An efficient modelling approach for forecasting financial time series data using support vector regression and windowing operators

Risul Islam Rasel*

Faculty of Computer Science and Engineering,
International Islamic University Chittagong,
154/A, College Road, Chittagong-4203, Bangladesh
Email: rasel.kmutnb@gmail.com
*Corresponding author

Nasrin Sultana

Faculty of Computer Science and Engineering,
University of Chittagong,
Chittagong-4331, Bangladesh
Email: nasrin_cse@cu.ac.bd

Phayung Meesad

Faculty of Information Technology,
King Mongkut's University of Technology North Bangkok,
1518 Piboolsongkram Rd. Bangsue, Bangkok 10800, Thailand
Email: pym@kmutnb.ac.th

Abstract: Forecasting or predicting stock market price and trend is regarded as a challenging task because of its chaotic nature. Stock market is essentially a nonlinear, non-parametric, noisy and deterministically chaotic system because of liquid money, stock adequacy, human behaviour, news related to the stock market, gambling, international money rate and so on. Since it is an emerging sector of the business and also many people are related to this sector, many researchers and experts have their interest to work on this area in order to describe this chaotic system for pattern recognition purpose of the trend. Many researches have been already done and still studies are ongoing for better solutions. Artificial intelligence and machine learning are the major techniques that have been applied in different studies. The aim of this study is also using machine learning algorithm for recognising the pattern of stock market trend in order to predict stock price. For that, support vector regression is used as machine learning technique and different widowing operators are used as input selection technique or data pre-processing steps. Experiments are undertaken on three different stock indexes; Dhaka Stock Exchange (DSE), from Bangladesh, S&P 500 stock index and IBM index.

Keywords: machine learning intelligence; data miming; support vector regression; SVR; financial time series data; financial time series modelling; predicting stock price; stock market; DSE index; S&P 500 index; IBM index.

Reference to this paper should be made as follows: Rasel, R.I., Sultana, N. and Meesad, P. (2015) 'An efficient modelling approach for forecasting financial time series data using support vector regression and windowing operators', *Int. J. Computational Intelligence Studies*, Vol. 4, No. 2, pp.134–150.

Biographical notes: Risul Islam Rasel is a Lecturer in Computer Science and Engineering at International Islamic University Chittagong (IIUC). He completed his Master's in Information Technology, an MSc (IT) from KMUTNB by getting Royal Thai Government Scholarship. Before that, he pursued his Bachelor degree in Computer Science and Engineering from University of Chittagong, Bangladesh. He worked as a Graduate Research Assistant at KMUTNB from 2012 to 2013. He has some publications in IEEE conference proceedings and Springer. His research interest is in data mining, text mining and machine learning.

Nasrin Sultana is a Lecturer in Computer Science and Engineering at University of Chittagong. She did her MSc and BSc in Computer Science and Engineering from University of Chittagong, Bangladesh. Her research interest is in data mining, text mining and machine learning.

Phayung Meesad is an Associate Professor and Head of Department of IT in KMUTNB. He did his PhD and MS in Electrical Engineering from Oklahoma State University, USA. His research interest is in data mining, text mining, fuzzy system, decision support system and machine learning.

This paper is a revised and expanded version of a paper entitled 'Predicting stock market price using support vector regression' presented at the International Conference on Informatocs, Electronocs & Vision (ICIEV 2013), University of Dhaka, Bangladesh, 17–18 May 2013.

1 Introduction

The stock market is a trading place where shares are issued and traded either through exchanges or over-the-counter markets. It is also known as the equity market. Stock market is one of the most vital areas of a market economy as it provides companies with access to capital and investors with a slice of ownership in the company and the potential of gains based on the company's future performance. The stock market is one of the most important sources for companies to raise money. This allows businesses to be publicly traded, or raise additional financial capital for expansion by selling shares of ownership of the company in a public market. The liquidity that an exchange affords the investors gives them the ability to quickly and easily sell securities. This is an attractive feature of investing in stocks, compared to other less liquid investments. Some companies actively increase liquidity by trading in their own shares.

So, nowadays, predicting stock market trend is become more demanding job to the people who are directly related to the stock market. It is a challenging job to predict the exact price of the stock because of the chaotic behaviour the market. But, stock market trend analysis able to create a view about the market to the people, depending on that people can decide whether they will invest or sale their share. It also helps the share market authority take proper decision about the market to avoid financial crisis in stock market.

Forecasting or predicting stock price has regarded as a challenging task because of the chaotic nature of stock market (Kim, 2003). The trend or price depends on many things like liquid money, adequate of stocks, human behaviours, news related to stocks etc all together control the stock situation. Stock markets are essentially nonlinear, non-parametric, noisy and deterministically chaotic system (Ince and Trafalis, 2007; Lucas et al., 2010; Lu et al., 2009; Lee, 2009; Kara et al., 2011). The behaviour can be analysed by using technical tools, parametric pricing methods, or combining of these methods (Lu et al., 2005; Kannan et al., 2010). Machine learning intelligence is the most superior data mining technique that has been applied before in many research for analysing the trend of stock market in order to describe and understand the pattern of stock trend and based on that learning forecasting the up coming trend and price (Lucas et al., 2010; Kannan et al., 2010; Chen and Ho, 2005; Ni et al., 2011; Wang et al., 2012). Neural network (NN) and support vector machine (SVM) are the most usable technique among the all, since both are standard machine learning techniques for predicting financial time series data (Kim, K.J. (2003; Lucas et al., 2010; Kannan et al., 2010; Hu and Pang, 2008; Chen et al., 2005). For that reason in this study, support vector regression (SVR) is used as machine learning approach for analysing the pattern with the help of different input selection techniques, like different windowing operators.

SVM are used for classification, regression and outlier detection (Xue-Shen et al., 2010; Debasish et al., 2007; Hsu et al., 2010). Regression analysis is a major application that helps to analyse time series data. It attempts to minimise the generalisation error bound instead of minimising the observed training error, so that generalised performance can be achieved (Thissena et al., 2003). This generalisation error bound is the combination of training error and a regularisation term that controls the complexity of the hypothesis space (Cao, 2003; Smola and Schölkopf, 2004).

Kim (2003) had applied back propagation NN and case base reasoning in his study. Lucas et al. (2010) did comparative experiments by using SVM and artificial NN. In this experiment, they had used 15 days exponential moving average (EMA15) and relative difference in percentage of price (RDP) as the data pre-process or input selection technique for the SVM and NN. Many other recent research studies have also been done by using these two useful superior data mining techniques (Kazem et al., 2013; Zhou et al., 2008; Mager et al., 2008; Kato et al., 2010; Meesad and Rasel, 2013a, 2013b).

So, the motivation from those works is applied to this study by using SVR as machine learning technique and different windowing operators as input selection technique for SVR model. Since this work combines SVR and windowing operators, so proposed models are named as Win-SVR model.

2 Methodology

This study is to propose a new application of combining SVR and different windowing operators. SVR is used for recognising the pattern of financial time series data and windowing operators are used as optimised input selection techniques for SVR models.

2.1 Support vector regression

In SVM regression, the input **X** is first mapped onto a *m*-dimensional feature space using some fixed (nonlinear) mapping, and then a linear model is constructed in this feature space (Ince and Trafalis, 2007; Lu et al., 2009; Pai and Lin, 2005). Using mathematical notation, the linear model (in the feature space) $f(x, \omega)$ is given by:

$$f(x,w) = \sum_{j=1}^{m} w_j g_j(x) + b \tag{1}$$

where $g_j(x)$, $j = 1, \ldots, m$ denotes a set of nonlinear transformations, and b is the 'bias' term. Often the data are assumed to be zero mean (this can be achieved by pre-processing), so the bias term is dropped. The quality of estimation is measured by the loss function $L(y, f(x, \omega))$. SVM regression uses a new type of loss function called ε – insensitive loss function proposed by Vapnik (Kim, 2003; Ince and Trafalis, 2007; Lucas et al., 2010; Lu et al., 2009; Kannan et al., 2010; Lu et al., 2009; Pai and Lin, 2005; Huang et al., 2005; Lee, 2009; Kara et al., 2011):

$$L_\varepsilon\left(y, f(x,\omega)\right) = \begin{cases} 0 & \text{if } \left|y - f(z,\omega)\right| \le \varepsilon \\ \left|y - f(x,\omega)\right| - \varepsilon & \text{if otherwise} \end{cases} \tag{2}$$

The empirical risk is:

$$R_{emp}(\omega) = \frac{1}{n}\sum_{i=1}^{n} L_\varepsilon\left(y_i, f\left(x_i,\omega\right)\right) \tag{3}$$

SVM regression perform linear regression in the high dimension feature space using ε– insensitivity loss and, at the same time tries to reduce model complexity by minimising $\|w\|^2$. This can be described by introducing slack variables ξ_i and ξ_i^* where $i = 1, \ldots, n$ to measure the deviation of training sample outside ε– sensitive zone (Lucas et al., 2010; Debasish et al., 2007; Smola and Schölkopf, 2004).

$$\min_{\omega,b,\xi_i,\xi_i^*} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \tag{4}$$

$$s.t.\begin{cases} y_i - f\left(x_i,\omega\right) \le \varepsilon + \xi_i^* \\ f\left(x_i,\omega\right) - y_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, i = 1, \ldots\ldots, n \end{cases} \tag{5}$$

This optimisation problem can transform into the dual problem and solution is given by:

$$f(x) = \sum_{i=1}^{n_{sv}}\left(\alpha_i - \alpha_i^*\right) K\left(x_i, x\right) \tag{6}$$

$$s.t. \quad 0 \le \alpha_i^* \le C, \quad 0 \le \alpha_i \le C$$

where n_{sv} is the number of support vector (SVs) and the kernel function

$$K\left(x, x_i\right) = \sum_{i=1}^{n} g_j(x) g_j\left(x_i\right) \tag{7}$$

SVM generalisation performance depends on a good setting of kernel parameters C, ε and kernel parameters (Debasish et al., 2007; Smola and Schölkopf, 2004). The problem of optimal parameter selection is further complicated by the fact that SVM model complexity (and hence its generalisation performance) depends on all three parameters. Existing software implementations of SVM regression usually treat SVM meta-parameters as user-defined inputs. Selecting a particular kernel type and kernel function parameters is usually based on application-domain knowledge and also should reflect distribution of input (**x**) values of the training data.

2.2 Widowing operators

2.2.1 Normal rectangular windowing

Windowing operator transforms a given example set containing series data into a new example set containing single valued examples. For this purpose, windows with a specified window and step size are moved across the series and the attribute value lying horizon values after the window end is used as label which should be predicted. In contrast to the *Series2WindowExamples* operator, this operator can also handle multivariate series data. In order to specify the dimension which should be predicted, one must use the parameter 'label dimension' (counting starts at 0). If we want to predict all dimensions of our multivariate series we must setup several process definitions with different label dimensions, one for each dimension.

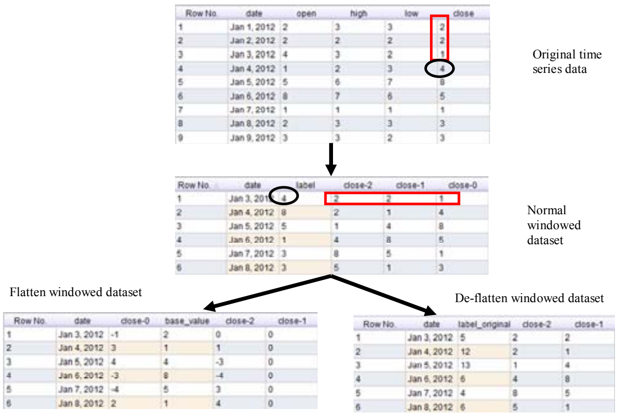
2.2.2 Fatten windowing

This operator performs three basic tasks. First, it removed all attributes lying between the time point zero (attribute name ending '–0') and the time point before horizon values. Second, it transforms the corresponding time point zero of the specified label stem to the actual label. Last, it represents all values relative to the last known time value for each original dimension including the label value.

2.2.3 De-Flatten windowing

This operator performs a simple task. It adds the values of the base value special attribute to both the label and the predicted label (if available) so the original data (range) is restored. After that, it removes the base value special attribute. The input example set must contain the base value special attribute created by the mentioned pre-processing operator. If a label and/or a predicted label is provided, the transformation is performed for the label and/or the predicted label (e.g., for performance calculations).

Figure 1 Transforming time series dataset into windowed dataset using windowing operators (see online version for colours)



3 Experiment design

3.1 *Experiment dataset*

To undertake the experiments three stock exchange indexes are selected as domain of the study. Two are international indexes; S&P 500 and IBM index whereas one local index from Bangladesh; Dhaka Stock Exchange (DSE). Experiment dataset are collected from these three stock exchanges. S&P 500 and IBM index dataset are collected from Google finance and DSE dataset is collected from local authority. All datasets have five attributes; one special attribute date as ID and four regular attributes; open price, high price, low price and close price. The dataset summery is given in Table 1.

Table 1 Experiment dataset summery

Index name	*Total data*	*Time period*	*Training data*	*Testing data*
DSE	822	Jan 2009–June’2012	700 (85%)	122 (15%)
S&P 500	878	Jan 2010–June 2013	754(85%)	124 (15%)
IBM	245	Aug 2009–Aug 2010	188 (80%)	57 (20%)

3.2 *Model flow chart*

The model has two parts; one is training part and another is test part. Model starts with the training part to train the model with training dataset and based on that training test the model output for evaluating the performance. The steps are given below:

3.2.1 Training phase

Step 1 Read the training dataset from local repository.

Step 2 Apply windowing operator to transform the time series data into a generic dataset. This step will convert the last row of a window within the time series into a label or target variable. Last variable is treated as label.

Step 3 Accomplish a cross validation process of the produced label from windowing operator in order to feed them as inputs into SVR model.

Step 4 Select kernel types and select special parameters of SVR (*C*, epsilon (+/–), g, etc).

Step 5 Run the model and observe the performance (accuracy).

Step 6 If performance accuracy is good than go to step 7, otherwise go to step 4.

Step 7 Exit from the training phase and apply trained model to the testing dataset.

3.2.2 Testing phase

Step 1 Read the testing dataset from local repository.

Step 2 Apply the training model to test the out of sample dataset for price prediction.

Step 3 Produce the predicted trends and stock price.

Figure 2 illustrate the steps of the win-SVR models. Two major sections are here, train the model and based on that training test the model. Training model is the significantly important part because it helps the model to recognise the pattern and helps to produce good prediction result. Training part has two sub section; one is input selection or data pre-process where different windowing operators work to produce optimised inputs for machine learning algorithm and another is applying machine learning technique where SVR works to recognise the input pattern and based on that produce outputs.

Figure 3 shows the tree structure of Win-SVR model using three different windowing operators; rectangular or normal windowing, flatten windowing, and de-flatten windowing. The three models have a common part; is windowing. First, windowing should apply and then transform the windowing to flatten and de-flatten category. Each operator has some important parameters in common which are very important to be determined very carefully to produce good windowing results; like attribute selection, role selection, horizon selection, step ahead selection, training window size, testing window size, label selection based on attribute selection. The process of using or applying different windowing is different from each other, but it is undoubtedly said that the parameter should be arrange with proper combination. More importantly, a validation process should be applied in order to validate the windowing performance. There are many validation techniques for window validation; like sliding window validation, cross validation, sliding window validation with cumulative training, fixed order split validation. Different validation has different application and advantages and disadvantages. In this study sliding window validation is used.

Figure 2 Model flow chart

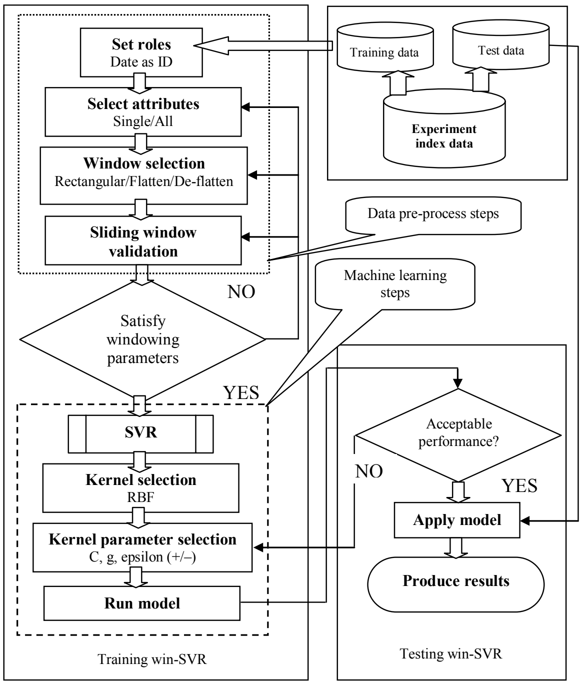
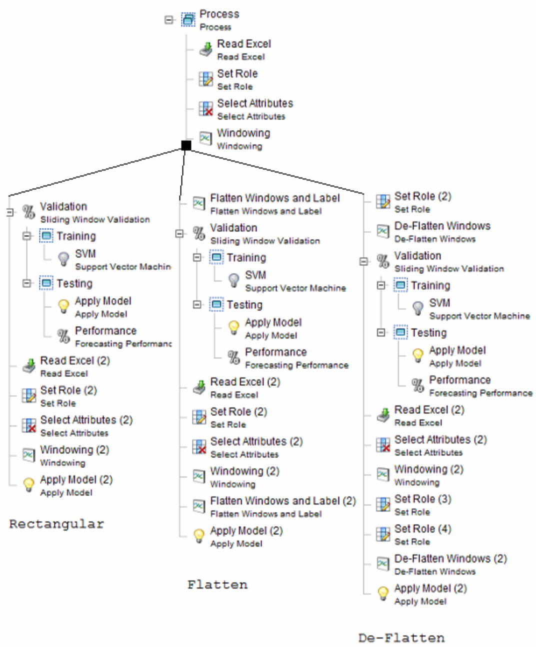


Figure 3 Tree structure of the models using three different windowing (see online version for colours)



4 Experiments and results

4.1 Data pre-process and optimised input selection

Before using data as input to the machine learning, dataset should be pre-processed to feed optimised input to the machine learning algorithm. In DSE, S&P 500 and IBM all indexes have six main attributes. One attribute is volume of the index, which is not relevant to the experiment issues. So, volume field is eliminated from the main dataset to prepare experiment dataset. Furthermore, S&P 500 is a one index; but DSE has almost 500 companies in list with different price index. So, only one well-known group of companies is selected and than filtered the dataset for this company. It is a random basis selection technique, so any company dataset can be selected as researcher wish. After doing that, the dataset is split into two groups; training dataset and testing dataset. The dataset is than feed into different windowing operator to eliminate the missing data and to produce the optimise input for the machine learning algorithm. A tabular format of windowing dataset for machine learning which are used in this study is given in Table 2:

Table 2 Windowing parameter for all windowing operators

Windowing name	*Model*	*Window size*	*Step size*	*Training window*	*Testing window*
Rectangular	All	3	1	30	30
Flatten window	1 day	3	1	30	30
	5 days	8	1	30	30
	22 days	25	1	30	30
De-flatten window	All	5	1	30	30

4.2 SVR Kernel function parameter settings

After data pre-process, the optimised input feeds to the SVR model. To produce good output from SVR model, it is important to select appropriate kernel function and kernel parameter component settings. There are different types of kernel functions are there for SVR model. Every kernel function has some own important parameter for error boundary setting for global generalised performance. To get good result from the model this parameter value should be determined carefully. Here in this research radial basis function is used as kernel function which uses epsilon insensitive loss function to generalise the model performance. A tabular looks of the components value is given in Table 3 which is used in this study:

Table 3 Kernel function component settings

SVR model	*Kernel*	*C*	*g*	*ε*	*ε+*	*ε–*
1 day ahead	RBF	10,000	1	2	1	1
5 days ahead	RBF	10,000	1	2	1	1
22 days ahead	RBF	10,000	1	2	1	1

Table 4 Prediction results for S&P 500 stock data for the month Jan'13 to June'13

	Month	*1 day ahead*			*5 days ahead*			*22 days ahead*		
		Actual	*Predicted*	*Error*	*Actual*	*Predicted*	*Error*	*Actual*	*Predicted*	*Error*
Rectangular	Jan'13	26,700.0	26,384.0	316.1	20,847.9	19,692.4	1,155.5	---	---	---
	Feb'13	28,733.9	27,305.9	1,428.0	28,733.9	25,394.1	3,339.8	24,213.8	21,231.4	2,982.4
	Mar'13	31,016.6	27,128.2	3,888.4	31,016.6	26,556.9	4,459.7	31,016.6	21,765.2	9,251.4
	Apr'13	34,555.5	28,659.9	5,895.5	34,555.5	28,957.7	5,597.7	34,555.5	23,197.4	11,358.0
	May'13	36,076.5	27,664.7	8,411.8	36,076.5	28,502.2	7,574.3	36,076.5	23,083.5	12,993.0
	Jun'13	32375.5	25178.6	7,196.9	32,375.5	25,834.1	6,541.4	32,375.5	21,014.0	11,361.5
Flatten	Jan'13	26,700.0	26,625.8	74.2	20,847.9	20,655.5	192.4	---	---	---
	Feb'13	28,733.9	28,703.3	30.6	28,733.9	28,713.9	20.0	24,213.8	23,658.6	555.2
	Mar'13	31,016.6	30,915.2	101.4	31016.6	30,743.3	273.3	31,016.6	30,211.2	805.4
	Apr'13	34,555.5	34,496.3	59.1	34,555.5	34,422.5	133.0	34,555.5	34,047.3	508.2
	May'13	36,076.5	35,982.6	94.0	36,076.5	35,770.7	305.8	31,143.0	29,782.0	1,361.0
	Jun'13	30,769.2	30,837.9	–68.7	22,799.2	22,907.6	–108.5	##	##	##
De-latten	Jan'13	23,781.0	46,675.2	–22,894.2	17,905.1	33,397.2	–15,492.0	---	---	---
	Feb'13	28,733.9	54,904.4	–26,170.5	28,733.9	51,582.0	–22,848.1	21,192.2	28,383.7	–7,191.5
	Mar'13	31,016.6	56,202.8	–25,186.2	31,016.6	52,873.8	–21,857.2	31,016.6	45,504.1	–14,487.5
	Apr'13	34,555.5	60488.1	–25,932.7	34,555.5	56,264.1	–21,708.6	34,555.5	52,975.3	–18,419.8
	May'13	36,076.5	60,004.5	–23,927.9	36,076.5	55,653.3	–19,576.7	36,076.5	54,327.0	–18,250.5
	Jun'13	30,769.2	51,811.9	–21,042.7	30,769.2	47,978.0	–17,208.8	30,769.2	46,973.1	–16,203.9

Notes: N/A* flatten window 1st removed, all attributes lying between the time point zero (attribute name ending '–0') and the time point before horizon values. Second, it transforms the corresponding time point zero of the specified label stem to the actual label. Last, it represents all values relative to the last known time value for each original dimension including the label value. So, it produces result exactly after from horizon selected.

(##) – Lack of input data for flatten windowing.

(---) – 22 days skipped for 22 days ahead prediction.

Table 5 Prediction results for DSE stock data for the month Jan'12 to June'12

	Month	1 day ahead			5 days ahead			22 days ahead		
		Actual	*Predicted*	*Error*	*Actual*	*Predicted*	*Error*	*Actual*	*Predicted*	*Error*
Rectangular	Jan'12	3,846.7	3,924.9	–78.2	2,973.6	3,115.8	–142.2	---	---	---
	Feb'12	3,310.9	4,252.3	–941.4	3,310.9	3,849.9	–539.0	2,976.2	3,387.5	–411.3
	Mar'12	4,015.8	4,109.4	–93.6	4,015.8	3,994.9	20.9	4,015.8	3,280.6	735.2
	Apr'12	5,279.5	5,182.1	97.4	5,279.5	5,055.1	224.4	5,279.5	4,253.6	1,025.9
	May'12	4,315.7	4,613.3	–297.6	4,315.7	4,763.1	–447.4	4,315.7	4,505.4	–189.7
	Jun'12	3,437.0	4,513.2	–1,076.2	3,437.0	4,004.1	–567.1	3,437.0	4,300.0	–863.0
Flatten	Jan'12	3,846.7	3,883.7	–37.0	2,973.6	3,112.3	–138.7	---	---	---
	Feb'12	3,310.9	3,306.5	4.4	3,310.9	3,319.7	–8.8	2,976.2	3,494.8	–518.6
	Mar'12	4,015.8	3,976.7	39.1	4,015.8	3,899.6	116.2	4,015.8	3,482.5	533.3
	Apr'12	5,279.5	5,242.2	37.3	5,279.5	5,139.4	140.1	5,279.5	4,413.4	866.1
	May'12	4,315.7	4,417.0	–101.3	4,315.7	4,601.0	–285.3	3,958.6	4,321.1	–362.5
	Jun'12	3,437.0	3,447.8	–10.8	2,604.1	2,651.1	–47.0	##	##	##
De-flatten	Jan'12	3,410.7	6,680.7	–3,270.0	2,551.7	4,721.2	–2,169.5	---	---	---
	Feb'12	3,310.9	9,756.1	–6,445.2	3,310.9	12,826.9	–9,516.0	5,506.4	2,981.4	2,525.0
	Mar'12	4,015.8	7,778.7	–3,762.9	4,015.8	8,705.9	–4,690.1	4,015.8	10,189.4	–6,173.6
	Apr'12	5,279.5	10,330.9	–5,051.4	5,279.5	8,832.3	–3,552.8	5,279.5	8,072.5	–2,793.0
	May'12	4,315.7	9,527.1	–5,211.4	4,315.7	8,535.5	–4,219.8	4,315.7	8,890.2	–4,574.5
	Jun'12	3,437.0	10,381.4	–6,944.4	3,437.0	13,218.6	–9,781.6	3,437.0	9,310.5	–5,873.5

Notes: N/A* flatten window 1st removed all attributes lying between the time point zero (attribute name ending '–0') and the time point before horizon values. Second, it transforms the corresponding time point zero of the specified label stem to the actual label. Last, it re-represents all values relative to the last known time value for each original dimension including the label value. So, it produces result exactly after from horizon selected.
(##) – Lack of input data for flatten windowing.
(---) – 22 days skipped for 22 days a-head prediction.

4.3 Experiment results

The overall experiments are done in several ways, such as applying three models; 1 day ahead and 5 days ahead for predicting short-term period and for predicting long-term period 22 days ahead model is applied. These three models again built with three different windowing operators. So, each operator has three individual models and total nine models are applied to the experiment datasets. Though there are nine different models but the basic model structure is only three. Figure 3 shows the basic tree structure of these models. Three models produce three different types of result and there is a significant difference between one models of result to another model result. The difference and accuracy of result mostly depends on the selection of windowing operator type with produce the optimised input for the SVR models. The SVR kernel parameters are same in all models, since these values are acquired in an iterative model execution process. A sample results Table 4 is given, where results for S&P 500 for the month of January 2013 to June 2013 are shown with the actual index closing value.

Table 5 shows the prediction results for DSE index data using three different windowing techniques with SVR for the month of January 2012 to June 2012.

4.4 Mean average percentage error

Performance evolution or error rating is the important part for any research study. It helps the researchers to evaluate their experiment's results. Many types for performance evolution techniques have been used in different research study based on the research experiment design. Here in this study mean absolute percentage error (MAPE) was used. MAPE helps to show the gap between absolute and predicted value very clearly. So, in many financial research analysis researchers preferred MAPE for evaluating their model performance (Meesad and Rasel, 2013a, 2013b; Pai and Lin, 2005).

The MAPE is also known as mean absolute percentage deviation (MAPD). It measures the accuracy of a method for constructing fitted time series values in statistics, specifically in trend estimation. It usually expresses accuracy as a percentage, and is defined by the formula:

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

Here MAPE stands for mean average percentage error (MAPE) between actual share prices (*A*) and predicted share price (*P*). And '*n*' is the number of days to take into count.

The difference between *A* and *P* is divided by the Actual value *A* again. The absolute value in this calculation is summed for every fitted or forecasted point in time and divided again by the number of fitted points *n*. multiplying by 100 makes it a percentage error (Lee, 2009).

Table 6 shows the error committed during the execution of different model for the experiment datasets. Outputs from the different model are tabulated with their respected actual price value and than calculated the MAPE for the testing dataset for the two domain indexes. Here in Table 6, MAPE value is given in average manner for the month January 2013 to June 2013 for S&P 500 index, January 2012 to June 2012 for DSE index and June 2010 to August 2010 for IBM index.

Table 6 MAPE (error) for S&P 500, DSE index and IBM index

Window type	*Model*	*Index Name*		
		S&P 500	*DSE*	*IBM*
Rectangular	1 day ahead	0.65	0.65	0.02
	5 days ahead	0.74	0.48	0.57
	22 days ahead	1.43	0.82	3.22
Flatten	1 day ahead	0.01	0.08	0.01
	5 days ahead	0.03	0.19	0.47
	22 days ahead	0.14	0.74	0.21
De-flatten	1 day ahead	3.99	6.84	5.84
	5 days ahead	3.86	8.28	6.51
	22 days ahead	2.45	5.40	6.19

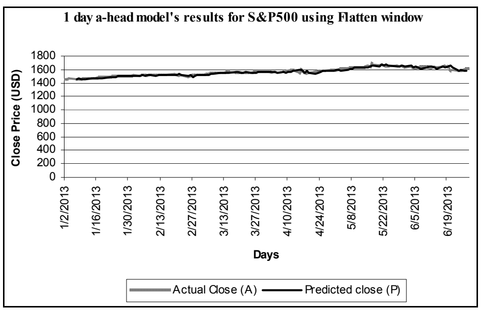
Notes: S&P 500: Average MAPE is calculated and tabulated for the month of Jan’13 to June’13.
DSE: Average MAPE is calculated and tabulated for the month of Jan’12 to June’12.
IBM: Average MAPE is calculated and tabulated for the month of Jun’10 to Aug’10.

From Table 6, it can be clearly state that the proposed Win-SVR models can apply to any stock index data for the purpose of predicting stock, if and only if they have same type of attributes that are used in the models. Moreover, the error rate among S&P 500 index, IBM index and DSE index is quite similar, which indicate that results produced from different indexes are depend on dataset that are used into model, indexes are independent.

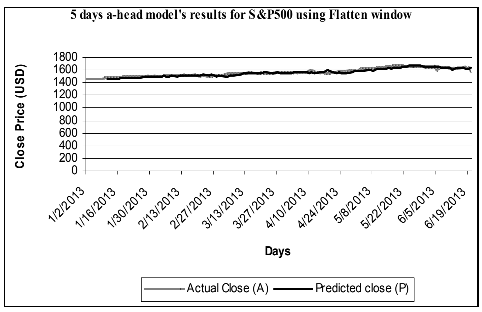
4.5 Graphical illustration of experiment results

4.5.1 Result illustration for S&P 500 index (Jan’13 to Jun’13)

Figure 4 (a) 1 day ahead result for S&P 500 index using flatten windowing (b) 5 days ahead result for S&P 500 index using flatten windowing (c) 22 days ahead result for S&P 500 index using flatten windowing

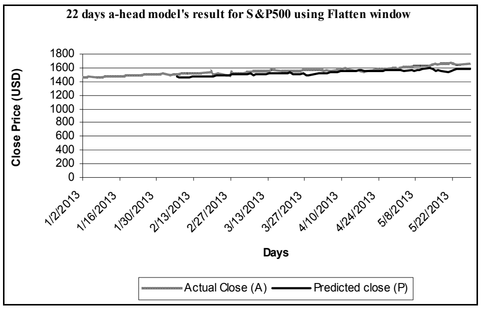


(a)



(b)

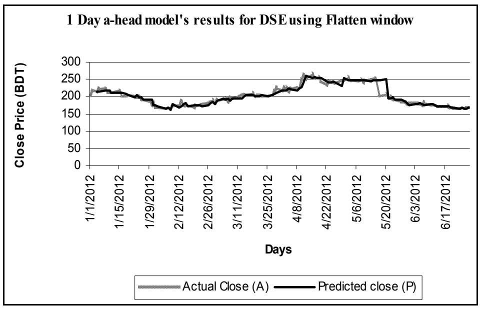
Figure 4 (a) 1 day ahead result for S&P 500 index using flatten windowing (b) 5 days ahead result for S&P 500 index using flatten windowing (c) 22 days ahead result for S&P 500 index using flatten windowing (continued)

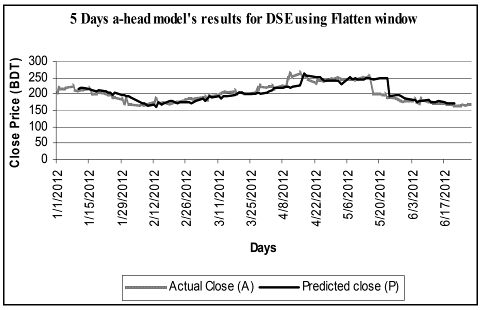


(c)

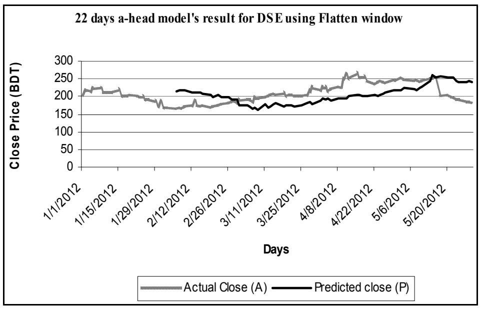
4.5.2 Result illustration for DSE index (Jan'12 to Jun'12)

Figure 5 (a) 1 day ahead result for DSE index using flatten windowing (b): 5 days ahead result for DSE index using flatten windowing (c) 22 days ahead result for DSE index using flatten windowing





(a) (b)



(c)

Figure 4(a), Figure 4(b), Figure 4(c) illustrates the outputs come from Win-SVM model for S&P 500 indexes. In these three models flatten windowing operator produces good prediction other than rectangular and de-flatten window. Though rectangular window can also produce good results, but flatten windowing is better option between these two (ref:

Table 6). De-flatten is the worst option to use among these three types of windowing operator because the result comes this are very erroneous with the actual price.

Similarly, Figure 5(a), Figure 5(b), Figure 5(c) shows the three best result from three models for DSE index. The short term prediction results are good by using flatten windowing and for long term period prediction, flatten windowing is also the best option among the three windowing operators. Here is also same case study found that rectangular and flatten window can produce good prediction results, whereas de-flatten windowing is the worst option.

5 Conclusions

5.1 Discussions

The aim of this study is to introduce an improved SVM modelling technique for forecasting financial time series data for stock market. As described before the domain of this study are S&P 500 index; is an international stock index, and another is DSE, Bangladesh; a local index from Bangladesh. Three basic models are built by using three different windowing; namely Normal rectangular windowing operator, Flatten windowing operator and de-flatten windowing operator. Models are applied for short period of time prediction; like 1 day ahead (daily) and 5 days ahead (weekly) and for long time period prediction; like 22 days ahead (monthly). After undertaking iterative experiments, a good combination of SVR kernel parameter's values are found (ref: Table 2) and also windowing component's values are determined to feed the input to the machine learning algorithm; SVR (ref: Table 1). Finally, it is found that both rectangular and flatten windowing operator are the good option for using as data pre-process or input selection technique for machine learning algorithm; SVR, among these three operators are used in this study. To do the experiments and analysis rapid miner 5.3 version; a free java-based data analysis tools is used.

5.2 Limitation and scopes

In this research work only three types of windowing operators are used. Still there is a scope to try other available windowing operators and windowing functions to produce optimised inputs for machine learning algorithm to analyse the trend pattern in order to produce more reliable outputs. Furthermore, for doing this study only price value are used. News related to stock market or other factors are not taken into account in this study. So, text mining or ontological analysis could be good option to add with this study to improve the prediction results.

References

Cao, L. (2003) 'Support vector machines experts for time series forecasting', *Neurocomputing*, Vol. 51, pp.321–339.

Chen, K-Y. and Ho, C-H. (2005) 'An improved support vector regression modeling for Taiwan Stock Exchange market weighted index forecasting', *The IEEE International Conference on Neural Networks and Brain*, pp.1633–1638.

Debasish, B., Srimanta, P. and Dipak, C.P. (2007) 'Support vector regression', *Neural Information Processing – Letters and Reviews*, Vol. 11, No. 10, pp.203–224.

Hsu, C-W., Chang, C-C. and Lin, C-J. (2010) *A Practical Guide to Support Vector Classification*, Initial version: 2003, Last updated version: 2010.

Hu, Y. and Pang, J. (2008) 'Financial crisis early warning based on support vector machine', *International Joint Conference on Neural Networks*, pp.2435–2440.

Huang, W., Nakamori, Y. and Wang, S-Y. (2005) 'Forecasting stock market movement direction with support vector machine', *Journal of Computers & Operations Research*, Vol. 32, pp.2513–2522.

Ince, H. and Trafalis, T.B. (2007) 'Kernel principal component analysis and support vector machines for stock price prediction', in: *IIE Transections*, Vol. 39, No. 6, pp.2053–2058.

Kannan, K.S., Sekar, P.S., Sathik, M.M. and Arumugam, P. (2010) 'Financial stock market forecast using data mining techniques', *Proceedings of the International Multiconference of Engineers and Computer Scientists*, pp.555–559.

Kara, Y., Boyacioglu, M.A. and Baykan, Ö.K. (2011) 'Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul Stock Exchange', Journal of Expert Systems with Applications, Vol. 38, pp.5311–5319.

Kato, R., Yata, N. and Nagao, T. (2010) 'Stock market prediction using classifier system based on incident pattern of wave template', *SICE Annual Conference 2010*, 18–21 August, pp.1176–1179.

Kazem, A., Sharifi, E., Hussain, F.K., Saberi, M. and Hussain, O.K. (2013) 'Support vector regression with chaos-based firefly algorithm for stock market price forecasting', *Journal of Applied Soft Computing*, Vol. 13, pp.947–958.

Kim, K.J. (2003) 'Financial time series forecasting using support vector machines', *Journal of Neurocomputing*, Vol. 55, pp.307–319.

Lee, M-C. (2009) 'Using support vector machine with a hybrid feature selection method to the stock trend prediction', Journal of Expert Systems with Applications, Vol. 36, pp.10896–10904.

Lu, C-J., Chang, C-H., Chen, C-Y., Chiu, C-C. and Lee, T-S. (2009) 'Stock index prediction: a comparison of MARS, BPN and SVR in an emerging market', *Proceedings of the IEEE IEEM*, pp.2343–2347.

Lu, C-J., Lee, T-S. and Chiu, C-C. (2009) 'Financial time series forecasting using independent component analysis and support vector regression', *Journal Decision Support Systems*, Vol. 47, pp.115–125, Elsevier.

Lucas, K., Lai, C., James, N. and Liu, K. (2010) 'Stock forecasting using support vector machine', *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics*, pp.1607–1614.

Mager, J., Paasche, U. and Sick, B. (2008) 'Forecasting financial time series with support vector machines based on dynamic kernels', *IEEE Conference on Soft Computing in Industrial Applications (SMCia/08)*, 25–27 June, pp.252–257.

Meesad, P. and Rasel, R.I. (2013a) 'Dhaka Stock Exchange trend analysis using support vector regression', *Advances in Intelligence Systems and Computing*, Vol. 209, pp.135–143, Springer.

Meesad, P. and Rasel, R.I. (2013b) 'Predicting stock market price using support vector regression', *International Conference on Informatocs, Electronocs & Vision (ICIEV)*.

Ni, L-P., Ni, Z-W. and Gao, Y-Z. (2011) 'Stock trend prediction based on fractal feature selection and support vector machine', *Journal of Expert Systems with Applications*, Vol. 38, pp.5569–5576.

Pai, P-F. and Lin, C-S. (2005) 'A hybrid ARIMA and support vector machines model in stock price forecasting', *Omega*, Vol. 33, pp.497–5005, Elsevier.

Smola, A.J. and Schölkopf, B. (2004) 'A tutorial on support vector regression', *Statistics and Computing*, Vol. 14, pp.199–222.

Thissena, U., van Brakela, R., de Weijerb, A.P., Melssena, W.J. and Buydensa, L.M.C. (2003) 'Using support vector machines for time series prediction', *Chemometrics and Intelligent Laboratory Systems*, Vol. 69, pp.35–49.

Wang, Y., Wang, B. and Zhang, X. (2012) 'A new application of the support vector regression on the construction of financial conditions index to CPI prediction', *Journal of Procedia Computer Science*, Vol. 9, pp.1263–1272.

Xue-Shen, S., Zhong-Ying, Q., Da-Ren, Y., Qing-Hua, H. and Hui, Z. (2007) 'A novel feature selection approach using classification complexity for SVM of stock market trend prediction', *14th International Conference on Management Science & Engineering*, pp.1654–1659.

Zhou, J., Bai, T., Zhang, A. and Tian, J. (2008) 'Forecasting share price using wavelet transform and LS-SVM based on chaos theory', in: *IEEE Coference on Cybernetics and Intelligent System*, pp.300–304.