

Sliding-Window based Stock price Forecasting expert system using LSSVR

By

Rohith Kumar K -2015103527

Umapathi C-2015103613

Naveen Kumar M-2015103519

A project report submitted to the

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

in partial fulfillment of the requirements for

the award of the degree

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Project Guide

Dr. Angelin Gladston

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY, CHENNAI – 25

January 2019

Abstract

Time series forecasting has been widely used to determine the future prices of stock, and the analysis and modeling of finance time series importantly guide investors' decisions and trades. In addition, in a dynamic environment such as the stock market, the nonlinearity of the time series is pronounced, immediately affecting the efficacy of stock price forecasts. Thus, this paper proposes an intelligent time series prediction system that uses sliding window metaheuristic optimization for the purpose of predicting the stock prices of Taiwan construction companies one step ahead. It may be of great interest to home brokers who do not possess sufficient knowledge to invest in such companies. The system has a graphical user interface and functions as a stand-alone application. The developed hybrid system exhibited outstanding prediction performance and it improves overall profit for investment performance. The proposed model is a promising predictive technique for highly nonlinear time series, whose patterns are difficult to capture by traditional models.

Problem Statement

Time series forecasting consists in a research area designed to solve various problems, mainly in the financial area. It is noteworthy that this area typically uses tools that assist in planning and making decisions to minimize investment risks. This objective is obvious when one wants to analyze financial markets and, for this reason, it is necessary to assure a good accuracy in forecasting tasks. Machine learning (ML) is coming into its own that can play a key in a wide range of critical applications. In machine learning, support vector machines (SVMs) have many advanced features that are reflected in their good generalization capacity and fast computation. They are also not very sensitive to assumptions about error terms and they can tolerate noise and chaotic components. Notably, SVMs are increasingly used in materials science, the design of engineering systems and financial risk prediction. Also, most methods that are in use are only applicable to a small portion of stock markets and usually such models do not generalize well to all stocks. Additionally, existing libraries are highly efficient in obtaining the optimal hyper parameters to be used in LSSVM and other algorithms.

Since time series data can be formulated by regression analysis, LSSVR is very efficient when applied to the issue at hand. However, the efficacy of LSSVR strongly depends on its tuning hyper parameters, which are the regularization parameter and the kernel function. Inappropriate settings of these parameters may lead to significantly poor performance of the model. Therefore, the evaluation of such hyper parameters is a real world optimization problem. Because the performance of SVR-based models strongly depends on the setting of its hyper parameters, they used to be set in advance based on the experience of practitioners, by trial-and-error, or using a grid search algorithm. Thus, finding the optimal values of regularization and kernel function parameters for SVR-based models is an important and time-consuming step. Therefore, a means of automatically finding the hyper parameters of SVR, while ensuring its generalization performance, is required.

Related Works

Investors in stock market primarily traded stocks based on intuition before the advent of computers. The continuous growth level of investing and trading necessitate a search for better tools to accurately predict the market in order to increase profits and reduce losses. Statistics, technical analysis, fundamental analysis, time series analysis, chaos theory and linear regression are some of the techniques that have been adopted to predict the market direction. However, none of these techniques has been able to consistently produce correct prediction of the stock market, and many analysts remain doubtful of the usefulness of many of these approaches. However, these methods represented a base-level standard which neural networks must outperform to command relevance in stock market prediction. Although the concept of artificial neural networks (ANN) has been around for almost half a century, only in the late 1980s could one ascertain that it gained significant use in scientific and technical presentations. There are quite a lot of research works on the application of neural networks in economics and finance . According to , White published the first significant study on the application of the neural network models for stock market forecasting. Following White's study, several research efforts were carried out to examine the forecasting effectiveness of the neural network models in stock markets. Among the earlier studies, the work in and can be mentioned. However, in another contribution, Yoda in investigated the predictive capacity of the neural network models for the Tokyo Stock Exchange. In neural network models was used to forecast various US stock returns. Also, in neural network models was used to select the stock from the Canadian companies.

The impact of fundamental analysis variables has been largely ignored. In this work, we explore the combination of the technical analysis and fundamental analysis variables for stock market prediction with the objective of attaining improved stock market prediction.

Proposed System Architecture and Dataset

Decision to buy or sell a stock is very complicated since many factors can affect stock price. This work presents a novel approach, based on least squares support vector regression (LSSVR), to constructing a stock price forecasting expert system, with the aim of improving forecasting accuracy. The intelligent time series prediction system that uses sliding-window metaheuristic optimization is a graphical user interface that can be run as a stand-alone application. The system makes the prediction of stock market values simpler, involving fewer computations, than that using the other method that was mentioned above . Additionally, the proposed system automatically fetches the latest stock data for any given company and date range

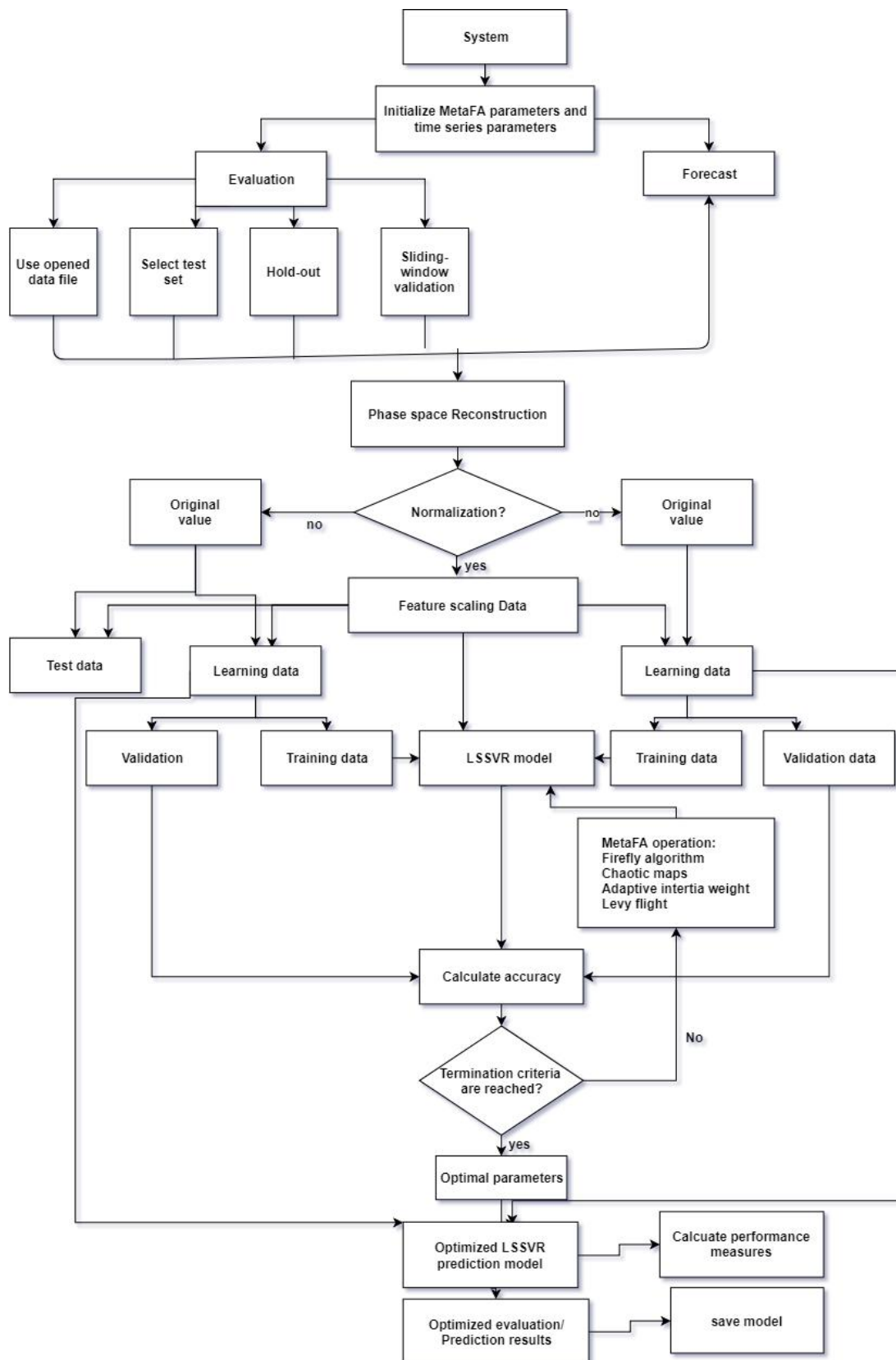
Historically prices were taken from Yahoo! Finance, a publicly accessible website, as they were by Six years (October 5, 2011 to May 31, 2017) of daily data on five company stocks were downloaded from Quandl.

The data were closing stock prices.

The features of the dataset includes:

- Date
- Open
- High
- Low
- Close

Architecture Diagram



Detailed Module design

Preprocessing

Many real world data is dirty and needs to be cleaned before used in code. Here we use data Preprocessing for our stock price data which contains unknown data, categorical data. Preprocessing of our data includes below steps :

1. Elimination of missing data
2. Conversion of Categorical Data
3. Splitting data into training and test data sets

Elimination of missing data

In the elimination of missing values we used **na.omit()** function, which returns the object with incomplete cases removed.

Conversion of Categorical Data

In our dataset, the date field contains the data in string type, but for our manipulations we need a stock data in date type. So that we define a **wantDate()** function which convert a string type date into date type.

Splitting data into training and test data sets

CUR represents the current stock data. All prediction is done from this day forward. All data from 1:CUR is the known stock value upto today.

Let ratio of TEST:TRAIN be 60:40% split .So set CUR 60% of DATA_SIZE

Window size. It has been observed that if we wish to predict stock for X days in advance, the window size must have proportionally equal or greater than X days of past data [Window size starts from 1]. The training set of the data is of size CUR starting from 1

Sliding Window Method

As suggest in, the learning dataset used in this study was collected within a sliding window. Block Diagram depicts the sliding-window and phase space construction. Since the forecast is one step ahead (hence the term, “one-step ahead forecasting”), the forecast horizon is 1. In the first validation, the working window includes p historical observations ($x_1, x_2, x_3, \dots, x_p$) which are used to forecast the next value x_{p+1} . In the second validation, the oldest value x_1 is removed from the window and the latest value x_{p+1} is added, keeping the length of the sliding window constant at p . The next forecast value will be x_{p+2} . The window continues to slide until the end of the dataset is reached. If the number of observations is N , then the total number of validations is $(N-p)$.

The algorithm for the sliding window is given as follows:

Algorithm : sliding Window

Input : data [stock data]

Output : A data frame of a lagged stock data

1. $LAG \leftarrow 1$
2. $y \leftarrow$ remove first LAG rows from data
3. reset row indices of y
4. $x \leftarrow$ remove last LAG rows from data
5. $train \leftarrow$ merge (x,y) into a dataframe
6. rename column name to x and y
7. return train

IMPLEMENTATION DETAIL (30%)

- Dataset containing stock price are uploaded
- Preprocessing had been done on the dataset
- Preprocessed dataset was given into the phase space reconstruction and the lagged stock data was extracted.

Now the lagged stock data was ready for training

Snapshots

Preprocessing Output

Date	Open	High	Low	Close
25-01-2019	11376.77	11441.22	11256.76	11278.82
24-01-2019	11345.29	11375	11302.25	11356.87
23-01-2019	11436.08	11447.59	11314.04	11333.77
22-01-2019	11474.84	11474.99	11366.19	11420.51
21-01-2019	11429.7	11496.21	11394.17	11455.68
18-01-2019	11423.06	11437.43	11360.53	11408.06
17-01-2019	11420.57	11438.23	11347.59	11408.86
16-01-2019	11406.31	11435.41	11379.66	11392.41
15-01-2019	11264.64	11397.93	11264.64	11387.91
14-01-2019	11324.31	11327.42	11189.33	11235.24
11-01-2019	11346.37	11353.65	11241.11	11292.14

Fig . Sample Stock Data

Sliding Window Output :




		x		y	
	1	864.45		861.95	
	2	861.95		861.10	
	3	861.10		856.90	
	4	856.90		855.50	
	5	855.50		859.55	
	6	859.55		865.20	
	7	865.20		866.35	
	8	866.35		869.95	
	9	869.95		909.05	
	10	909.05		909.30	
	11	909.30		902.75	
	12	902.75		892.35	
	13	892.35		882.15	

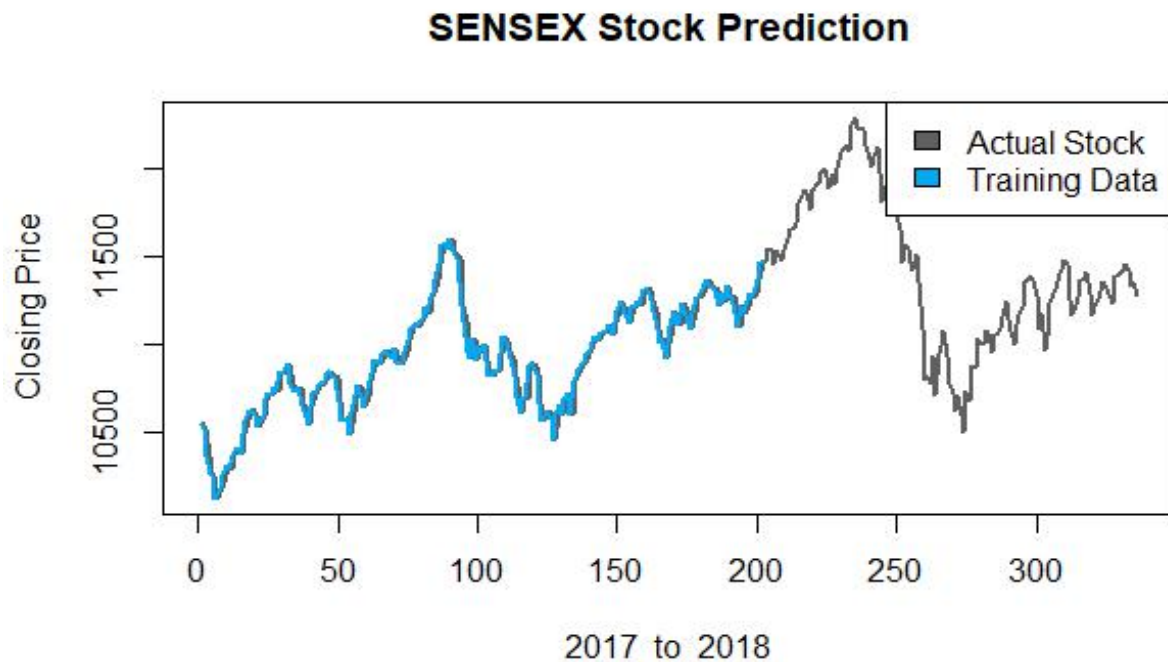
Fig . Lagged Stock Data

Console UI

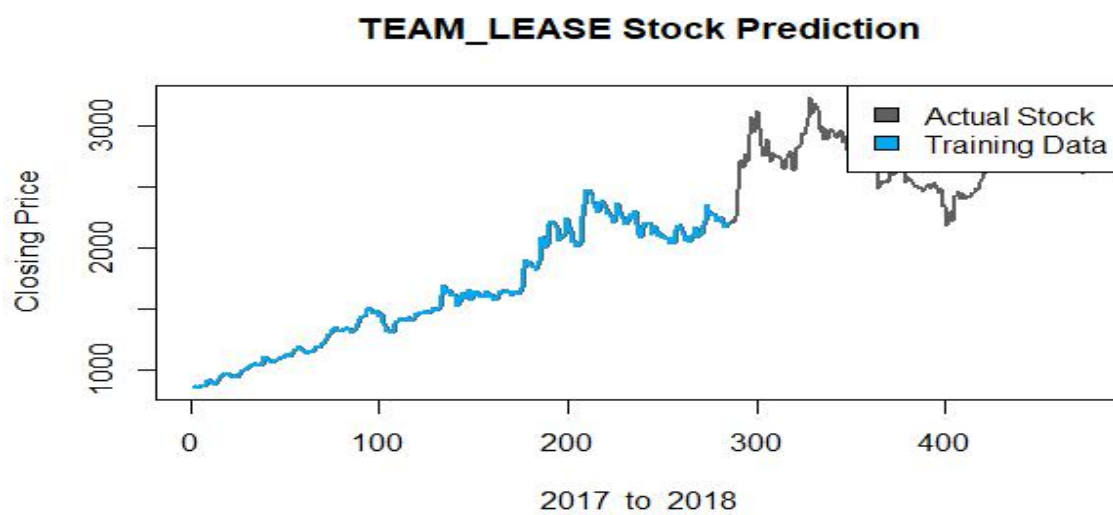
```
welcome to Stock Market Prediction
BY Rohith Kumar Umapathi Naveen Kumar
Choose the dataset you wish to predict the stock for
1. SENSEX          2. TEAM LEASE
3. UJJIVAN         4. LUX INDUSTRIES
5. MAX INDIA       6. THYROCARE
Enter your choice : 1
Do you wish to use default date range? (y|n) y
Fetching data for COMPANY
Predict stock for how many days in advance? (1 - 189 ) : 60
> |
```

Intermediate Output (Graphs)

Sensex data plots



Time Lease data plot



Metrics for Evaluation

R-square

R-squared tells us what percent of the prediction error in the y variable is eliminated when we use least-squares regression on the x variable.

As a result, r^2 is also called the **coefficient of determination**.

Many formal definitions say that r^2 tells us what percent of the variability in the y variable is accounted for by the regression on the x variable.

It seems pretty remarkable that simply squaring r gives us this measurement. Proving this relationship between r and r^2 is pretty complex, and is beyond the scope of an introductory statistics course.

$$R^2 = 1 - \left(\frac{\sum (actual - predict)^2}{\sum (actual - mean(actual))^2} \right)$$

References

1. C. M. Anish and B. Majhi Hybrid nonlinear adaptive scheme for stock market prediction using feedback FLANN and factor analysis J. Korean Statist. Soc. 2016.
2. R. Dash and P. Dash Efficient stock price prediction using a self evolving recurrent neuro-fuzzy inference system optimized through a modified differential harmony search technique Expert Syst .2016.
3. Bhattacharya, A. Konar, and P. Das, Secondary factor induced stock index time-series prediction using self-adaptive interval type-2 fuzzy sets ,2016.
4. J.-S. Chou, K.-H. Yang, J.P.Pampang, and A.-D. Pham, Evolutionary metaheuristic intelligence to simulate tensile loads in reinforcement for geosynthetic-reinforced soil structures, Comput. Geotechnics , 2015.
5. J.-S. Chou and A.-D. Pham, Smart artificial firefly colony algorithm based support vector regression for enhanced forecasting in civil engineering , Comput.-Aided Civil Infrastructure Eng, 2015.
6. D. Saini, A. Saxena, and R. C. Bansal, Electricity price forecasting by linear regression and SVM, in Proc. Int. Conf. Recent Adv. Innov. Eng, 2016.
7. J. Wang, R. Hou, C. Wang, and L. Shen, Improved v-support vector regression model based on variable selection and brain storm optimization for stock price forecasting, Appl. Soft Comput, 2016.
8. A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, Decision tree and SVM-based data analytics for theft detection in smart grid, IEEE Trans. Ind. Informat, 2016.
9. E. Avci, "Forecasting Daily and Sessional Returns of the Ise-100 Index with Neural Network Models", Dogus University Dergisi, vol. 2, no. 8, 2007.
10. T. Yamashita, K. Hirasawa, and J. Hu, "Application of Multi-branch Neural Networks to Stock Market Prediction", In Proc. of the International Joint Conference on Neural Networks, Montreal, 2005.