Multi-Model Integration for Long-Term Time Series Prediction

Zifang Huang, Mei-Ling Shyu, James M. Tien
Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL, USA
z.huang3@umiami.edu, shyu@miami.edu, jmtien@miami.edu

Abstract

Long-term (multi-step-ahead) time series prediction is a much more challenging task comparing to the short-term (one-step-ahead) time series prediction. This is due to the increasing uncertainty and the lack of knowledge about the future trend. In this paper, we propose a multi-model integration strategy to 1) generate predicted values using multiple predictive models; and then 2) integrate the predicted values to generate a final predicted value as the output. In the first step, a k-nearest-neighbor (k-NN) based least squares support vector machine (LS-SVM) approach is used for long-term time series prediction. An autoregressive model is then employed in the second step to combine the predicted values from the multiple k-NN based LS-SVM models. The proposed multi-model integration strategy is evaluated using six datasets, and the experimental results demonstrate that the proposed strategy consistently outperforms some existing predictors.

Keywords: long-term time series prediction, multi-model integration, k-nearest-neighbor, least squares support vector machine (LS-SVM), autoregressive model

1. Introduction

Time series prediction has been widely studied because of its general applications in transportation prediction [4], power prediction [12][16], health care study [6], etc. The essence of time series prediction is to predict future values of a time series based on the patterns or knowledge learnt from the past sequential values of the time series. Time series prediction tasks can be broadly divided into two groups: short-term and long-term time series prediction. Short-term time series prediction is to predict the next value one step ahead, while long-term time series prediction is to predict future values multi-step ahead. Comparing to short-term time series prediction, long-term time series prediction is much more difficult. Especially, when the prediction horizon increases, the uncertainty of the future trend also increases, and it becomes harder to model and capture the inherent relationships of a time series.

For short-term time series prediction, there have been plenty of classical time series prediction approaches, such as exponential smoothing [11], linear regression [13], autoregressive model (AR) [21], autoregressive integrated moving average (ARIMA) [24], support vector machines (SVM) [20], artificial neural networks (ANN) [10][24], Kalman filter [3], and fuzzy logic [10]. In order to utilize these short-term time series prediction models for long-term time series prediction, there are two approaches: recursive approach and direct approach [5][18]. The recursive approach trains one prediction model by optimizing the prediction performance at the next time step, and then iterates the same model using the previously predicted values as a part of the input to generate the prediction for a higher horizon. This approach suffers from the error propagation problem. On the other hand, the direct approach trains one prediction model for each prediction horizon by optimizing the prediction performance at each prediction horizon. This approach needs to train multiple models, so it takes a longer time in the training stage; while it avoids the error accumulation problem. The direct approach usually outperforms the recursive approach on the prediction accuracy aspect. A multi-input multi-output local learning (LL-MIMO) approach [1][23], which is used as one of the comparison approaches in the experiment, predicts the future values as a whole simultaneously. However, it could still be decomposed into multiple independent models, and thus it can be considered as a direct approach.

Due to the difficulties that arise in long-term time series prediction, not every model that works for short-term time series prediction would work well in long-term prediction. For example, the autoregressive model is a linear prediction method that attempts to predict the next value based on the previous observations [21], which is typically applied to autocorrelated time series data. Because of its linear nature, it is not able to achieve a good prediction precision if the time series contain non-linear components. In the case of long-term time series prediction, the mapping function is usually non-linear. Therefore, the autoregressive model is not preferable. In addition, the Kalman filter [3] is an optimal recursive filter for linear functions subjected to Gaussian noise. In order to build a suitable Kalman filter, the mechanism which generates the time series should be known, or at least enough information of the mechanism should be available to model the dynamics of the target. This limitation narrows the application of the Kalman filter method. Meanwhile, the recursive approach has to be applied to conduct long-term time series prediction in the case of the Kalman filter. Error propagation makes the Kalman filter unreliable, especially when the prediction horizon is high. The least squares support vector machine (LS-SVM), a fundamental approach for classification and function estimation, has been successfully applied to time series prediction [7][22], since the kernel trick extends the LS-SVM theory to a nonlinear technique without an explicit construction of the nonlinear mapping function. The most frequently used kernels in LS-SVM are linear kernel, polynomial kernel, radial basis function (RBF) kernel, and multilayer perceptron (MLP) kernel.

In this paper, a multi-model integration framework utilizing LS-SVM for long-term time series prediction is proposed. In our proposed framework, a k-nearest-neighbor (k-NN) based LS-SVM approach is first employed to conduct long-term time series prediction, where the k-NN method is utilized to obtain a smaller training dataset to train an LS-SVM regressor for each given testing instance. The distance function used in the k-NN method integrates the Euclidean distance and the dissimilarity of the trend of a time series. The direct approach for long-term time series prediction is adopted to avoid the error propagation problem. Next, an autoregressive model is employed to combine the predicted values from multiple k-NN based LS-SVM models, which makes the prediction more robust and reliable. Several experiments are conducted to evaluate the proposed multi-model integration strategy, and the experimental results demonstrate that the proposed strategy performs better than those predictors used in the performance comparisons.

The rest of the paper is organized as follows. Section 2 illustrates the proposed multi-model integration framework for long-term time series prediction. The experiments and discussions of the experimental results are presented in Section 3. The paper is concluded in Section 4.

2. Proposed Framework

Given a time series d for training with $Tlength$ data points, represented as (x_1 ,..., $x_{Tlength-1}$, $x_{Tlength}$), long-term time series prediction is to predict the future n values based on the most recent p observations, where $n > 1$ and $p \geq 1$. According to the direct prediction strategy, one prediction model f_i is obtained for each prediction horizon i, respectively as shown in Equation (1).

$$\overline{x}_{t+i} = f_i(x_{t-p+1}, \ldots, x_{t-1}, x_t), \qquad 1 \leq i \leq n. \quad (1)$$

As mentioned earlier, in order to avoid error propagation and to achieve higher prediction precision values, the direct prediction strategy for long-term time series prediction is adopted in the proposed framework. The prediction problem becomes how to construct the mapping function f_i. Figure 1 shows the proposed multi-model integration framework for long-term time series prediction. As can be seen from this figure, it constructs the prediction function using a k-NN based LS-SVM method, and then integrates the results using an autoregressive model to generate the output. The details of each component are described in the following subsections.

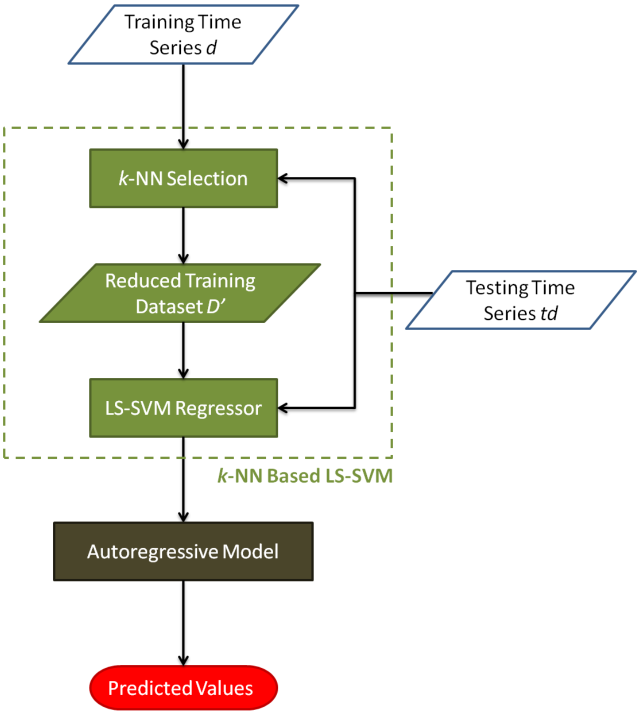

Figure 1. System architecture of the proposed multi-model integration framework

2.1. k-NN Based LS-SVM Component

As shown in Figure 1, the k-NN based LS-SVM component utilizes the k-NN method to dynamically select k instances from the training time series d to form a reduced training dataset D' for each testing instance. D' is then employed to train the LS-SVM regressors. This is based on the observation that analyzing the instances, which have similar inputs, could render a better prediction model in terms of better capturing the relationship between the inputs and the corresponding outputs. Therefore, instead of the whole training dataset, the training instances which are closer to the testing instance are used for training. The training time series d is formed by the most recent $Tlength$ data points, which enables the prediction model to capture the latest pattern. Even though the instance selection process takes some time, it reduces the size of the training data for LS-SVM. Parameter k is usually much smaller than the number of instances in the training dataset, and thus instance selection significantly decreases the complexity of training the LS-SVM regressor. Therefore, the k-NN based LS-SVM method is able to obtain more reliable prediction results in a more efficient manner.

For the k-NN method, the selection of the distance metric is crucial, and it directly influences which instances are selected as neighbors. We define a distance metric specifically for evaluating the similarity of time series segments, which incorporates both the Euclidean distance and the similarity of the trend of a time series. The trend of a time series is described by a vector, which is the first derivative of the time series. The detailed definition of this distance metric and an example of the instance selection can be found in [8].

The length of the input vector in each training/testing instance is p. The most recent p_1 values are used as the input of the k-NN method, where $p_1 \leq p$. Similarly, the most recent p_2 values are used as the input of the following LS-SVM regressor component, where $p_2 \leq p$ and $p = MAX(p_1, p_2)$. For an n-step ahead prediction problem, the k-NN based LS-SVM component is set to generate a prediction over a horizon n', which is larger than the required prediction horizon n, *i.e.*, $n' > n$. Accordingly, n' regressors are trained in order to predict n'-step ahead. In this case, the value of the time series at time $t + n$ would have been predicted $n' - n + 1$ times at time t. It is considered that these multiple predicted values may provide additional information to generate a more robust final predicted value. Hence, an autoregressive model is then employed to integrate these multiple predicted values to generate the n-step ahead prediction result for that testing instance. The details of the autoregressive model are illustrated in the following subsection.

2.2. Autoregressive Model for Integration

There are some existing studies on combining prediction results from individual models. Due to the fact that there is no single prediction model that is able to outperform all other methods on any time series, a combination of multiple models becomes the solution to make the prediction model more general, which is able to predict well for a group of time series. There are different opinions on which types of the models should be combined, different methods with distinct nature or very similar models. Some studies claim that these are not many values added in combining the models which are not significantly different, because the models access the same information set and capture similar patterns [2]. On the contrary, some other researchers claim that it is also important to combine forecasts from very similar models [25]. We agree that even when very similar models are combined, the model uncertainty could be generally reduced. Some of our primary study has validated this claim [8].

In the proposed framework, we obtain the predicted values over a horizon n' from the previous k-NN based LS-SVM component. Each predicted value is corresponding to a unique LS-SVM regressor. n' LS-SVM regressors are trained using the same input vectors but generating different outputs, as described in Equation (1). Let the current time be T. The testing input vector is $< x_{T-p+1}$, ..., x_{T-1}, $x_T >$, and the length of the vector is p. k-NN based LS-SVM component generates n' predicted values, which can be denoted as $< \overline{x}_{T+1}$, ..., $\overline{x}_{T+n}$, ..., $\overline{x}_{T+n'} >$, where the prediction value at time $T + n$ is the expected output. The prediction for the value at time $T + n$ is also conducted in the previous $n' - n$ testing instances. Instead of simply returning $\overline{x}_{T+n}$ as the output for the prediction at horizon n, we integrate the previous h predictions of the time series value at time $T + n$, where $0 \leq h \leq n' - n$, together with the current prediction $\overline{x}_{T+n}$ by using an autoregressive model. The autoregressive model takes $h + 1$ predicted values as the input and generates the final output. In order to better illustrate how the approach works, an example is given in Table 1.

Table 1. An example of autoregressive model for integration

1	x_1	x_2	$\overline{x}_3$	$\overline{x}_4$	$\overline{x}_5$	$\overline{x}_6$
2	x_2	x_3	$\overline{x}'_4$	$\overline{x}'_5$	$\overline{x}'_6$	$\boldsymbol{\overline{x}'_7}$
3	x_3	x_4	$\overline{x}''_5$	$\overline{x}''_6$	$\boldsymbol{\overline{x}''_7}$	$\overline{x}''_8$
4	x_4	x_5	$\overline{x}'''_6$	$\boldsymbol{\overline{x}'''_7}$	$\overline{x}'''_8$	$\overline{x}'''_9$

In this example, the first column is the instance index. The values of the parameters are $p = 2$, $n = 2$, $n' = 4$, and $T = 5$, which means that the prediction takes the past 2 values ($p = 2$) as the input vector, predicts the next 4 values ($n' = 4$), while the goal is to predict the next 2 values ($n = 2$), and the current time is 5 ($T = 5$). The current testing instance is instance $T - p + 1 = 4$, and the input vector is (x_{T-1}, x_T) = (x_4, x_5). h can be an integer within the range of $[0, 2]$. Let the autoregressive model that is employed to combine the predicted values be f_a.

- If $h = 0$, it renders the prediction value at time $T + n = 7$, *i.e.*, $\overline{x}_7'''$ as the final prediction value at horizon $n = 2$.
- If $h = 1$, it integrates the previous one prediction of the time series value at time $T + n = 7$, *i.e.*, $\overline{x}_7''$, with the current prediction $\overline{x}_7'''$ as the final output, which is $f_a(\overline{x}_7'', \overline{x}_7''')$.
- If $h = 2$, it integrates the previous two predictions of the time series value at time $T + n = 7$, *i.e.*, $\overline{x}_7'$ and $\overline{x}_7''$, with the current prediction $\overline{x}_7'''$ as the final output, which is $f_a(\overline{x}_7', \overline{x}_7'', \overline{x}_7''')$.

The autoregressive model is a linear prediction formula that is commonly used to predict an output of a system based on the previous outputs. In the proposed framework, parameter h determines the order of the autoregressive model. For a given h, the corresponding order of the autoregressive model should be $h + 1$. An autoregressive model of order $h + 1$ is defined in Equation (2).

$$x_t = c + \sum_{i=1}^{h+1} \varphi_i x_{t-i} + \varepsilon, \qquad (2)$$

where $\varphi_1, \cdots, \varphi_{h+1}$ are the parameters of the model, c is a constant and ε is white noise. There are many ways to estimate the coefficients. In the experiments, we estimate the coefficients by minimizing the root mean square error in the training process. The training instances for the autoregressive model are formed by the predicted values (as the input) and the corresponding true value (as the output). Take the dataset given in Table 1 as an example, and let h be 2. Accordingly, the order of the autoregressive model is 3. For instance, the input of one training instance is ($\overline{x}_6'$, $\overline{x}_6''$, $\overline{x}_6'''$), and the output is the real value at time 6, which is x_6. In reality, there could be many more training instances with the progression of time. The time series is much longer than the one given in this simple example. A set of the training instances can be used to estimate the coefficients for the autoregressive model. Given an input of a testing instance, for example, ($\overline{x}_7'$, $\overline{x}_7''$, $\overline{x}_7'''$), the trained autoregressive model f_a returns a value as the final prediction for time 7.

The input values to this integration component are all predicted values from the k-NN based LS-SVM method, and thus a linear function is more suitable to combine these values than a non-linear function. We compared autoregressive model with some other methods, such as LS-SVM, nonlinear autoregressive moving average, and the autoregressive model is able to achieve the highest prediction precision.

3. Experiments and Results

We conduct experiments on various datasets to evaluate the performance of the proposed framework. The datasets used in the experiments are described in Section 3.1. Section 3.2 introduces the two measurements used to evaluate the performance. Comparison results with LL-MIMO [1], LS-SVM [19][22], AR [21] and autoregressive moving average (ARMA) [24] methods are presented in Section 3.3. The experiments are conducted on an Intel Core 2 machine with two 2.66 GHz CPUs and 3.25 GB of RAM.

3.1. Datasets

Three types of datasets are used in the comparative experiments: the Mackey-Glass time series benchmark, four time series provided by NNGC1 competition, and a chaotic laser time series. These datasets are selected because of their diverse sequential patterns.

The Mackey-Glass time series [15] is generated by the following delayed differential equation:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1 + x(t-\tau)^{10}} - bx(t). \qquad (3)$$

This time series is generally used to evaluate and compare the performance of time series prediction approaches [5][14][17]. For the experiments, 2201 data points are generated with an initial value $x(0) = 1.2$, where $a = 0.2$, $b = 0.1$, and $\tau = 17$ by using the 4th order Runge-Kutta method. The last 2000 data points of the time series are used in the experiments.

NNGC1 competition [4] provides diverse non-stationary, heteroscedastic transportation time series data with different structures and frequencies. The datasets are frequently used in the related publications as well. Four longest series collected hourly from the provided datasets are used in the experiments. The length of each time series is 1742.

The chaotic laser time series is a univariate time record of a single observed quantity, measured in a physics laboratory experiment. It comprises the measurements of the intensity pulsations of a single-mode Far-Infrared-Laser NH_3 in a chaotic state [9]. The length of the time series is 1000.

3.2. Evaluation Measures

Two measurements, namely the root mean squared error ($RMSE$) and the fit measure (FIT), are used to evaluate the performance of the prediction models. Let X be the real time series, and $\overline{X}$ be the predicted time series obtained at the prediction horizon n. The length of both X and $\overline{X}$ is m. $RMSE$ is the square root of the variance, which is defined in Equation (4). $RMSE$ is closely related to the value range of the time series data. For this error measure, a smaller value implies a better performance.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{m} (x_t - \overline{x}_t)^2}{m}}. \tag{4}$$

Let the mean value of X be $mean(X)$. The fit measure is defined in Equation (5).

$$FIT = 100 \times (1 - \frac{\|X - \overline{X}\|_2}{\|X - mean(X) \cdot 1_v\|_2})\%, \tag{5}$$

where $1_v =< 1; \ldots ; 1 >$, and the length of 1_v is m. FIT reaches the maximum value (100%) when the prediction $\overline{X}$ exactly matches with the real time series X, *i.e.*, the prediction error $RMSE$ is 0. Other than this ideal situation, FIT is always a number smaller than 100%. FIT describes how fit the prediction is to the real time series, and thus a larger FIT value implies a better prediction performance.

3.3. Experimental Results

We present the results of the performance comparison with LL-MIMO, LS-SVM, AR and ARMA methods. In the experiment, the prediction horizon n is set to 20. For the proposed framework, n' is set to 30 for all the time series. Grid searching was done to tune the rest of the parameters one by one within a preset value range. The parameters include the length of the training time series $Tlength$, the length of the input vector for the k-NN component p_1, the length of the input vector for LS-SVM model p_2, the parameter k in the k-NN approach, γ and σ used in LS-SVM with the RBF kernel, and the parameter h in the autoregressive model. To conduct a fair comparison, the parameters required in LL-MIMO, LS-SVM, AR and ARMA methods are set to be the same values as the ones used in the proposed framework. Table 2 shows the selected parameter values for each dataset.

Table 2. Preset parameter values

Dataset	$Tlength$	p_1	p_2	k	γ	σ	h
Mackey-Glass	700	15	30	80	30	50	1
NNGC1-1	600	20	20	70	10	10	3
NNGC1-2	600	20	20	60	5	10	7
NNGC1-3	600	35	25	110	10	10	1
NNGC1-4	600	30	20	70	5	10	5
Chaotic laser	700	6	25	70	20	17	7

The comparison results on six time series datasets at the prediction horizon $n = 20$ are reported in Table 3 and Table 4, in terms of $RMSE$ and FIT, respectively. As we can see from these two tables, the proposed multi-model integration approach can always achieve the lowest prediction errors in terms of $RMSE$, and the highest fit measure in terms of FIT. Mackey-Glass time series is a synthetic data series without any noise, and the chaotic laser time series is measured in a physics laboratory experiment. Both of the time series datasets have steady sequential patterns. The prediction results for these two time series datasets are relevantly better than the results for the datasets from the NNGC1 competition, which are collected from daily traffic, and thus contains a lot of noise and uncertainties. Also, the NNGC1 data represents a much more dynamic and complicated model, which renders more challenges for the prediction; while the proposed approach is still able to generate fair prediction results on this real-world dataset. The $RMSE$ is related to the value range of the time series, and therefore, its value differs a lot among different time series datasets for one prediction model. FIT is a relative measure, and it is independent from the absolute values of a time series. In this sense, FIT is a better measurement to evaluate different algorithms across multiple datasets. On average, the proposed multi-model integration framework preforms 75.7% better than the LL-MIMO method and 130.2% better than the AR method in terms of FIT.

We present some of the 20-step ahead prediction results by all five approaches for each of the three types of time series in Figure 2 to Figure 4. The figures include the results for Mackey-Glass time series, NNGC1-4 time series (representing the NNGC1 time series), and Chaotic laser time series. Figure 2 shows some of the 20 steps ahead prediction results for time series Mackey-Glass. To make the plot clear, only the 250 predicted data points out of the 1281 testing data are shown in the figure. The corresponding real values of the time series are also plotted. As shown in Figure 2, the predicted time series by LS-SVM and the proposed framework are very close to the real time series; while the proposed multi-model integration framework achieves a lower error measure referring to Table 3. The prediction result of AR goes off the pattern the most. Comparing to AR, ARMA includes a moving average part, which incorporates the prediction error in constructing the prediction model. This makes ARMA more capable and faster in following the trend of a time series. It can be also observed from Table 4 that ARMA always outperforms AR.

Figure 3 shows some of prediction results for the NNGC1-4 time series. The first 50 predicted data points out of the 1123 testing data points are shown in the plot. It is observed that the predicted values by the proposed multi-model integration approach follow the real values quite closely, and it outperforms all the rest approaches significantly. The LL-MIMO, AR and ARMA approaches fail to predict and preserve the trend of the time series dataset, especially from $x = 32$ to $x = 50$. The traditional LS-

Table 3. Performance in terms of $RMSE$

Dataset	LL-MIMO	LS-SVM	AR	ARMA	Multi-Model Integration
Mackey-Glass	0.0735	0.0080	0.1610	0.0854	**0.0015**
NNGC1-1	6594.2	4039.9	6605.4	6497.9	**3362.8**
NNGC1-2	155.84	113.10	158.60	155.91	**101.12**
NNGC1-3	7462.1	4771.5	7286.4	6788.7	**4265.9**
NNGC1-4	2394.7	1683.0	2553.6	2542.5	**1433.8**
Chaotic laser	20.528	5.4761	26.978	9.7876	**2.0260**

Table 4. Performance in terms of FIT (in $\%$)

Dataset	LL-MIMO	LS-SVM	AR	ARMA	Multi-Model Integration
Mackey-Glass	67.25	96.44	28.22	61.91	**99.33**
NNGC1-1	29.43	56.77	29.31	30.46	**63.67**
NNGC1-2	30.63	49.65	29.40	30.59	**54.57**
NNGC1-3	34.05	57.83	35.61	40.01	**62.03**
NNGC1-4	40.10	57.90	36.13	36.41	**63.88**
Chaotic laser	48.20	86.18	31.93	75.30	**95.18**

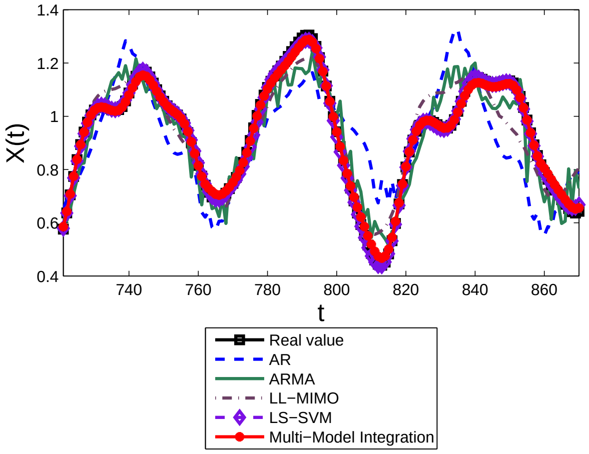

Figure 2. Prediction results for Mackey-Glass time series

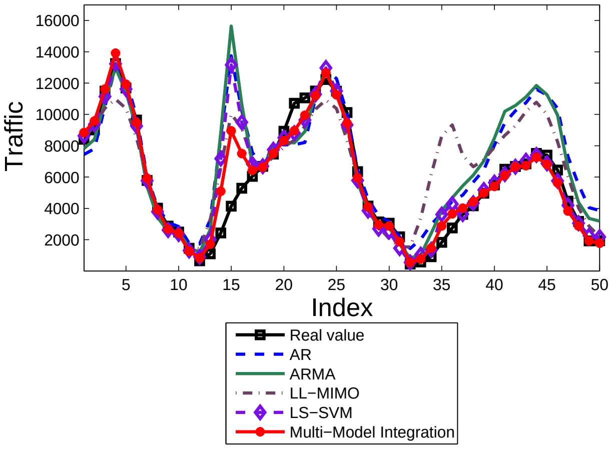

Figure 3. Prediction results for NNGC1-4 time series

SVM approach generates fair prediction results, but it consumes a much longer time because of using a large training dataset, which makes it impractical to do prediction in real time. On average, it took 28.76 seconds for the traditional LS-SVM approach to execute a 20-step ahead prediction, while the proposed approach only requires 0.54 seconds. NNGC1 time series contain some dynamic pattern changes, and there are delays for the prediction models to adapt to the new pattern. Therefore, the overall FIT measures for NNGC1 time series are lower comparing to the other two datasets. Figure 4 shows a part of the prediction results for chaotic laser time series. The first 25 predicted data points out of the 381 testing data points are shown in the plot. The prediction by our framework again follows the real values very well. LS-SVM and ARMA methods give fair prediction results as well, while the prediction results by LL-MIMO and AR shift away from the real time series.

4. Conclusions

In this paper, a multi-model integration framework is proposed to perform long-term time series prediction. The

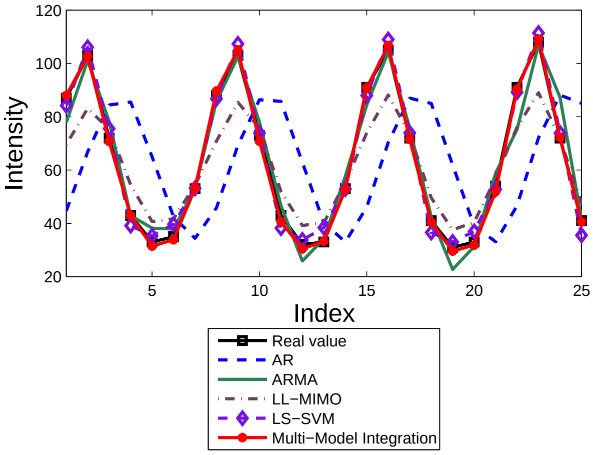

Figure 4. Prediction results for chaotic laser time series

main idea of the proposed framework is to consider utilizing multiple predicted values to generate a more robust final prediction. For this purpose, an autoregressive model is employed to integrate those multiple predictive values generated by a group of k-NN based LS-SVM models. The proposed framework incorporates the non-linear k-NN based LS-SVM models with a linear autoregressive model seamlessly to effectively reduce the uncertainty of the predictive model and decrease the prediction error. The experimental results have validated that the proposed framework constantly outperforms the LL-MIMO, LS-SVM, AR and ARMA approaches with lower prediction errors and higher fit measures under various time series datasets. Although LS-SVM approach generates fair prediction results, high computational cost makes it impractical to perform prediction in real time. The proposed multi-model integration framework significantly reduces the computational complexity by downsizing the training dataset. Therefore, the proposed framework is able to perform long-term time series prediction efficiently and effectively.

References

[1] G. Bontempi. Long term time series prediction with multi-input multi-output local learning. *The 2nd European Symposium on Time Series Prediction*, pages 145–154, 2008.

[2] M. P. Clement and D. F. Hendry. *Forecasting economic times series*. Cambridge University Press, 1998.

[3] R. Cristi and M. Tummala. Multirate, multiresolution, recursive kalman filter. *Signal Process*, 80(9):1945–1958, 2000.

[4] S. F. Crone. Artificial neural network & computational intelligence forecasting competition. February 2010. `www.neural-forecasting-competition.com/`.

[5] L. J. Herrera, H. Pomares, I. Rojas, A. Guilln, A. Prieto, and O. Valenzuela. Recursive prediction for long term time series forecasting using advanced models. *Neurocomputing*, 70(16-18):2870–2880, 2007.

[6] N. Homma, M. Sakai, and Y. Takai. Time series prediction of respiratory motion for lung tumor tracking radiation therapy. *The 10th WSEAS international conference on Neural networks*, pages 126–131, 2009.

[7] Z. Huang and M.-L. Shyu. k-NN based LS-SVM framework for long-term time series prediction. *2010 IEEE International Conference on Information Reuse and Integration*, pages 69–74, 2010.

[8] Z. Huang and M.-L. Shyu. Long-term time series prediction using k-NN based LS-SVM framework with multi-value integration. *Recent Trends in Information Reuse and Integration, Chapter 9*, pages 191–209, 2012.

[9] U. Huebner, N. B. Abraham, and C. O. Weiss. Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH_3 laser. *Phys. Rev. A*, 40(11):6354–6365, 1989.

[10] J.-S. Jang. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3):665–685, 1993.

[11] R. H. Jones. Exponential smoothing for multivariate time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):241–251, 1966.

[12] A. Kusiak, H. Zheng, and Z. Song. Short-term prediction of wind farm power: A data mining approach. *IEEE Transactions on Energy Conversion*, 24(1):125–136, 2009.

[13] K. Lin, Q. Lin, C. Zhou, and J. Yao. Time series prediction based on linear regression and SVR. *The 3rd International Conference on Natural Computation*, 1:688–691, 2007.

[14] P. Liu and J. Yao. Application of least square support vector machine based on particle swarm optimization to chaotic time series prediction. *IEEE International Conference on Intelligent Computing and Intelligent Systems*, 4:458–462, 2009.

[15] M. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.

[16] M. Maralloo, A. Koushki, C. Lucas, and A. Kalhor. Long term electrical load forecasting via a neurofuzzy model. *The 14th International CSI Computer Conference*, pages 35–40, 2009.

[17] K. Meng, Z. Dong, and K. Wong. Self-adaptive radial basis function neural network for short-term electricity price forecasting. *Generation, Transmission Distribution, IET*, 3(4):325–335, 2009.

[18] H. H. Nguyen and C. W. Chan. Multiple neural networks for a long term time series forecast. *Neural Computing & Applications*, 13:90–98, 2004.

[19] K. Pelckmans, J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, L. Lukas, B. D. Moor, and J. Vandewalle. LS-SVMlab toolbox user's guide. *ESAT-SCD-SISTA Technical Report*, pages 1–106, 2003.

[20] N. Sapankevych and R. Sankar. Time series prediction using support vector machines: A survey. *IEEE Computational Intelligence Magazine*, 4(2):24–38, 2009.

[21] S. Soltani, D. Boichu, P. Simard, and S. Canu. The long-term memory prediction by multiscale decomposition. *Signal Processing*, 80(10):2195 – 2205, 2000.

[22] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Farrer Road, Singapore, 2002.

[23] S. B. Taieb, G. Bontempi, A. Sorjamaa, and A. Lendasse. Long-term prediction of time series by combining direct and MIMO strategies. *International Joint Conference on Neural Networks*, pages 3054–3061, 2009.

[24] G. Zhang. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50:159–175, 2003.

[25] H. Zou and Y. Yang. Combining time series models for forecasting. *International Journal of Forecasting*, 20(1):69–84, 2004.