Applied Neuro-Fuzzy using Support Vector Approximation for Stock Prediction

Tong Srikhacha* and Phayung Meesad**

ABSTRACT

In general case, stock pricing pattern is similar to a noisy pattern with a slow changing curve. The global prediction techniques such as support vector (SV) show good enveloped prediction patterns but they tend to delay the prediction. Fuzzy methods have better local optimizing and show significant within training sets. Unfortunately, these sometimes give the surface oscillation effect at the output. Combining our previous prediction models, output component base (OCB) and output-input iteration (OII), results in significant compromise for stock prediction.

Keyword: output component base, support vector, fuzzy, anfis, output-input iteration

1. INTRODUCTION

Solving a 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]

A prominent time series model like Autoregressive Integrated Moving Average (ARIMA) model was introduced by Box and Jenkins [2]. This time series analysis is attractive because it can capture complex arrival patterns, including those that are stationary, nonstationary, and seasonal ones [3]. This complicated technique requires an advanced level of knowledge, labor-intensive tasks, and sophistication only available from specialists [4]-[6]. However, it is more suitable if one who does not familiar with complex forecasting models can use a simple equation like applied exponential smoothing model for forecasting. [7]

In case of advance methods, artificial intelligences are alternative good techniques that can be applied in forecasting by using prior distributed data learning. The main methods are Bayesian, fuzzy models, and neural networks.

Bayesian method assumes that past data follows a particular probability distribution [8] with simple formula 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 averse pattern is not good enough [9]. Fuzzy models are particularly well suited for human decision making. Most fuzzy applications are popular in control and engineering, and even larger potential exist in business and finance [10]. Neural networks are very efficient adaptive forecasting models and superior to time series because of their excellent performances of treating non-linear data with self learning capability. However, with the effects of "black box," slow convergence, local optimal solution, it occasionally produces some very wild forecasting values [5] which strongly limited 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. The result gives better than any single algorithm, or sometimes very similar to the best prediction method [13].

Studying the historical background of abnormal behaviors of patterns is effective in prediction. Dealing with unusual, there are some effects when data are not applicable to an entire range of applications. Special neural networks and prediction systems need to have an individual standard as well [14].

Windowing is a method to discover optimal local patterns, which concisely describe the main trends in a time series 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].

Some forecasting methods, such as neural networks and ARIMA model, which are based on historical data, cannot find out the hidden remarkable features or signal decomposition behind the data. 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].

Some disadvantages of NNs, such as the ease over training or the lack of training data, the ease of relapse into local minimization, have not been solved until 1995, when Cortes and Vapnik [19] introduced a new theory named Support Vector Machine (SVM). This concept is applied to a regression model for approximate prediction, and it was named Support Vector Regression (SVR).

The successful impact of forecasting is tied directly to one's need, understanding, and involvement of forecasting. In this paper, we propose a prediction model which is a combination of Output Component-Based Support Vector Regression (OCB-SVR) model and Output-Input Iteration based on ANFIS (OII-ANFIS). This applied-SVR model with a technique for mapping is better prediction in jittering domain

* Department of Information Technology, Faculty of Information Technology, KMITNB

** Department of Teacher Training in Electrical Engineering, Faculty of Technical Education, KMITNB

between the historical space and target space.

A well known neuro-fuzzy system like Aadaptive Neuro-Fuzzy Inference System (ANFIS) has limitation in surface oscillation effects but it provides good performance in system learning. If such global optimizing as SV can hint the approximate of prediction value to local optimizing in case of neuro-fuzzy, we expect that the combined techniques will give the compromised to these benefits in the prediction focus.

The remains of this paper are organized as follows. Section 2 and 3 describe the OCB-SVR and OII-ANFIS model respectively. The proposed system design of the applied model is described in Section 4. The experimental performance testing with real stock pricing are given in Section 5. Finally, concluding remarks and further studies are given in Section 6.

2. OCB-SVR MODEL

The OCB-SVR [20] model was derived from the fact that the time series data with the pattern $trnX_i$ in Historical space (H-space) may be different both shape and size even providing similar patterns. In this case, if SVR is trained, the resulting minimum error of the SVR does not always guarantee to provide accurate output prediction.

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)

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

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

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 in (2).

In testing mode, the SVR output is based on suitable weighting factors of training data in H-space, $trn = \{(X_1,Y_1),(X_2,Y_2),...,(X_n,Y_n) \in \Re^l \times \Re\}$. 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 domain. Another factor is an optimized effect that it gives approximate outputs, which tend to be delay outputs. Another disadvantage is that it is most critical in prediction focused.

Optimization of parameters of the SVR model is depended on support vector points selected from representative multivariate in H-space. Because of a constraint when solving the quadratic equation, there are some dimensions of input variables scattering nearby root values.

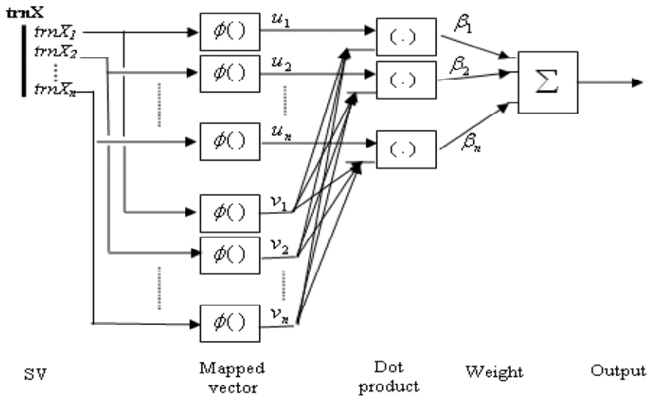

***Figure 1**: SVR structure model in training mode.*

The analysis decomposition of based inputs on output components discloses significant hidden patterns. The jittering effect can be selected to final output as representative mapping from H- to T-space. We propose a prediction model called Output Component Based–SVR (OCB-SVR) to handle the jittering effect. Figure 2 shows the network diagram of OCB-SVR model.

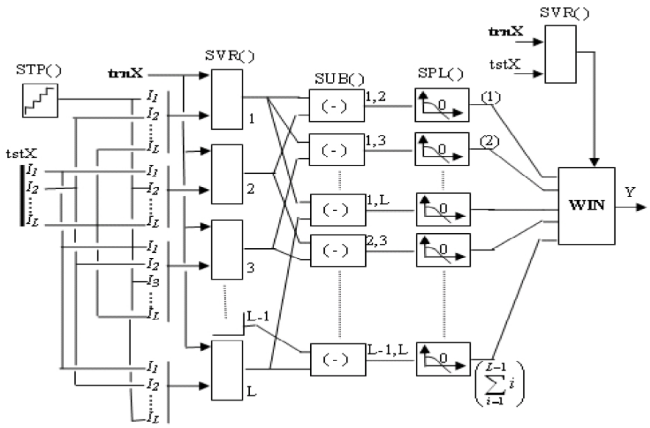

***Figure 2**: OCB-SVR structure model.*

In Figure 2, SVR is 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; and 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 $tstX_j$ have $\{I_1, I_1,..., I_L\}$ format. The evaluations of each input component are performed for curve fitting functions in SPL. The expected values of each compared dimension will be close to 0. The SVR function is defined as ε - SVR in (3).

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

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

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

The result of stepping 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 evaluated output value of $tstX_j$.

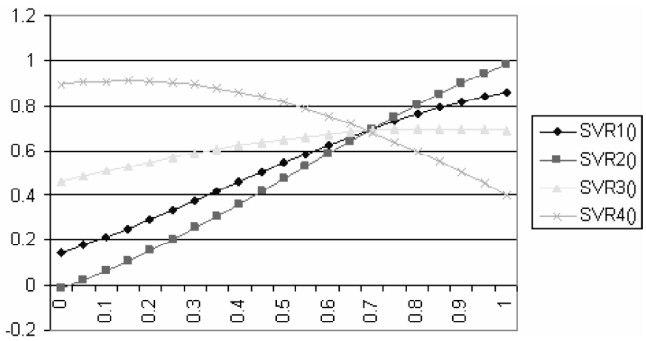

***Figure 3**: Example of completed intersection graph of stepping value of SVR function*

To compare the results of each input vector dimension, each output pair is compared using SUB function in (5). Figure 4 shows example result of SUB function.

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

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 $\sum_{i=1}^{L-1} i$

Figure 4 illustrates a case example of SUB functions. The case example in Figure 4 uses input vector dimension L = 4. This number resulting of compared value is 6, more compared dimensions.

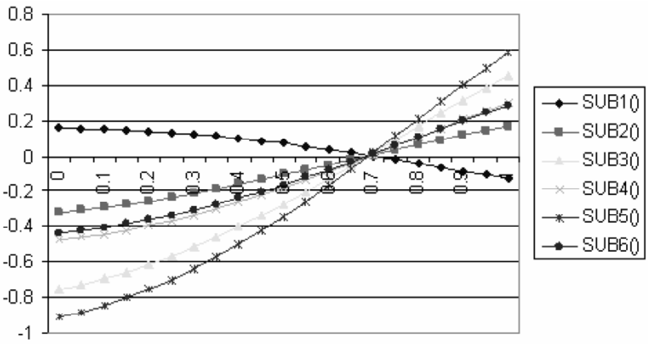

***Figure 4**: Example of SUB function graph lines.*

However, in practice, all of graph lines are not a properly intersection. This effect comes from optimizing results that $tstX_j$ is incompletely represented by $trnX$. In general case, incomplete matching between T- and Hspace is resulted from the uncontrolled effect of instant environment variables.

If high accuracy is needed, an increase of precision of stepping in (4) is required. However, this leads to a large consumption of the processing time. To reduce this effect we use cubic spline interpolation technique by defining knot point (x, y) with x = SUB, y = input step values and evaluate the result at point $x = 0$. The example interpolation points near $x = 0$ of each line are shown in Figure 5. All outputs of SPL function are candidate values. The final output is processed and selected by WIN function shown in Figure 6.

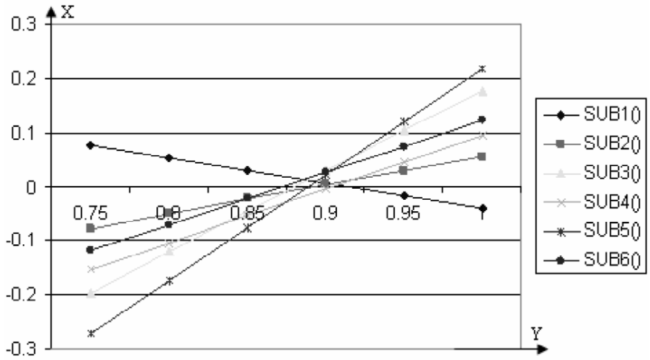

***Figure 5**: Example of uncompleted result of SUB function graph lines.*

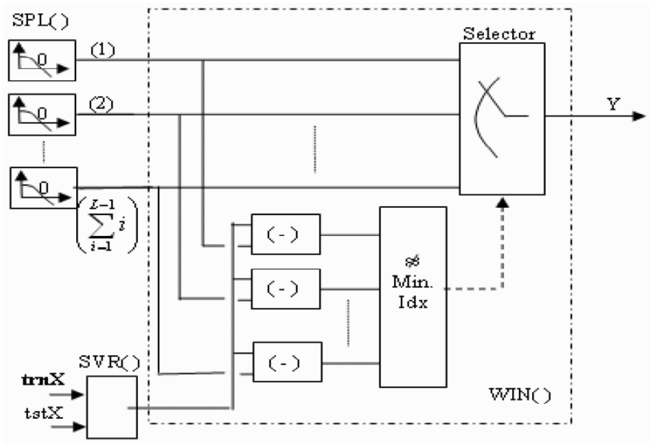

***Figure 6**: Structure of WIN function.*

During prediction processes in T-space, the actual output value of the system is not known. So, all outputs from SPL function can represent OCB-SVR outputs. At this state, the suitable value may refer to standard SVR output for comparisons in WIN function. The closest output of SPL to the SVR reference is selected to represent the final output of OCB-SVR model. Figure 6 is equivalent to (6).

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

where *idx* is the index at the lowest comparison between SUB and SVR. The description of OCB-SVR algorithm in (6) assumes that all SVR models are already trained with a standard SVR for pre-defined β parameter.

3. OII-ANFIS MODEL

ANFIS [21] constructs a fuzzy inference system (FIS) whose membership function parameters are adjusted using either a backpropagation algorithm alone or a combination with a least square method. This allows system to learn from the data that they are modeling.

For the fuzzy inference process, we have been referring to TSK or Takagi-Sugeno-Kang method. Introduced in 1985 [22], it is similar to the Mamdani method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant. In this paper we use only linear mode.

The other advantage technique for automatic generate fuzzy rules and related initial parameters, we use Subtractive clustering method [23]. This results in that the model can learn very fast.

The Output-Input-Iteration (OII) model is applied the benefit of ANFIS by reduction the surface oscillation effect. 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 testing mode. This case may reduce output surging. The evaluated results give the prediction values in final of OII model.

The concepts of OII can be explained by (7) to (15). The input dimension with th i^{th} records can be represented by (7).

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

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

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

The processes inside OII function is composted of ANFIS or Takagi-Sukeno (TS) fuzzy inference system in testing mode. The input pattern is rearranged as equation (9).

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

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

The different value between the TS fuzzy output and the forced input $V^{(\mathrm{stp})}$ will be changed according to (10) to (12).

$$V_i^{(\mathrm{dif})} = V_i^{(\mathrm{out})} - V^{(\mathrm{stp})} \qquad (10)$$

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

$$V_{\mathrm{r+1}}^{(\mathrm{stp})} = V_{\mathrm{r}}^{(\mathrm{stp})} + V^{(\mathrm{inc})} \qquad (12)$$

where i represents the i^{th} iteration, $V^{(\mathrm{low})} \le V^{(\mathrm{stp})} \le V^{(\mathrm{hgh})}$ and $V^{(\mathrm{inc})} = \dfrac{V^{(\mathrm{hgh})} - V^{(\mathrm{low})}}{n}$.

Precision output can increase by iteration the processes at new smaller value of $V^{(\mathrm{inc})}$ that it is related directly to $V^{(\mathrm{low})}$ and $V^{(\mathrm{hgh})}$ in (13) and (14), respectively. With this method the output will be close to the root value when the iteration number is increased.

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

$$\mathrm{new}V^{(\mathrm{hgh})} = \left(\mathrm{OII} + V^{(\mathrm{inc})}\right) \qquad (14)$$

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

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

In case of the same sign, the solution can be determined by the average between these values. This case is affected from the H-space and T-space nonmatching and cannot give $V(\mathrm{dif})$ curve intersection with the root axis. So, the average value is most suitable for this case.

4. APPLIED MODELS

OCB and OII design model can be combined with ANFIS and SVR to create 5 complex models: M-ANFIS, M-OII, S-ANFIS, S-OII, O-ANFIS, and O-OII. The prefix M stands for "Mixed" input of both SVR and OCB to ANFIS or OII model. The prefix S is only combined SVR with input to ANFIS or OII. The O is also the same prefix meaning. The applied models are illustrated in Figure 7. Figure 7 represents all 5 candidate models. SV (Support Vector) is SVR or OCB or both. STP stands for a stepping function. SUB is subtraction and PD is closest pair-wise distance function.

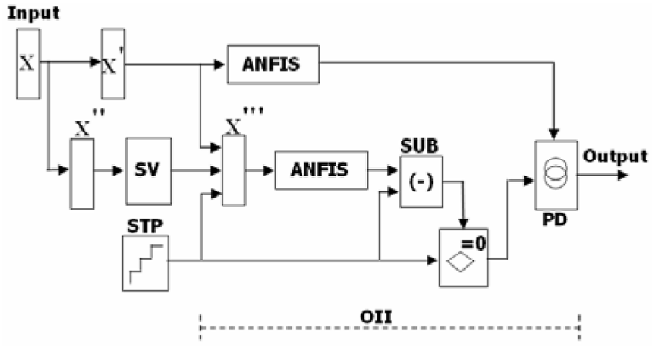

Figure 7: Applied model

The main focus is based on stock prediction and given input as (16) to (19).

$$ {}_{i} = \{C_i, H_i, L_i, O_i\} \tag{16} $$

$$ {}'_{i} = \frac{1}{1.3 \cdot C_i}\{C_i, C_{i-1}, \mathbf{L}, C_{i-k}, H_i, \mathbf{L}, H_{i-k}, L_i, \mathbf{L}, L_{i-k}, O_i, \mathbf{L}, O_{i-k}\} \tag{17} $$

$$ {}''_{i} = \frac{X_i}{\max} = \frac{1}{\max}\{C_i, H_i, L_i, O_i\} \tag{18} $$

$$ {}'''_{i,j} = \left\{ {}'_{i,j}, Y_i^{(IN)}, Y_j^{(STP)} \right\} \tag{19} $$

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

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

where $i = 1\mathrm{L}k$, j is ${}'_i$ dimension, k is the number of back records as present records (including the present record). $Y^{(IN)}$ 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 *Ci* in (17) is used as a maximum value of *ith* 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 7 is generally given multiple root values. To find these solutions, we select hybrid secant false position method [24]. This compromised method is generally guarantee 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 *ith* record is selected from the closest pair-wise distance between reference ANFIS output and candidate of $Y^{(STP)}$ at root state.

5. EXPERIMENTAL CASES AND DISCUSSION

To evaluate our applied models, we tested with real 5-stock data sets from the group of bank sectors in Thailand stock markets: BBL (Bangkok Bank), KTB (Krung Thai Bank), SCB (Siam Commercial Bank), TISCO (TISCO BANK), and TMB (TMB BANK).

Each data set is divided into two groups: training data and testing data, 50/50 each. Training data were collected from 04/1/2005 to 30/6/2005, and test data were collected from 04/7/2005 to 30/12/2005. Both data sets have the same format i.e., {close, high, low, open}.

The statistics measurement techniques using in this paper are mean squared error (MSE) [25], Theil's UStatistic statistic (U-Stat) [24], and Regression Error Characteristic curve (REC) [26], [27].

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 case of error is calculated using the absolute deviation (AD), the AOC is close to the mean absolute deviation (MAD).

To compare the deviation strength between SVR and OCB we test with all five data sets by max value changing in (18). The average strength of close pricing is shown in figure 8. SVR is slightly better when max value is in between 1.3 and 1.5 but OCB shows overall compromising strength resistance to outputs. Others give the same results when testing with high, low, and open as show in parts of Figures 10b to 10f.

In the second test, we used training data of BBL's high pricing from a group of records selected by most similarity of a pair-wise distance technique, Euclidean measurement. In case strength comparison between ANFIS and OII, we used (17) and (19) by forced $Y^{(IN)}$. The comparison graph is shown in Figure 9. The output value *Yout* to the actual value *Yr* is related with $Y^{(IN)}$ to *Yr*. Both ANFIS and OII have better result than linear curve. OII is better because its slope is lower.

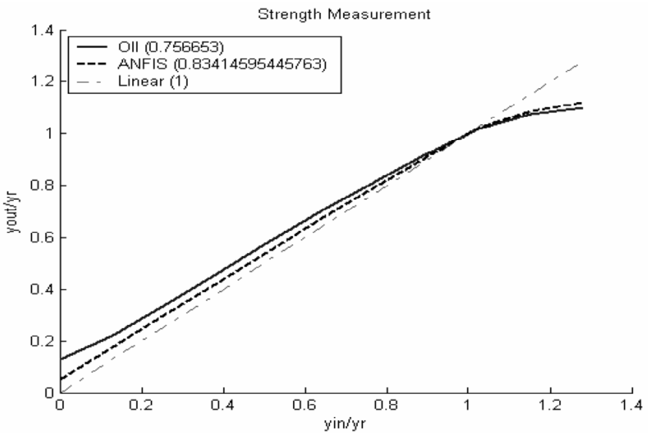

Figure 9*: Strength comparison between SVR and OCB of TMB closing price.*

The average tests of total 5 time series of stock prices are shown in Table 1. There are contrast error values of average high prices between ANFIS and OII. ANFIS gives mse and U-Stat as 0.96 and 2.06, respectively, but OII shows an inverse value when comparing the rest. This is only an effect from BBL's high price tested by surface oscillation effect at output results. However, the overall averages still generally significant that OII gives better performance.

Table 1*: Average error comparison of ANFIS and OII model with test data sets.*

		Mse	U-Stat
C	ANFIS OII	4.952301 1.8192114	2.68904 1.95962
H	ANFIS OII	0.9603752 1.2588804	2.06394 1.8123
L	ANFIS OII	1.9285486 0.82701	2.49488 1.75514
O	ANFIS OII	0.4477304 0.3649216	1.004276 1.047524

Figure 10 demonstrates all strength comparisons of 6 applied models and including 2 reference models. Within 4 dimensions on close, high, low, and open price testing, the results show significant that O-OII is the best compromising of strength outputs and the second is OANFIS. These are resulted from OCB as reference inputs. The worst case is SVR. M-types give average results.

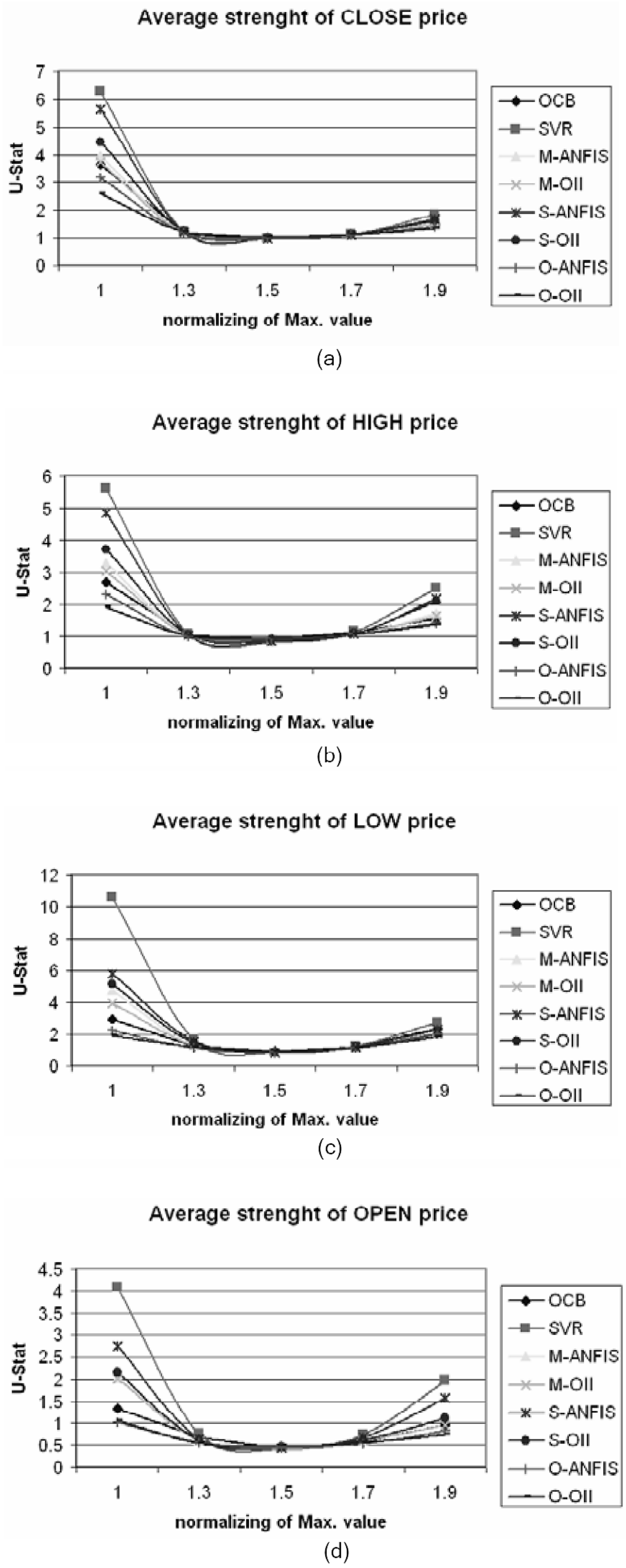

Figure 10*: Total strength comparison*

The other point of view is shown in Figure 11 by using REC graph. This TMB's low price test sample with max value setting to 1.0 shows that SVR is the worst case and gives AOC equal to 0.232. It is related to S-type; both S-ANFIS (0.176) and S-OII (0.145) have significant better curve. O-OII is good, which is the same as OANFIS and the origin OCB.

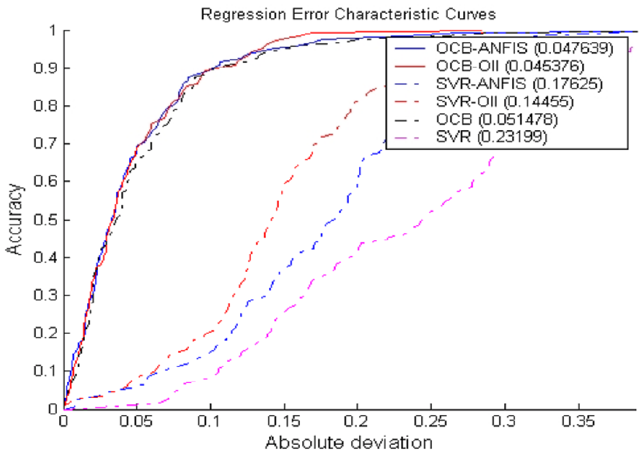

Figure 11: *Example of REC graph of TMB's LOW pricing testing (max. value =1)*

6. CONCLUSION AND FURTHER STUDY

Prediction of open price generally gives low prediction error because closing price values in the input part tend to show closer similarity to the next open price value. This effect is directly guideline to SV regression for the good performance.

Within train data limitation, OCB shows general compromised strength output and its characteristic is transferred to applied model both O-OII and O-ANFIS.

The suitable normalized maximum values for OCB and SVR usage are between 1.3 and 1.5 times of the maximum values of the training sets. This is guaranteed that the models give low prediction errors.

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

Our further research still focuses on stock prediction. The benefit form both OCB and OII are based on our developing techniques with extra procedure control such as jittering network, stock rules and their constraint, and adaptive learning control. These combined techniques shall provide better performances.

7. REFERENCES

[1] Panella, M., Rizzi, A. "Baseband Filter Banks for Neural Prediction," *CIMCA & AIWTIC '06*, Nov 2006.

[2] Pankratz, A. *Forecasting with Univariate Box-Jenkins Models: Concepts and Cases*, John Wiley & Sons. New York. 1983.

[3] Tran, N., Reed, D. A. "ARIMA Time Series Modeling and Forecasting for Adaptive I/O Prefetching," *The 15th International Conference on Supercomputing*, Jun 2001.

[4] Harrison, H. C., Qizhong, G. "An Intelligent Business Forecasting System," *The 1993 ACM Conference on Computer Science*, Mar 1993.

[5] Wu, S., Lu, R. "Combining Artificial Neural Networks and Statistics for Stock-Market Forecasting," *ACM Conference on Computer Science*, Mar 1993.

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

[7] Meesad, P., Srikhacha, T. "Universal Data Forecasting with an Adaptive Approach and Seasonal Technique," *CIMCA & AIWTIC '06*, Nov 2006, pp. 66.

[8] Clay, G. R., Grange, F. "Evaluating Forecasting Algorithms and Stocking Level Strategies using Discrete-Event Simulation," *The 1997 Winter Simulation Conference*. 1997.

[9] Dangelmaier, W., *et al.*, "Risk Averse Shortest Path Planning in Uncertain Domains," *CIMCA & AIWTIC '06*, Nov 2006, pp. v-xxii.

[10] Negnevitsky, M. "Artificial Intelligence: a Guide to Intelligent SystemV, *1st edit. Addison Wesley*. 2002. pp. 315-316.

[11] Rongjun, L., Zhibin, X. "A Modified Genetic Fuzzy Neural Network with Application to Financial Distress Analysis", *CIMCA*, Nov 2006, pp. 120.

[12] Liu, F., *et al.*, "A Neural Network Based Short Term Electric Load Forecasting in Ontario Canada", *CIMCA & AIWTIC '06*, Nov 2006, pp. 119.

[13] Wang, Y., Shen, Y. "Stock Predictor Algorithm: A Control Method Dealing With Distributed Control Systems," *CIMCA & AIWTIC '06*, Nov. 2006, pp. 49.

[14] Insung, J., *et al.*, "Two Phase Reverse Neural Network Based Human Vital Sign Prediction System", *CIMCA & AIWTIC '06*, Nov 2006, pp. 135.

[15] Papadimitriou S., Yu P. "Optimal Multi-Scale Patterns in Time Series Streams," *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, 2006.

[16] Huanmei W., *et al.* "Subsequence Matching on Structured Time Series Data," *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, 2005.

[17] Xiaopeng X., *et al.* "Fast time series classification using numerosity reduction," *Proceedings of the 23rd International Conference on Machine Learning ICML '06*, 2006.

[18] Huang, L., Zhong, J. "ICA-based Potential Significant Feature Extraction for Market Forecast," *CIMCA & AIWTIC '06*, Nov 2006.

[19] Mao, X., Yang, J., "Time Series Prediction Using Nonlinear Support Vector Regression Based on Classification," *CIMCA & AIWTIC '06*, Nov 2006, pp. 13.

[20] Meesad, P., Srikhacha, T. "Data Prediction by Support Vector Regression with a Decomposition Method," *ECTI '07*.

[21] Jang, J.-S. R., "ANFIS: Adaptive-Network-based Fuzzy Inference Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, pp. 665-685, May 1993.

[22] Sugeno, M., "Industrial applications of fuzzy control", *Elsevier Science Pub. Co.*, 1985.

[23] Chiu, S., "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent & Fuzzy Systems*, Vol. 2, No. 3, Sept. 1994.

[24] Pizer., M.S., Wallace.,L.V., "To Compute Numerically: Concepts and Strategies.", *Little Brown & Company Ltd*., p.81,107-108.

[25] Makridakis, S. et al. "Forecasting: Methods and Application", *Third Edition, John Wiley & Son, Inc*., p.43, 48-50.

[26] Bi, J., Bennett, K.P.: Regression Error Characteristic Curves. *The 20th International Conference on Machine Learning (ICML)*, Washington, DC (2003) 43"50.

[27] Aloisio, C. P.,Gerson Z. "Applying REC Analysis to Ensembles of Particle Filters", *IJCNN 2007*.