Thai Journal of Mathematics
Volume 14 (2016) Number 3 : 553–563
http://thaijmath.in.cmu.ac.th
ISSN 1686-0209

Improving Stock Price Prediction with SVM by Simple Transformation: The Sample of Stock Exchange of Thailand (SET)

P. Chatvorawit[†], P. Sattayatham[†] and B. Premanode[‡,1]

[†]School of Mathematics, Suranaree University of Technology
Nakhon Ratchasima, Thailand
e-mail : piyatat@gmail.com (P. Chatvorawit)
psattayatham@gmail.com (P. Sattayatham)

[‡]Center for Bio-inspired Technology, Imperial College London,
London, UK
e-mail : premanode@gmail.com (B. Premanode)

Abstract : A stock price at any particular time is represented by its closing price, i.e., the last traded price in stock market prior to that particular time. By nature of a closing price movement, it is a non-stationary data, which is not suitable for prediction. Closing price difference, closing change, is a simple transformation to make a stationary data. However, with the price movement rules of SET, a range of closing change is not the same for a difference price interval. There are some rules to set a limitation and pattern for price movement. A tick size is the smallest amount that stock price can change. For SET market, a tick size is varied for various price intervals, e.g., a tick size is 0.01 Baht per step for price between 0.01 Baht and 2.00 Baht. We calculate a number of tick different in price change, called tick change, which is also a stationary data and a range remains the same for difference price interval. From our experiment on 50 stocks listed in SET50, prediction by closing change based SVM gave the best MAPE result comparing to closing price and tick change based SVM. Although, the overall performance of closing change is better than tick change; there are 10 out of 50 stocks that tick change did better than closing change, and only 5 out of 50 stocks that have

[1]Corresponding author.

more than 10 precents difference in results. Both closing change and tick change achieved a lower prediction error comparing to using closing price model.

Keywords : support vector machine; stock prediction; SET50.
2010 Mathematics Subject Classification : 62M20.

---

1 Stock Prediction

A stock market is a network of trading transaction of companies shares. The origin of the stock market can be traced back to around the 12th century in France, at that time people traded with agricultural debts in their communities. In the present day, there are stock markets in almost every country.

A share is traded at an agreed price between two parties, so a stocks price can change rapidly and dramatically depending on the demand and supply in the stock market. A change in the shares price affects the companys value directly, and a company can raise funds easily by selling more shares to the public. There are many people and companies making profits from buying and selling shares in the market. All of these need strategies to increase the chance to make profit and reduce risk. The stock index was introduced as a method to measure the market value.

Each stock market has its unique characteristic. Two stock markets may have different factors that affect their indices. A prediction methodology that works well for one stock market may not do well in the other stock markets. In order to get the preferable result from prediction, we need to study that particular stock market. In our case, we are going to work on the stock market of Thailand (SET).

There are several researches in predicting the SET index (Sutheebanjard and Premchaiswadi [1],[2]). A neural network is quite a popular technique for forecasting from its learning ability. It does not require human intervention in extracting and formulating any functions or models for prediction, it can learn by itself from the training data (Chaigusin et al. [3]; Sutheebanjard and Premchaiswadi [2]). However, it has some significant drawbacks. It suffers from local-optima as it may converge to the local minima in- stead of the global minima. Its convergence rate is quite slow (Sutheebanjard and Premchaiswadi [1]). An evolution strategy is also popular in forecasting SET index. It is an optimization technique based on a natural evolution process (Rimcharoen et al. [4]; Sutheebanjard and Premchaiswadi [1]). There is also a research on using soft computing techniques, namely; fuzzy logic, neural networks, and probabilistic reasoning (Chaigusin et al. [3],[5]) to forecast the SET index. While traditional computing that we are familiar with, called hard computing, deals with precision, certainty, and exact solution; soft computing on the other hand has been invented to work on imprecision, uncertainty, and approximation.

Nevertheless, predicting the stock prices of individual companies are not attracted much attention from the public, since there are a lot of different factors that can affect the individual stock, e.g., news, politics, rumors, or even stock prices of the other companies.

This paper will show that with simple transformation of stock price, we can improve the performance of stock price prediction with Support Vector Machine (SVM). SVM was developed by Vapnik et al. (1995). Its popularity has been increasing due to many attractive features, and its promising empirical performance in a variety of applications such as pattern recognition, regression estimation, time series prediction etc. And we also introduce a new representation of stock price, tick change, which is calculated based on a characteristic of stock price movement. The main purpose of this present study is to investigate a simple price transformation can improving stock price prediction, and introduce a new transformation for using with stock forecasting.

2 Support Vector Machine

A Support Vector Machine (SVM) is one of the popular approaches in machine learning. (T. Fletcher [6]). It is used mainly in classification tasks and regression analysis. Its basic concept is to divide a set of data into two groups by drawing a separation boundary between two sets of data. It is a supervised learning, which means we need to present some sets of training data along with classified results in order to allow SVM to determine an appropriated separation function. After it is trained with the training data, it is used to predict which group the input data belongs to. This makes SVM a non-probabilistic binary linear classifier.

Let $\{x_i, y_i\}$ be the set of training data, where x_i is the input and y_i is the class of input. y_i can be only one of the two class, i.e., $y_i \in \{1, -1\}$. The training data is of the form:

$$\{x_i, y_i\} \text{ where } x_i \in R, y_i \in \{1, -1\} \text{ for } i = 1, ..., L.$$

The main concept of SVM is to find the line that divides these points into two groups, i.e., finding the parameters w, b such that

$$w \cdot x_i + b \geq 1 \text{ if } y_i = 1$$

$$w \cdot x_i + b \leq -1 \text{ if } y_i = -1.$$

These equations can be combined into:

$$y_i(w \cdot x_i + b) - 1 \geq 0.$$

These parameters w, b can have more than one set of value. The task is to find the most appropriated line. Furthermore, if there is noise in the training data, i.e., there exists the point of the first group that locates in another side of the separation boundary. In this case, we can use the relax version of SVM called Soft Margin Model. It introduces the slack variable into the equation to allow some error.

The equation now is the form:

$$w \cdot x_i + b \geq 1 - \epsilon_i \text{ if } y_i = 1$$

$$w \cdot x_i + b \leq -1 + \epsilon_i \text{ if } y_i = -1$$

where $\epsilon_i \geq 0$. These equations can be combined into:

$$y_i(w \cdot x_i + b) - 1 + \epsilon_i \geq 0.$$

There are many situations that a straight line cannot separate the data, i.e., the data is not linearly separable. In these cases, SVM will map the data from the input space into feature space (which usually has a high dimension than the input space) by using Kernel function, and then do all the work in this new feature space instead (where the data is now linearly separable). So the key is to find a suitable Kernel function to apply. There are many Kernels functions that have been constructed until now, which are suitable for different tasks (Karatzoglou et al. [7]):

(1) Linear kernel

$$k(x, x') = \langle x, x' \rangle.$$

(2) Gaussian Radial Basis Function (RBF) kernel

$$k(x, x') = e^{-\sigma \|x - x'\|^2}.$$

(3) Polynomial kernel

$$k(x, x') = (a \cdot \langle x, x' \rangle + b)^c.$$

(4) Bessel function of the first kind kernel

$$k(x, x') = \frac{Bessel^n_{(v+1)} \sigma \|x - x'\|}{(\|x - x'\|)^{-n(v+1)})}.$$

(5) Laplace Radial Basis Function (RBF) kernel

$$k(x, x') = e^{-\sigma \|x - x'\|}.$$

The Gaussian RBF, Laplace RBF, and Bessel kernel are used in general case when there is no prior knowledge about the dataset. The linear kernel is normally used for text categorization, while polynomial kernel is normally used in image processing. (Karatzoglou et al. [7]).

There is also a version of SVM for regression model that is called Support Vector Regression (SVR). The basic concept is quite similar to the linear SVM except that instead of finding the separation line, SVR tries to find the line that the input data is not differed from this line more than the threshold. The equation is the form:

$$|(w \cdot x_i + b) - y_i| \leq \epsilon.$$

3 Methodology

In this section, we are going to discuss the mechanism that we employ in this paper. The data that we use to train and predict the stock price is a closing price on daily basis. We used R programming with kernlab library in our experiments.

At any given period, there are four points of stock price that are interested, i.e., open, close, high, low. Usually, closing price is used for representing a stock price at any given time. Therefore, we use closing price as a base of stock data for training and prediction in this paper.

Generally, a closing price is non-stationary data, which is not suitable for forecasting (T. Iordanova [8]). Therefore, we are going to transform this closing price into stationary data before feeding into prediction models. A closing price difference, e.g., $x_i - x_{i-1}$, is a stationary series; we call this *closing change*. One disadvantage of this calculation is a loss of one observation.

Although a level of demand and supply in the stock market determines stock price, its price cannot be changed freely, i.e., there is a minimum amount that stock price can change, called *tick size* or *price spread*. In SET, a tick size is varied for difference price intervals, e.g., a tick size is 0.01 Baht for price between 0.01 Baht and 2.00 Baht. This results in changing of range of closing change for stock that its price changes from one interval to the other interval, which may result in variance changing over time. We calculate a number of ticks in closing change; we call this *tick change*, which is stationary and a range remain the same for any price interval.

Market Price Level (Baht)	Spread (Baht)
Less than 2	0.01
2 up to less than 5	0.02
5 up to less than 10	0.05
10 up to less than 25	0.10
25 up to less than 100	0.25
100 up to less than 200	0.50
200 up to less than 400	1.00
400 up	2.00

Table 1: The current price spread in effective at the time of writing this paper. Available at `http://www.set.or.th/en/products/trading/equity/tradingsystem_p5.html`.

Let $close_i$ be closing price of i^{th} day

$tickCount_i$ be a number of tick of closing price of i^{th} day

$change_i$ be closing change of i^{th} day

$tickChange_i$ be tick change of i^{th} day

$$[a_i, a_{i+1}] \text{ be an } i^{th} \text{ price interval}$$
$$t_i \text{ be a tick size of an } i^{th} \text{ price interval}$$

$$change_i = close_i - close_{i-1}$$
$$tickCount_i = \sum_{j=0}^{n} \frac{a_{j+1} - a_j}{t_j} + \frac{close_i - a_{n+1}}{t_{n+1}}; close_i \in [a_{n+1}, a_{n+2}]$$
$$tickChange_i = tickCount_i - tickCount_{i-1}$$

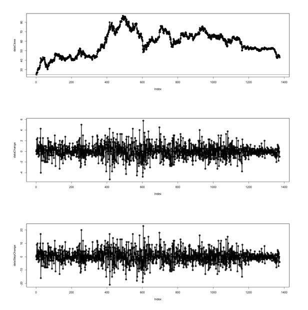
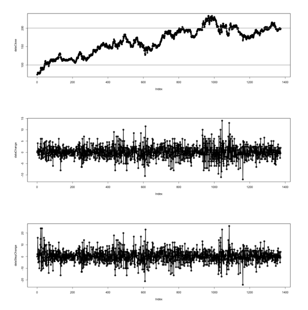
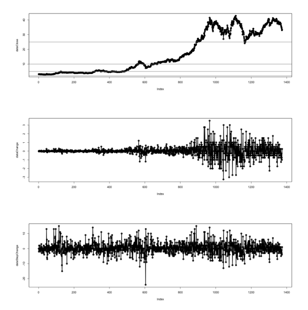
(A)	(B)	(C)
		

Table 2: Chart of closing price (top), closing change (middle), and tick change (bottom) from three stocks in SET50.

Table 2 illustrates charts from closing price, closing change, and tick change from three stocks. The horizontal line in closing price charts indicate the point where tick size is changed. For stock (A), the tick size remains the same for an entire sample period, so the closing change chart and tick change chart are very similar to each another. Stock (B) has three difference tick sizes in the sample data, but the majority of its data is in a signle tick size interval. So its closing change and tick change charts are quite similar to one another. Stock (C) has four difference tick sizes interval in the dataset, and each tick size interval has a significant amount of data. So its closing change and tick change charts are quite difference from one another. So its variance is likely to be changed overtime when tick size is changed.

We used closing price, closing change, and tick change for the experiment in this paper. Furthermore, we only uses historical data of individual stock price alone for forecasting its price without any other data, because we need to determine the relation between each data, which may result in difference set of related data for difference stock.

In oreder to show that the simple price transformation can improve the prediction performance, we construct SVM with a simple model

$$x_n = f(x_{n-1}, x_{n-2}, ..., x_{n-i})$$

with the Gaussian RBF, Laplace RBF, and Bessel kernel which are suitable for dataset with no prior knowledge. (Karatzoglou et al. [7]) We use a week of historical data, e.g.,the past 5 days, and do prediction for a week ahead, e.g., 5 days ahead by using a prediction result to predict a result of the next day and repeating this until we get 5 days ahead results. So the equation becomes:

$$x_n = f(x_{n-1}, x_{n-2}, ..., x_{n-5}).$$

The data used for experiment in this paper were closing price data of stocks in SET50, and retrieved from E-Finance on November 13, 2014. The data was from March 30, 2009 to November 13, 2014, except for some stock that was added to SET50 after March 30, 2009. Therefore, the number of record is varied, ranging from around 300 records to 1,300 records. We used 70% of data for training, 30% for validation.

Closing Price	Closing Change	Tick Change
23.2	0.4	4
23.1	-0.1	-1
23.3	0.2	2
24.8	1.5	15
24.7	-0.1	-1
25.25	0.55	4
25.75	0.5	2
25.25	-0.5	-2
24.9	-0.35	-2
25	0.1	1
25.5	0.5	2
27	1.5	6
27.25	0.25	1

Table 3: The sample data used for training and testing.

The error was measured by Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), and R-squared (R^2). MAPE measures an error in percentage. It is commonly used in quantitative forecasting methods (Sutheebanjard & Premchaiswadi [1][2]). MSE measures the average of the squares of the errors. R-squared is a number that indicates how well the data fit the prediction model.

$$MAPE = \frac{\sum_{i=1}^{n} |\frac{a_i - p_i}{a_i}|}{n} \times 100$$

$$MSE = \frac{\sum_{i=1}^{n} (a_i - p_i)^2}{n}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (a_i - p_i)^2}{\sum_{i=1}^{n} (a_i - \bar{a})^2}$$

where a_i is an actual value, p_i is a predicted values, and n is a number of data.

4 Result

In our experiment, we repeated the training and testing 3 times, and averaged the results. We used SVM implementation from kernlib package in R programming.

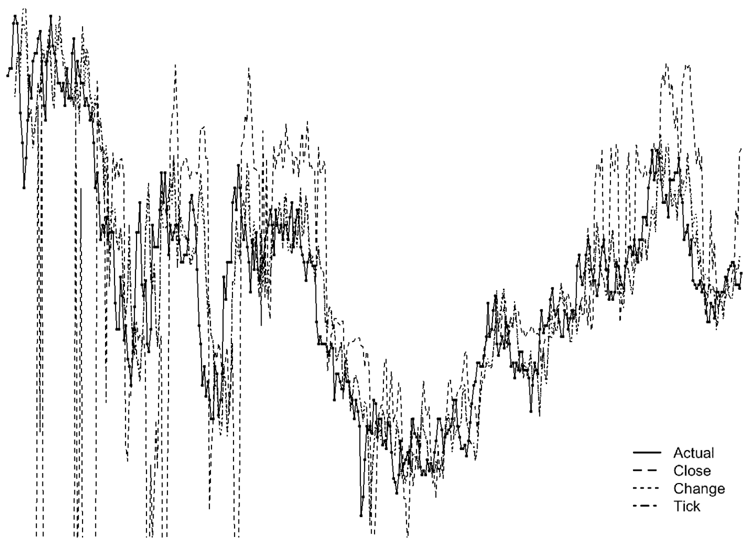


Figure 1: Prediction result for 5 days ahead of a single stock in SET50 by closing price, closing change, and tick change based SVM with Gaussian RBF kernel.

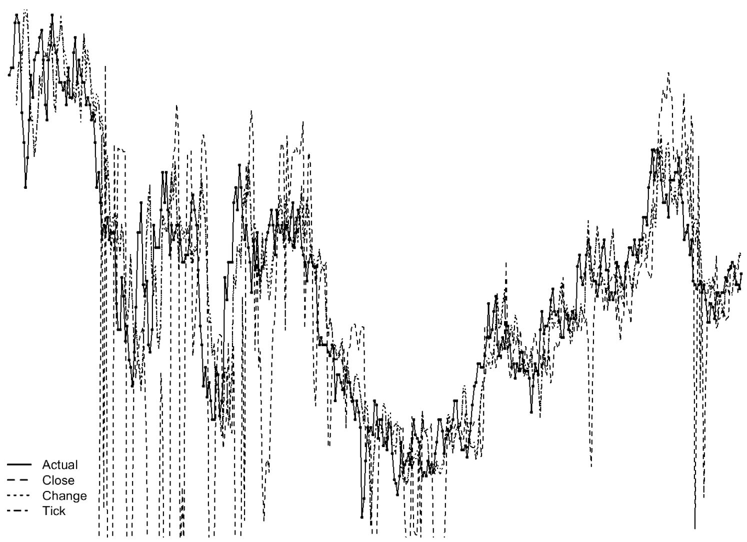


Figure 2: Prediction result for 5 days ahead of a single stock in SET50 by closing price, closing change, and tick change based SVM with Laplace RBF kernel.

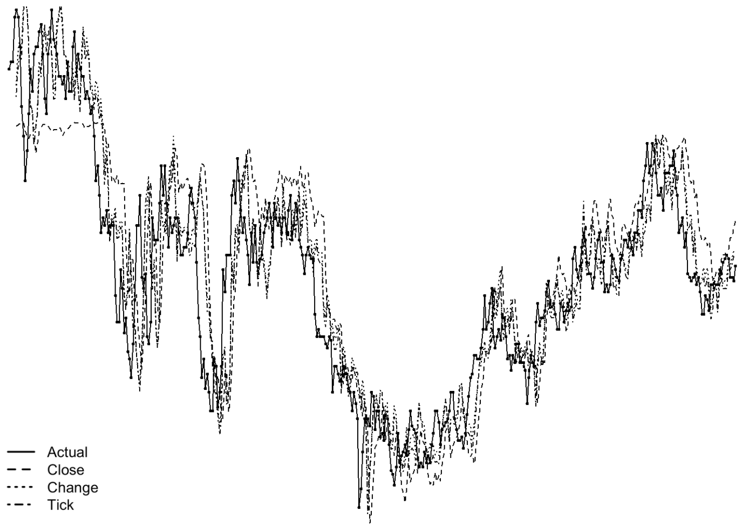


Figure 3: Prediction result for 5 days ahead of a single stock in SET50 by closing price, closing change, and tick change based SVM with Bessel kernel.

The following tables show an average MAPE, MSE and R^2 from all experiment categorized by stock price type (Close price, Close Change, Tick Change) and kernel that used for prediction.

MAPE	Gaussian	Laplace	Bessel
Close	18.54153929	21.84193574	10.77907097
Change	2.66570006	2.64632871	2.64683537
Tick	2.79002912	2.76139014	2.73096759

Table 4: MAPE result of closing price, closing change, and tick change based SVM.

MSE	Gaussian	Laplace	Bessel
Close	970.2716956842	1225.1545308773	517.5365986714
Change	16.1729432405	15.8376041233	16.1403386799
Tick	17.1004931258	16.6654468704	16.8431728395

Table 5: MSE result of closing price, closing change, and tick change based SVM.

R^2	Gaussian	Laplace	Bessel
Close	-4.3822652578	-5.4872399377	-1.2667564799
Change	0.8724079398	0.874983206	0.8757349303
Tick	0.8641715173	0.867491841	0.8700150013

Table 6: R^2 result of closing price, closing change, and tick change based SVM.

Clearly, from the result tables, model based on closing change and tick change yield better results than closing price model. Prediction model based on closing change gives slightly better result than tick change. Bessel kernel gives the best result for closing price, and tick change based model. For closing change based model, Laplace RBF kernel gives the best result, while Bessel kernel give a slightly higher error than that of Laplace RBF kernel. There are 10 out of 50 stocks that tick change model gives better result than closing change. Both closing change and tick change models give about the same result. There are 5 out of 50 stocks that the difference in results between them is greater than 10 percents.

5 Conclusion

In this paper, we created a new data set by using a certain characteristic of stock price movement, called tick change, and used it to create prediction model with SVM. Although, from our experiment with 50 stocks listed in SET50, prediction by closing change based SVM gave the best results; tick change did give about the same results as closing change model. Both closing change and tick change based SVM achieved a lower prediction error comparing to using closing price to create prediction model directly.

We only used historical data of stock to forecast a price of that particular stock in this experiment. We can improve our model by including information of the other relevant stocks, market index, or some kind of measurement for building a forecasting model. However, to achieve this, it would require analysis and strategies to identify which information is relevant and have impact on that particular stock. Obviously, each stock has its own set of relevant factors, so a mechanism to select relevant factors is necessary.

Another possible improvement is to use multi-model SVM model for prediction. At any time, a trend of stock price is not necessary the same, so we can build a SVM model based on this trend. By combine with closing change and tick change as the basis, we believe this will yield an even better result.

References

[1] P. Sutheebanjard, W. Premchaiswadi, Forecasting the Thailand stock market using evolution strategies, AAMJAF 6 (2) (2010) 85-114.

[2] P. Sutheebanjard, W. Premchaiswadi, Stock Exchange of Thailand Index prediction using Back Propagation Neural Networks, 2nd International Conference on Computer and Network Technology, 2010.

[3] S. Chaigusin, C. Chirathamjaree, J. Clayden, The Use of Neural Networks in the Prediction of the Stock Exchange of Thailand (SET) Index, Computational Intelligence for Modelling Control & Automation, 2008.

[4] S. Rimcharoen, D. Sutivong, P. Chongstitwatan, Prediction of the Stock Exchange of Thailand Using Adaptive Evolution Strategies, Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence, 2005.

[5] S. Chaigusin, C. Chirathamjaree, J. Clayden, Soft computing in the forecasting of the stock exchange of Thailand (SET), Management of Innovation and Technology, ICMIT 2008.

[6] T. Fletcher, Support Vector Machines Explained, Tutorial Paper, University College London, 2009.

[7] A. Karatzoglou, D. Meyer, K. Hornik, Support vector machine in R, Journal of Statistical Software 15 (9) (2006) file:///C:/Users/LONG/Downloads/v15i09.pdf.

[8] T. Iordanova, Introduction To Stationary And Non-Stationary Processes, 2009. Available at http://www.investopedia.com/articles/trading/07/stationary.asp.

(Received 17 June 2016)
(Accepted 6 July 2016)