Development of a Model for Predicting the Direction of Daily Price Changes in the Forex Market Using Long Short-Term Memory

Watthana Pongsena, Prakaidoy Sitsayabut, Nittaya Kerdprasop, and Kittisak Kerdprasop

***Abstract*—Forex is the largest global financial market in the world. Traditionally, fundamental and technical analysis are strategies that the Forex traders often used. Nowadays, advanced computational technology, Artificial Intelligence (AI) has played a significant role in the financial domain. Various applications based on AI technologies particularly machine learning and deep learning have been constantly developed. As the historical data of the Forex are time-series data where the values from the past affect the values that will appear in the future. Several existing works from other domains of applications have proved that the Long-Short Term Memory (LSTM), which is a particular kind of deep learning that can be applied to modeling time series, provides better performance than traditional machine learning algorithms. In this paper, we aim to develop a powerful predictive model targeting to predicts the daily price changes of the currency pairwise in the Forex market using LSTM. Besides, we also conduct an extensive experiment with the intention to demonstrate the effect of various factors contributing to the performance of the model. The experimental results show that the optimized LSTM model accurately predicts the direction of the future price up to 61.25 percent.**

***Index Terms*—Long short-term memory, time-series data analytic, foreign exchange market, forex.**

I. Introduction

Forex or foreign exchange is the largest global financial market where all currencies are traded. Since this market has been established, the number of Forex traders is rapidly growing [1]. This leads to huge amount of investment around the world that makes trading volume in the market exceeds five trillion US dollars per day [2]. In the Forex market, a pair of two-exchangeable currencies is defined as a quotation where a currency, which is quoted in relation, is called base currency. Another currency, which is used as the reference, is called counter currency. All currency pairs in the Forex market are systematically defined by concatenating the ISO currency codes of the base currency and the counter currency. For example, a symbol “EURUSD” is the indication of the Euro against the US dollar.

Consider the quotation of EURUSD traded at a quotation of 1.5000, EUR is the base currency and USD is the quote currency. The quotation of EURUSD 1.5000 has the meaning that 1 Euro is exchangeable to 1.5 US dollars. If the EURUSD quotation rises from 1.5000 to 1.5100, that means the relative value of the Euro has increased. This could be because either value of the Euro has strengthened, or the value of US dollar has weakened, or it could be because of both cases, and vice versa if the EURUSD quote drops from 1.5000 to 1.4900. To trade in the Forex market, traders open a buy order of the EURUSD quotation, if they believe that the quotation will be raise up. On the contrary, traders open a sell order of EURUSD if they have confidence that the quotation will be dropped.

Due to the fact that Forex becomes a significant financial market, several researches have been conducted based on different types of trading methods and strategies, which are commonly used for analyzing the movement of the market. Traditional trading methods categorized as fundamental and technical analyses are popular and widely used among traders. Fundamental analysis traders are those making their trading decision based on economic and financial information such as news, financial reports, economic announcements, social and political forces [3]. On the contrary, technical analysis traders use pricing charts to visualize historical data, and then apply technical indicators to study the price movement as it is the easiest way to forecast the market direction [4]. Among many existing trading indicators, they can be classified into four categories: momentum, volatility, trend, and volume.

Momentum is a group of indicators used by many technical traders as they are simple and easy to use. These indicators show the difference between today's closing price and the closing price of the previous *N* days. In general, momentum refers to the price that moves continuously until it is clearly identified as a trend. Momentum indicators indicate a positive trend if the uptrend will continue to remain constant. A negative trend occurs when the downtrend will continue continuously [5]. The popular momentum indicators include the Relative Strength Index [6], Stochastic Oscillator [7], Williams% R [8], Awesome Oscillator [9], True strength index [10], and Money Flow Index [11].

Volatility indicators are very useful for investors to check the potential of the reversal price in the market. The indicators evaluate the fluctuations of the market price. A strong upward trend indicates a decrease in volatility, while a strong downward trend indicates an increase in volatility.

Manuscript received August 13, 2019; revised June 26, 2020.

Watthana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop are with the School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000 Thailand (e-mail: pongsena@hotmail.com, nittaya.k@sut.ac.th, kerdpras@sut.ac.th).

Prakaidoy Ditsayabut is with the School of Biotechnology, Suranaree University of Technology, Nakhon Ratchasima 30000 Thailand (e-mail: prakaidoy_sut@hotmail.com).

doi: 10.18178/ijmlc.2021.11.1.1015

The tendency of a price reversal often occurs when the volatility increases. There are various volatility indicators, such as Average True Range [12] and Bollinger Bands [13].

Trend is a group of indicators that are used to measure the direction or trend of price movements. Indicators in this group include Moving Average Convergence Divergence. [14], Average Directional Movement Index [15], Triple Exponential Average [16], Commodity Channel Index [17], and Exponential Moving Average [18].

Volume indicators are used to measure how a specified financial asset has a trading rate at a given time. Investors often use this information to increase profits and reduce investment risks. The popular volume indicator is Force Index [19]. Although, trading based on technical analysis is the easiest way for investors to visualize the movement of the market price, they are difficult the use in the real-world situations as the market values are changing over the time.

From the last two decades, Artificial Intelligence (AI) is widely applied in several domains particularly the financial domain to achieve smart decision. Many existing research works are based on the popular AI algorithms including Support Vector Machine (SVM) [20], Artificial Neural Network (ANN) [21], Recurrent Neural Network (RNN) [22]. For instance, in the year 2003 to 2004, Kamruzzaman and Sarker conduct several researches based on SVM and ANN [23]-[27]. For ANN, the data from the Moving Average (MA) indicator are used as input data for training the models. Then, the trained models are tested with 6 different currency exchange data, consisting of USD, GBP, JPY, SGD, NZD, CHF versus AUD. The results show that all ANN models provide better performance than the Autoregressive Integrated Moving Average (ARIMA) model. In addition, they also perform a performance evaluation of the developed model using SVM. The model is used to forecast the exchange rate of the six currencies compared to the AUD currency. They also examine the effects of three different kernel functions: linear, polynomial radial basis, and curve function. These three functions affect the performance of the SVM model. The SVM models that use the polynomial kernel provide the best results on trend predictions, whereas the SVM models that use radial basis and polynomial kernel functions provide equally good results. The conclusion of this research is similar to Cao *et al*. [28], who state that parameter configurations have a significant effect on the performance of the SVM model.

In the next two years, Eng *et al*. [29] report empirical evidence that the ANN model can be used to predict foreign exchange rates. They developed the ANN model by using economic fundamentals data as input for training their ANN model. The result shows that economic fundamentals are important to the exchange rate movement. However, the hidden relationship of these factors does not affect the predictive performance of the ANN model, even if they had been used as input variables. Eng and his research team also discuss that the frequency of economic fundamentals which are updated quarterly could affect the performance of the model. The economic indicators that are updated more frequently may result in more effective predictions of ANN.

As the historical data of the Forex market is a time-series data, in 2014, Stankeviciene *et al*. [30] develop a model using the RNN method, which is suitable for analyzing time-series data. They use the highest and the lowest daily pricing data for training and testing the performance of their model. The performance of their model shows that the model based on today's highest and lowest pricing data cannot be used to predict future trend. Therefore, they modified the model by converting the data from the original format to be the standard deviation, skewness and kurtosis statistics which significantly improve the efficiency of the model. However, the fact that RNN method has a significant limitation, called vanishing gradient problem [31], when the range of the input data increase. For this reason, LSTM which is the extension of the RNN, can be used to solve this issue.

In 2008, Huisu *et al*. [32] use a rolling window LSTM model to predict the price of Bitcoin. The input data for this LSTM model contains macroeconomic data, global currency ratios, and blockchain data. In addition, they also compared their model performance with the model-based machine learning principles such as SVR, LR and NN. The results indicate that the rolling window LSTM model accurately predicts the latest bitcoin prices more efficient than other machine learning models mentioned above. However, there are various factors to be considered when applying the LSTM modeling method such as the input length for the LSTM model, the appropriate amount of the training data, and the frequency for updating the model to keep the model robust. In this paper, we conduct the experimentation to demonstrate the effect of these factors on our LSTM model.

The rest of this research is organized as follows. Section II demonstrates how the problem is formulated. Section III describes the data and the methodology used for conducting this research. The empirical results are discussed in section IV. Finally, section V represents our conclusions and makes suggestions for future research.

II. Problem Formulation

LSTM is the technique suitable for time-series prediction where the previous data affect the value that will happen in the future. In this research we aim to use several previous daily data to predict whether the price change of tomorrow will increase or decrease, as schematically shown in Fig. 1. The horizontal axis in the figure represents time-frame unit, while the vertical axis is the price value that we would like to forecast.

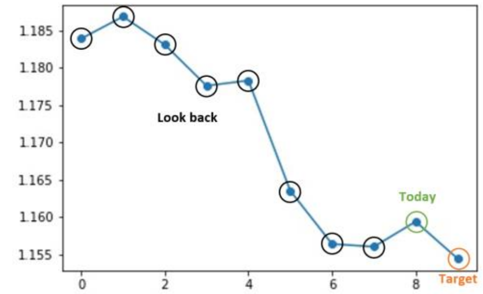

Fig. 1. Problem formulation for LSTM modeling.

Therefore, prior to the training phase to build a model, there are three sub-problems that must be addressed:

1) How many periods should we lookback as the input length for the LSTM modeling?
2) How long of the historical data should we use as the training data for developing the model?
3) When to update the model to keep the model robust to future price and time changes?

III. Materials and Methods

A. Long Short-Term Memory

As LSTM method is the extension of the RNN, we describe the fundamental of the RNN first. RNN is a machine learning method that is very useful for time-series analysis, in which the previous event has affected the event that will happen in the future [33]. The principle of RNN is that it sends input to each node in the network, and each node has mathematical function inside the node to perform some calculation. The computational output value will be forwarded to the next node for further calculation as demonstrated in Fig. 2.

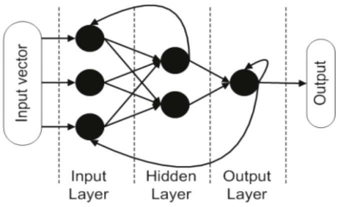

Fig. 2. Structure of the RNN model.

RNN method uses the feed-forward network where the output value will be sent to the next node without being interested in remembering that value. Doing so creates a problem when dealing with back-propagation (BP) or the calculation of errors after the operation of each node ends because BP needs to go back many steps through multiple nodes. This problem called Vanishing Gradient Problem [34]. Note that the gradient helps to know that the change of the initial value affects the outcome, which greatly affects the learning of the entire system as illustrated in Fig. 3.

To overcome the vanishing gradient problem, LSTM uses memory cell state to recognize the previous data. The architecture of LSTM composes of three kinds of nodes, called gates, including input gate, forget gate, and output gate. Input gate considers whether to allow new input or not. Forget gate deletes data when it is not important. Output gate creates output at the current time.

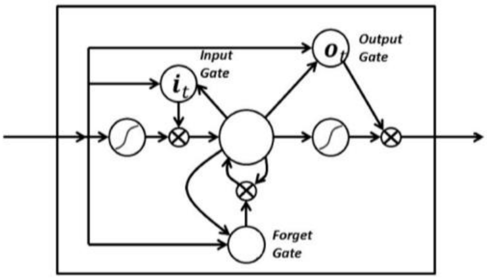

Fig. 3. Three types of gates in LSTM including input gate, forget gate and output gate.

The LSTM models can be used to predict various types of data as follows.

1) One-to-one sequence prediction

This model is suitable for the problem of predicting the order in which we want to predict the next step by entering data in current step (as illustrated in Fig. 4). For example, we use this model to predict the possible next word in the sentence.

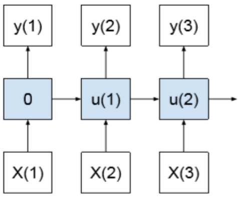

Fig. 4. Structure of the one-to-one sequence prediction of LSTM.

2) One-to-many sequence prediction

This model (Fig. 5) is suitable for sequential prediction problems that we want to generate sequential output for each step of the input time. For example, we use this model to predict the sequence of words from a single image or forecast the observation set from a single event.

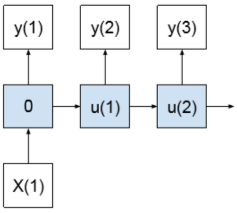

Fig. 5. Structure of the one-to-many sequence prediction of LSTM.

3) Many-to-one sequence prediction

This model (Fig. 6) is suitable for predicting problems that require multiple input time steps to predict the next single step. For example, the next real value forecast in the time series that determines the sequence of data entry observations.

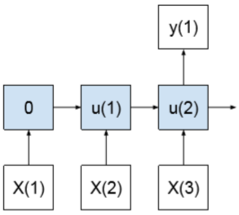

Fig. 6. Structure of the many-to-one sequence prediction of LSTM.

4) Many-to-many sequence prediction

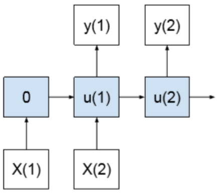

Fig. 7. Structure of the many-to-many sequence prediction of LSTM.

This model (Fig. 7) is suitable for predicting the sequence that requires multiple input-time steps to predict the sequence of time-output steps. This is often called sequence-to-sequence or seq2seq type problems. For example, we use this model to summarize a word document to be a shorter word sequence or classify the order of audio data into a sequence of words.

In this research, we use the One-to-Many Sequence Prediction of LSTM to build a predictive model as it is fit to our formulated problem.

B. Dataset

TABLE I: THE EXAMPLE OF HISTORICAL DATA OF EURUSD

Date	Open	High	Low	Close	Volume
2015.01.04	1.19114	1.19530	1.18672	1.1946	6936
2015.01.05	1.19495	1.19753	1.18864	1.19366	37558
2015.01.06	1.19361	1.19683	1.18486	1.18715	38120
2015.01.07	1.1872	1.18954	1.18013	1.18327	40248
2015.01.08	1.18321	1.18466	1.17537	1.17913	36991
2015.01.09	1.17905	1.18443	1.17621	1.18387	63687
2015.01.11	1.18523	1.18691	1.18357	1.18689	4385
2015.01.12	1.18676	1.18704	1.17856	1.18312	73332
2015.01.13	1.18309	1.1859	1.17531	1.17761	79404
2015.01.14	1.17759	1.18449	1.17267	1.17832	79148

The data used for analyzing in this research are exported from a Forex trading platform called Meta Trader 4 (MT4) of FXPro [35]. The time-frame is daily quotation between EUR and USD starting from 2st January 2013 to 29th December 2018 (approximately 1,500 records). Table I shows the example of the historical data of the EURUSD.

C. Development of the LSTM Model

In this section, the process of how we develop the LSTM models is demonstrated. The process consists of three sub-processes including data transformation, model building and performance optimizing of the LSTM model, and performance evaluation as illustrated in Fig. 8.

This process consists of three steps including data transformation, building and optimizing performance of the LSTM model, and performance evaluation. The detail of each step demonstrated as the following.

1) Data transformation

To formulate the problem as the classification problem for LSTM model, the dataset needs to be transformed from continuous to be an ordinal number. We start with creating a new variable (column) that represents the movement of the daily price with possible value either 0, 1, or 2. If the price movement is "Up", its representation is "2". For the "Down" movement, we represent it as "1". The number "0" is for "Neural" movement.

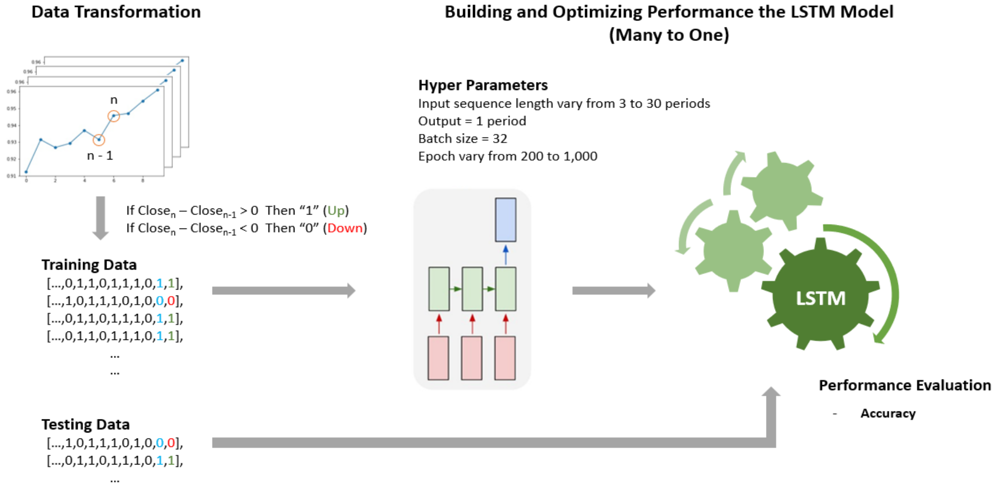

Fig. 8. Process of the LSTM model development.

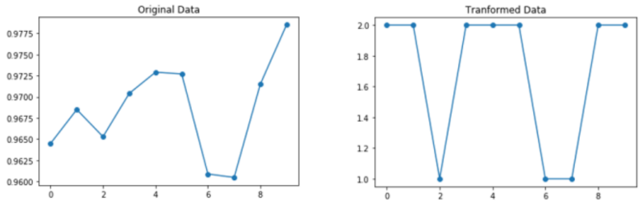

Fig. 9. Data transformation process.

Fig. 9 shows the result of the data transformation process. The original sequence [1.14660, 1.14940, 1.14380, 1.14140, 1.15201, 1.14700, 1.15190, 1.16300, 1.17350, 1.17840] is transformed to be sequence [1, 2, 1, 1, 2, 1, 2, 2, 2, 2].

2) Building and optimizing performance of the LSTM model

The LSTM models are implemented using Python with TensorFlow and Keras [36]–[38], which is a powerful machine learning and deep learning framework provided by Google. To obtain the effective model, we divide the model building and model optimization process into three phases as follows.

Phase 1: Find suitable input length of the sequence for LSTM modeling

In this phase, we evaluate the performance of the LSTM models based on different input length varying from 3 to 30 days using the historical data from 2013 to 2017 as a training data, while the testing data is the historical data of the year 2018. At the end of this phase, the performances of all models are evaluated in terms of accuracy of the prediction.

Phase 2: Find the suitable period to be used for training and testing the model (long period vs. short period of the training data).

In this phase, the dataset is divided into training and testing set using sliding-windows technique [37]. This technique is used for preventing the model from the overfitting situation. For the long period of the training data, five-year data are used as testing data starting from the year 2013, and data in the next year is used for evaluating performance of the model. Then, we move forward for 1 year until the end of the year 2018. For the short period of the training data, one-year data are used for training, and the next one-year data are used as testing data. Then, the time-frame has been slid to the next year. After the performances of all models are evaluated in terms of accuracy, we compare the average accuracy of the two types of the LSTM models to find the suitable period to be used for training and testing the model. The input length of the sequence for LSTM model is based on the results of the experiment in phase 1.

Phase 3: Evaluate performances of the models, which are updated annually vs. monthly.

In the final phase, we evaluate the performances of the models which are updated annually compared with the models which are updated monthly using data of the year 2018. The period used for training and testing the model is based on the results from the previous phase.

3) Performance evaluation

The performance of the proposed LSTM model is evaluated in terms of accuracy of the prediction. The accuracy can be computed using the equation 1.

$$\text{Accuracy} = (TP + TN) / (TP + FN + FP + TN) \qquad (1)$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. TP, TN, FP, and FN are derived from the Confusion Matrix table, which is the result of performance evaluation of the models. The structure of the Confusion Matrix table is shown in Table II.

TABLE II: CONFUSION MATRIX TABLE

		Actual Data	
		Positive	**Negative**
Predicted Data	**Positive**	*TP*	*FP*
	Negative	*FN*	*TN*

IV. RESULTS AND DISCUSSIONS

In this section, the result of the experimentation based on the three phases described in Section III are demonstrated as follows.

A. Find Suitable Input Length of the Sequence for LSTM Model

Fig. 10 presents the prediction accuracy of the LSTM models based on different input length varying from 3 to 30 periods. As is illustrated in Fig. 10, the models that use 9, 10 and 12 periods as input data for training the models provide the best accuracy equally at 52 percent. Based on these results, it could be concluded that the suitable input length of the sequence for LSTM model should range between 9 and 12. Therefore, we select the 10 periods of data as the input length for training the model in the next phase.

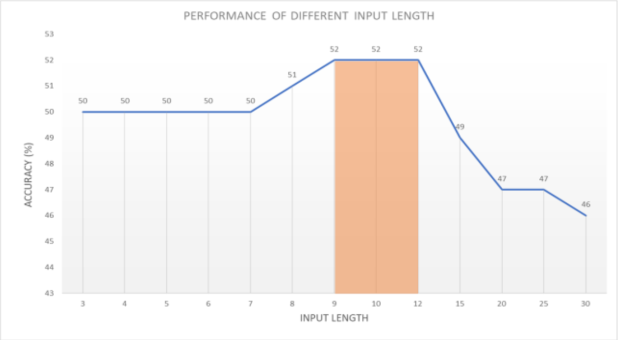

Fig. 10. Performance of the LSTM model based on different input length varying from 3 to 30 periods.

B. Find the Suitable Period to Be Used for Training and Testing the Model

For the long period of training data, we use historical data comprising totally five years, while data in the next one year are used to test the performance of the model. Train-test schemes are listed as follows.

1) Training data from 2013 to 2017 – Testing data 2018
2) Training data from 2012 to 2016 – Testing data 2017
3) Training data from 2011 to 2015 – Testing data 2016
4) Training data from 2010 to 2014 – Testing data 2015

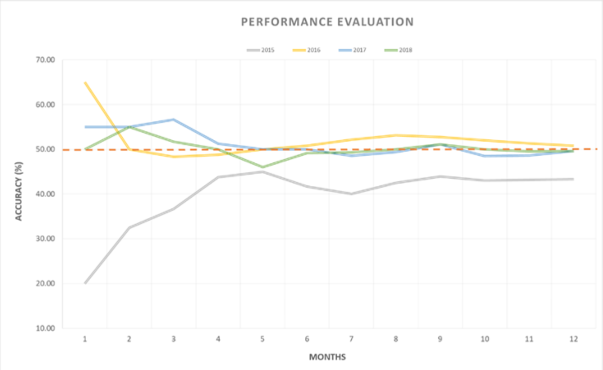

Fig. 11. Performance of the LSTM model based on long period of the training data.

Fig. 11 shows accuracy of the LSTM model based on long period of the training data. As can be seen in Fig. 11, at the first 4 months in the year 2016, 2017, and 2018, the accuracy of the models is over 50 percent. After that the accuracy is fluctuated near and around the 50 percent threshold line. For the year 2015, the performance of the model is less than 50 percent.

For the short period of training data, we use historical data totally one year for training the models and use data in the next one year for testing the performance of the models. These train-test schemes are summarized as follows.

1) Training data from 2017 – Testing data 2018
2) Training data from 2016 – Testing data 2017
3) Training data from 2015 – Testing data 2016
4) Training data from 2014 – Testing data 2015

Fig. 12 reveals the accuracy of the LSTM model based on short period of the training data. It can be seen from Fig. 12 that the accuracies of the models testing during the year 2015 to 2018 are over 50 percent in the first two months. After that accuracies for all testing year are slightly drop after three months.

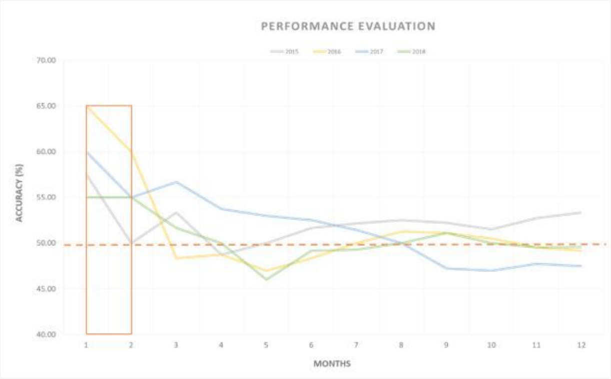

Fig. 12. Performance of the LSTM model based on short period of the training data.

Based on the results shown in Fig. 11 and 12, it could be concluded that training the model with short period of the training data is better than long period of the training data. For the next phase, we thus use short period of data to train the model.

C. Performances of the Models, Which Are Updated Monthly, Quarterly, and Annually

In this phase of experimentation, we select data from the year 2017 as training data and use data from the year 2018 as a testing set. The purpose of this phase is to compare the performances of the models that are updated monthly, quarterly, and annually. Note that updating the model monthly and quarterly mean that we re-train the model in every month and three months, respectively, with a new monthly data, then test the performance of the re-trained model.

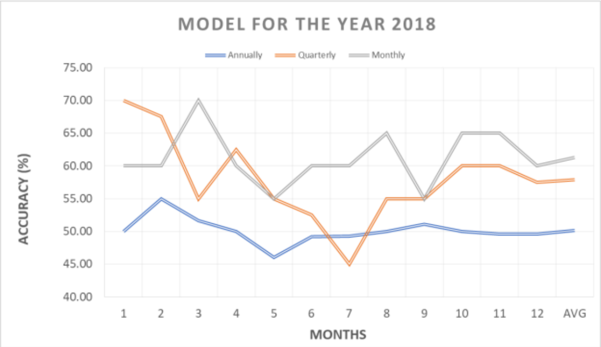

Fig. 13. Average accuracy of the model using one-year data as the training and testing data vs. the model which are updated quarterly.

Fig. 13 shows the average accuracy of the model using one-year data as the training and testing data compared to the model which is updated quarterly. As illustrated in the figure, the model updated monthly provides the best accuracy at 61.25 percent, followed by the model updated quarterly and annually at 57.92 and 50.11 percent, respectively. These findings indicate that re-train the model by feeding new data frequently can enhance performance of the model.

V. Conclusion

In this research, we develop the deep learning model-based LSTM algorithm targeting to predict the direction of daily price changes in the Forex market. Despite the efficiency of LSTM model on learning time-series data, developing such high-efficient model requires the consideration of various factors such as the input length appropriate for training the LSTM model, the appropriate amount of the training data, and the frequency for updating the model to keep the model robust and effective. Therefore, not only creating an LSTM model, we also conduct an extensive series of experiments aiming to demonstrate the effect of these factors on the performance of the LSTM model.

The experimental results lead to the following conclusions. Firstly, the suitable input length of the sequence for creating the accurate LSTM model should range between 9 and 12 periods. Secondly, training the model with short period of the training data is better than training with the long period of data. Finally, feeding new data by re-train the model on a monthly basis can increase the performance of the model.

Based on these experiments, our final LSTM model can accurately predict the direction of daily price changes in the Forex market at 61.25 percent. For future works, we aim to enhance the performance of the model by tuning-up the structure of LSTM model. In addition, we expect to implement the trading system that can automatically provide a daily signal according to the predicting signal from the proposed model.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Watthana Pongsena conducted the research and wrote the paper. All authors had approved the final version of the paper.

Acknowledgment

The authors are grateful to National Research Council of Thailand, Ministry of Science and Technology, and Suranaree University of Technology for granting the research funds.

References

[1] W. Pongsena, P. Ditsayabut, N. Kerdprasop, and K. Kerdprasop, “An analysis of the co-movement of price change volatility in forex market,” in *Proc. the World Congress on Engineering*, 2018, vol. 1.

[2] L. L. Jacque, *Management and Control of Foreign Exchange Risk*, Springer Science & Business Media, 2013.

[3] J. S. Abarbanell and B. J. Bushee, “Fundamental analysis, future earnings, and stock prices,” *J. Account. Res*., vol. 35, no. 1, pp. 1–24, 1997.

[4] A. W. Lo, H. Mamaysky, and J. Wang, “Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation,” *J. Finance*, vol. 55, no. 4, pp. 1705–1765, 2000.

[5] M. H. Pee, “Trend detection index,” T*ech. Anal. Stock. Commod.*, vol. 19, no. 10, pp. 54–61, 2001.

[6] T. T.-L. Chong and W.-K. Ng, “Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30,” *Appl. Econ. Lett.*, vol. 15, no. 14, pp. 1111–1114, 2008.

[7] K. S. Kannan, P. S. Sekar, M. M. Sathik, and P. Arumugam, “Financial stock market forecast using data mining techniques,” in *Proc. the International Multiconference of Engineers and Computer Scientists*, 2010, vol. 1, p. 4.

[8] R. C. Williams, “Oscillator circuit,” Google Patents, 2000.

[9] S. Russell and V. Yoon, “Heterogeneous agent development: A multi-agent system for testing stock trading algorithms,” in *Proc. AMCIS*, 2005, p. 283.

[10] A. Gorgulho, R. Neves, and N. Horta, "Applying a GA kernel on optimizing technical analysis rules for stock picking and portfolio composition," *Expert Syst. Appl*., vol. 38, no. 11, pp. 14072–14085, 2011.

[11] H. Chen, T. T.-L. Chong, and X. Duan, "A principal-component approach to measuring investor sentiment," *Quant. Financ*., vol. 10, no. 4, pp. 339–347, 2010.

[12] R. K. Nayak, D. Mishra, and A. K. Rath, "A naive SVM-KNN based stock market trend reversal analysis for Indian benchmark indices," *Appl. Soft Comput.*, vol. 35, pp. 670–680, 2015.

[13] J. Bollinger, "Using bollinger bands," *Stock. Commod*., vol. 10, no. 2, pp. 47–51, 1992.

[14] G. Appel, "Become your own technical analyst: How to identify significant market turning points using the moving average convergence-divergence indicator or MACD," *J. Wealth Manag*., vol. 6, no. 1, pp. 27–36, 2003.

[15] T. Chenowethl, Z. Obradovic, and S. Lee, "Technical trading rules as a prior knowledge to a neural networks prediction system for the S&P 500 index," in *Proc. IEEE Technical Applications Conference and Workshops. Northcon/95. Conference Record*, 1995, p. 111.

[16] J. K. Hutson, "TRIX-triple exponential smoothing oscillator," *Tech. Anal. Stock. Commod*., pp. 105–108, 1983.

[17] D. R. Lambert, "Commodity channel index: Tool for trading cyclic trends," *Tech. Anal. Stock. Commod.*, vol. 1, p. 47, 1983.

[18] U. A. Müller, "Specially weighted moving averages with repeated application of the ema operator," *Intern. Doc. UAM. 1991--10-14*, Olsen Assoc. Switz., 1991.

[19] A. Elder, *Trading for a Living: Psychology, Trading Tactics, Money Management*, vol. 31, John Wiley & Sons, 1993.

[20] D. Meyer and F. H. T. Wien, "Support vector machines," *Interface to libsvm Packag. e1071*, p. 28, 2015.

[21] A. H. Moghaddam, M. H. Moghaddam, and M. Esfandyari, "Stock market index prediction using artificial neural network," *J. Econ. Financ. Adm. Sci.*, vol. 21, no. 41, pp. 89–93, 2016.

[22] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," arXiv Prepr. arXiv1605.05101, 2016.

[23] J. Kamruzzaman and R. Sarker, "Comparing ANN based models with ARIMA for prediction of forex rates," *Asor Bull*., vol. 22, no. 2, pp. 2–11, 2003.

[24] J. Kamruzzaman and R. A. Sarker, "Forecasting of currency exchange rates using ANN: A case study," in *Proc. International Conference on Neural Networks and Signal Processing*, 2003, vol. 1, pp. 793–797.

[25] J. Kamruzzaman and R. A. Sarker, "ANN-based forecasting of foreign currency exchange rates," *Neural Inf. Process. Rev*., vol. 3, no. 2, pp. 49–58, 2004.

[26] J. Kamruzzaman and R. A. Sarker, "Application of support vector machine to forex monitoring," *IEEJ Trans. Electron. Inf. Syst*., vol. 124, no. 10, pp. 1944–1951, 2004.

[27] J. Kamruzzaman, R. A. Sarker, and I. Ahmad, "SVM based models for predicting foreign currency exchange rates," in *Proc. Third IEEE International Conference on Data Mining*, 2003, pp. 557–560.

[28] D.-Z. Cao, S.-L. Pang, and Y.-H. Bai, "Forecasting exchange rate using support vector machines," in *Proc. International Conference on Machine Learning and Cybernetics*, 2005, vol. 6, pp. 3448–3452.

[29] M. H. Eng, Y. Li, Q.-G. Wang, and T. H. Lee, "Forecast forex with ANN using fundamental data," in *Proc. 2008 International Conference on Information Management, Innovation Management and Industrial Engineering*, 2008, vol. 1, pp. 279–282.

[30] J. Stankevičien, N. Maknickien, and A. Maknickas, "Investigation of exchange market prediction model based on highlow daily data," in *Proc. the 8th International Scientific Conference "Business and Management*, 2014, pp. 15–16.

[31] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 6, no. 02, pp. 107–116, 1998.

[32] J. Huisu, J. Lee, H. Ko, and W. Lee, "Predicting bitcoin prices by using rolling window LSTM model," *ACM*, July 2018.

[33] V. L. Bernard and J. K. Thomas, "Evidence that stock prices do not fully reflect the implications of current earnings for future earnings," *J. Account. Econ.*, vol. 13, no. 4, pp. 305–340, 1990.

[34] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," *CoRR, abs/1211.5063*, vol. 2, 2012.

[35] M. Wong, L. Chan, and W.-K. Wong, "Automated gold trading with MT4," *Wong Manyat, Chan LCWJ Wong WK Autom. Gold Trading with MT4. Finamatrix J*., 2014.

[36] A. Gulli and S. Pal, *Deep Learning with Keras*, Packt Publishing Ltd, 2017.

[37] L. Rampasek and A. Goldenberg, "Tensorflow: Biology's gateway to deep learning?" *Cell Syst.*, vol. 2, no. 1, pp. 12–14, 2016.

[38] R. Shanmugamani, "Deep learning for computer vision: Expert techniques to train advanced neural networks using TensorFlow and Keras," Packt Publishing Ltd, 2018.

Watthana Pongsena is a Ph.D. student with the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his B.E. and M.E. in computer engineering from Suranaree University of Technology in 2008 and 2012, respectively. His research of interest includes software engineering, data mining, artificial intelligence, and human-computer interaction.

Prakaidoy Sitsayabut is a Ph.D. student in the School of Biotechnology, Institute of Agricultural Technology, Suranaree University of Technology, Thailand. She graduated with a bachelor degree of animal production of technology from Suranaree University of Technology in 2006. After graduation, in 2012, she graduated with the master degree from the School of Biotechnology, Institute of Agricultural Technology, Suranaree University. Her research of interest includes animal reproductive technology and data mining.

Nittaya Kerdprasop is an associate professor and the head of Data Engineering Research Unit, School of Computer Engineering, Suranaree University of Technology, Thailand. She received her B.S. in radiation techniques from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. Her research of interest include data mining, artificial intelligence, logic, and constraint programming.

Kittisak Kerdprasop is an associate professor at the School of Computer Engineering, Chair of the School, and the head of Knowledge Engineering Research Unit, SUT. He received his bachelor degree in mathematics from Srinakarinwirot University, Thailand, in 1986, MS in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. His current research includes machine learning and artificial intelligence.