วารสารวิจัยสหวิทยาการไทย
Journal of Thai Interdisciplinary Research
Volume 11, Number 2, Pages 42 - 52

A cascade model of support vector regression and adaptive neuro-fuzzy inference system for next day stock price prediction

Phayung Meesad[1,*] and Tong Srikhacha[2]

[1]Faculty of Information Technology, King Mongkut's University of Technology North Bangkok
Bangkok 10800, Thailand

[2]TOT Public Company Limited, Bangkok 10210, Thailand

Abstract

Stock pricing is one of the challenging tasks in prediction due to noisy patterns with a slow changing curve. Global prediction techniques such as support vector (SV) show good enveloped prediction patterns but it tends to delay the prediction. Fuzzy prediction methods have better local optimizing and show significantly within training sets. Unfortunately, these sometimes generate surface oscillation effects in the output. This includes both global and local stock price rules with filtering of existing prediction models, output component base (OCB) and output-input iteration (OII) models, resulting in significant compromise for stock prediction.

Keywords: Stock prediction, time series data, hybrid neuro-fuzzy system, support vector regression, output component base, output-input iteration, cascade model

บทคัดย่อ

การทำนายราคาหุ้นเป็นงานที่มีความท้าทาย เนื่องจากรูปแบบที่มีข้อมูลอาจมีการรบกวน และมีการเปลี่ยนแปลงข้อมูลแบบเส้นโค้งอย่างช้า เทคนิคการทำนายแบบเชิงกว้าง เช่น ซัพพอร์ตเวกเตอร์แสดงให้เห็นรูปแบบการทำนายที่ดี แต่ก็มีแนวโน้มในการทำนายข้อมูลที่ล้าหลัง วิธีการทำนายแบบฟัซซีมีความเหมาะสมเชิงพื้นที่ที่ดีกว่า และแสดงนัยสำคัญภายในชุดการฝึกสอน แต่น่าเสียดายที่เทคนิควิธีฟัซซีบางครั้งให้ผลเอาต์พุตที่กระเพื่อม การวิจัยครั้งนี้พิจารณากฎราคาหุ้นทั้งเชิงกว้างและเชิงพื้นที่ร่วมกับการกรองของโมเดลการคาดการณ์ที่มีอยู่ ได้แก่ โมเดลฐานองค์ประกอบเอาต์พุต และโมเดลการทำซ้ำเอาต์พุต-อินพุต ให้ผลลัพธ์การพยากรณ์ที่ดีกว่าแบบดั้งเดิม

คำสำคัญ: การทำนายราคาหุ้น, ข้อมูลอนุกรมเวลา, ระบบไฮบริดแบบนิโรฟัซซี, การถดถอยแบบเวกเตอร์สนับสนุน, ฐานองค์ประกอบเอาต์พุต, การทำซ้ำเอาต์พุตอินพุต, โมเดลแบบคาสเคด

1. Introduction

Solving the prediction problem is a critical task in those real-world applications where accuracy is strictly related to the management of economic resources. The price to pay in this case is usually an increased complexity of the computing system. Thus, the trade-off between cost and benefits must always be taken into account [1].

Autoregressive Integrated Moving Average (ARIMA) model was introduced by Box and Jenkins [2] and seems to be a prominent method. It is attractive because it can capture complex arrival patterns, including those that are stationary, non-stationary, and seasonal [3]. However, tasks are labor-intensive and sophistication, which requires experts in the domain [4-6]. However, it is more

*Corresponding author; e-mail: pym@kmutnb.ac.th

suitable if one is not familiar with complex forecasting models to use a simpler method like an applied exponential smoothing model [7].

Alternatively, artificial intelligent techniques can be applied in forecasting by using prior knowledge learned directly from data. The main methods are Bayesian, fuzzy models, and neural networks. Bayesian method assumes that past data follows a particular probability distribution [8] with a simple formula which can provide a good prediction. However, it requires a substantially larger inventory investment. This is much more complex and expensive [4] and impacts the risk adverse pattern which is not good enough [9]. Fuzzy models are particularly well suited for human decision making including business and finance areas [10]. Neural networks are suitable as forecasting models for time series because of nonlinear mapping and learning abilities. However, with the effects of "black box," slow convergence, and local optimal solution, it occasionally produces some very wild forecasting values [5], which strongly limit its applications in practice [11]. If the training dataset does not cover the full range of operating conditions, the model may perform badly when deployed [12]. Hybrid systems can be applied to overcome these drawbacks. This provides better results than any single algorithm, or sometimes very similar to the best prediction method [13]. Special neural networks and prediction systems need to have an individual standard as well [14].

In time series forecasting, windowing is a method to discover optimal local patterns, which concisely describe the main trends in time series data at multiple time scales. It can capture meaningful patterns in a variety of settings [15]. Also, an internal structure within the time series data can be used to improve the performance [16].

In some cases, a large fraction of the training data can be discarded to improve the performance while maintaining high accuracy [17]. Forecasting is not based on only historical data but also the understanding of the significant factors hidden in the data; thereby it greatly improves the forecast accuracy [18]. Cortes and Vapnik [19] introduced a new theory named Support Vector Machine (SVM) that is very suitable to a regression model for time series prediction known as Support Vector Regression (SVR) [20-22].

The next day stock prices, a type of time series data, are highly temporal and dynamic changing. The SVR can provide global enveloped value of prediction; however, it tends to generate large error when the domain of testing input is shifted from training patterns [23]. The Adaptive Neuro-Fuzzy Inference System (ANFIS) [24] can be applied to general prediction. It is called fast convergent learning when initial parameters are appropriated and prearranged. The limitations found in this case are inconsistent training patterns and large shifting domains between training and testing patterns [25].

In this paper, contributing to time series prediction, the author proposes cascade models which are a combination of Output Component-Based Support Vector Regression (OCB) [23] model and Output-Input Iteration (OII) [25] based on Adaptive Neuro-Fuzzy Inference System (ANFIS) [24]. The remainder of this paper is organized as follows. Section 2 describes the related works. Section 3 illustrates the proposed cascade model for prediction of support vector machine and nero-fuzzy network. Section 4 shows the simulations results. Finally, discussion and conclusion are given in sections 5 and 6.

2. Related works

2.1 Support vector regression with output component base approximation

The Output Component-Based Support Vector Regression (OCB) model [23] is derived from the fact that the time series data with the training data trnXi in historical space (H-space) are different from the testing data in both shape and size even if it comes from the same source. In the case of domain pattern shifting, if SVR [20, 21] is trained, the result minimum error of the SVR does not always guarantee to provide accurate output in prediction.

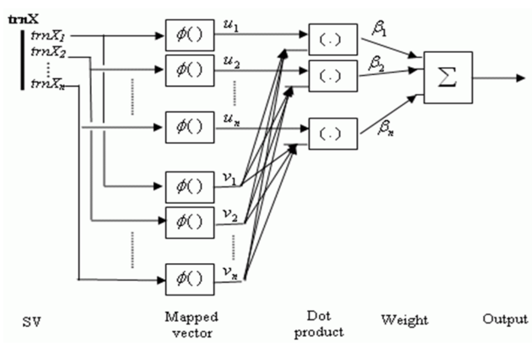

Figure 1. SVR structure model in training mode.

Figure 1 shows SVR model in training mode in which inputs of both mapping vector sets come from the same training data. If $\phi()$ is a Gaussian function for training, the user needs to define the constant σ. Another two predefined parameters required are C and ε for SVR's risk optimized solution in (1) and (2).

$$R = \frac{1}{2}\|\mathsf{w}\|^2 + C\left(\sum_{i=1}^{n}\phi\left(y_i - f\left(\mathsf{x}_i, \mathsf{w}\right)\right)\right). \quad (1)$$

$$|y - f(\mathsf{x}, \mathsf{w})|_\varepsilon = \begin{cases} 0, & |y - f(\mathsf{x}, \mathsf{w})| \le \varepsilon, \\ |y - f(\mathsf{x}, \mathsf{w})| - \varepsilon, & \text{otherwise,} \end{cases} \quad (2)$$

where w is the weighting parameter in which it is optimized equivalent to β in Figure 1; C is a balancing value between maximized margin and minimized empirical risk; and ε is an insensitive error found from (2). In the testing mode, the SVR output is based on suitable weighting factors of training data in H-space

$$trn = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n) \in \Re^l \times \Re\}.$$

The adjustment depends on appropriate parameters: C, ε, and σ. However, when it is evaluated with test data ($tstX$), the results may not provide good accuracy. Fortunately, the enveloped output is similar to real patterns in the Target space (T-space). The major effect comes from partial matching between H-space and T-space domains.

In case of non-complete matching between H-space and T-space, the composite analysis of input processing to output components may disclose significant hidden patterns. The jittering effects can be introduced at the final output as representative mapping from H- to T-space. Output Component Based-SVR (OCB) was proposed in [23] to handle the jittering effects. Figure 2 shows the network diagram of OCB model.

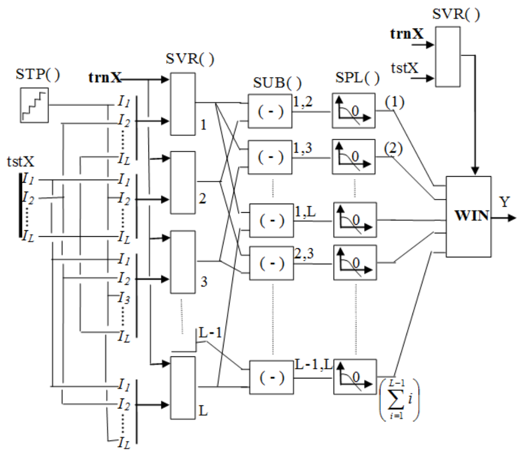

Figure 2. OCB structure model.

From Figure 2, SVR illustrates a standard ε-SVR module. STP stands for a stepping function. SUB stands for a subtractive function. SPL represents a Spline function used to evaluate approximate root values. WIN function represents a selector, which selects an appropriate winner output. Training data $trnX$ contains a set of records with L dimensional data. Both $trnX_i$ and $tstX_i$ have $\{I_1, I_2, \ldots, I_L\}$ format. The evaluations of each input component are performed for curve fitting functions in SPL. The expected values of each compared dimension are close to 0. The SVR function is defined as ε-SVR in (3).

$$SVR_i = \beta \cdot k(trnX \cdot vtstX_i) + b, \quad (3)$$

$$vtstX_i = \begin{cases} tstX_j; & i \ne j \\ 0 \ldots 1; & i = j \end{cases} \quad (4)$$

where $\beta = \alpha - \alpha^*$ and α, α^* are Lagrange multiplier pairs, and b is a constant. Gaussian kernel function $k()$, b is close to 0 by a constraint of $1 \le i < j \le L$.

The result of changing stepping values of each input component in (4) gives output SVR_i in (3), which is a directly related value. In case of complete mapping, all output graph lines of each component input will have the same intersection point as shown in Figure 3. This point can represent the most suitable output of $tstX_j$.

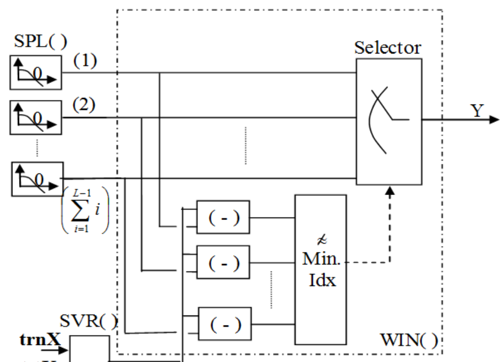

Figure 3. Structure of WIN function.

To compare the results of each dimension of input, each output pair is compared using SUB function in (5).

$$SUB_k = SVR_i - SVR_j \quad (5)$$

where

$$k = \begin{cases} j - i; & i = 1 \\ j + i + \sum_{m=1}^{L-1}(L - m); & i > 1 \end{cases}$$

and the number of SUB value is equal to $\sum_{i=1}^{L-1} i$.

If higher accuracy in prediction is needed, an increase of step resolution in (4) is required. However, this leads to a large consumption of the processing time. To reduce this effect cubic Spline interpolation technique [26] can be used by defining knot point (x, y) with $x =$ SUB, $y =$ input step values. All outputs of SPL function are candidate values for final outputs which is processed and selected by WIN function as shown in Figure 3. During prediction processes in T-space, the actual output value of the system is not known. So, all outputs from SPL function can be representatives of OCB outputs. At this state, a suitable output may refer to standard SVR output selected by WIN function in Figure 3. The closest SPL output to the SVR reference is selected to represent the final output of OCB model. Figure 3 can be represented by (6).

$$WIN = SPL_{(idx\,\min(|SUB_i - SVR(tstX,trnX)|))} \quad (6)$$

where idx is the index at the lowest value compared between SUB and SVR. The description of OCB algorithm in (6) assumes that all SVR models are already trained with a standard SVR for predefined the weighting β parameter.

The fact that T-space may shift from H-space, closest optimizing design of OCB tends to provide so better approximation than best optimizing with H-space as SVR model. In next section, the introduction of the OII model which it is based on strength extended input to ANFIS model is discussed.

2.2 Output-Input-Iteration applied to ANFIS model

The Output-Input-Iteration (OII) applies the benefit of ANFIS [24] by reducing the surface oscillation effects. The ideas come from extended overlapping of membership inputs by adding an extra value as a part of input dimensions. The extra input dimension is actual output in training mode and leaves the room for evaluation in the test mode. This case may reduce output surging because of this extra input dimension assistance. The evaluated results give the prediction values in the final stage of OII model.

The concepts of OII can be explained by (7) to (15). The input dimension with n records can be represented by (7).

$$x_{\theta,i} = [x_1, x_2, \ldots, x_n]_\theta\,. \quad (7)$$

OII output can be evaluated by using iterative ANFIS process with stepping values in parts of input dimensions. The stepping focuses on input dimension one by one in each record whereas the rest of the parts are kept as origin values as shown in (8) and (9).

$$x_{\theta,i}|_{i=1}^{n} = [x_1, x_2, \ldots, x_n, y]_\theta\,. \quad (8)$$

The processes inside OII function is composed of ANFIS or Takagi-Sukeno (TS) [24] fuzzy inference system in the test mode. The input pattern $x_{\theta,i}^{(f)}$ is rearranged in the format as in (9).

$$x_{\theta,i}^{(f)} = \begin{bmatrix} \left(V_1^{(\text{stp})}, x_2, \ldots, x_n, V_1^{(\text{stp})}\right)_1 \\ \left(x_1, V_2^{(\text{stp})}, \ldots, x_n, V_2^{(\text{stp})}\right)_2 \\ \vdots \quad \ldots \quad \vdots \\ \left(x_1, x_2, \ldots, V_n^{(\text{stp})}, V_n^{(\text{stp})}\right)_n \end{bmatrix}_\theta \quad (9)$$

where $V_i^{(\text{stp})}$, $i = 1, 2, \ldots, n$, is a step value, increased by $V^{(inc)}$ in each i step; $V^{(\text{stp})}$ is increased and related to (12) with the initial value $V^{(\text{low})} = 0, V^{(\text{hgh})} = 1$, and $V^{(\text{stp})} = 0.1$. These include ANFIS's initial parameters $\{k, c, \sigma\}$, which are created during ANFIS training phase.

The value $V_i^{(\text{dif})}$, which is the difference between the TS fuzzy output $V_i^{(\text{out})}$ and the forced input $V_i^{(\text{stp})}$ will be changed according to (10) to (12).

$$V_i^{(\text{dif})} = V_i^{(\text{out})} - V_i^{(\text{stp})} \quad (10)$$

$$V_i^{(\text{out})} = \text{TS}\left(x_{\theta,i}^{(f)}, \{k, c, \sigma\}\right) \quad (11)$$

$$V_{r+1}^{(\text{stp})} = V_r^{(\text{stp})} + V^{(\text{inc})} \quad (12)$$

where i represents the i^{th} iteration, $V^{(\text{low})} \leq V^{(\text{stp})} \leq V^{(\text{hgh})}$ and $V^{(\text{inc})} = \left(V^{(\text{hgh})} - V^{(\text{low})}\right)/n$. Each iteration, output precision can be increased by choosing a smaller value of $V^{(\text{inc})}$ that is related directly to $V^{(\text{low})}$ and $V^{(\text{hgh})}$ in (13) and (14), respectively. With this method the output will close to the root value when the iteration number is increased.

$$\text{new } V^{(\text{low})} = \left(\text{OII} - V^{(\text{inc})}\right) \quad (13)$$

$$\text{new } V^{(\text{hgh})} = \left(\text{OII} + V^{(\text{inc})}\right). \quad (14)$$

The final OII output can provide the last iteration in the process with decision according to (15). If the neighbors on the left and right of minimum different values have different signs, then the solution meets at zero point between these neighbor values.

$$Y^{\text{out}} = \text{OII}\left(x, \{k, c, \sigma\}\right) = \text{recur}\begin{cases} V^{(\text{stp})}_{\left(\arg\min\left|V_i^{(\text{dif})}\right|\right)} & ;\left(\forall V_i^{(\text{dif})} > 0\right) \vee \left(\forall V_i^{(\text{dif})} < 0\right) \\ \frac{1}{2}\left(V^{(\text{stp})}_{\left(\arg\min\left(\forall V_i^{(\text{dif})} > 0\right)\right)} + V^{(\text{stp})}_{\left(\arg\min\left(\forall V_i^{(\text{dif})} < 0\right)\right)}\right) & ;\ \text{otherwise.} \end{cases} \quad (15)$$

In the case of same sign, the solution can be determined by the average between these values. This case is affected from the H-space and T-space non-matching and cannot give curve intersection with the root axis. So, the average value is most suitable for this case.

The advantage of both OCB and OII is that they can support each other for better approximate prediction. To evaluate combined models, the next section describes the general cascade model which is used in section 4.

3. The proposed cascade models of SVR and ANFIS

OCB and OII design models can be combined with ANFIS and SVR to create 6 more complex models: M-ANFIS, M-OII, S-ANFIS, S-OII, O-ANFIS, and O-OII. The prefix M stands for "Mixed" inputs of both SVR and OCB to ANFIS or OII model. The prefix S is only the combination of SVR with inputs and the prefix O is only the combination of OCB with input to ANFIS or OII.

The applied model is illustrated in Figure 4 and represents all 6 candidate models. SV is represented by SVR, OCB, or both according to the prefix S, O, or M respectively. STP stands for a stepping function. SUB is subtraction and PD is closest pair-wise distance function.

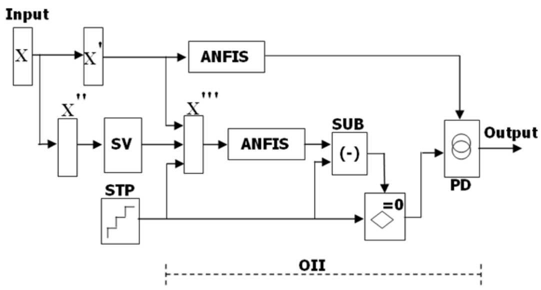

Figure 4. The proposed cascade model.

To describe this representative figure, the example of M-OII model: the SV box is replace with two sub-box functions (SVR and OCB). The same input (X'') is sent to SVB and OCB function. Then, these two outputs from SV box are added with others as dimension extension to X'' box. To describe S-OII model, the SV box has only SVR function. In this case, there is only one output from SV box direct to X'' box. The same way of O-OII model, the SV box has only OCB function.

In the case of ANFIS suffix; the example of M-ANFIS model, the OII part changes to simple ANFIS model. The main focus is based on stock prediction and given input as (16) to (19).

$$X_i = \{C_i, H_i, L_i, O_i\} \quad (16)$$

$$X_i' = \frac{1}{1.3 \cdot C_i}\{C_i, C_{i-1}, \ldots, C_{i-k}, H_i, \ldots, H_{i-k}, L_i, \ldots, L_{i-k}, O_i, \ldots, O_{i-k}\} \quad (17)$$

$$X_i'' = \frac{X_i}{\max} = \frac{1}{\max}\{C_i, H_i, L_i, O_i\} \quad (18)$$

$$X_{i,j}''' = \left\{Y_{i,j}', Y_i^{(IN)}, Y_j^{(STP)}\right\} \quad (19)$$

$$Y_{i,j}' = \begin{cases} X_{i,j}' & ; i \neq j \\ X_{i,j}' = Y_i^{(STP)} & ; i = j \end{cases} \quad (20)$$

$$Y_i^{(IN)} = \begin{cases} SVR_i^{(OUT)} & ;\text{for S type} \\ OCB_i^{(OUT)} & ;\text{for S type} \\ \left\{SVR_i^{(OUT)}, OCB_i^{(OUT)}\right\} & ;\ \text{for M type} \end{cases} \quad (21)$$

where C, H, L, and O are represented to stock values of Close, High, Low, and Open, respectively. $i = 1, 2, \ldots, k$, j is X_i' dimension, k is the number of previous data points (including the present data point), which are reformatted to be an input pattern. $Y^{(IN)}$ stands for $SVR^{(OUT)}$ in case of S-type or $OCB^{(OUT)}$ in O-type. It is combined into 2 dimensions for M-type usage. $Y^{(STP)}$ is a stepping value between 0 and 1.

The value of 1.3 time of C_i in (17) is used as a maximum value of i^{th} record for ANFIS usage and OII in (19). In case of OCB and SVR, it is suitable to use a normalized value based on max value in (18) because their original solution is depended on the classification principle.

Because of non-linear effect in OII parts, the result of SUB function in Figure 4 is generally given multiple root values. To find these solutions, the author selected a hybrid secant false position method [27]. This compromised method generally guarantees output performance by partition found in a ranking scale.

At the given state of root values, $Y^{(STP)}$ is represented candidate to PD function. The final output for i^{th} record is selected from the closest pair-wise distance between reference ANFIS output and candidate of $Y^{(STP)}$ at root state.

The design models from sections 2 and 3 are already to be tested and evaluated. The results are described in next section.

4. Simulations results

In this section, each separate model is tested to find its limitation and to select the appropriate types and constraints applied in a cascade model.

The subsection 4.1 defines data sets and tool for measuring. The separated modules are compared between each pair of OCB and SVR and OII and ANFIS.

The subsection 4.2.1 shows the tested results between OCB and SVR model strength with changed parameters; k days of stock price to extend present input dimension and inappropriate normalizing unit values. The OCB shows better prediction on average test results.

The subsection 4.2.2 compares the testing result of OII and ANFIS model by extra extended input with expected output. The average prediction slope of the difference between prediction output and actual output divided by the difference between guideline inputs used as based comparison. More strength output is better representative model for our cascade model. The testing results show that the OII model is significantly more stable than the ANFIS model.

These detail results are described in subsection 4.2. The final cascade model testing is shown in subsection 4.3. The testing results confirm that the applied OCB and OII as a cascade model provides an average result better than SVR and ANFIS.

4.1 Data preparation for training and testing of the proposed cascade model

The data sets for training and testing are collected from 10 stocks of Thailand. These 10 examples of stock prices consist of 2 stock groups and one total of SET index. The first group is bank group. There are 5 examples: BBL (Bangkok Bank plc.), KTB (Krung Thai Bank Plc.), SCB (The Siam Commercial Bank), TISCO (Tisco Bank Plc.), and TMB (TMB Bank Plc.). The second group is petrochemical group. There are 4 examples; IRP (Indorama Polymers Plc.), PTTCH (PTT Chemical Plc.), TPC (Thai Plastic and Chemical), and VNT (Viny Thai Plc.). All stock data for training the system are collected during 3rd January to 28th February 2006. The test data are collected during 1st September 2006 to 28th February 2007.

The statistics measurement techniques using in this paper are mean squared error (MSE) [28], Theil's U-Statistic statistic (U-Stat) [28], and Regression Error Characteristic curve (REC) [29, 30]. MSE is made positive by squaring each error, and then the squared errors are averaged. The smaller error value, the better forecasting is compared with the same time series. U-Stat allows a relative comparison of formal forecasting methods with naïve approach. If U-Stat is equal to 1, the naïve method is as good as the forecasting technique being evaluated. The smaller the U-Stat, the better forecasting technique is relative to naïve method. REC is a technique for evaluation and comparison of regression models that facilitates the visualization of the performance of many regression functions simultaneously in a single graph. A REC graph contains one or more monotonically increasing curves (REC curves) each corresponding to a single regression model. REC curves plot the error tolerance on the x-axis and the accuracy of a regression function on the y-axis. Accuracy is defined as the percentage of points predicted within the error tolerance. A good regression function provides a REC curve that climbs rapidly towards the upper-left corner of the graph. In other words, the regression function achieves high accuracy with a low error tolerance. The area over the REC curve (AOC) is a biased estimate of the expected error for a regression model. In the case of an error calculated using the absolute deviation (AD), the AOC is close to the mean absolute deviation (MAD).

4.2 Separated module testing

4.2.1 OCB and SVR model testing

The purpose of this testing is to confirm the benefit of OCB system design in section 2 and input defined in (18). To test the models, they were defined into 2 input models based on pre-arranged data inputs for both OCB and SVR testing. The first input model uses k days of stock price to extend present input dimension with normalization unit by using (22). To control unit value, this case uses 1.3 time of close price Ci in present time i.

$$X_i' = \frac{1}{1.3 \cdot C_i}\{C_i, C_{i-1}, \ldots, C_{i-k}, H_i, \ldots, H_{i-k}, L_i, \ldots, L_{i-k}, O_i, \ldots, O_{i-k}\}. \quad (22)$$

The second input model uses only present values of $\{O_i, H_i, L_i, O_i\}$ readjusted to unit values divided by max value; $V_{max} = m \cdot \max(test\ data)$

$$X_i'' = \frac{X_i}{V_{\max}} = \frac{1}{V_{\max}}\{C_i, H_i, L_i, O_i\}. \quad (23)$$

Both OCB and SVR model are tested with SET index data with (3) to (6) and case by case of input format as in (22) and (23). For SET data, the result of each prediction type testing case within (22) by adjusting k = 2, 3, and 4 is shown in Table 1. This is significant as the number of k-day back does not guarantee the accuracy of prediction output. It is not acceptable especially in the SVR model. It is worth noting that more k days used will cost more processing time and higher un-acceptable errors. There is only lowest price test case with m = 1.5 that SVR gives mse = 234.37 lower than OCB with mse = 240.46. However, the average ratio of AOC between OCB:SVR is 6.2609:6.2978. This means that OCB is slightly better than SVR. This is not a similar result of tool usage because the tool is sensitive to temporary value of error change but AOC calculates the average of area over the curve. This means that AOC is generally not sensitive to transient errors.

The lowest price prediction of SET data is shown in Figure 5. The blue error line of OCB gives an average error slightly closer to zero level than SVR. This is re-confirmed that the use of AOC tool is meaningful in comparisons.

The lowest price prediction of SET data is shown in Figure 5. In this case, the SVR has slightly better result in the area of transient data. However, OCB gives an average error slightly better than SVR (OCB: AOC = 6.2609 and SVR: AOC =.6.2978). The results from these tests significantly show that the OCB gives an average prediction better than SVR in terms of variation of normalizing inputs. So, OCB with reformatted input using (23) is preferred to use in a further combined system.

4.2.2 OII and ANFIS model testing

In this subsection, a test comparison of the ANFIS core system with the OII core system is described. The tests include control factors, input types, and data clustering types defined in next subsection. The results of final comparison between OII and ANFIS are described in 4.2.2.2.

Table 1. Error comparison results when changing k.

		k = 2			k = 3			k = 4		
		AOC	**Mse**	**ustat**	**AOC**	**mse**	**Ustat**	**AOC**	**Mse**	**Ustat**
CLOSE	OCB	**18.85**	**805.79**	**2.1728**	**24.926**	**1339.1**	**2.8353**	**73.778**	**7178.6**	**6.3501**
	SVR	60.475	3885.3	4.6621	71.904	5498.2	5.526	99.66	10424	7.5863
HIGH	OCB	**28.442**	**1309**	**5.5979**	**45.263**	**2885.6**	**8.2594**	**57.048**	**4125.7**	**9.8859**
	SVR	87.036	7740.3	13.43	99.612	10146	15.312	85.359	7467.2	13.218
LOW	OCB	**45.304**	**3836.4**	**3.9617**	**102.34**	**14053**	**7.7381**	**104.89**	**14085**	**7.8079**
	SVR	117.1	14099	7.7667	136.88	19378	9.0858	179.5	33183	11.873
OPEN	OCB	**22.37**	**1361.5**	**3.139**	**55.117**	**5334.3**	**6.289**	**80.859**	**9053.8**	**8.0815**
	SVR	131.72	17610	11.158	159.48	25940	13.484	163.38	27204	13.788

Table 2. Error comparison results when changing m.

		m=1			m=1.5			m=2		
		AOC	**mse**	**ustat**	**AOC**	**mse**	**Ustat**	**AOC**	**mse**	**ustat**
CLOSE	OCB	**6.9814**	**201.3**	**1.0726**	**7.0351**	**192.72**	**1.0316**	**9.1284**	**270.19**	**1.2314**
	SVR	96.866	10262	7.4141	8.3139	212.71	1.0909	30.402	1108	2.4964
HIGH	OCB	**6.9684**	**121.17**	**1.6936**	**4.1187**	**34.657**	**0.92479**	**7.1824**	**111.28**	**1.6637**
	SVR	66.199	4867.5	10.486	4.2087	35.13	0.93236	17.738	353.51	2.9357
LOW	OCB	**6.4242**	**319.68**	**1.2348**	**6.2609**	240.46	0.99964	**10.023**	**354.58**	**1.2269**
	SVR	77.028	6583.5	5.2009	6.2978	**234.27**	**0.98806**	26.407	935.91	2.0194
OPEN	OCB	**4.8411**	**80.656**	**0.766**	**3.0462**	**19.148**	**0.38773**	**6.0688**	**53.367**	**0.63246**
	SVR	70.286	5602.3	6.1781	3.382	22.075	0.41637	32.381	1084.9	2.8207

4.2.2.1 Control factor, input pattern type, and data clustering type defining

There are 2 input models according to prediction types as defined in (22) and (23). The equation (23) is suitable for support vectors in both OCB and SVR because of their original classification design model. However, it is not compromised if the input type in (59) is defined for neuro-fuzzy, both OII and ANFIS. Neuro-fuzzy is based on local optimizing model. It is a better learning system if learning data sets are unique; otherwise, the inconsistent records will degrade system learning especially in the area of neighbors of these conflicting records. So, equation (22) is preferred to use as a re-formed input model for testing both OII and ANFIS.

For system learning, there are some factors to be concerned such as the processing time and membership function of fuzzy nodes. The system should learn the data quickly with acceptable prediction errors. The processing time is controlled by the number of input dimensions or k day back data. To reduce time in the learning process, one can select the constant k to be 3. A larger k value which does not give significant prediction accuracy compared to the cost of time. The other concerning factors needed to be designed before system learning are membership functions (MF) of fuzzy nodes, which are applied for both inputs and outputs. The appropriated MF rules per total number of training data are found from experience illustrated in the next subsection.

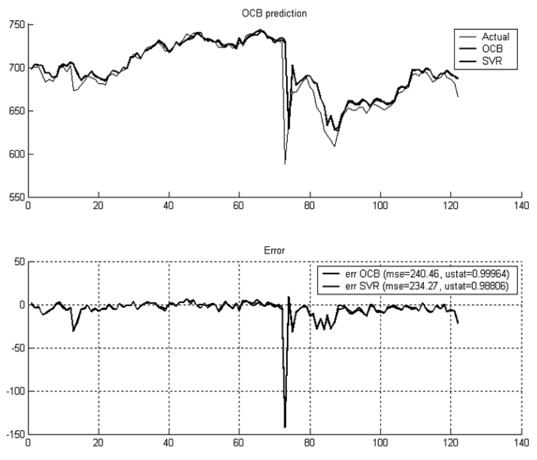

Figure 5. Lowest price error graph of OCB and SVR testing eq. (23) with m = 1.5.

4.2.2.2 OII and ANFIS testing

To compare 4 time series prediction models $\{C, H, L, O\}$ when using OII and ANFIS as prediction models, test comparison tools using the average prediction slope of the difference between prediction output $Y^{(S)}$ and actual output $Y^{(R)}$ were divided by the difference between guideline input $Y^{(I)}$ and $Y^{(R)}$ as shown in (24).

$$Y_{out}/Y_{in} = \overline{\sum_{j=1}^{n} \overline{\sum_{i=1}^{n} \left| \frac{(Y_{i,j}^{(S)} - Y_j^{(R)})}{(Y_i^{(I)} - Y_i^{(R)})} \right|}} \quad (24)$$

where i is step value in $[0 \ldots 1]$ and n is the total number of i for $Y_i^{(S)}$.

The comparison results between OII and ANFIS using training data of stock name SET as fuzzy learning and testing are shown in Figure 6. The suffix C, H, L, and O are represented to approximate type of Close, High, Low, and Open, respectively.

These testing controls of MF rule per train data from 0.1 to 1. From experience, the auther found that the appropriated MF rule/training data value is between 0.4 and 0.5. However, there is un-normal result in case of ANFIS model for predicting lowest price (ANFIS_L).

The OII model shows more stability than ANFIS model. Additional details of several tests of cascade models are described in the next section.

4.3 Testing model complexity

The test in this subsection is to select and confirm hybrid models of Figure 7. The test uses max in (18) or (23) for 10 data sets collected from 4.1. The comparison is based on average errors in comparison by using Ustat of total test data in case of m variation.

Because of processes limitation of ANFIS [24] in fuzzy rule layer may provide all zero outputs and cannot calculate outputs at normalization layer. In this case, output value of ANFIS is set to 0. To reduce this effect, it is suitable to separate the testing result into 2 groups: non-effect and effect of ANFIS limitation.

The first group consists of 7 examples of data; BBL, KTB, SCB, TISCO, TMB, PTTCH, and TPC. The rests are IRP, VNT, and SET, which receive ANFIS limitation effect and need to be filtered before calculating the average errors in Ustat tool. The average results of Ustat for all 7 examples of stock price data in first test group are shown in Table 3. Each model is evaluated by controlling m value for all 4-dimension prediction.

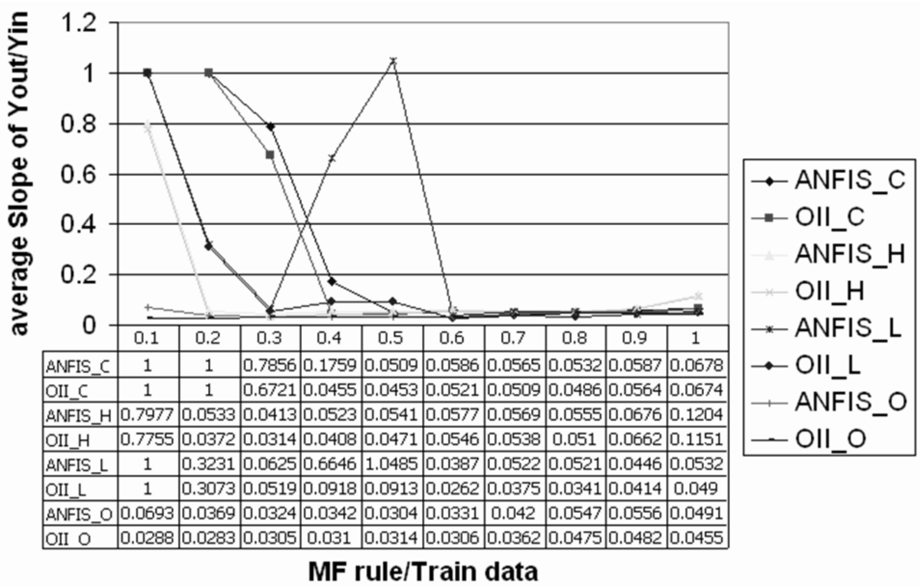

Figure 6. Strength input comparison between OII and ANFIS with SET data.

The results in Table 3 show that the combined techniques of both O-ANFIS and O-OII models are the two best out of 8 compared models, providing lowest average Ustat. With rank ordered from small to large error, others are S-OII, M-OII, S-OII, and S-ANFIS, respectively. SVR shows better results when m is defined between 1.3 and 1.5. However, by changing m, both O-ANFIS and O-OII show generally better in prediction than others.

In the case of testing in the second group with ANFIS limitation effects, the results are similar and confirm the result of first group after filtering prediction error point and substitute with the latest value of each prediction type. Table 4 shows average Ustat measurement before filtering and Table 5 gives the results after filtering.

Table 3. Average Ustat of testing results.

	Ustat			
	CLOSE	HIGH	LOW	OPEN
OCB	1.428	2.496	1.380	0.724
SVR	3.230	3.840	2.324	1.843
M-ANFIS	2.149	5.059	1.511	1.552
M-OII	1.941	4.769	1.466	1.363
S-ANFIS	4.888	7.764	3.542	3.425
S-OII	5.576	7.355	3.561	3.348
O-ANFIS	4.601	7.744	3.487	3.277
O-OII	5.037	7.752	3.499	3.279

It is confirmed that these candidate models using in NFSV system require data jittering for N networks. Each jittering set is defined for each fuzzy learning network. It is easy to reject un-similar neighbor intermediate output.

Table 4. Average Ustat of Second group testing with ANFIS limitation effect.

	Ustat			
	CLOSE	HIGH	LOW	OPEN
OCB	1.428	2.496	1.380	0.724
SVR	3.230	3.840	2.324	1.843
M-ANFIS	1.919	2.448	1.341	1.001
M-OII	1.630	1.946	1.218	0.774
S-ANFIS	1.670	2.660	1.182	0.900
S-OII	1.613	1.953	1.208	0.814
O-ANFIS	**1.219**	**1.403**	**1.090**	**0.720**
O-OII	**1.242**	**1.408**	**1.104**	**0.719**

Both O-ANFIS and O-OII models are the best combined model and should be selected as a candidate models, as shown in Table 5 at the last two columns.

Table 5. Second group testing with filtered ANFIS limitation effect.

	Ustat			
	CLOSE	HIGH	LOW	OPEN
OCB	1.696	1.305	1.365	0.889
SVR	3.054	3.101	2.271	1.463
M-ANFIS	1.942	1.897	1.323	0.952
M-OII	1.597	1.712	1.158	0.922
S-ANFIS	1.629	1.802	1.184	0.872
S-OII	1.521	1.773	1.126	0.857
O-ANFIS	**1.140**	**1.075**	**0.975**	**0.741**
O-OII	**1.158**	**1.079**	**0.982**	**0.760**

5. Discussion

SVR and OCB are original designs based on classification and applied regression for prediction applications. SVR seeks for global optimum while OCB seeks for near optimum. The advantages of classification techniques provide good prediction results similar to actual patterns. However, SVR has a major problem in providing suitable offset output when it has to process from inappropriate normalizing inputs. In the case of OCB, this effect can be reduced. It is especially reduces more if SVR generates poor reference outputs.

Fuzzy model both ANFIS and OII types have major disadvantages in local optimizing effects. After the system has learned, fuzzy models may not provide defuzzification outputs in test datasets or T-space. However, OII shows slight improvement.

Both OCB and OII have their limitations in time consumption based on iterative calls for their original models. However, these two techniques show significant compromising for stock prediction.

Fortunately, both lowest and open price of next day prediction show good accuracy because they yield benefits from classification in OCB output guidelines. The latest close price is a major effect to induce outputs of OCB. This value is close to the next day open and low price.

In the case of close and highest price of next day prediction, it does not show generally appropriate result because there are external factors directly involved with investor decision. The historical pattern does not guarantee the prediction results.

6. Conclusion and further study

In this paper, cascade models for stock prediction based on the combination of fuzzy logic and neural network concepts are proposed. There are 2 main applied models: OCB and OII types proposed herein. OCB is applied for classification of support vector technique and OII is based on local optimizing fuzzy model. These techniques have both advantages and disadvantages. However, they can support each other when they are synergistically combined to become a hybrid intelligent system. The OCB provides good enveloped output pattern but provides poor offset values. OII uses enveloped output patterns as expansion input guideline. This can reduce local optimizing effects in the case of fuzzy learning. This research result focused on only limitation of comparison to the original model ANFIS and SVR with financial Thailand stock price during 1st September 2006 to 28th February 2007. The domain of testing was quite specific about limitations.

References

[1] Panella, M. & Rizzi, A. (2006). Baseband filter banks for neural prediction. **CIMCA & AIWTIC'06**.

[2] Pankratz, A. (1983). **Forecasting with univariate Box-Jenkins models: Concepts and cases**. John Wiley & Sons, New York.

[3] Tran, N. & Reed, D. A. (2001). ARIMA time series modeling and forecasting for adaptive I/O perfecting. **The 15th International Conference on Supercomputing**.

[4] Harrison, H. C. & Qizhong, G. (1993). An intelligent business forecasting system. **The 1993 ACM Conference on Computer Science**.

[5] Wu, S. & Lu, R. (1993). Combining artificial neural networks and statistics for stock-market forecasting. **ACM Conference on Computer Science**.

[6] Robinson, L. (1997). Exploring time series analysis using APL. **ACM SIGAPL APL Quote Quad**, Vol. 27, No. 3.

[7] Meesad, P. & Srikhacha, T. (2006). Universal data forecasting with an adaptive approach and seasonal technique. **CIMCA & AIWTIC'06**, p. 66.

[8] Clay, G. R. & Grange, F. (1997). Evaluating forecasting algorithms and stocking level strategies using discrete-event simulation. **The 1997 Winter Simulation Conference**.

[9] Dangelmaier, W., Klöpper, B., Wienstroer, J., & Döring, A. (2006). Risk averse shortest path planning in uncertain domains. **CIMCA & AIWTIC'06**, p. v-xxii.

[10] Negnevitsky, M. (2002). **Artificial intelligence: A guide to intelligent system**. 1st Ed. Addison Wesley, p. 315-316.

[11] Rongjun, L. & Zhibin, X. (2006). A modified genetic fuzzy neural network with application to financial distress analysis. **CIMCA'06**, p. 120.

[12] Liu, F., Findlay, R. D., & Song, Q., A neural network based short term electric load forecasting in Ontario Canada. **CIMCA & AIWTIC'06**, p. 119.

[13] Wang, Y. & Shen, Y. (2006). Stock predictor algorithm: A control method dealing with distributed control systems. **CIMCA & AIWTIC'06**, p. 49.

[14] Jung, I., Park, S. C., & Wang, G. N. (2006). Two phase reverse neural network based human vital sign prediction system. **CIMCA & AIWTIC'06**, p. 135.

[15] Papadimitriou, S. & Yu, P. (2006). Optimal multi-scale patterns in time series streams. **Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data**.

[16] Wu, H., Salzberg, B., Sharp, G. C., Jiang, S. B., Shirato, H., & Kaeli, D. (2005). Subsequence matching on structured time series data. **Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data**.

[17] Xi, X., Keogh, E., Shelton, C., & Wei, L. (2006). Fast time series classification using numerosity reduction. **Proceedings of the 23rd International Conference on Machine Learning ICML'06**.

[18] Huang, L. & Zhong, J. (2006). ICA-based potential significant feature extraction for market forecast. **CIMCA & AIWTIC'06**.

[19] Cortes, C. & Vapnik, V. N. (1995). Support vector networks, machine learning, p. 273-297.

[20] Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. **In Advances in neural information processing systems, MIT press**, vol. 9, p.155-161.

[21] Vapnik, V., Golowich, S. E., & Smola, A. (1997). Support vector method for function approximation, regression estimation, and signal processiong. **Advances in Neural Information Processing Systems**, vol. 9.

[22] Mao, X. & Yang, J. (2006). Time series prediction using nonlinear support vector regression based on classification. **CIMCA & AIWTIC'06**, p. 13.

[23] Meesad, P. & Srikhacha, T. (2007). Data prediction by support vector regression with a decomposition method. **ECTI'07**.

[24] Jang, J. S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference systems. **IEEE Transactions on Systems, Man, and Cybernetics**, 23 (3), 665-685.

[25] Meesad, P. & Srikhacha, T. (2007). Applied neuro-fuzzy using support vector approximation for stock prediction. **NCCIT'07**.

[26] Prenter, P. M. (2008). Splines an variational methods, John Wiley & Son, Inc., p. 78-87.

[27] Pizer, M. S. & Wallace, L. V. (1983). **To compute numerically: Concepts and strategies**. Little Brown & Company Ltd., p.81, 107-108.

[28] Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). **Forecasting: Methods and application**. 3rd Ed, John Wiley & Son, Inc.

[29] Bi, J. & Bennett, K. P. (2003). Regression error characteristic curves. **The 20th International Conference on Machine Learning (ICML), Washington, DC**, p. 43-50.

[30] Aloisio, C. P. & Gerson, Z. (2007). Applying REC analysis to ensembles of particle filters. **IJCNN'07**.