To enhance the performance of stock price forecasting, this work proposes the method by combining the seasonal trend decomposition using the Loess (STL) method and long-short-term memory (LSTM) model. The proposed models are experimentally evaluated on 50 stocks collected from Forex, commodity, and New York Stock Exchange (NYSE) stock market. The experimental results show that LSTM-STL-price enhances the forecasting performance compared to single LSTM model.

Keywords: Stock price forecasting, seasonal trend decomposition using the Loess (STL), long-short-term memory (LSTM)

1. INTRODUCTION

Financial markets are a complex, evolving, nonlinear dynamic system[1], and stock price forecasting is a well-studied classical problem[2]. Worldwide, significant stock investments are made. Finance is influenced by various factors, including political events, socioeconomic conditions, corporate financial news, natural disasters, and traders' expectations; as a result, forecasting stock market price activity presents numerous challenges. However, the primary impediment to stock price forecasting is the dual effect of factors and time on the predicted value. Additionally, securities-related time series data are inherently noisy, non-stationary, and chaotic[3]. Noise refers to the inability to use complete historical data from financial markets to observe the relationship between future and historical price. Nonstationarity demonstrates that the distribution of financial time series is inherent changes over time. Furthermore, in the presence of unknown chaos, financial time series are stochastic in the short run but not in the long run. Therefore, machine learning and deep learning were used in popularity to generate more complex nonlinear models of stock prices, and model-based approach have become standard approaches to solve the stock price forecasting problem. However, neural networks have a disadvantage in that they require a large amount of training data and take a long time to train. As a result, data preprocessing becomes a critical step in optimizing the model's performance.

Likewise, rather than relying on a single model, we incorporate methodologies time series preprocessing by decomposition using the Loess (STL) method based on long short-term memory (LSTM) to enhances the forecasting performance.

This study is structured as follows. Section 2 introduces a related work. Section 3 provides an overview of STL and LSTM. Sections 4 and 5 exhibit and discuss the dataset and experimental results, respectively. Finally, in Section 6, concluding remarks are offered.

2. RELATED WORKS

The stock price has long been a source of concern for investors and researchers, and there is an abundance of literature devoted to the examination and forecasting of stock prices. In the past, a company's financial information was hypothesized to be systematically reflected at current prices[4]. Fama (1965) discussed empirical evidence supporting the random walk theory and proposed the Efficient Market Hypothesis (EFM). According to Malkiel and Fama, the EFM states that asset prices reflect all available information. Many economists believe that stock prices are partly affected by the psychology of

*ptbao@sgu.edu.vn; phone 84 28 38354409; fax 84 28 38305568

the participants[5]. Based on this belief, the behavioral finance field has been developed to address stock market behavior. Behavioral finance has become the foundation of two classic approaches to familiar stock price analysis - technical analysis and fundamental analysis.

The other methodology is time series analysis to extract the significant statistical characteristics of time series data. These methods typically use past information on the variable to capture or predict forecast future trends and behaviors. The behavior and past patterns are used to predict future changes because of their time-dependence and nonlinear characteristics. It is possible to decompose time series into four components (trend, cyclical, seasonal and irregular fluctuations). However, economic and business time series show strong seasonality and cyclic behaviors followed by the regular transition of business activities. Thus, the research on non-stationary in economic time series has emerged and attracted modern econometric analysis. The traditional time series forecasting methods such as Autoregression (AR), Moving Average (MA), Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), SARIMA or Vector Autoregression Moving-Average (VARMA) have been widely applied has been implemented in a variety of situations because of their efficiency level[6,7].

Since the 2000s, Zou et al. proposed a Bayesian network for the price-earnings ratio (P/E ratio) forecast[8]. The results show that Bayesian networks give a better forecast accuracy by 15% - 20% than traditional time series forecast methods. Although ARIMA models are relatively efficient than previous complex models, Adebiyi et al. examined the performance of ARIMA and Artificial Neural Network (ANN) with published stock data from the New York Stock Exchange (NYSE)[9] and revealed the outperformance of ANN over the ARIMA model.

The deep learning method is applied in stock price forcasting fields such as artificial neural network (ANN), recurrent neural network (RNN), and especially long short-term memory (LSTM) has gained popularity for time series forecasting.

Additionally, rather than utilizing a single model, hybrid models that incorporate methodologies such as time series preprocessing, and optimization are frequently utilized in studies on stock price forecasting. Other studies using a hybrid LSTM and Gated Recurrent Unit (GRU) models which improved performance on the S&P 500 index dataset from 1950 to 2016[10].

Another study by Nelson et al. proposed the LSTM model by using technical indicators to forecast stock price trends and applied it on the dataset of stocks in Brazil from 2008 to 2015 and achieved the highest results for the Banco Bradesco SA ticker with an accuracy of 55.9%[11].

Milosevic simultaneously applied deep learning and feature extraction using machine learning algorithms and experiments on 1739 stocks in the S&P 1000, FTSE 1000, and S&P Europe 350 portfolios from 2012 to 2015 to forecast stock price trends. The results show the superiority of the deep learning method when combined with the Random Forest to extract the features of the dataset when bringing the accuracy up to 76.2%[12]. In one recent work, Shen and Shafiq apply the feature expansion (FE) with recursive feature elimination (RFE), followed by principal component analysis (PCA) to build a feature engineering procedure and assembling with the LSTM model[13].

He et al. developed a novel seasonal-trend decomposition-based dendritic neuron model (STLDNM) using the most appropriate preprocessing by using seasonal-trend decomposition based on loess (STL)[14]. The newly constructed STLDNM not only has significant application potential in the financial sector but also gives innovative inspirations for academics working on complex time series prediction problems. Another approach using variational mode decomposition (VMD) and LSTM network. The forecasting errors of VMD-LSTM are more evenly distributed around zero and have a smaller fluctuation range than those of the other five models, confirming the model's superior predictive performance and adaptability to different datasets[15].

3. DATASET

The time series price dataset is 5-min Bid price of 50 stocks collected from the Forex market, commodity market, and U.S. stock market from 2017-2019, which was retrieved history of all data feeds from Dukascopy's Historical Data Feed. Each observation with features containing the open price, highest price, lowest price, close price, and volume information. Stock examples are displayed in **Figure 1.**

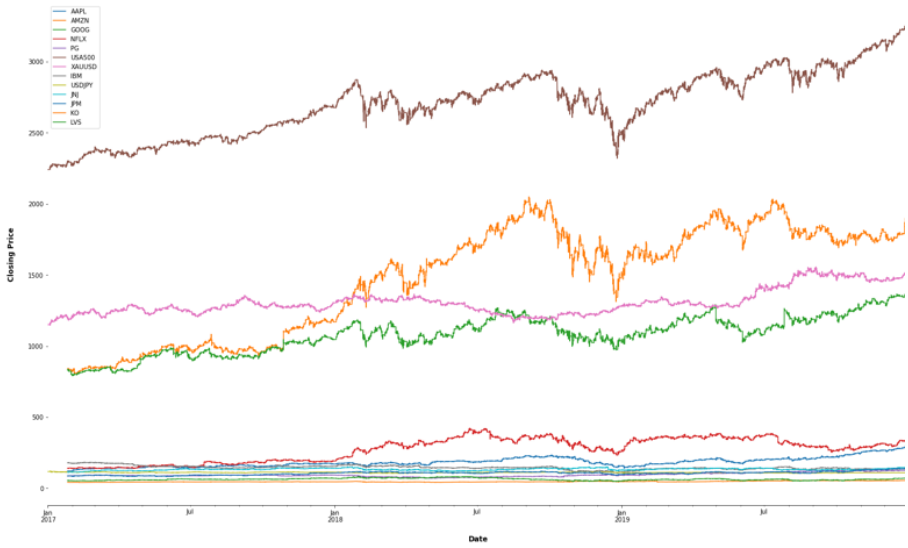


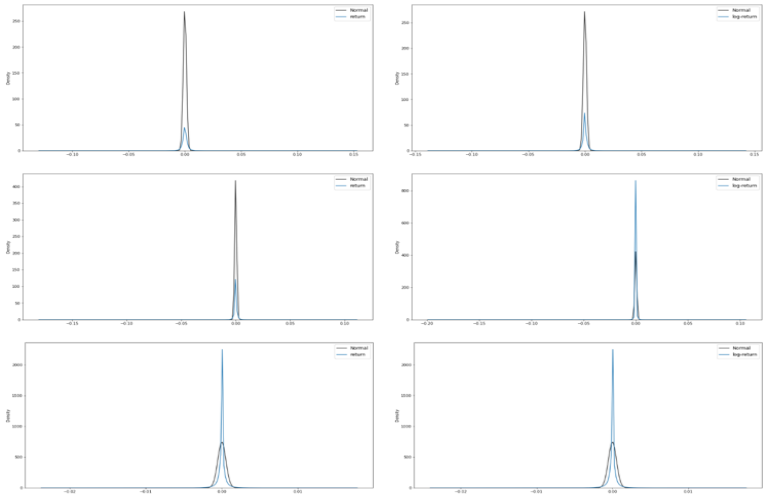
Figure 1. Closing price of tickers on Commodities, Forex and Stock Market

It is known that the price series in the financial markets have some typically statistical properties. Three important univariate properties of price series are (i) unit-root property, (ii) the fat tail phenomenon and (iii) the volatility clustering[16,17]. First, the prices show the unit-root property. The test statistic of all prices series for close price series is greater than critical values (negative value) with p-value approach, all p-value is greater than 0.05. Therefore, financial prices follow a random walk or are nonstationary.

It shown that the return has a higher excess kurtosis than the closure price, and the kurtosis value distribution of the returns is more prone to outliers than the normal distribution. Because the return distribution is a leptokurtic distribution, we can infer a fat tail occurrence. Moreover, the skewness value for the selected stock's price, return, and log return is skewed to the left or negative. This indicates a longer or fatter tail on the distribution's left side. If we model financial returns using the normal distribution, we will underestimate the number and magnitude of crashes and booms.

For example,

Figure 2 shows the distribution of return and log return of TSLA, FB stock and NASDAQ-100 index within two years. With the US market index, the empirical density function has a higher peak and fatter tails than that of the corresponding normal distribution. For a selected individual stock, the peak of the normal distribution is higher and the tail is wider than the corresponding normal density.



(a)

(b)

(c)

Figure 2. Comparison of return and log returns with normal distribution of Tesla (a), Facebook (b) and NASDAQ 100 Index (c).

Quiescence and turbulence tend to occur in clusters cause the presence of highly significant autocorrelation in the absolute or squared returns, despite the absence of autocorrelation in the price returns.

The autocorrelation of the selected index, commodity, and stock returns is frequently negative or extremely close to zero, as shown in **Figure 3**. This trend is consistent with the financial price return series and supports the presence of clustering.

(a)

(b)

(c)

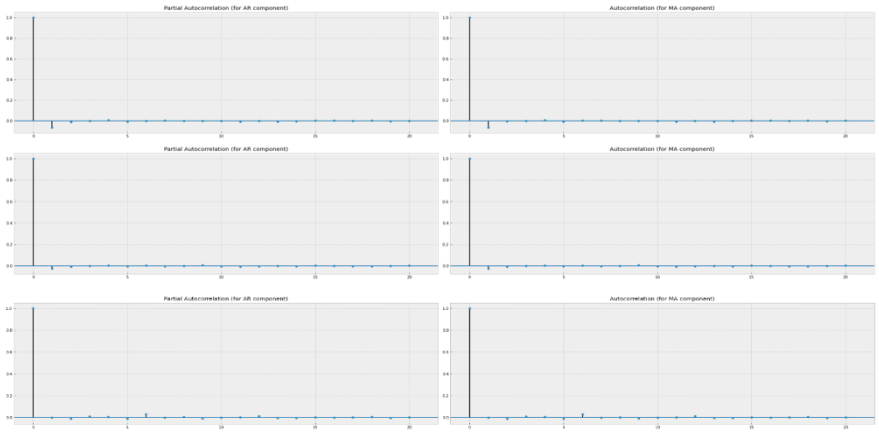


Figure 3. Partial Autocorrelation and Autocorrelation for returns of S&P 500 (a), Gold (b) and Apple (c) Stock

For the financial prediction problem, it requires to construct a function $f: \mathbb{R}^d \longrightarrow \mathbb{R}$. And Mean Squared Error (MSE), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are the most popular metrics for the regression task. They are mathematically expressed as

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \quad (1)$$

$$MSE = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2 \quad (2)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2} \quad (3)$$

4. PROPOSED METHOD

4.1 Time Series Decomposition

Time series can be decomposed into four components as trend, cyclical, seasonal, and irregular fluctuations. Time series data y_t can be decomposed as an additive decomposition equation (4).

$$y_t = S_t + T_t + R_t \quad (4)$$

where y_t is the time series

S_t is the seasonal component

T_t is the trend-cycle component

R_t is the remainder component, all at period t

Classical decomposition algorithm

1. Estimate $\widehat{T_t}$ by using simple moving average with seasonal period m
2. Detrend the series $y_t - \widehat{T_t}$
3. Average the detrended values to estimate the seasonal component for each season $\widehat{S_t}$
4. Compute the remainder component by subtracting the estimated seasonal and trend-cycle components

$\widehat{R_t} = \widehat{y_t} - \widehat{T_t} - \widehat{S_t}$

Unlike the classical method, Seasonal and Trend decomposition using Loess or mostly called STL decomposition methods efficiently handle various types of seasonality and the smoothness of the trend-cycle, which was proposed first time in 1990[18] by contributing to a decomposition method robust to anomalies.

STL decomposition algorithm

The algorithm consists of two loops. An outer loop - robustness coefficients are calculated to minimize the effect of outliers, and an inner loop - the trend component and seasonal component are iteratively updated using LOESS smoothing. The trend and seasonal components are updated in the inner loop using the following procedure.

1. Detrending is the first step. Obtained at the kth pass $y_t - T_t^{(k)}$
2. Smooth the subseries of data at each location of the seasonal cycle using LOESS. To obtain a preliminary seasonal component, the outcome is denoted at $\tilde{S}_t^{(k+1)}$
3. Filtering the smoothed seasonality $\tilde{S}_t^{(k+1)}$ using a low-pass filter and obtained $\tilde{T}_t^{(k+1)}$
4. Detrending of Smoothed Cycle-Subseries $S_t^{(k+1)} = \tilde{S}_t^{(k+1)} + \tilde{T}_t^{(k+1)}$
5. Deseasonalizing - The original series y_t is subtracted from the seasonal component $S_t^{(k+1)}$ to produce the seasonally adjusted series $y_t - S_t^{(k+1)}$
6. Smooth the deseasonalized series using LOESS to obtain the trend component of this pass $T_t^{(k+1)}$

For time series price datasets, besides using low, high, open, and close prices, technical indicators are widely used by professionals and scholars in stock trading. Therefore, we created a list of technical indicators, which are Simple Moving Average (SMA), Moving Average (MA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Stop and Reverse (SAR), Commodity Channel Index (CCI), ATR (Average True Range) using a mathematical formula to derive a series of data points from Close price.

Time series is an aggregate of three components (trend, seasonal and residual), which leads to decomposing the time series of Close price into three components – seasonal, trend by using the STL decomposition method and add these components to the forecasting model as features.

4.2 LSTM

Long-short term memory (LSTM)[19] is of a recurrent type. One advantage of the LSTM model over a standard recurrent network model is its ability to capture autoregressive structures of any length. The difficulty in training RNNs that are incapable of learning across long input sequences of data due to exploding gradient or vanishing gradient difficulties is caused by the explosion of the long-term components. As a result, an LSTM architecture with gates was proposed to illustrate this issue.

A typical LSTM unit consists of a cell and three gate architectures, as illustrated in **Figure .** The cell remembers values for an infinite number of time intervals, and the three gates control the flow of data into and out of the cell. In time t of LSTM model, the input is c_{t-1}, h_{t-1}, x_t, where x_t is the input vector at time t; c_{t-1}, h_{t-1} is output of previous state and c_t, h_t is the output at time t where c is cell state, h is a hidden state.

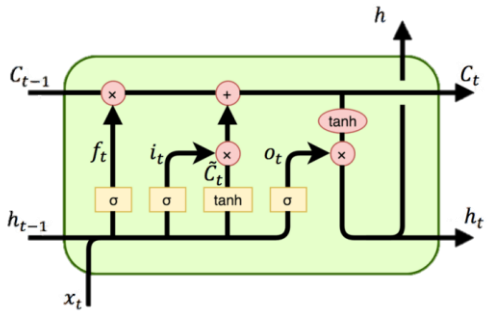


Figure 4. LSTM architecture

The LSTM determines f_t, i_t, o_t corresponding to forget gate, output gate and input gate and the candidate state of memory $\tilde{c_t}$, respectively. The three gates have values in the range $0 < f_t, i_t, o_t < 1$ where b_f, b_i, b_o are bias parameters.

The forget gate f_t determines which data should be discarded or retained. The σ is used to convey information from the prior hidden state and the current input. The closer the value is to 0, the more likely it is to be forgotten, and the closer it is to 1, the more likely it is to be retained.

$$f_t = \sigma\left(U_f * x_t + W_f * h_{t-1} + b_f\right) \tag{5}$$

The input gate i_t is used to update the state of the cell by inputting the previous concealed state and the current input into σ. This determines which values are updated by either changing them to be between 0 and 1 or by passing the hidden state and current input to the $tanh$, which squishes values between -1 and 1 to assist in network regulation. Then multiply the output of $tanh$ by the output of σ. The σ output will determine which information from the $tanh$ output should be retained.

$$i_t = \sigma(U_i * x_t + W_i * h_{t-1} + b_i) \tag{6}$$

By inputting the previous hidden state and the current input into a sigmoid function, the output gate o_t determines the next hidden state. Then, calling the $tanh$ function with the newly updated cell state and multiplying the $tanh$ output by the σ output determine what information the concealed state should contain. The hidden state is returned as the output. The changed cell state and concealed value are then passed forward to the following steps.

$$o_t = \sigma(U_o * x_t + W_o * h_{t-1} + b_o) \tag{7}$$

Candidate memory $\tilde{c_t}$ could be added to the state

$$\tilde{c_t} = tanh(U_c * x_t + W_c * h_{t-1} + b_c) \tag{8}$$

5. EXPERIMENTAL RESULT

The experiment was conducted on Google Colab Pro, Intel(R) Xeon(R) CPU @ 2.30GHz with 26.30 GB RAM and Tesla P100-PCIE-16GB with CUDA 10.1 and the deep learning framework is Keras.

The dataset is divided into 85% for training and 15% for test sets. And split the training data into training and validation with 80:20. The stock price is predicted at time t as a function of the stock price at time $t-1$, $t-2, t-3, \ldots, t-m$, where m is the window size of past stock prices.

Two model trained in this work are presented in **Table 1** result are shown in **Table 2**. LSTM-price and LSTM-STL-price are trained based on historical stock price and technical indicator to predict closing price. It should be noted that LSTM-STL-price is a hybrid model of LSTM and STL decomposed method. The target was normalized across the company's stock, and the output is the stock price of all companies in the dataset.

Three layers of LSTM were used in this work, with the dense layer and flatten layer serving as the final layers for dimensional transformation. The first and second hidden layers' activation functions are tanh, while the third hidden layer's activation function is relu. The window size is 10, and the batch size is 64. In multiple training iterations, the optimizers used during model fitting is Adam, EarlyStopping, and ReduceLROnPlateau callbacks are configured in this model.

The corresponding performance of each model for all 50 selected prices is described in **Appendix A** and **Appendix B**. The results show a certain similarity between these metrics. It is verified that LSTM-STL-price model performed better across all metrics with the lowest value. The experimental results prove the effectiveness of hybrid LSTM-STL over the single model LSTM.

Table 1. Description of model

Model	**Description**
LSTM-price	LSTM model with price and technical indicator as input
LSTM-STL-price	LSTM model with price, technical indicator and decompose to seasonal and trend using STL as input

Table 2. Performance of proposed model and LSTM.

Model	**MSE**	**MAE**	**RMSE**
LSTM-price	0.049454	0.135156	0.187166
LSTM-STL-price	0.046634	0.125776	0.181006

6. CONCLUSION

The research has provided an overview of the stock price forcasting problems and an application of deep learning to these problems. The result indicates the effectiveness of combining LSTM to STL method - a versatile and robust time series decomposition method when compared with a single LSTM architecture. This proposed model takes of time series by analyzing and fitting the characteristics of time series components while single LSTM models are based on the technical indicator and historical observations. In conclusion, the experimental results verified that the proposed hybrid approach model improved accuracy and provided outperformance result.

REFERENCES

[1] Huang, Wei and Nakamori, Yoshiteru and Wang, Shou-Yang, "Forecasting stock market movement direction with support vector machine," Computers & operations research, vol. 32, no. 10, pp. 2513-2522, 2005.

[2] Patel, Jigar and Shah, Sahil and Thakkar, Priyank and Kotecha, Ketan, "Predicting stock market index using fusion of machine learning techniques," Expert Systems with Applications, vol. 42, no. 4, pp. 2162-2172, 2015.

[3] Deboeck, Guido J, Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets, John Wiley & Sons, 1994.

[4] Malkiel, Burton G and Fama, Eugene F, "Efficient capital markets: A review of theory and empirical work," The journal of Finance, vol. 25, no. 2, pp. 383-417, 1970.

[5] Shiller, Robert J, "From efficient markets theory to behavioral finance," Journal of economic perspectives, vol. 17, no. 1, pp. 83-104, 2003.

[6] Wei, William W.S., Time series analysis: Univariate and multivariate methods, 2nd edition, Pearson Education, Inc, 2006.

[7] Montgomery, Douglas C and Jennings, Cheryl L and Kulahci, Murat, Introduction to time series analysis and forecasting, 2nd edition., John Wiley and Sons, 2015.

[8] Zuo, Yi and Kita, Eisuke, "Stock price forecast using Bayesian network," Expert Systems with Applications, vol. 39, no. 8, pp. 6729-6737, 2012.

[9] Adebiyi, Ayodele Ariyo and Adewumi, Aderemi Oluyinka and Ayo, Charles Korede, "Comparison of ARIMA and artificial neural networks models for stock price prediction," Journal of Applied Mathematics, vol. 2014, 2014.

[10] Hossain, Mohammad Asiful and Karim, Rezaul and Thulasiram, Ruppa and Bruce, Neil DB and Wang, Yang, "Hybrid deep learning model for stock price prediction," in 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2018.

[11] Nelson, David MQ and Pereira, Adriano CM and de Oliveira, Renato A, "Stock market's price movement prediction with LSTM neural networks," in 2017 International joint conference on neural networks (IJCNN), 2017.

[12] Milosevic, Nikola, "Equity forecast: Predicting long term stock price movement using machine learning," Journal of Economics Library, vol. 3, no. 2, p. 288, 2016.

[13] Jingyi Shen & M. Omair Shafiq, "Short-term stock market price trend prediction using a comprehensive deep learning system," Journal of big Data, vol. 7, no. 1, pp. 1-33, 2020.

[14] He, Houtian and Gao, Shangce and Jin, Ting and Sato, Syuhei and Zhang, Xingyi, "A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction," Applied Soft Computing, vol. 108, p. 107488, 2021.

[15] Niu, Hongli and Xu, Kunliang and Wang, Weiqing, "A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network," Applied Intelligence, vol. 50, no. 12, pp. 4296-4309, 2020.

[16] Tsay, Ruey S, Analysis of financial time series, vol. 543, John wiley & sons, 2005.

[17] Lux, Thomas and Marchesi, Michele, "Volatility clustering in financial markets: a microsimulation of interacting agents," International journal of theoretical and applied finance, vol. 3, no. 4, pp. 657-702, 2000.

[18] Cleveland, Robert B and Cleveland, William S and McRae, Jean E and Terpenning, Irma, "STL: A seasonal-trend decomposition," J. Off. Stat, vol. 6, no. 1, pp. 3-73, 1990.

[19] Hochreiter, S., & Schmidhuber, J., "Long short-term memory.," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[20] Gan, Christopher and Lee, Minsoo and Yong, Hua Hwa Au and Zhang, Jun, "Macroeconomic variables and stock market interactions: New Zealand evidence," Investment management and financial innovations, vol. 3, no. 4, pp. 89-101, 2006.

[21] Gebhard, Lukas and Hamborg, Felix, "The POLUSA Dataset: 0.9 M Political News Articles Balanced by Time and Outlet Popularity," in Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, 2020.

[22] M. G. Kendall, "The Analysis of Economic Time-Series—Part I: Prices," Journal of the Royal Statistical Society: Series A (General), vol. 116, no. 1, pp. 11-25, 1953.

[23] Yıldırım, D. C., Toroslu, I. H., & Fiore, U., "Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators," Financial Innovation, vol. 7, no. 1, pp. 1-36, 2021.

[24] Moosavi, Sobhan and Samavatian, Mohammad Hossein and Nandi, Arnab and Parthasarathy, Srinivasan and Ramnath, Rajiv, "Short and long-term pattern discovery over large-scale geo-spatiotemporal data," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.

[25] Chang, Shao-Chi and Chen, Sheng-Syan and Chou, Robin K and Lin, Yueh-Hsiang, "Weather and intraday patterns in stock returns and trading activity," Journal of Banking & Finance, vol. 32, no. 9, pp. 1754-2766, 2008.

[26] J. a. S. R. a. M. C. D. Pennington, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014.

[27] C. a. G. E. Hutto, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in Proceedings of the International AAAI Conference on Web and Social Media, 2014.

[28] R. Geoghegan, Time series analysis and its applications: with R examples., Springer, 2006.

[29] J. M. Zurada, Introduction to artificial neural systems, vol. 8, West St. Paul, 1992.

[30] D. E. a. H. G. E. a. W. R. J. Rumelhart, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533-536, 1986.

[31] Osborne, Maury FM, "Brownian motion in the stock market," Operations research, vol. 7, no. 2, pp. 145-173, 1959.

APPENDIX

APPENDIX A. Performance of model LSTM-price of all stocks on dataset

Instruments	Ticker	MSE		MAE		RMSE	
		Normalize Price	Price	Normalize Price	Price	Normalize Price	Price
Forex	EUR/USD	0.1383	0.0059	0.3121	0.0642	0.3718	0.0765
	AUD/USD	0.0045	0.0001	0.0500	0.0072	0.0668	0.0097
	GBP/USD	0.0553	0.0031	0.1891	0.0446	0.2351	0.0555
	USD/CAD	0.0575	0.0017	0.1867	0.0323	0.2398	0.0414
	USD/JPY	0.0381	4.4919	0.1593	1.7306	0.1951	2.1194
Commodities	XAG/USD	0.0861	1.9314	0.2293	1.0863	0.2934	1.3898
	XAU/USD	0.0119	1830.6048	0.0920	36.0444	0.1092	42.7856
Indices	S&P 500	0.1782	103076.3398	0.4179	317.8721	0.4221	321.0550
	DJIA	0.2538	14701541.7484	0.5017	3818.6774	0.5037	3834.2590
	US-TECH 100	0.1770	1528162.0168	0.4111	1208.2099	0.4207	1236.1885
Stocks	AAPL	0.0202	255.7718	0.0742	8.3584	0.1420	15.9929
	AMGN	0.1168	401.2084	0.3187	18.6748	0.3418	20.0302
	AMZN	0.1658	257367.3849	0.3885	484.0362	0.4072	507.3139
	AVGO	0.0563	860.0876	0.2158	26.6824	0.2372	29.3273
	AZN	0.0136	4.9595	0.0487	0.9285	0.1167	2.2270
	BA	0.0407	3283.7544	0.0938	26.6370	0.2018	57.3041
	BABA	0.0630	780.0181	0.1738	19.3291	0.2511	27.9288
	BAC	0.0354	4.2103	0.0800	0.8726	0.1882	2.0519
	BBD	0.0022	0.0983	0.0165	0.1095	0.0472	0.3136
	C	0.0141	14.6567	0.0599	1.9337	0.1187	3.8284
	CMCSA	0.0722	153.9998	0.1216	5.6141	0.2687	12.4097
	CS	0.0008	0.0772	0.0195	0.1901	0.0285	0.2779
	CSCO	0.0175	13.5458	0.0540	1.5005	0.1324	3.6805

CVX	0.0156	17.6078	0.0504	1.6915	0.1249	4.1962
DIS	0.0013	3.2420	0.0217	1.0999	0.0355	1.8006
FB	0.0179	160.7235	0.0654	6.1927	0.1338	12.6777
GILD	0.0075	6.3420	0.0388	1.1288	0.0865	2.5183
GOOG	0.1768	43776.9453	0.4036	200.8831	0.4204	209.2294
GOOGL	0.1826	42732.9812	0.4094	198.0201	0.4273	206.7196
GS	0.0035	52.8450	0.0225	2.7791	0.0590	7.2695
HD	0.0210	180.9723	0.0708	6.5688	0.1450	13.4526
IBM	0.0257	150.7851	0.1006	7.7073	0.1603	12.2795
INTC	0.0277	18.9994	0.0729	1.9074	0.1666	4.3588
ITUB	0.0011	0.0946	0.0137	0.1284	0.0328	0.3076
JNJ	0.0217	30.0479	0.0730	2.7163	0.1473	5.4816
JPM	0.0504	71.0139	0.1174	4.4045	0.2246	8.4270
KO	0.0065	1.4546	0.0353	0.5287	0.0804	1.2061
LVS	0.0116	13.2159	0.0509	1.7207	0.1075	3.6354
MSFT	0.0258	160.1354	0.0999	7.8736	0.1606	12.6545
NFLX	0.0411	3309.0271	0.0990	28.0915	0.2026	57.5241
PBR	0.0080	2.4719	0.0355	0.6239	0.0896	1.5722
PCLN	0.0552	23847.1631	0.1921	126.2536	0.2349	154.4253
PFE	0.0237	5.6178	0.0723	1.1138	0.1539	2.3702
PG	0.0004	1.1474	0.0130	0.6623	0.0210	1.0712
T	0.0316	7.9392	0.1036	1.6412	0.1779	2.8177
TSLA	0.0422	1902.8185	0.1703	36.1720	0.2054	43.6213
V	0.0230	241.8959	0.0863	8.8394	0.1518	15.5530
VALE	0.0254	1.8106	0.0672	0.5675	0.1594	1.3456
WFC	0.0025	1.3350	0.0294	0.6834	0.0497	1.1554
XOM	0.0036	2.1804	0.0286	0.7005	0.0604	1.4766

APPENDIX B. Performance of model LSTM-STL-price of all stocks on dataset

Instruments	Ticker	MSE		MAE		RMSE	
		Normalize Price	Price	Normalize Price	Price	Normalize Price	Price
Forex	EUR/USD	0.1452	0.0061	0.3207	0.0660	0.3811	0.0784
	AUD/USD	0.0035	0.0001	0.0426	0.0062	0.0592	0.0086
	GBP/USD	0.0489	0.0027	0.1752	0.0413	0.2212	0.0522
	USD/CAD	0.0546	0.0016	0.1899	0.0328	0.2337	0.0404
	USD/JPY	0.0369	4.3499	0.1562	1.6962	0.1920	2.0856
Commodities	XAG/USD	0.0097	1490.2150	0.0815	31.9425	0.0985	38.6033
	XAU/USD	0.0849	1.9055	0.2233	1.0579	0.2914	1.3804
Indices	S&P 500	0.1592	92103.9813	0.3945	300.0082	0.3990	303.4864
	DJIA	0.2312	13393193.3294	0.4787	3643.9554	0.4808	3659.6712
	US-TECH 100	0.1648	1423103.0497	0.3907	1148.1105	0.4059	1192.9388
Stocks	AAPL	0.0176	223.0186	0.0552	6.2191	0.1326	14.9338
	AMGN	0.0235	80.6257	0.0645	3.7778	0.1532	8.9792
	AMZN	0.1725	267758.3049	0.3995	497.7274	0.4153	517.4537
	AVGO	0.0553	845.2678	0.2137	26.4259	0.2351	29.0735
	AZN	0.0135	4.9071	0.0441	0.8409	0.1161	2.2152
	BA	0.0496	3999.9356	0.1424	40.4216	0.2227	63.2450
	BABA	0.0517	639.3163	0.1105	12.2920	0.2273	25.2847
	BAC	0.0332	3.9409	0.0747	0.8148	0.1821	1.9852
	BBD	0.0023	0.1027	0.0203	0.1351	0.0483	0.3204
	C	0.0122	12.7504	0.0502	1.6203	0.1107	3.5708
	CMCSA	0.0742	158.3566	0.1246	5.7532	0.2725	12.5840
	CS	0.0006	0.0569	0.0135	0.1312	0.0245	0.2385
	CSCO	0.0188	14.5618	0.0559	1.5556	0.1372	3.8160
	CVX	0.0174	19.6086	0.0615	2.0658	0.1318	4.4282
	DIS	0.0012	3.1945	0.0130	0.6611	0.0353	1.7873
	FB	0.0147	132.0170	0.0536	5.0758	0.1213	11.4899

GILD	0.0066	5.6035	0.0403	1.1744	0.0813	2.3672
GOOG	0.1924	47659.0881	0.4223	210.1888	0.4387	218.3096
GOOGL	0.1865	43640.8274	0.4141	200.2980	0.4319	208.9039
GS	0.0106	160.5027	0.0810	9.9873	0.1027	12.6690
HD	0.0236	202.6986	0.0696	6.4599	0.1535	14.2372
IBM	0.0227	132.9062	0.0868	6.6492	0.1505	11.5285
INTC	0.0269	18.4264	0.0666	1.7427	0.1640	4.2926
ITUB	0.0009	0.0794	0.0127	0.1190	0.0301	0.2818
JNJ	0.0199	27.5305	0.0632	2.3531	0.1410	5.2469
JPM	0.0472	66.5002	0.0975	3.6607	0.2173	8.1548
KO	0.0056	1.2585	0.0276	0.4135	0.0748	1.1218
LVS	0.0099	11.2695	0.0412	1.3931	0.0993	3.3570
MSFT	0.0218	135.2103	0.0674	5.3138	0.1476	11.6280
NFLX	0.0379	3055.7541	0.0890	25.2725	0.1947	55.2789
PBR	0.0054	1.6781	0.0277	0.4870	0.0738	1.2954
PCLN	0.0687	29670.2581	0.2209	145.2145	0.2621	172.2506
PFE	0.0222	5.2540	0.0692	1.0650	0.1489	2.2922
PG	0.0004	1.1060	0.0092	0.4721	0.0206	1.0517
T	0.0273	6.8578	0.0743	1.1766	0.1653	2.6187
TSLA	0.0434	1959.0755	0.1733	36.7935	0.2084	44.2614
V	0.0205	215.3181	0.0699	7.1607	0.1432	14.6737
VALE	0.0289	2.0614	0.0665	0.5614	0.1701	1.4358
WFC	0.0031	1.6665	0.0291	0.6753	0.0556	1.2909
XOM	0.0021	1.2742	0.0189	0.4621	0.0461	1.1288