https://doi.org/10.48048/tis.2022.3045

A Gaussian Process Regression Model for Forecasting Stock Exchange of Thailand

Kamonrat Suphawan, Ruethaichanok Kardkasem and Kuntalee Chaisee*

Data Science Research Center, Department of Statistics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand

(*Corresponding author's e-mail: kuntalee.chaisee@cmu.ac.th)

Received: 1 April 2021, Revised: 6 July 2021, Accepted: 16 July 2021

Abstract

A stock price index measures the change in several share prices, which can describe the market and assist investors in deciding on a specific investment. Thus, foreseeing the stock price index benefits investors in creating a better investment strategy. However, forecasting the stock price index can be challenging due to its non-linearity, non-stationary and high uncertainty. Gaussian process regression (GPR) is an attractive and powerful approach for prediction, especially when the data fluctuates over time with fewer restrictions. Besides, the GPR gains advantages over other forecasting techniques as it can offer predictions with uncertainty to provide margin errors. In this study, we evaluate the use of GPR to predict the stock price of Thailand (SET). The SET data are divided into 2 datasets; the data in the year 2015 - 2020 and the data in the year 2020 due to the massive change during the COVID-19 pandemic. The prediction results from the GPR are then compared to the machine learning approaches, artificial neural network (ANN) and recurrent neural network (RNN) using evaluation scores; the root mean square error (RMSE), the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the Nash-Sutcliffe efficiency (NSE). The results indicate that the GPR is superior to the ANN and RNN for both datasets as it provides a high prediction accuracy. Moreover, the results suggest that the GPR is less sensitive to the number of input lags in the model. Therefore, the GPR is more favorable for the prediction of SET than the ANN and RNN.

Keywords: Stock price index, Gaussian process regression, Artificial neural network, Recurrent neural network

Introduction

The prediction of stock price index movement remains an attractive and challenging task for researchers and investors. Estimating the stock price or the stock index can provide the best trading strategy in advance that brings a high profit or reduce a potential loss. The data of stock prices and stock indexes are non-linear and non-stationary that can be troublesome to establish the methods for predicting future values. Over the past several years, many techniques are developed to apply in stock price and stock index forecasting [1-4].

Statistical analysis of time-series, particularly autoregressive models, is one of the most broadly used methods for predicting future values and illustrating future value movement based on historical data. However, it relies on the data pattern, which might not be mathematically simple to establish. Machine learning, such as an ANN and RNN, is a suitable candidate to substitute for time-series analysis. The machine learning methods can automatically learn a complex structure like trends and seasonality of a time-series by forming inputs and outputs without assuming the transition function from inputs to outputs. Another approach, which has been gained popularity in statistical modeling and machine learning because it is a powerful method with few restrictions, is GPR. GPR is a Bayesian approach intended to model a non-linear regression that can learn an unknown transition function in a regression problem with fewer conditions in the model's choice. As a result, GPR and machine learning have been applied in diverse research fields. For example, in environmental research [5], GPR is used to predict $PM_{2.5}$ and PM_{10} in the Seoul Metropolitan area. The GPR provided a reasonable degree of accuracy. In [6], GPR and machine learning models are used to forecast the drought index in Iran. They found that the GPR provided more accurate results than the other models based on RMSE and R^2. In agricultural research [7], GPR and the modification of machine learning methods are used to evaluate diet nutrient digestibility. They found that the GPR is superior to other methods. In [8], the GPR model with the sparse procedure in computation is used for forecasting non-stationary tourism series in Hong Kong. The proposed sparse GPR model showed that its forecasting capability is superior to ARMA and support vector machine. In [9], the GPR and ANN are studied for intraday distribution grid load forecasting. The results showed that the GPR provided good forecasting values. Furthermore, it can offer a confidence interval that is a strong motivation for using GPR in grid load forecasting.

In stock market and finance, time-series modeling and machine learning are mainly used for stock price prediction [10-12]. In [13], the GPR is applied to forecast foreign exchange returns volatility. They suggested that the multivariate GP provided the best approximation. According to the literature review, GPR and machine learning modeling is attractive and potentially provide good accuracy in time-series data, notably when it fluctuates over time with relaxing mathematical assumptions. Besides, the GPR can offer the variability and margin errors of the predictive values so that it is regarded as a powerful tool to illustrate the uncertainty from the forecasting. In this paper, we evaluate the performance of the GPR for forecasting Stock Exchange Thailand (SET) in 2015 - 2020. The SET data are divided into 2 datasets; the total data and data in the year 2020 because of the sharp drop in the stock prices in 2020 due to the COVID-19 pandemic that might affect the prediction. Then the efficiency of the GPR is determined using RMSE, MAE, MAPE and NSE scores compared to the machine learning methods; ANNs and RNNs. The 95 % confidence intervals for the mean predictive values are provided to illustrate the uncertainty of the forecast.

Materials and methods

Data

This study uses the SET index from 5 January 2015 to 30 June 2020, excluding closed market days. **Figure 1** illustrates the time-series plot of SET over the period of study. The primary statistical descriptions of the annual SET index are shown in **Table 1**.

Table 1 Descriptive statistics of the SET index.

Year	N	Mean	SD	Min	1st Q	Median	3rd Q	Max	Skew
2015	243	1460.3	89.5	1261.6	1389.1	1483.2	1526.5	1615.8	–0.22
2016	244	1433.0	84.0	1224.8	1390.4	1444.2	1502.4	1552.6	–0.68
2017	244	1612.3	62.0	1535.5	1568.8	1577.8	1672.8	1753.7	0.86
2018	245	1720.5	71.2	1548.3	1667.9	1722.0	1784.9	1838.9	–0.25
2019	244	1639.9	43.8	1548.6	1613.9	1636.0	1664.9	1740.9	0.38
2020	123	1354.4	156.9	1024.4	1260.3	1341.9	1513.4	1600.4	–0.19

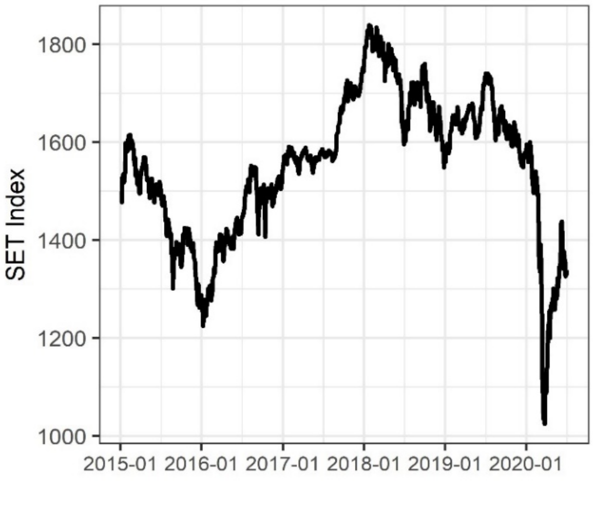

Figure 1 Time-series plot of the SET index from 5 January 2015 to 30 June 2020.

Figure 1 and **Table 1** show that the mean and the standard deviation of the SET index are varied over time. It shrank from 2015 to 2016 and grew to the highest value at the beginning of 2018 before declining afterward. In 2020, it dropped rapidly in February and March due to the impact of COVID-19 before showing a recovery in April. This situation creates a high variation. To evaluate the model, we divide the data into 2 sets; the 1st 80 % of the data is the training data. The remaining 20 % is the testing data. Due to the fluctuation in 2020, we also evaluate the model using data in 2020 separately. Therefore, we have 2 datasets to model; Data 1 is from 5 January 2015 to 30 June 2020, and Data 2 is from 2 January 2020 to 30 June 2020.

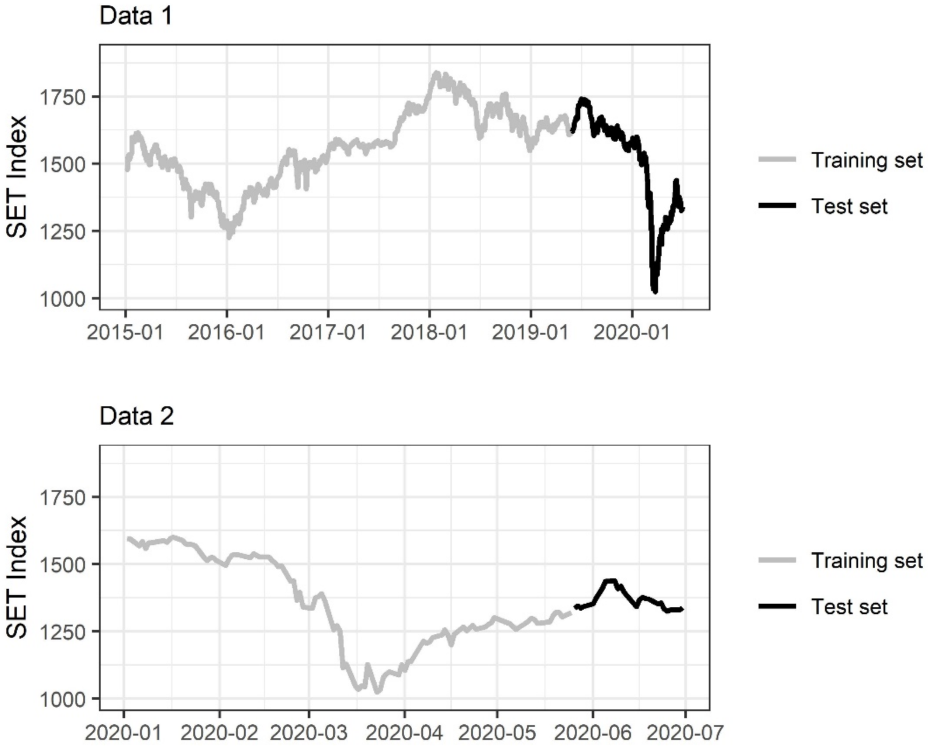

Figure 2 Time-series plot of Data 1 and Data 2.

Table 2 Datasets for modeling and model validation.

Datasets	Training		Testing	
	Study period	Days	Study period	Days
Data 1	5 January 2015 to 28 May 2019	1,074	29 May 2015 to 30 June 2020	269
Data 2	2 January 2020 to 25 May 2020	98	26 May 2020 to 30 June 2020	25

Gaussian process regression (GPR)

A Gaussian process (GP) is a stochastic process made up of a collection of random variables where each random variable has a Gaussian distribution, and any finite number of those random variables has a multivariate Gaussian distribution. The GP can be applied in the regression model, called Gaussian process regression (GPR). A regression model is defined as $\mathbf{y} = f(\mathbf{X}) + \epsilon$, where $\mathbf{y} = (y_1, y_2, \dots, y_n)$ is observation vector, $f(x)$ is an underlying function mapping from a collection of input vectors, $\mathbf{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_d})$ to output, where input vector $\boldsymbol{x}$ as a collection of inputs $\boldsymbol{x_i} = (x_1, \dots, x_n)', i = 1, \dots, d$, and $\boldsymbol{\epsilon}$ is the additive noise term assumed to be Gaussian with zero mean and constant variance, $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma_\epsilon^2 I)$, where I is an n-dimensional unit matrix. In the GPR, the function $f(x)$ is defined with a GP as:

$$\mathrm{f}(\mathbf{X}) \sim GP\big(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X}')\big). \tag{1}$$

The vector $\mathbf{f} = f(\mathbf{X})$ has a multivariate Gaussian distribution, $\mathbf{f}|\mathbf{X} \sim N(m(\mathbf{X}), K)$ where $m(\mathbf{X})$ is the mean function vector often set to be 0, and $K = k(\mathbf{X}, \mathbf{X}')$ is the covariance matrix. In this work, we use the squared exponential (SE) covariance, which is the most commonly used covariance function in GP modeling:

$$k_{SE}(x, x') = h^2 \exp\left[-\frac{1}{2}\left(\frac{x - x'}{l}\right)^2\right]. \tag{2}$$

We define $\theta = \{h^2, l\}$ to be a set of hyper-parameters for a given mean and covariance function. More details on GP covariance functions can be found in [14].

Hence, the distribution of the output $\boldsymbol{y}$ given $\mathbf{f}$ is $\mathbf{y}|\mathbf{f} \sim N(\mathbf{f}, \sigma_\epsilon^2 I)$ and $\mathbf{y}|\mathbf{X} \sim N(m(\mathbf{X}), K + \sigma_\epsilon^2 I)$. We are often interested in predicting new outputs given the training input matrix, $\mathbf{X}$, and training outputs, $\mathbf{y}$. We define the test input matrix as $\mathbf{X}^* = (\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_d^*)$, the test latent function variables $\mathbf{f}^* = f(\mathbf{X}^*)$ and the test output as $\mathbf{y}^* = (y_1^*, \ldots, y_m^*)$. We can write the joint distribution of the training and test latent variables, of the training outputs and latent variables, and the training and test outputs as:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} \sim N\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}^*) \end{bmatrix}, \begin{bmatrix} K & k(\mathbf{X}^*, \mathbf{X}) \\ k(\mathbf{X}, \mathbf{X}^*) & k(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix}\right), \tag{3}$$

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim N\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}^*) \end{bmatrix}, \begin{bmatrix} K + \sigma_\epsilon^2 I & k(\mathbf{X}^*, \mathbf{X}) \\ k(\mathbf{X}, \mathbf{X}^*) & k(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix}\right). \tag{4}$$

Bayes's rule is then applied to obtain the joint posterior distribution of training and test latent variables given the training outputs. The predictive distribution of $\mathbf{f}^*$ at given test inputs is shown as follows:

$$\mathbf{f}^*|\mathbf{X}^*, \mathbf{y}, \theta \sim N(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*),$$

$$\text{where} \quad \boldsymbol{\mu}^* = m(\mathbf{X}^*) + k(\mathbf{X}^*, \mathbf{X})(K + \sigma_\epsilon^2 I)^{-1}\big(\mathbf{y} - m(\mathbf{X})\big) \tag{5}$$

$$\boldsymbol{\Sigma}^* = k(\mathbf{X}^*, \mathbf{X}^*) - k(\mathbf{X}^*, \mathbf{X})(K + \sigma_\epsilon^2 I)^{-1} k(\mathbf{X}, \mathbf{X}^*). \tag{6}$$

The predictive distribution of the target outputs, $\mathbf{y}^*$, is

$$\mathbf{y}^*|\mathbf{X}^*, \boldsymbol{y}, \theta \sim N(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^* + \sigma_\epsilon^2 I). \tag{7}$$

The point predictions of GPR are the mean vector $\boldsymbol{\mu}^*$. The GPR prediction intervals at 95 % confidence level are $[\boldsymbol{\mu}^* - 1.96\, diag(\boldsymbol{\Sigma}^* + \sigma_\epsilon^2 I), \boldsymbol{\mu}^* + 1.96\, diag(\boldsymbol{\Sigma}^* + \sigma_\epsilon^2 I)]$. To obtain the predictive distribution, we need to determine the hyper-parameters, θ, and the variance, σ_ϵ^2, given a set of data. The classical approach is to maximize the joint marginal likelihood function of θ and σ_ϵ^2 whose logarithm is written as:

$$l(\theta, \sigma_\epsilon^2; \mathbf{y}) = -\frac{1}{2}\mathbf{y}^T(K + \sigma_\epsilon^2 I)^{-1}\mathbf{y} + \frac{1}{2} log|K + \sigma_\epsilon^2 I| + \frac{n}{2} log(2\pi) \tag{8}$$

with respect to θ and σ_ϵ^2. In this study, we use the package "*mlegp*" (Maximum Likelihood Estimates of Gaussian Processes) in R programming to get the GPR hyper-parameter, the variance estimates, the predictive means and variances of the target outputs in Eqs. (5) - (7). Alternatively, we can learn about the hyper-parameters using a Bayesian approach, inferring posterior distribution over hyper-parameters [15,16].

Artificial neural network (ANN)

The ANN technique attempts to learn output from input by replicating the human brain system, with neuron and nodes interconnection. The basic ANN consists of an input layer, hidden layers and an output layer. Each layer is connected with a node or neuron that takes the weighted sum of the inputs and passes it through a transfer function to become inputs of the next layer. The output of the preceding layer is then the summation of the inputs and their weights presented as follows:

$$a_j^l = f\left(b_j^l + \sum_{i=1}^{k} a_i^{l-1} w_{ji}^l\right). \tag{9}$$

The w_{ji}^l is the weight for connection from the i^{th} neuron in the $(l-1)^{th}$ layer to the j^{th} neuron in the l^{th} layer and k is the total number of neurons of the preceding layer. The b_j^l denotes the bias term of the j^{th} neuron in the l^{th} layer. The a_j^l is the output of the j^{th} neuron in the l^{th} layer, which is related to the outputs in the $(l-1)^{th}$ layer through the transfer function $f(.)$ as shown in Eq. (9). The ANN can approximate complex non-linear functions mapping the input to the output without information about the relationship between the data. However, the ANN results are sensitive to a choice of the input variables, number of neurons and number of hidden layers.

Recurrent neural network (RNN)

The RNN technique is a class of the ANN in which the output of the previous hidden neuron layer is saved and then fed back to be the input of the future hidden layer, together with the input. The RNN is suitable to model sequential or time-series data as the technique takes the time and sequence of the data into account. For each time step t, the hidden neuron states h_t and the output y_t are expressed as follows:

$$h_t = g_1(w_{xh}x_t + w_{hh}h_{t-1} + b_h), \tag{10}$$

$$y_t = g_2\left(w_{yh}h_t + b_y\right), \tag{11}$$

where x_t is the input at time step t, h_t, h_{t-1} represent the hidden neuron states at the time step t and $t-1$ respectively, w_{xh} and w_{hh} are the weight of input-hidden neurons and hidden-hidden neurons, b_h and b_y are the bias terms, and g_1, g_2 are the activation functions.

Performance evaluation

We evaluate the performance of the models using the RMSE, the MAE, the MAPE, and the NSE, defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(P_t - \widehat{P_t}\right)^2}, \tag{12}$$

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}\left|P_t - \widehat{P_t}\right|, \tag{13}$$

$$\text{MAPE} = \frac{1}{T}\sum_{t=1}^{T}\frac{|P_t - \widehat{P_t}|}{P_t}, \tag{14}$$

$$\text{NSE} = 1 - \frac{1}{T}\sum_{t=1}^{T}\frac{(P_t - \widehat{P_t})^2}{(P_t - \bar{P})^2}, \tag{15}$$

where P_t, and $\widehat{P_t}$ are observed and forecasted values at time t, respectively, and T is the total number of forecasts.

The RMSE, MAE and MAPE measure the average magnitude of the difference between forecast and corresponding observed values; smaller values indicate the better predictive performance of a model. The NSE measures 1 minus ratio of error variance from a model over the variance of the observed values.

The NSE closer to 1 indicates good predictive performance, whereas the NSE nearer to 0 suggests that a model has a similar predictive performance as the mean of the observed values. The NSE less than 0 indicates that the mean of observed values predicts better than the model.

Results and discussion

Determining input lags

The ANN, RNN, and GPR require inputs and outputs for modeling. The outputs usually mean either the current data or the predictive values in the future, while the inputs are obtained from the lagged data (previous data). Therefore, we need to determine the optimal number of lags for the inputs for these models. We use the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots of the training data to determine the number of lags used in inputs [17,18]. The number of significant lags (spikes) in the PACF plot gives the order–of the autoregressive (AR) model, referring to the number of lags of the series to be used as predictors. The ACF plot is used to identify the order–of the moving average (MA) model, referring to the number of forecast error lags that are related to predicting the series. Besides, we use the Akaike information criterion (AIC) to determine the number of lags from the autoregressive moving average (ARMA) model within the range of orders provided.

The ACF, PACF, and AIC plots for Data 1 and Data 2 are illustrated in **Figures 3** and **4**. The PACF plots suggest that the autocorrelation of lag 1 should be used as the input for both datasets. However, the AIC values of Data 1 decrease from lag 1 to lag 2 with the minimum at lag 2. This indicates that using lag 2 is more desirable than lag 1. The AIC plot for Data 2 suggests using lag 6 as it gives the minimum AIC value. This contrasts with the result from the PACF plot suggesting lag 1. Therefore, we decide to follow the suggestions from the PACF plots so that lag 1 and lag 2 are used for both datasets.

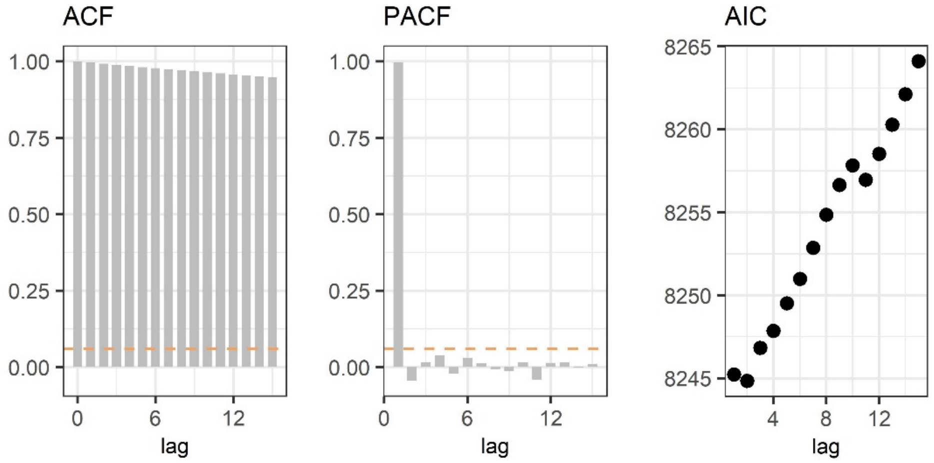

Figure 3 The ACF, PACF, and AIC plots of Data 1.

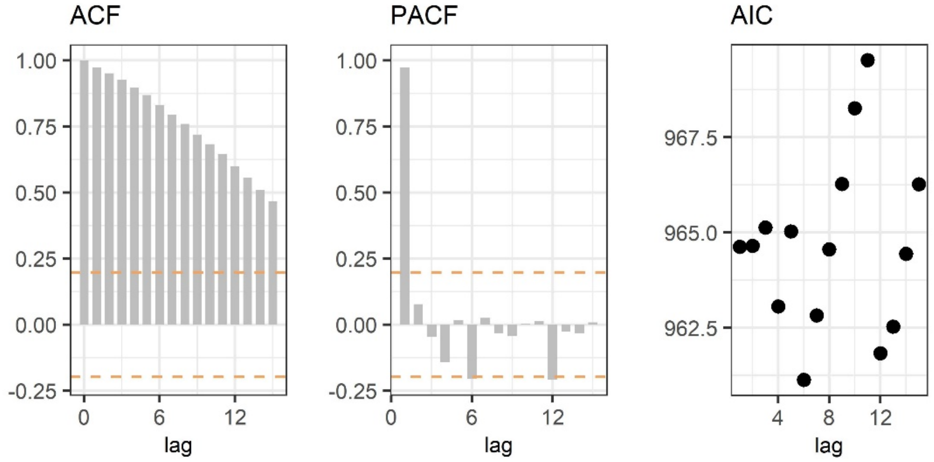

Figure 4 The ACF, PACF, and AIC plots of Data 2.

Forecasting results

This study uses GPR to predict SET data and compare the GPR predictive performance with ANN and RNN models. The data are normalized prior to modeling using min-max scaling, scaled to a fixed range of 0 to 1. The ANN and RNN models are performed with choices of hidden nodes and layers, whose model with the smallest AIC is then selected. The forecasting result, which is a point forecast value over the forecasting period, from all models are shown in **Figures 5** and **6**. For Data 1, all models using either lag 1 or lag 2 give similar results. For the ANN, at the beginning of the predictions, it performs well, but it cannot reach the good predictions at the low peak data. The RNN and the GPR results are similar and better than the ANN because they can get the low peak data. For Data 2, there is a slight difference in predictions from using lag 1 and lag 2, particularly in the ANN and the RNN. The ANN with lag 1 performs better than lag 2, while the RNN shows the opposite results. The GPR, on the other hand, shows the same forecasting results in both lags.

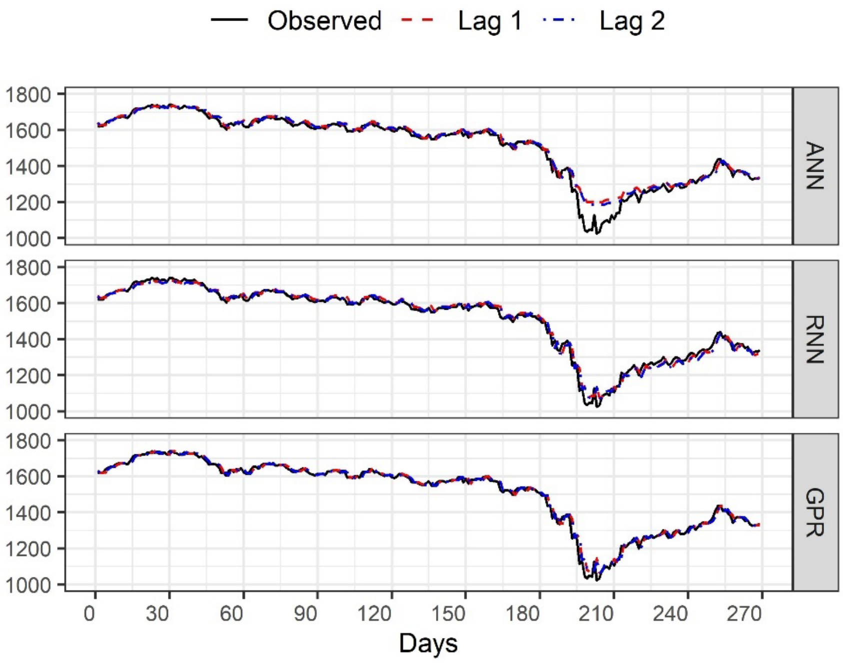

Figure 5 Forecasting results of Data 1.

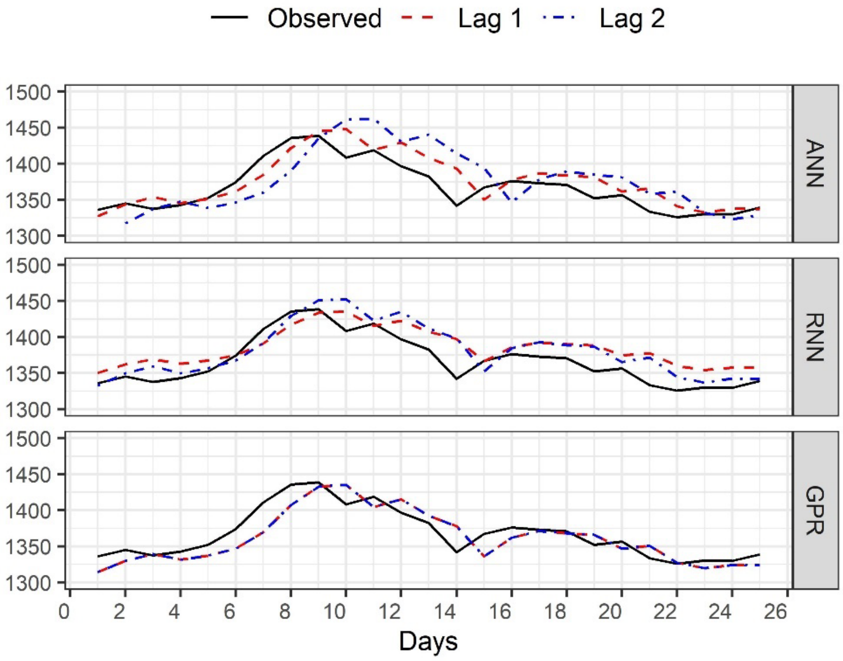

Figure 6 Forecasting results of Data 2.

Comparisons

In this section, we compare the overall performance of the 3 models using the model performance evaluation and the residuals. The RMSE, MAE, MAPE, and NSE values for Data 1 and Data 2 are shown in **Tables 3** and **4**. The results agree with the forecasting results. They indicate that the GPR outperforms the other models.

Table 3 Model performance evaluation for Data 1.

Data set	Score	Data 1					
		Lag 1			Lag 2		
		ANN	RNN	GPR	ANN	RNN	GPR
Training	RMSE	15.1518	18.8820	**11.1766**	16.1903	19.3082	**11.1804**
	MAE	11.1011	14.1238	**8.1838**	12.0071	13.9796	**8.1861**
	MAPE	0.7092	0.8920	**0.5307**	0.7656	0.8849	**0.5318**
	NSE	0.9875	0.9807	**0.9932**	0.9858	0.9798	**0.9932**
Testing	RMSE	38.0618	23.6204	**21.6062**	35.4978	23.3312	**21.6064**
	MAE	20.5061	16.8695	**13.2431**	20.2259	16.5256	**13.2432**
	MAPE	1.6029	1.2065	**0.9687**	1.5538	1.1871	**0.9687**
	NSE	0.9565	0.9833	**0.9860**	0.9622	0.9837	**0.9860**

Table 4 Model performance evaluation for Data 2.

Data set	Score	Data 2					
		Lag 1			Lag 2		
		ANN	RNN	GPR	ANN	RNN	GPR
Training	RMSE	35.0473	59.0180	**31.1368**	45.2917	36.5227	**31.2879**
	MAE	25.0001	49.7384	**21.0440**	34.1862	26.2834	**21.1799**
	MAPE	1.9171	3.8357	**1.6709**	2.7018	2.0214	**1.6831**
	NSE	0.9591	0.8840	**0.9677**	0.9309	0.9551	**0.9670**
Testing	RMSE	20.2176	24.7183	**19.1915**	33.6488	22.7355	**19.1913**
	MAE	**15.0893**	21.2099	15.9251	27.7526	17.6295	**15.9248**
	MAPE	**1.1021**	1.5623	1.1599	2.0208	1.2898	**1.1598**
	NSE	0.6336	0.4523	**0.6698**	–0.0150	0.5366	**0.6698**

It is essential to investigate the distributions of the errors as it can show the variability of the performance of the models. The forecast errors for Data 1 and Data 2 are displayed in **Figure 7**. We can see that the range of the GPR forecast errors is smaller than the errors from the ANN and RNN. Furthermore, the GPR errors are closer to zero than the others. The GPR also produces less extreme forecast errors.

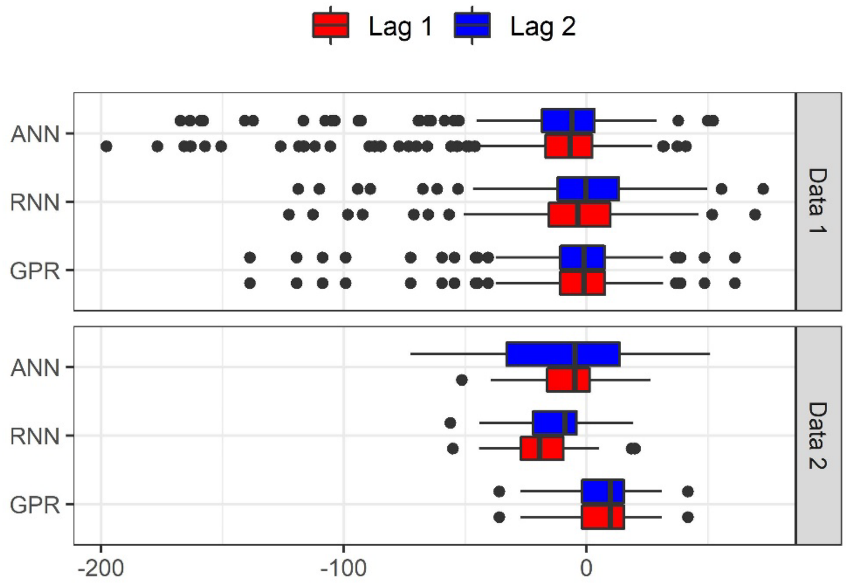

Figure 7 The boxplots of forecast errors.

GPR prediction with intervals

The evaluation scores indicate that GPR outperforms the ANN and RNN models. In this section, we focus on the ability of GPR in prediction with uncertainty. We note that the GPR produces similar forecasting outcomes for lag 1 and lag 2 inputs. For Data 1, we model with a large number of observations. This results in overfitting and yields too narrow GPR prediction intervals. For Data 2, when the data are at the low peak, the GPR prediction intervals are wider than the other periods. **Figure 8** illustrates the GPR performance with 95 % prediction intervals from 26 May 2020 to 28 June 2020, which is the overlapping forecasting area in Data 1 and Data 2. Thus, the effect of using a different number of observations in modeling can be compared. It can be seen that the predictive means of Data 1 are closer to the observed values than those of Data 2. Moreover, the prediction intervals of Data 1 modeling are narrower than those of Data 2. However, the prediction intervals of Data 2 modeling still cover the observed values.

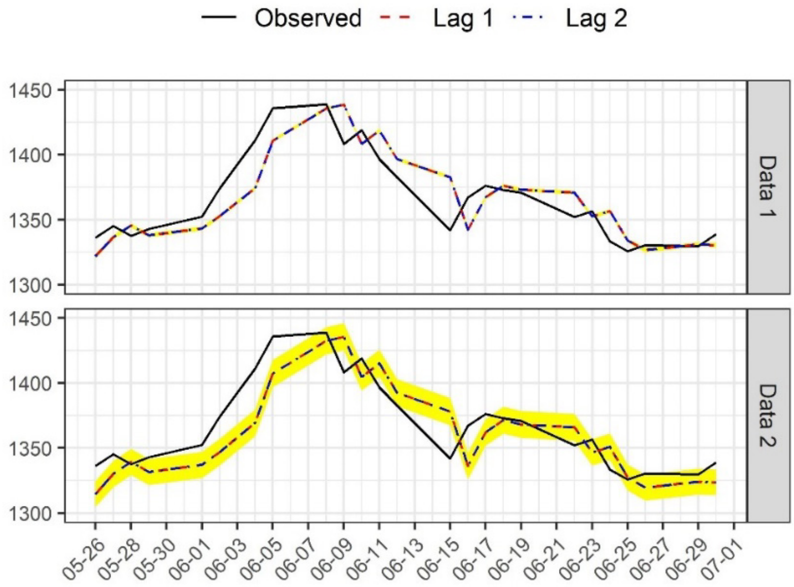

Figure 8 GPR forecasting values with 95 % prediction intervals.

Conclusions

This study presents the power of using GPR to predict the SET index from 2015 to 2020. Due to the COVID-19 pandemic, the stock prices fell drastically in the year 2020. As a result, the daily SET index is divided into 2 datasets: the data in years before and after the COVID-19 outbreak. The performance of the GPR model is compared with ANN and RNN for separate datasets. The findings reveal that the GPR could capture the tendency of the daily SET index better than ANN and RNN, especially the stock plunge during the early stage of the pandemic. On the other hand, prediction errors of the GPR are significantly lower than those of ANN and RNN for both datasets. Furthermore, the GPR forecasts are accompanied by a confidence interval of the prediction results. Therefore, it can be concluded that GPR is an exceptional method in time-series stock exchange forecasting. However, the GPR with Gaussian distribution assumption in the noises could be sensitive to extreme values or outliers in the data. In future studies, the implementation of GPR that can handle such extreme observations should be further discussed. Furthermore, the hybrid model that combined GPR with other machine learning methods can be proposed to obtain more reliable probabilistic prediction results with quantification of uncertainty.

Acknowledgments

This research is supported by the Data Science Research Center, Department of Statistics, Faculty of Science, Chiang Mai University, Thailand.

References

[1] Y Kara, MA Boyacioglu and OK Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Syst. Appl.* 2011; **38**, 5311-9.

[2] J Patel, S Shah, P Thakkar P and K Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Syst. Appl*. 2015; **42**, 259-68.

[3] W Li and J Liao. A comparative study on trend forecasting approach for stock price time series. *In*: Proceedings of the 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification, Xiamen, China. 2017, p. 74-8.

[4] W Long, Z Lu and L Cui. Deep learning-based feature engineering for stock price movement prediction. *Knowl. Based Syst*. 2019; **164**, 163-73.

[5] J Jang, S Shin, H Lee and IC Moon. Forecasting the concentration of particulate matter in the Seoul Metropolitan Area using a Gaussian process model. *Sensors* 2020; **20**, 3845.

[6] P Ghasemi, M Karbasi, AZ Nouri, MS Tabrizi and HM Azamathulla. Application of Gaussian process regression to forecast multi-step ahead SPEI drought index. *Alex. Eng. J.* 2021; **60**, 5375-92.

[7] Q Fu, W Shen, X Wei, P Zheng, H Xin and C Zhao. Prediction of the diet nutrients digestibility of dairy cows using Gaussian process regression. *Inf. Process. Agric.* 2019; **6**, 396-406.

[8] Q Wu, R Law and X Xu. A sparse Gaussian process regression model for tourism demand forecasting in Hong Kong. *Expert Syst. Appl.* 2012; **39**, 4769-74.

[9] N Dkhili, S Thil, J Eynard and S Grieu. Comparative study between Gaussian process regression and long short-term memory neural networks for intraday grid load forecasting. *In*: Proceedings of the IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe, Madrid, Spain. 2020, p. 232-7.

[10] AA Ariyo, AO Adewumi and CK Ayo. Stock price prediction using the ARIMA model. *In*: Proceedings of the UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge. 2014, p. 106-12.

[11] S Selvin, R Vinayakumar, EA Gopalakrishnan, VK Menon and KP Soman. Stock price prediction using LSTM, RNN and CNN-sliding window model. *In*: Proceedings of the International conference on Advances in Computing, Communications and Informatics, Karnataka State, India. 2017, p. 1643-7.

[12] I Jahan and S Sajal. Stock price prediction using Recurrent Neural Network (RNN) algorithm on time-series data. *In*: Proceedings of the Midwest Instruction and Computing Symposium, Minnesota. 2018, p. 1-6.

[13] B Liu, I Kiskin and S Roberts. An overview of Gaussian process regression for volatility forecasting. *In*: Proceedings of the International Conference on Artificial Intelligence in Information and Communication, Fukuoka, Japan. 2020, p. 681-6.

[14] CE Rasmussen and CK Williams. *Gaussian processes for machine learning*. The Massachusetts Institute of Technology press, Cambridge, Massachusetts, 2006.

[15] MK Titsias, N Lawrence and M Rattray. Markov chain Monte Carlo algorithms for Gaussian processes. *In*: Proceedings of the Inference and Estimation in Probabilistic Time-Series Models, Cambridge. 2008, p. 98-106.

[16] M Heinonen, H Mannerstrom, J Rousu, S Kaski and H Lahdesmaki. Non-stationary Gaussian process regression with Hamiltonian Monte Carlo. *In*: Proceedings of the International Conference on Artificial Intelligence and Statistics, Cadiz, Spain. 2016, p. 732-40.

[17] C Huang, Z Zhang and A Bensoussan. Forecasting of daily global solar radiation using wavelet transform-coupled Gaussian process regression: Case study in Spain. *In*: Proceedings of the IEEE Innovative Smart Grid Technologies - Asia, Melbourne, Australia. 2016, p. 799-804.

[18] S Zhu, X Luo, X Yuan and Z Xu. An improved long short-term memory network for streamflow forecasting in the upper Yangtze River. *Stoch. Environ. Res. Risk Assess*. 2020; **34**, 1313-29.