MODEL SELECTION OF ENSEMBLE FORECASTING USING WEIGHTED SIMILARITY OF TIME SERIES

Agus Widodo and Indra Budi

Faculty of Computer Science, Universitas Indonesia, Kampus Baru UI Depok, Jawa Barat, 16424, Indonesia

E-mail: agus.widodo@ui.ac.id

Abstract

Several methods have been proposed to combine the forecasting results into single forecast namely the simple averaging, weighted average on validation performance, or non-parametric combination schemas. These methods use fixed combination of individual forecast to get the final forecast result. In this paper, quite different approach is employed to select the forecasting methods, in which every point to forecast is calculated by using the best methods used by similar training dataset. Thus, the selected methods may differ at each point to forecast. The similarity measures used to compare the time series for testing and validation are Euclidean and Dynamic Time Warping (DTW), where each point to compare is weighted according to its recentness. The dataset used in the experiment is the time series data designated for NN3 Competition and time series generated from the frequency of USPTO's patents and PubMed's scientific publications on the field of health, namely on Apnea, Arrhythmia, and Sleep Stages. The experimental result shows that the weighted combination of methods selected based on the similarity between training and testing data may perform better compared to either the unweighted combination of methods selected based on the similarity measure or the fixed combination of best individual forecast.

Keywords: *ensemble forecasting, model selection, time series, weighted similarity*

Abstrak

Beberapa metode telah diajukan untuk menggabungkan beberapa hasil *forecasting* dalam *single forecast* yang diberi nama *simple averaging*, pemberian rata-rata dengan bobot pada tahap validasi kinerja, atau skema kombinasi non-parametrik. Metode ini menggunakan kombinasi tetap pada *individual forecast* untuk mendapatkan hasil final dari *forecast*. Dalam *paper* ini, pendekatan berbeda digunakan untuk memilih metode *forecasting*, di mana setiap titik dihitung dengan menggunakan metode terbaik yang digunakan oleh dataset pelatihan sejenis. Dengan demikian, metode yang dipilih dapat berbeda di setiap titik perkiraan. *Similarity measure* yang digunakan untuk membandingkan deret waktu untuk pengujian dan validasi adalah *Euclidean* dan *Dynamic Time Warping* (DTW), di mana setiap titik yang dibandingkan diberi bobot sesuai dengan keterbaruannya. Dataset yang digunakan dalam percobaan ini adalah data *time series* yang didesain untuk *NN3 Competition* dan data *time series* yang di-*generate* dari paten-paten USPTO dan publikasi ilmiah PubMed di bidang kesehatan, yaitu pada Apnea, Aritmia, dan *Sleep Stages*. Hasil percobaan menunjukkan bahwa pemberian kombinasi bobot dari metode yang dipilih berdasarkan kesamaan antara data pelatihan dan data pengujian, dapat menyajikan hasil yang lebih baik dibanding salah satu kombinasi metode *unweighted* yang dipilih berdasarkan *similarity measure* atau kombinasi tetap dari *individual forecast* terbaik.

Kata Kunci: *perkiraan ansambel, kesamaan tertimbang, seleksi model, time series*

1. Introduction

Methods for predicting the future values based on past and current observations have been pursued by many researchers and elaborated in many literatures in recent years. Several methods proposed to improve the prediction's accuracy include data pre-processing, enhancing theprediction's methods, and combining those methods.

Meanwhile, several prediction methods have been studied and used in practice. The most common ones are linear methods based on autoregressive models of time series, as stated by

This paper is the extended version from paper titled "Model Selection For Time Series Forecasting Using Similarity Measure" that has been published in Proceeding of ICACSIS 2012.

Romera et al. [1] and Makridakis et al. [2]. More advanced approaches apply nonlinear models based mainly on artificial neural networks (NNs), support vector machine (SVM), and other machine learning methods as studied by Siwek et al. [3], Crone and Kourentzes [4], Huang et al. [5], and Zang et al. [6].

Another common prediction approach is to train many networks and then pick the one that guarantees the best prediction on out-of-sample (verification) data, as done by Siwek et al. [3]. A more general approach is to take into account some best prediction results, and then combine them into an ensemble system to get the final forecast result as suggested by Huang et al. [5] and Armstrong et al. [7]. Poncela et al. [8] combine several dimensional reduction methods for prediction and then use ordinary least squares for combination, while Siwek et al. [3] combine prediction results from neural networks using dimensional reduction techniques.

However, previous literatures calculate the weight of the predictors at once using all training data. In our previous study [9], every future point is predicted by the best predictors used by similar training dataset. In other words, every point may be predicted by different predictors.

In this study, researcher extend our previous work by considering the weight of each point in time series to compare such that the most recent point get more weight than the point at the past. In addition, more dataset from patent and online publication are included in the experiment besides the dataset from NN3 Competition.

Thus, this paper aims to explore the use of weighted similarity measure as a method for selecting predictors that would be used for forecast combination. Our hypothesis is that the best methods used in training and validation will be suitable for similar time series used in testing. Furthermore, researcher expect that the most recent point in the time series carries more important information than the distant point to predict the future point.

Several combination methods are described by Timmerman [10], such as by least squares estimators of the weights, relative performace weight, minimization of loss function, non-parametric combination, and pooling several best predictors. Time-varying method is also discussed where the combination weight may change over time.

Recently, Poncela et al. [8] combine several dimensional reduction methods, such as principal component analysis, factor analysis, partial least squares and sliced inverse regression, for prediction, using ordinary least squares. The dataset comes from the Survey of Professional Forecasters, which provides forecasts for the main US macroeconomic aggregates. The forecasting results show that partial least squares, principal component regression and factor analysis have similar performances, and better than the usual benchmark models. Mixed result is found for sliced inverse regression which shows an extreme behavior.

Meanwhile, Siwek et al. [3] combine prediction results from neural networks using dimensional reduction techniques, namely principal component analysis and blind source separation. In this paper, all of the predictors are used to form the final outcome. The ensemble of neural predictors is composed of three individual neural networks. The prediction data generated by each component of the ensemble are combined together to form one forecasted pattern of electricity power for 24 hours ahead. The best results have been obtained with the application of the blind source separation method by decomposing the data into streams of statistically independent components and reconstructing the noise-omitted time series.

Meanwhile time series similarity has been widely employed in several fields, namely the gene expression, medical sequences, image, among others. The most common method to find the time series similarity is computing their distances. These distances are usually measured by Euclidean distance. Vlancos [11] describes several variation of this distance computations exist to accommodate the similarity of some parts of the series, namely the Dynamic Time Warping, and Longest Common Subsequence.

Others used likelihood to find similarity, such as Hassan [12], who uses Hidden Markov Model to identify similar pattern including time series. It is suggested that the forecast value can be obtained by calculating the difference between the current and next value of the most similar training series, and add that differences to the current value of the series to forecast. However, in this paper, the similarity measure is not used to directly compute the next value, but to select the most suitable predictors to compute that value.

As stated in [13], a time series is sequence of observations in which each observation x_t is recorded a particular time t. A time series of length t can be represented as a sequence of $X=[x_1,x_2,\dots,x_t]$. Multi-step-ahead forecasting is the task of predicting a sequence of h future values, X_{t+1}^{t+h}, given its p past observations, X_{t-p+1}^{t}, where the notation X_{t-p}^{t} denotes a segment of the time series $[x_{t-p},x_{t-p+1},\dots,x_t]$.

Time series methods for forecasting are based on analysis of historical data assuming that past patterns in data can be used to forecast future data points [14]. Furthermore, the multi-step-ahead prediction task of time series can be achieved by either explicitly training a direct model to predict several steps ahead, or by doing repeated one-step ahead predictions up to the desired horizon. The former is often called as direct method, whereas the latter is often called as iterative method.

The iterative approach is used and the model is trained on a one-step-ahead basis in [15]. After training, the model is used to forecast one step ahead, such as one week ahead. Then the forecasted value is used as an input for the model to forecast the subsequent point. In the direct approach, a different network is used for each future point to be forecasted. In addition, a parallel approach is also discussed in [15]. It consists of one network with a number of outputs equal to the length of the horizon to be forecasted.

The network is trained in such a way that output number k produces the k-step-ahead forecast. However, it was reported that this approach did not perform well compared to the two previous methods. Thus, direct approach is used in this paper as our previous experiment [16] indicates that even though direct approach is slightly better than iterative but it takes a lot less time to compute.

Several reasons of combining the forecasts are summarized by Timmerman [10]. First argument is due to diversification. One model is often suited to one kind of data. Thus, the higher degree of overlap in the information set, the less useful a combination of forecasts is likely to be. In addition, individual forecasts may be very differently affected by structural breaks in time series. Another related reason is that individual forecasting models may be subject to misspecification bias of unknown form. Lastly, the argument for combination of forecasts is that the underlying forecasts may be based on different loss functions. A forecast model with a more symmetric loss function could find a combination of the two forecasts better than the individual ones.

The forecast combination problem generally seeks an aggregator that reduces the information in a potentially high-dimensional vector of forecasts to a lower dimensional summary measure. Poncela *et al.* [8] denotes that one point forecast combination is to produce a single combined 1-step-ahead forecast f_t at time t, with information up to time t, from the N initial forecasts; that is

$$f_t = w_t' y_{t+1|t} \qquad (1)$$

where w_1 is the weighting vector of the combined forecast, $y_{t+1|t}$ is N dimensional vector of forecasts at time t. A constant could also be added to the previous combining scheme to correct for a possible bias in the combined forecast. The main aim is to reduce the dimension of the problem from N forecasts to just a single one, f_t.

Various integration methods may be applied in practice. In this paper, we will compare methods based on the averaging, both simple and weighted on predictor's performance. In the Averaging Schema, the final forecast is defined as the average of the results produced by all different predictors. The simplest one is the ordinary mean of the partial results. The final prediction of vector x from M predictors is defined by:

$$x = \frac{1}{M} \sum_1^M x_i \qquad (2)$$

This process of averaging may reduce the final error of forecasting if all predictive networks are of comparable accuracy. Otherwise, weighted averaging shall be used.

The accuracy of weighted averaging method can be measured on the basis of particular predictor performance on the data from the past. The most reliable predictor should be considered with the highest weight, and the least accurate one with the least weight. The estimated prediction is calculated as

$$x = \sum_1^M w_i x_i \qquad (3)$$

where w_i is weight associated with each predictor. One way to determine the values of the weights (i=1, 2, …, M) is to solve the set of linear equations corresponding to the learning data, for eaxample, by using ordinary least squares. Another way is using relative performance of each predictor [10], where the weight is specified by:

$$w_i = \frac{1/MSE_i}{\sum_{i=1}^M 1/MSE_i} \qquad (4)$$

In this weighted average, the high performance predictor will be given larger weight and vice versa.

Franses [17] states that the prediction methods that need to be combined are those which contribute significantly to the increased accuracy of prediction. The selection of prediction models in the ensemble is usually done by calculating the performance of each model toward the hold-out sample.

In addition, Andrawis et al. [15] use 9 best models out of 140 models to combine. The combination method used in their study is simple average. Previously, Armstrong [7] states that only five or six best models are needed to get better prediction result. Our previous study [18] on the use of Neural Network for forecast combination also suggests that selecting few best models are crucial for improving the forecasting result.

To measure the distance between time series, the difference between each point of the series can be measured by Euclidean Distance. The Euclidean Distance between two time series $Q = \{q_1, q_2, \ldots, q_n\}$ and $S = \{s_1, s_2, \ldots, s_n\}$ is:

$$D(Q,S) = \sqrt{\sum_{i=1}^{n}(q_i - s_i)^2} \quad (5)$$

This methods is quite easy to compute, and take complexity of $O(n)$.

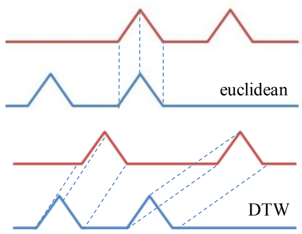

Figure 1. Two time series to compare.

Meanwhile, Dynamic Time Warping (DTW) [19] allows acceleration-deceleration of signals along the time dimension. For two series $X = x_1, x_2, \ldots, x_n$, and $Y = y_1, y_2, \ldots, y_n$, each sequence may be extended by repeating elements such that the Euclidean distance can calculated between the extended sequences X' and Y'. For example, for two time series in figure 1, it is exactly the same for DTW, whereas it is not for euclidean. It shall also be noted that the compared time series must be first centered and then normalized by its standard deviation to get uniform scale.

The mean squared error (MSE) of an estimator is one of many ways to quantify the difference between values implied by an estimator and the true values of the quantity being estimated. Let $X=\{x_1, x_2,..x_T\}$ be a random sample of points in the domain of f, and suppose that the value of $Y=\{y_1, y_2,..y_T\}$ is known for all x in X. Then, for all N samples, the error is computed as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 \quad (6)$$

An MSE of zero means that the estimator predicts observations with perfect accuracy, which is the ideal. Two or more statistical models may be compared using their MSEs as a measure of how well they explain a given set of observations.

2. Methodology

The steps to conduct this experiment are as follows: (1) read and scale the time series so that they have equivalent measurement (2) construct matrices of input and output for training as well as for testing, (3) run the prediction algorithms, which includes (a) machine learning methods, namely Neural Network, and Support Vector Regressions, (4) select the best models of the training data which is most similar with the testing data, (5) combine the forecasting results (6) record and compare the performance of the prediction. The steps of (1) comparing time series, (2) selecting best models (3) applying those methods, and (4) combining the forecasts are illustrated in figure 2.

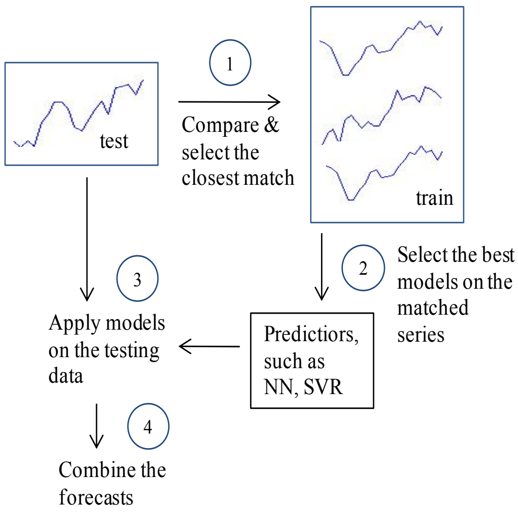

Figure 2. Steps to forecast using the combinations of selected models.

The assignment of linear combination of weight is given by multiplying the difference of each point by linearly or nonlinearly increasing weight. The difference itself is calculated either by Euclidean or DTW. The nonlinearly increasing weight can be calculated using polinomial function, such as square. Thus, the most recent point will get quite large weight while the distant point will get otherwise.

Neural network for regression. Neural Network is well researched regarding their properties and their ability in time series prediction [20]. Data are presented to the network as a sliding window [21] over the time series history, as shown in figure 1. The neural network will learn the data during the training to produce valid forecasts when new data are presented. Figure 3 shows the predicting future value using neural network.

The general function of NN, as stated in [21] is as follows:

$$f(X,w) = \beta_0 + \sum_{h=1}^{H} \beta_h g\left(\gamma_0 + \sum_{i=0}^{I} \gamma_{hi} x_i\right) \quad (7)$$

where $X = [x_0, x_1, ..., x_n]$ is the vector of the lagged observations of the time series and $w = (\beta, \gamma)$ are the weights. I and H are the number of input and hidden units in the network and $g(.)$ is a nonlinear transfer function [12]. Default setting from Matlab is used in this experiment, that is *'tansig'* for hidden layers, and *'purelin'* for output layer, since this functions are suitable for problems in regression that predict continuous values.

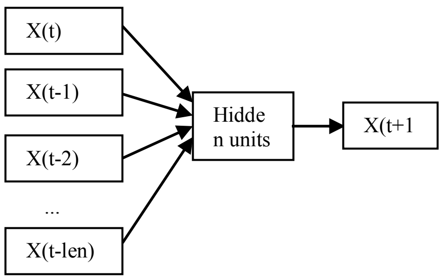

Figure 3. Predicting future value using neural network.

Support Vector Regression (SVR) is a Support vector machines (SVM) for regression which represents function as part of training data, often called as support vectors. Muller et al. [22] stated that SVM deliver very good performance for time series prediction. Given training data $\{(x_1, y_1), K, (x_1, y_1)\} \subset X \times R$, where X is the input pattern, SVM would seek function $f(x)$ that has maximum deviation ε from target value y_i. A linear function f can be written as:

$$f(x) = \langle w, x \rangle + b \text{ with } w \in X,\ b \in R \quad (8)$$

A flat function can be achieved by finding small w by minimizing norm, $\|w\|^2 = \langle w, w \rangle$. Technique which enable SVM to perform complex nonlinear approximation is by mapping the original input space into the higher dimensional space through a mapping Φ, at which each data training x_i is replaced by $\Phi(x_i)$. The explicit form of Φ does not need to be known, as it is enough to know inner product in the feature space, which is called the kernel function, $K(x,y) = \Phi(x)\cdot\Phi(y)$. Such function needs to obey Mercer's condition. Some kernel functions which if often used are Gausian Radial Basis Function, Polynomial or Linear [23].

The dataset used in this experiment is 7 quarterly time series of the output of motor vehicles taken from Time Series Forecasting Competition for Computational Intelligence (http://www.neural-forecastingcompetition.com/NN3). In addition, other dataset are generated from the frequency of USPTO's patents and PubMed's scientific publications on the field of health, namely on Apnea, Arrhythmia, and Sleep Stages. These frequencies are obtained by querying the USPTO and Pubmed online database from the year 1976 until 2010, which means 35 years. Thus, the total number of time series used is 13, each of which exhibits different pattern. Figure 4 and 5 shows the fluctuating pattern of those time series.

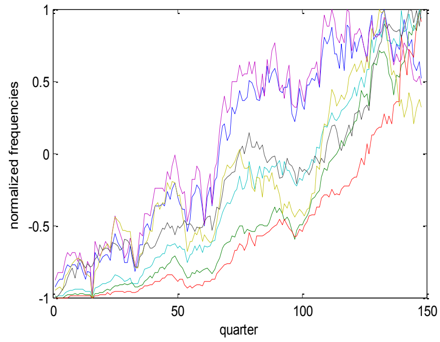

Figure 4. Seven dataset form NN3 Competition.

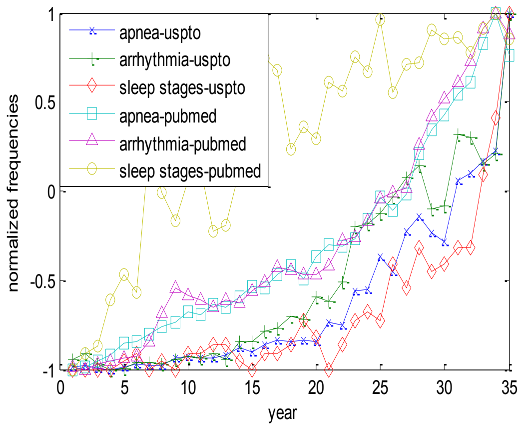

Figure 5. Six dataset form USPTO and Pubmed.

The task of this NN3competition is to predict the future values of the next 2 consecutive years or 8 consecutive quarters. The number of time series used in this experiment is 7 series, each one of them has a length of 148 quarters. Meanwhile, from the the 6 series we would like to predict the future values of 5 year ahead.

In this experiment, the 8 output for testing for NN3 data is the series from quarter 141 to 148, since the actual prediction output has not been provided yet. The testing output is the sliding window of series between quarters 9 to 140. The series for training output is from the quarter 133 to 140, whereas the one for training input is the sliding window of series between quarters 1 to 132. The input matrix of training is two dimensional matrix having the row size of the length of time series and the column size of the number of samples. Thus, having 8 values to predict, the vector y_{test} consists of 8 values, and the matrix x_{test} consists of $m\times 8$ series, where m is the sliding window. The value of m is determined while constructing the training dataset, namely the x_{train} and y_{train}, whose matrix's size are $m\times n$ and n. The shorter the value of m the larger the dataset (which is n) that can be constructed, and vice versa. The example of x_{train} as a sliding window is shown in figure 6. Similar matrix construction is done for time series from the USPTO and Pubmed.

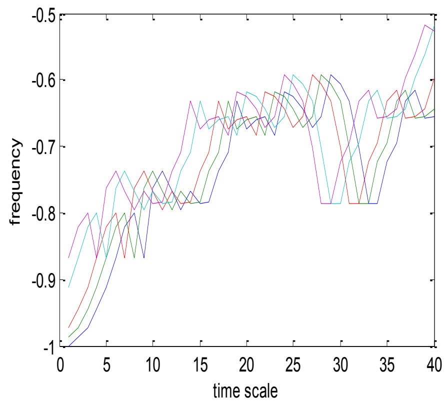

Figure 6. Example of sliding window of training dataset.

For performance evalution, MSE is mainly used for out-of-sample predictions, namely on the testing and validation dataset. MSE is also employed to evaluate the forecasting combination results using the simple average, median, and weighted average on individual performance, and ranking based on the individual performance.

Median is sometimes preferable than average as it is not easily affected by outliers. Similarly, ranking based on the individual performance is a better choice if all hypothesis from individual predictor need to be considered as the weighted average on individual performance based on the inverse MSE (mean squared error) might give very large weight on some predictors and very small or even zero to the others. This ranking method is similar to Borda count [24], at which each voter (predictor) rank orders the candidates (selected predictors). If there are N candidates, the first-place candidate receives N−1 votes, the second-place candidate receives N−2, with the candidate in i th place receiving N−i votes. The candidate ranked last receives 0 votes.

This experiment is conducted on computer with Pentium processor Core i3 and memory of 4GB. The main software used is Matlab version 2008b. The Matlab's command used to perform the NN is *newff*'. To normalise data into the range of -1 to 1, the command used is *mapminmax*.

The toolbox for Support Vector Regression is provided by Gun [25], whereas toolbox for Hidden Markov Model comes from Ghahramani [26]. The Bayesian toolbox is provided by Drugowitsch [13], and the statistical toolbox, namely Holt and Winter's method, is available from Kourentzes [4]. Meanwhile, the DTW toolbox for time series similarity measure is available from Felty [27].

3. Results and Analysis

The first experiment in this study is to compare the performance of each predictor. There are 2 predictors used, namely (1) Neural Network having its hidden node set to 1, 2, 4, 6, 10, 15 and 20, and (2) Support Vector Regression (SVR) using kernel radial basis function (RBF) of sigma's width of 1, 2, 3, 5, 10 and 15, kernel polynomial of degree 2, and kernel linear. Hence, there are totally 15 models by differentiating the parameters of those predictors.

Smaller sigma value in SVR implies smaller variance which fits the data tighter. Smaller sigma value implies smaller variance, hence fits the data tighter. The SME on training is smaller than that of bigger sigma, but SME on testing tends to be bigger as the model tends to overfit.

TABLE I
FORECASTING PERFORMANCE USING MSE AMONG TIME SERIES

No	Predictors	Average MSE of 7 NN3 series	3 USPTO series	3 Pubmed series
1	NN (HN 1)	0.1879	0.8921	0.6887
2	NN (HN 2)	0.1897	0.5735	0.5800
3	NN (HN 4)	0.1595	0.7281	0.5561
4	NN (HN 6)	0.1329	0.7295	0.6181
5	NN (HN 10)	0.4598	0.9026	0.4846
6	NN (HN 15)	0.5893	1.0883	1.1368
7	NN (HN 20)	0.2140	0.8452	0.4308
8	SVR (RBF 1)	0.5112	0.3008	0.3552
9	SVR (RBF 2)	0.1590	0.5622	0.0886
10	SVR (RBF 3)	0.0600	0.8402	0.0465
11	SVR (RBF 5)	0.1556	1.0992	0.0319
12	SVR (RBF 10)	0.1881	1.2447	0.0333
13	SVR (RBF 15)	0.1173	1.2746	0.0366
14	SVR (Poly 2)	0.1998	0.3643	0.1990
15	SVR (Linear)	3.8483	1.2992	0.0407
	Average	0.4782	0.8496	0.3551

Similarly, using polynomial as kernel of higher degree tends to overfit, hence yield poor generalisation error. Kernel polynomial of degree

2 is chosen as degree 1 means linear regression and degree higher than 3 tends to overfit.

Table I indicates that time series from USPTO is the most difficult to predict, as, on average, they have the highest Mean Squared Error (MSE), followed by those from NN3 and Pubmed. On those series, SVR using kernel RBF of moderate sigma's width, namely 3 and 5, gives the best result for NN3 and Pubmed series. Similarly, SVR using kernel polinomial of low degree also yields the best result for the other series.

As illustrated in figure 7, based on MSE measure on 7 NN3 time series, the best models are SVR of RBF kernel having sigma=3 and 15 and NN of hidden nodes=6. This figure also indicates that the tightly fitted curve will not yield good performance, such as SVR having sigma=1, or NN having large hidden nodes. As expected, SVR linear also yiled unsatisfactory result as it approximate the fluatuation by linear line.

The second experiment in this study is to select the predictors that perform best on training time series similar to testing time series to be predicted. The similarity between those series is calculated using Eucledian Distance and DTW. The performance of all possible number of best models is shown in table II for Euclidean similarity, DTW similarity and without similarity, respectively. By selecting the best models without similarity, the best models are determined by all training samples at once. For example, suppose that only 3 best model s are selected. In this experiment, since the first 3 best predictors are SVR of RBF having 3 and 15, and NN having hidden nodes of 3, these three predictors will be used to predict all 8 future points.

By contrast, using similarity measure, the best models are determined by the training sample that is similar to the testing data. If the best models 3, then those 3 models are not always SVR of RBF having 3 and 15, and NN having hidden nodes of 3. Instead, they would be the best model used by the training data that is similar to the particular testing data.

Table II shows the ensemble methods clearly outperform the individual pedictor. For example, the average MSE of model selection without similarity measure for NN3 dataset is 0.148 whereas that of individual predictor is 0.478. Similar observation can be noted for USPTO & Pubmed dataset. Furthermore, similarity between training and testing dataset to select predictors also improves the prediction accuracy, which is in line with our previous finding [9]. This paper tries to investigate whether giving weight to the compared time series would improve the prediction accuracy. Table II indicates that the average MSE on Weighted Eucledian is lower than the plain Eucledian on both NN3 and USPTO & Pubmed dataset. 'No-sim' means selecting best models without similarity measure, 'Euclid' means using Eucledian distance to compare two time series, and 'Weightd Euclid' means linearly weighted Euclidean distance measure.

Table III further elaborate the use of distance and aggregation measure. Besides Eucledian, DTW is also used to compare the time series. In this experiment, DTW slightly outperform the Euclidean. Table III shows that using Average, Median, Inverse MSE or Rank as combination method, the use of similarity measure always improve the accuracy except for Euclidean combined by Inverse MSE. This table also confirms that the use of weight both for Euclidean and DTW always improve the accuracy. This accuracy still can be improved, although not significantly, by using nonlinear weight. Squared weighted Eucledian and DTW also yield lower MSE than the linear ones.

TABLE II
MSE ON COMBINATION OF METHODS USING EUCLIDEAN DISTANCE

	NN3			USPTO & Pubmed		
Number of best models	No-sim	Euclid	Weightd euclid	No-sim	Euclid	Weightd euclid
1	0.589	0.179	0.159	0.282	0.475	0.352
2	0.217	0.111	0.082	0.262	0.452	0.325
3	0.179	0.096	0.073	0.263	0.330	0.298
4	0.153	0.088	0.061	0.313	0.307	0.284
5	0.130	0.078	0.060	0.357	0.277	0.283
6	0.111	0.075	0.053	0.385	0.270	0.293
7	0.104	0.068	0.049	0.392	0.249	0.303
8	0.104	0.062	0.049	0.379	0.231	0.290
9	0.101	0.066	0.056	0.324	0.227	0.295
10	0.118	0.075	0.066	0.313	0.227	0.277
11	0.115	0.072	0.066	0.295	0.240	0.259
12	0.107	0.072	0.068	0.288	0.248	0.259
13	0.067	0.073	0.073	0.300	0.256	0.260
14	0.066	0.084	0.085	0.301	0.278	0.269
15	0.063	0.063	0.063	0.302	0.302	0.283
Avg	0.148	0.084	0.071	0.317	0.291	0.289

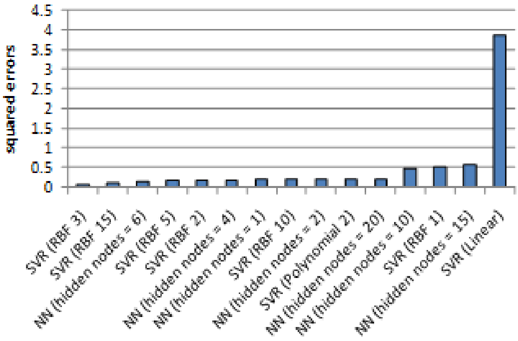

Figure 7. Performance of individual predictors for NN3 time series.

Figure 8 also shows that using combination of methods selected based on the similarity between training and testing data may lead into better prediction result compared to the combination of all methods. Table II presents the detail of performance of the combination of those methods, which actually perform fairly well compared to the individual forecast.

Even though it is not in a stark contrast, the combination of selected methods using similarity measure performs better than the best methods without similarity measure as the average MSE of combination without similarity is higher than the that using similarity measure. Likewise, the use of weighted similarity measure offer opportunity to increase the accuracy of the prediction.

TABLE III
AVERAGE MSE ON COMBINATION OF BEST MODELS USING DIFFERENT AGGREGATION MEASURE

	Avg	Median	Inv MSE	Rank
No sim	0.1483	0.1458	0.1093	0.1134
Euclidean	0.0842	0.0893	0.1235	0.0935
DTW	0.0613	0.0672	0.0653	0.0602
w-Euclidean	0.0708	0.0767	0.0864	0.0769
w-DTW	0.0609	0.0667	0.0654	0.0599
w2-Euclidean	0.0737	0.0777	0.0855	0.0784
w2-DTW	0.0596	0.0652	0.0649	0.0585

'w-Euclidean' means linearly weighted Euclidean distance measure, 'w2-Euclidean' means squared weighted Euclidean distance measure.

The chart in figure 8 is decreasing and level off when the number of predictors combined reaches 7 out of 15. Thus, the optimum number of models to combine turns out to be about less than 50% of all models.

Lastly, the most often used models as the best models are depicted in figure 9. To sort the predictors, each predictor is weighted based on its rank. Since there is 15 models used, the weight assignd is 1 until 15 for the least to the best model. For instance, if a predictor is twice selected as best model, 3 times 4th best, then its score would be 2x15+3x12, and so on. It turns out that SVR with kernel RBF having sigma of 3, 2 and 15 is the most often selected as best model.

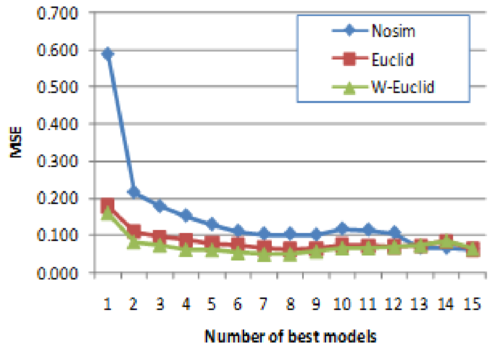

Figure 8. Average performance of forecast combination using models selected by euclidean and weighted eucledian similarity compared to the one using best models without similarity measure.

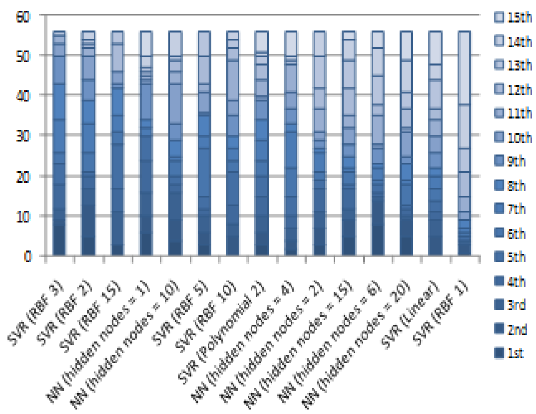

Figure 9. The most often selected models.

4. Conclusion

The experimental result shows that the weighted combination of methods selected based on the similatity between training and testing data may perform better than unweighted ones. Nevertheless, this unweighted combination of methods may perform better than the fix combination of best models without similarity measure. In addition, those combinations of selected models are certainly better than the average of individual predictors.

Our other observation shows that the optimum number of models to combine is about less than fifty percent of the number of models. Smaller number of models to combine may not provide enough diversification of method's capabilities whereas greater number of models may select poor performing models.

In this paper, the proposed method is also performed to other dataset to enhance its generality. However, for future works, this method shall be tested against many other time series data to confirms its feasibility. There are also many possibilities of employing different predictors other than NN and SVR. There are similarity methods other than the Euclidean and DTW that may suit better for comparing testing and training of time series dataset. In addition, other methods to assign weight can be explored further.

References

[1] E. Gonzalez-Romera, M.A. Jaramillo-Moran, & D. Carmona-Fernandez, "Monthly electric energy demand forecasting based on trend extraction," *IEEE Transations on Power Systems*, vol. 21, pp. 1946–1953, 2006.

[2] S. Makridakis, S.C. Wheelwright, & V.E. McGee, *Forecasting: Methods and Applications*, 2nd ed., John Wiley & Sons, United States of America, 1983.

[3] K. Siwek, S. Osowski, & R. Szupiluk, "Ensemble Neural Network Approach For Accurate Load Forecasting In A Power System," *Int. J. Appl. Math. Comput. Sci.*, vol. 19, pp. 303–315, 2009.

[4] S.F. Crone & N. Kourentzes, *Forecasting Seasonal Time Series with Multilayer Perceptrons – an Empirical Evaluation of Input Vector Specifications for Deterministic Seasonality*, Lancaster University Management School, UK, 2007.

[5] C. Huang, D. Yang, & Y. Chuang, "Application of wrapper approach and composite classifier to the stock trend prediction," *Elsevier, Expert Systems with Applications,* vol. 34, pp. 2870–2878, 2008.

[6] G.Q. Zang, B.E. Patuwo, & M.Y. Hu, "Forecasting with artificial neural network: The state of the art," *International Journal of Forecasting*, vol. 14, pp. 35-62, 1998.

[7] J.S. Armstrong, *Principles of Forecasting: A Handbook for Resarchers and Practitioners,* Kluwer Academic Publishers, Norwel, 2001.

[8] P. Poncela, J. Rodrıgueza, R. Sanchez-Mangasa, & E. Senra, "Forecast combination through dimension reduction techniques," *International Journal of Forecasting*, vol. 27, pp. 224–237, 2011.

[9] A. Widodo & I. Budi, "Model Selection For Time Series Forecasting Using Similarity Measure," *In Proceeding of ICACSIS (Advanced Computer Science and Information System) 2011,* pp. 221-226, 2011.

[10] A. Timmerman, Forecast Combinations, UCSD, http://www.banxico.org.mx/publicaciones-y-discursos/publicaciones/documentos-de-investigacion/banxico/%7B687AB152-CD27-993D-1FB8-C05468E33C30%7D.pdf, 2005, retrieved November 12, 2011.

[11] M. Vlachos, *A practical Time-Series Tutorial with MATLAB*, ECML PKDD, Portugal, 2005.

[12] M.R. Hassan, "Hybrid HMM and Soft Computing modeling with applications to time series analysis," Ph.D Thesis, Department of Computer Science and Software Engineering, The University of Melbourne, Australia, 2007.

[13] J. Drugowitsch, Bayesian Linear Regression, Laboratorie de Nuerosciences Cognitives, http://www.lnc.ens.fr/~jdrugowi/code/bayes_logit_notes_0.1.2.pdf, 2010, retrieved November 13, 2011.

[14] R. Ihaka, Time Series Analysis, Lecture Notes for 475.726, Statistics Department, University of Auckland, New Zealand, 2005.

[15] R.R. Andrawis, A.F. Atiya, & H. El-Shishiny, "Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition," *International Journal of Forecasting*, vol 27, pp. 672-688, 2011.

[16] A. Widodo & I. Budi, Multi-step-ahead Forecasting in Time Series using Cross-Validation, In International Conference On Information Technology and Electrical Engineering, Gajah Mada University, Yogyakarta, 2011.

[17] P.H. Franses, Model selection for forecast combination, Econometric Institute Report, Erasmus University Rotterdam, Rotterdam, 2008.

[18] A. Widodo & I. Budi, "Combination of Time Series Forecasts using Neural Network", *In Electrical Engineering and Informatics (ICEEI)*, pp. 1-6, 2011.

[19] D. Berndt & J. Clifford, "Using dynamic time warping to find patterns in time series" *AAAI-94 Workshop on Knowledge Discovery in Databases (KDD-94)*, pp. 359-370, 1994.

[20] K. Honik, "Approximation capabilities of multilayer feedforward network," *Neural Networks*, vol. 4, pp. 251-257, 1991.

[21] N. Mirarmandehi, M.M. Saboorian, & A. Ghodrati, "Time Series Prediction usi*ng Neural Networ*k", 2004.

[22] K.R. Muller, A.J. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, & V. Vapnik "Predicting Time Series with Support Vector

Machines" *In Proceedings of the 7th International Conference on Artificial Neural Networks*, pp. 999-1004, 1997.

[23] N. Cristianini, Support Vector and Kernel Machines, ICML, USA, 2001.

[24] R. Polikar, "Ensemble Based System in Decision Making," *IEEE Circuits And Systems Magazine*, vol. 6, pp. 21-45, 2006.

[25] S.R. Gunn, Support Vector Machines for Classification and Regression, Technical Report, University Of Southampton, Southampton, 1998.

[26] Z. Ghahramani, Machine Learning Toolbox, Version 1.0, University of Toronto, http://learning.cs.toronto.edu/index.shtml?section=home, 1996, retrieved November 14, 2011.

[27] T. Felty, Dynamic Time Warping, Matlab Central, http://www.mathworks.com/matlabcentral/fileexchange/6516-dynamic-time-warping, 2005, retrieved December 20, 2011.