Paper 11823-2016

Upgrade from ARIMA to ARIMAX to Improve Forecasting Accuracy of Nonlinear Time-Series: Create Your Own Exogenous Variables Using Wavelet Analysis

Woranat Wongdhamma, Ph.D., Oklahoma State University

ABSTRACT

This paper proposes a technique to implement wavelet analysis (WA) for improving a forecasting accuracy of the autoregressive integrated moving average model (ARIMA) in nonlinear time-series. With the assumption of the linear correlation, and conventional seasonality adjustment methods used in ARIMA (that is, differencing, X11, and X12), the model might fail to capture any nonlinear pattern. Rather than directly model such a signal, we decompose it to less complex components such as trend, seasonality, process variations, and noises, using WA. Then, we use them as exogenous variables in the autoregressive integrated moving average with explanatory variable model (ARIMAX). We describe a background of WA. Then, the code and a detailed explanation of WA based on multi-resolution analysis (MRA) in SAS/IML® software are demonstrated. The idea and mathematical basis of ARIMA and ARIMAX are also given. Next, we demonstrate our technique in forecasting applications using SAS® Forecast Studio. The demonstrated time-series are nonlinear in nature from different fields. The results suggest that WA effects are good regressors in ARIMAX, which captures nonlinear patterns well.

INTRODUCTION

In time-series prediction and forecasting applications, the statistical powers are mainly based on the underlying dynamics of the time-series itself (in case when explanatory or exogenous variables are unavailable or uncorrelated). As commonly known, the nonlinear or irregular behaviors embedded in the data always impede the predictability in both time-series and pattern recognition applications. This problem happens in many aspects. One of them is that, most of the time, we can only observe at most one of the process activities or its result, not all influences that comprise to the process. Thus, the characteristics of this observable data is always a result of couplings or interactions between several other unknown variables. This make the data patterns to be irregular and non-periodic. For example, we can easily observe a stock price but not all factors that influence the change in the price such as short or intermediate trends from investors and so on. If we assume that all influences comprising to the process are independent and we can directly observe their activities, most likely, the process could approximately be represented by a mathematical model that is built by a linear combination of the effects from all influence activities. This simplifies the problem from nonlinear system to the approximated linear system. Unfortunately, that is not always the case.

We may assume that there might be trend, seasonality, regular, and irregular patterns embedded in the data using the background or historical knowledge. These influences could be easily tested using several built-in functions in SAS® Forecast Studio. However, as a data scientist, we may not be an expert in the domain of the analyzed data; therefore, we may have limited understanding in the focused system. To tackle this problem, another way around is that, instead of trying to model one complicate time-series, we can decompose the data into several or many less complex time-series then study their patterns separately. To do this, the wavelet transformation could be used to decompose the signal to several components such as trends, seasonality, signal's variations, and noises based on their central dominant frequencies. These decomposed signals could be viewed as exogenous or explanatory variables. Each of them will be less complex and may correlate to the original time-series differently. This creates more regular patterns for the modeling techniques to learn, ensuing in more precise prediction or forecasting result.

WAVELET TRANSFORMATION: BACKGROUND

Wavelet decomposition is a modified short-time Fourier transform that represents the decomposed signals in both time and frequency domain through time windowing function or mother wavelet function [1]. Traditionally, the Fourier transform is normally used for analyzing the signal in frequency domain. However, in nonlinear time-series that contains short duration transients, Fourier transform failed to capture that behavior. When transformed the short transient in time domain to frequency domain, it corresponds to a damped and long-duration vibration [2]. This time-frequency localization advantage is a well-known characteristic of a wavelet transformation. In contrast to Fourier transform, which assumes the signal to be stationary, the wavelet analysis does not have such limitation so that it works well with the nonstationary time-series.

The wavelet transformation process comprises of two main phases, analysis or decomposition and synthesis or reconstruction phases. If the certain condition is met, the signal can be perfectly reconstructed using the coefficients obtained from the analysis or decomposition phase. With these reasons, the wavelet decomposition is popular in a signal denoising application. The user can selectively delete the decomposed coefficients corresponding to the noises and reconstruct the denoised signal back.

There are several mathematical methods that could be used to achieve a wavelet decomposition. The one that seems to be intuitively easy to understand is a multiresolution analysis (MRA) developed by Mallat in 1989 [3]. In general discrete wavelet transformation (DWT), the signal is passed through a series of high-pass filters (mathematical tool that allows only fast changing value data to pass, otherwise zero) and low-pass filters (passing slow changing value data, otherwise zero) as shown in Figure 1 below:

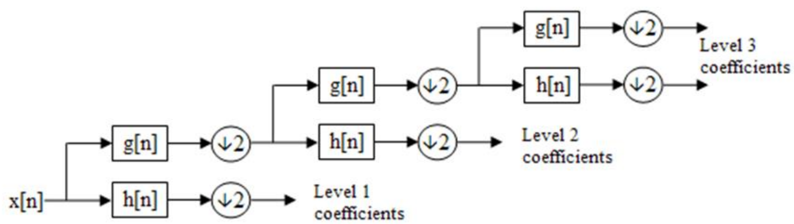

Figure 1. Discrete wavelet transform (DWT) using multiresolution analysis (MRA) with 3 level filter banks

The DWT procedure starts from feeding the time-series $x[n]$ to the half band low-pass filter with an impulse response $g[n]$ and half band high-pass filter with an impulse response h[n]. In mathematical expression, the filtering process is the convolution of the signal with the impulse response of the filter:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{1}$$

Regarding to the Nyquist theory, after passing the signal through either a half band low-pass filter or a half band high-pass filter, half of the samples could be eliminated. This denotes by the symbol ↓2 in the Figure 1. The result of the first high-pass filter is level 1 detail coefficients. Likewise, the result of the first low-pass filter is level 1 approximation coefficients. To perform a further analysis, the level 1 approximation coefficients are used as a signal to be passed through another set of half band low-pass and high-pass filters. In theory, the decomposition level could be done for n levels. However, in practice, the analysis levels depend on a number of samples of the original signal. It should be noted that because the decomposition process involves the downsampling with the factor of two. Thus, the number of samples required in the wavelet analysis must be the power of two.

In synthesis (reconstruction) phase, to be able to perfectly reconstruct the signal back from the wavelet coefficients in every decomposed level, the pair of low-pass and high-pass filters must form orthonormal bases. To satisfy that constrain, the relationship between them is [4]:

$$h[L-1-n] = (-1)^n \cdot g[n] \tag{2}$$

Where $h[n]$ is impulse response of a high-pass filter

$g[n]$ is impulse response of a low-pass filter

L is the filter length in number of sample

When the filter pair that satisfies equation 2 is used, the reconstruction process is exactly the reverse process of the analysis process. The coefficients at every level are upsampled with the factor of two then passed through the synthesis filter pairs. The relationship between the analysis and synthesis filters is that they are identical to each other but time reversal. There are many choices of the low-pass and high-pass filter pairs used in wavelet analysis. SAS/IML® [5] provides two choices of wavelet family, the Daubechies Extremal phase family, and the Daubechies Least Asymmetric family (Symmlet family). For further information, reference [4] provides very good information on theory and application of wavelet decomposition.

WAVELET DECOMPOSITION USING PROC IML

The goal of this section is to go through how a wavelet decomposition is processed in SAS/IML®. In this paper, we use SAS/IML® version 12.1 user guide [5] as a guideline, specifically Chapter 19 and 23 for a wavelet analysis. To start, after the time-series data was successfully imported into SAS library. We can activate PROC IML using the following command:

```
proc iml;
```

Unlike others, PROC IML works interactively inside its own shell. Once activated, the commands for computation are little different comparing to the normal commands. Now, we will start to do a wavelet decomposition using the commands bellow:

```
use SASUSER.Winddatamod; *indicate the dataset to be used;
    read all var{SPEED80M_M_S_}into SPEED;
    optn ={0,.,2,10};*SYMLET10;
        call wavft(decomp,SPEED,optn);
        call coefficientPlot(decomp,,,,,"Summary of wavelet decomposition's coefficient");
```

The first line in the code above is to indicate which dataset to be used. In this case, dataset $Winddatamod$ from a library $SASUSER$ is assigned. Next, we read all values in variable $SPEED80M\_M\_S\_$ into a variable name $SPEED$ in PROC IML shell. Then, the options for our wavelet decomposition is assigned. Briefly, there are 4 options needed to be declared before the wavelet decomposition could be executed.

The first element in vector $optn$ (opt[1]) indicates how the signal boundary is handled. One of the wavelet analysis limitations is that the analysis signal must have a number of data points (N) in the increment of 2^n where n=1,2,3…. SAS IML has a built-in function to handle this limitation using several options for padding the signal such as padding the signal by zero, the signal reflection, user specified number, and so on. In our case, we use zero padding because of its simplicity and manageable bias. However, to reduce the error introduced in the analysis process, it is suggested that the data should be format to the length of 2^n. In the next option, (opt[2]), the user can indicate the degree of the polynomial to be used in the data padding if the first option (opt[1]) is set to be 2. Since we use zero padding, this option will be ignored by PROC IML. For option 3, (opt[3]), the user must specify the method to be used for a decomposition. Symmlet family, (opt[3]=2), was chosen in our case because of its near symmetric property which is desirable in the reconstruction phase.

Finally, the last option, (opt[4]), chooses which wavelet family member to be used in the decomposition. Generally, the wavelet family member indicates how enlarged or compressed the wavelet base function is (the higher number indicates more compressed wavelet base function). The choice for choosing this number depends solely on the user's application. Some experiments may be needed before the final wavelet family member is chosen. For the demonstration, we use Symmlet10 in this case (opt[4]=10). For more information about the aforementioned options, please consult Chapter 19 and Chapter 23 in the SAS/IML® user manual [5].

After required options have been specified, we call a wavelet decomposition $(call\ wavft(\ldots..))$ on variable $SPEED$ and its decomposition information will be stored in variable $decomp$. To visually inspect the decomposition, $coefficientPlot$ call is used. The result is shown in the Figure 2. We can use the call $wavprint$ to see the summary of the composition also. From this plot, we look for the total number of the decomposed levels. In this case, we have a total of 18 decomposition levels (start level = 0 and top level =17). The lower levels are composed of lower frequencies (slow changing) components extracted from the original data. Likewise, the higher levels are composed of higher frequency (fast changing) components.

To interpret, we can look at the lower frequency levels to see if there is any trend or long-term seasonality, and at the higher frequency levels to look for noises or short time influences in the data.

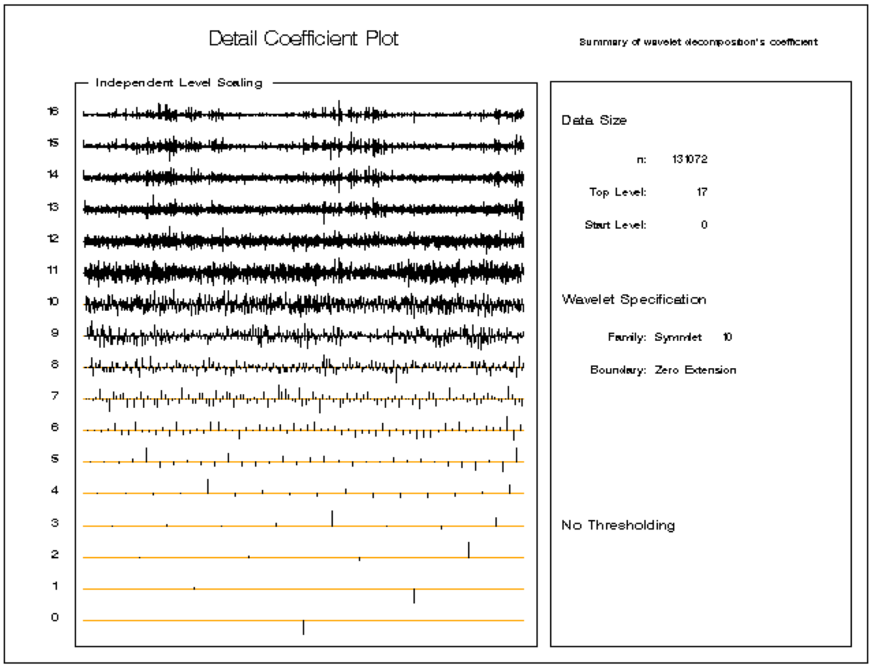

Figure 2. Detailed coefficient plot and decomposition summary

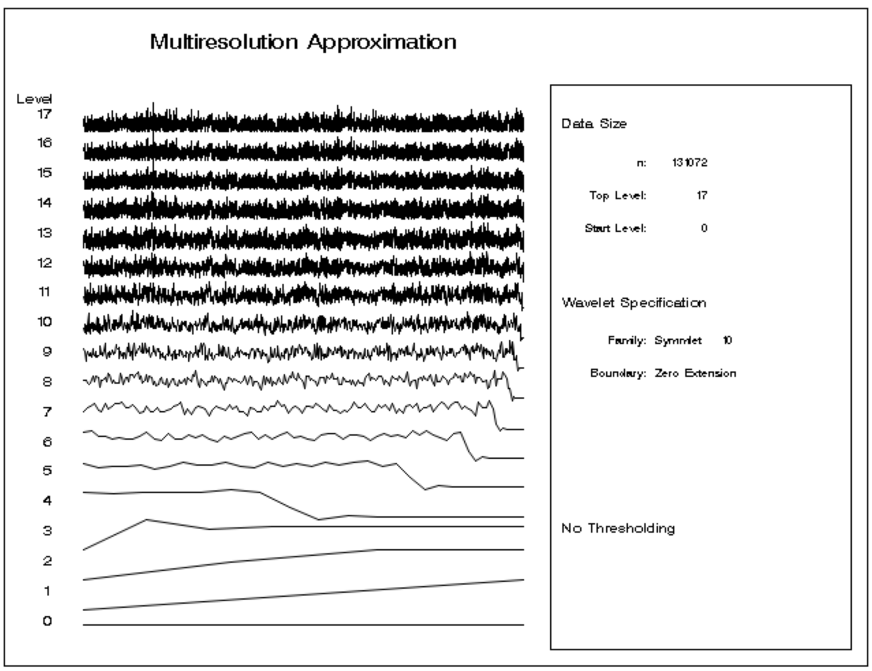

Figure 3. Multiresolution approximation of the signal corresponding to each decomposition level

Next, the wavelet decomposition based on a multiresolution analysis (MRA) is called using the command below:

```
call mraApprox(decomp,,0,,);
```

The result of the $mraApprox$ call is shown in Figure 3. The time-series in each level is corresponding to the reconstruction based solely on the detail coefficient of that particular level. If no loss was introduced in analysis and synthesis phases, the summation of every level shown in Figure 3 is our original signal. Unfortunately, to the best of our knowledge, with the PROC IML version used in this analysis (12.1), there is no direct way to obtain any decomposed time-series from $mraApprox$ call. However, with available commands in PROC IML, we can reconstruct each corresponding time-series based on the MRA concept from the process that will be described as follows.

By using the obtained coefficients in variable $decomp$ and its wavelet base function, we can reconstruct the original signal back or choose not to use some levels that corresponding to the noise in the signal in the reconstruction process. PROC IML has a built-in function to help eliminate noises in the signal with $wavift$ call which is the Inverse Fast Wavelet Transform (WAVIFT) via several thresholding methods such as hard, soft, and garrote thresholding methods (see [5] for more information). Now, to continue our decomposition, our next goal is to reconstruct the time-series corresponding to each decomposed level using MRA concept. Another way around is to manually keep each level of the reconstructed time-series by manually thresholding other non-desired level. It may sound simple but the $wavift$ call was not originally designed to do such task. First, the usage of $wavift$ call is as follows [5]:

```
call WAVIFT(result,decomp<,opt>< ,level>);
```

The options we used is the hard threshold which corresponds to the equation 3 below [5]:

$$\delta_T^{hard}(x) = \begin{cases} 0 \; if \; |x| \leq T \\ x \; if \; |x| > T \end{cases} \tag{3}$$

Intuitively, if the absolute value (magnitude) of the signal $(|x|)$ is smaller than or equal to the threshold value (T), that data point will be set to zero, but if it is larger than the threshold, it will be set to itself. Thus, we will use a very high threshold on the levels that we would like to eliminate. The code used in this case is as follows (still in PROC IML shell):

```
n=nrow(SPEED); *declare array size in proc iml;
     wind=j(n,18,0);
     effect=j(n,18,0);
     temp=j(n,1,0);
         opt=j(4,18,0);
         opt[1,]=1;
         opt[2,]=0;
         opt[3,]=100;
         opt[4,]=0:17;
         call wavift(buffer,decomp,opt[,1]);
     wind[,1]=buffer;
*Apply the threshold and reconstruct wavelet decomposed signal to all levels;
     do i=1 to 18;
         call wavift(buffer,decomp,opt[,i]);
         wind[,i]=buffer;
     end;
```

To thoroughly explain the code above, the calculation in PROC IML is done in a matrix fashion so that it is a good practice to declare a dimension of the matrix that will be used. For example, a number of data point in $SPEED$ is looked up and kept in variable n. Then, we will store the reconstruction results in the variable name $wind$ so that we declare the size of this variable to be n row and 18 columns which is corresponding to the decomposition levels.

For $WAVIFT$ call options, the first option is to specify that the hard thresholding method will be used. Then, we specify option two to be 0 to use the global user-defined threshold. For the third option, this is a threshold value $(T = 100)$ which is pretty high comparing to our signal. Finally, the last option will specify the number of levels that the thresholding will be applied to, starting from the highest level. This means that we cannot apply the hard threshold exclusively on each detail coefficient. Again, the calculation method to get the individual reconstructed time-series will be explained later. For now, we will apply the threshold to the detail coefficient and reconstruct the decomposed signal starting from the lowest level iteratively until we reach the highest level of the decomposition. This is done by do-loop in the code above.

In the first loop, we applied the threshold to the highest level at level 0, meaning that the thresholding was done to level 0 only. Therefore, the reconstructed signal in this loop is the signal that does not contain any effect from the detail coefficient at level 0. In the next loop, the thresholding was done to the highest level at level 1 and 0 so that the reconstructed signal does not contain any effect from level 0 and level 1 detail coefficients. The process is executed until we reach the last level. The reconstructed signals without the effects are stored in variable $wind$ in the hierarchy fashion (highest to lowest). Finally, we can obtain the exclusively reconstructed time-series from each level detail coefficient (effect) by:

```
*calculate effects;
do i=1 to 18;
effect[,i]=SPEED-wind[,i]-temp;
temp=temp+effect[,i];
end;
```

The idea is that, the first column of variable $wind$ is the time-series that does not contain any reconstructed component from level 0. Thus, if we subtract this time-series off the original signal ($SPEED$), what left is actually the reconstructed time-series exclusively from the level 0 detail coefficient (we will call this the effect 0). Thus, in the next iteration, the reconstructed time-series exclusively from the level 1 coefficient (effect 1) could be derived from subtracting the time-series that does not contain any reconstructed component from level 0 and 1 (second column of variable $wind$) and effect 0 from the original signal ($SPEED$). This process is executed until we obtain all effect time-series equal to the number of decomposed levels. An example of the effects with the original signal is shown in Figure 4 below:

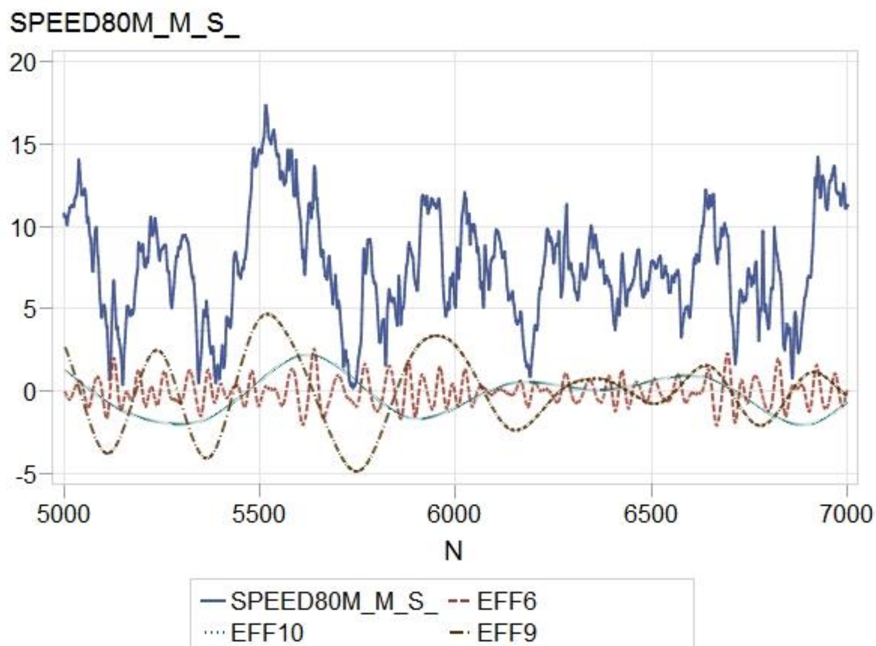

Figure 4. A plot between the original signals and some wavelet-decomposed components

Shown in Figure 4 above is a plot of the original time-series used in this example which is a wind speed (solid line) and some of its wavelet decomposed signals. We can see a nonlinear pattern that impedes the ability to accurately forecast this original time-series. Intuitively, we can think of this signal as a combination of many simple signals. Using the aforementioned wavelet decomposition method, we decomposed the original signal to many levels based on their center frequencies. This helps unfolding the very complicated original signal to much simpler signals so that we can analyze each of them individually then use their information to infer back to the original time-series. For example, the reconstructed time-series level 9 (EFF9) can capture a slow seasonal pattern in the original data well and is much simpler to analyze. Also, the EFF10 captures much slower trend of the data precisely. Furthermore, the EFF 6 captures much faster seasonal patterns. With these characteristics, these decomposed signals can be used as explanatory variables in ARIMAX model to improve the forecasting accuracy, especially in the long-term forecasting applications that will be demonstrated in the next sections.

The challenging task for the user after this is that they will have to look at each decomposed signal individually and infer their meanings based on the specific knowledge area. For example, in this case, we may interpret the EFF6 as a seasonality pattern effected by differences in day and night temperatures that influence the wind speed.

ARIMA MODEL

The well-known autoregressive integrated moving average (ARIMA) model was first introduced by Box and Jenkins [6]. In brief, the ARIMA model utilized the history information of a univariate stochastic time-series and use it to minimize the model's forecasting error. There are three components in the model, autoregressive process (AR), integration (I), and moving average process (MA). Originally, the model structure of ARMA (without integration (I) part) only allows to be used with the stationary time-series. Practically speaking, the (weak) stationary means that the mean, variance, and covariance of time-series remain constant over time. However, most of time-series shows trend over time. Therefore, the integration (I) or differencing can be used to remove such trend, making the time-series stationary.

An autoregressive process (AR), is a process that has a significant relationship with its history observations (previous time lags) and a moving average process (MA) is a process that has a significant relationship with its previous random errors. The complete ARIMA model is a linear combination of AR and MA processes or ARMA that their parameters derived from a time-series that becomes stationary through differencing with the form:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \tag{4}$$

We say that time-series y_t is an ARIMA (p, d, q) process where d indicates a number of differencing needed for a stationary, and p and q indicate a number of AR and MA parameters. A time information is indicated by a subscript t (i.e., t indicates a current observation and $t-1$ indicates a previous one time-step observation). Note that an augmented dickey fuller (ADF) test is normally used for testing a stationary.

To build an ARIMA model using Box and Jenkins approach, there are three iterative steps:

1. Identification stage; Identify a number of parameters needed for AR and MA processes. The idea is to match the empirical autocorrelation patterns with the theoretical ones. An autocorrelation function (ACF) plot and a partial autocorrelation function (PACF) plot are primary tools used for identifying a number of parameters for AR and MA processes respectively. Some other tools such as an extended autocorrelation function (EACF) may also be used. This is to come up with candidate models.

2. Estimation stage; The parameter coefficients for all candidate models are estimated in this stage. The standard method for a parameter estimation is a maximum likelihood estimation. Then, the best model is selected best on their Akaike information criterion (AIC) or Bayesian information criterion (BIC). Intuitively, AIC and BIC tell which model fits better to the data in term of the information loss.

3. Diagnostic stage; The residual of the selected model is tested against a white noise assumption (zero mean, no autocorrelation, normality, and constant variance). If the residual violates the white noise assumption, the model candidate is rejected and the next best model candidate is then tested.

ARIMAX MODEL

Intuitively, the autoregressive integrated moving average with explanatory variable (ARIMAX) model can be viewed as a time-series forecasting model using the multiple regression with ARIMA model that takes care of the residual's serial correlations. Therefore, to build ARIMAX model, one can follow the stepwise multiple regression method to develop a multiple regression model that well fits to the time-series. Then, build the ARIMA model to fit the residual of the regression model. The ARIMAX model has a form as follows:

$$y_t = \beta x_t + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \quad (5)$$

However, the process to build such model is iteratively complex because the result from each model building step can violate required assumptions. The reference [7] has concluded such assumptions as follows:

1. Both time-series used for building the regression model and the residual must be stationary. If the residual is not stationary, more differencing of original time-series is needed and the regression model must be rebuilt.
2. The final model's residual follows the white noise assumption.
3. The coefficients of each exogenous variable in the final model must be statistically significant. In some occasions, after building the ARIMA model for the regression model's residual, some regression's coefficients can become insignificant. Then, the least significant coefficient must be removed and all assumptions must be reevaluated.
4. There could be only one-way causal relationship from an exogenous variable to a dependent variable but not from a dependent variable to an exogenous variable (using Granger causality test). If the reverse causal relationship is found, that exogenous variable must be removed and all assumptions must be reevaluated.
5. The regression coefficients in the final model must display the same relationship (sign) with the correlation coefficients of the exogenous variables and the dependent variable (original time-series).
6. There is no multicollinearity found between exogenous variables in the final model.

In this and previous sections, we introduce a basic background of the ARIMA and ARIMAX model so that the reader can have intuitive idea about them and how the wavelet decomposed signals can be used to improve the forecasting accuracy. However, to develop a time-series forecasting model, SAS® Forecast Studio implements more sophisticated methods such as ARIMA with seasonality adjustments, and transfer function models which are more advanced topics that could not fit in this paper. For more information, please consult these SAS user guides [8, 9].

DATA DESCRIPTION AND PREPARATION

To demonstrate the proposed technique, we choose to use three complicated datasets from various fields with irregular or nonlinear patterns which are challenging for forecasting:

1. Heart rate variability (HRV) of a sleep apnea and cardiovascular patient from National Institutes of Health (NIH) with 72,000 samples (sampled every 10 seconds, accounting for 120 minutes of HRV). The HRV is normally used for detecting cardiovascular and sleep disorder diseases. The forecasted information of this data could help preventing the incoming attacks which could be intervened before they really occur.
2. Daily S&P 500 close values with 5,200 samples (January 3, 1995 – August 26, 2015, from Yahoo Finance). This time-series is quite different from the other two that it shows not only the nonlinearity patterns but also the nonstationary trait (ADF tests suggested with $p > 0.05$).
3. Wind speed collected at a wind turbine from National Renewable Energy Laboratory with 30,528 samples (sampled every 10 minutes) from January 1 to July 31, 2005 (station ID 00365 in Eastern Wind Dataset) [10].

To prepare the data for an analysis in SAS® Forecast Studio, the processes needed to be carried out before the analysis are 1) combine the DATE variable with TIME variable 2) format the combined variable to comply with Time ID Variable format that can be used in SAS® Forecast Studio, and 3) wavelet decomposition of each time-series using PROC IML [6]. The modeling process will be described in the next section.

METHODOLOGY

To recapitulate, the goal of this study is to demonstrate the usefulness of wavelet transformation in long-term time-series forecasting applications. To do so, we already explained the idea and code for a wavelet decomposition using SAS/IML® in previous sections. In this section, we integrate the wavelet decomposed data into the ARIMAX model as exogenous variables and compare its forecasting performances to the ones from the conventional model, ARIMA. To minimize a modeler's bias in modeling process, we use SAS® Forecast Studio that follows ARIMA and ARIMAX modeling assumptions rigorously to automatically build 1) ARIMA model from the nonlinear time-series alone, and 2) ARIMAX model built from the nonlinear time-series as dependent variable and its wavelet decompositions using PROC IML as exogenous variables. To summarize, the process is depicted in Figure 5 below:

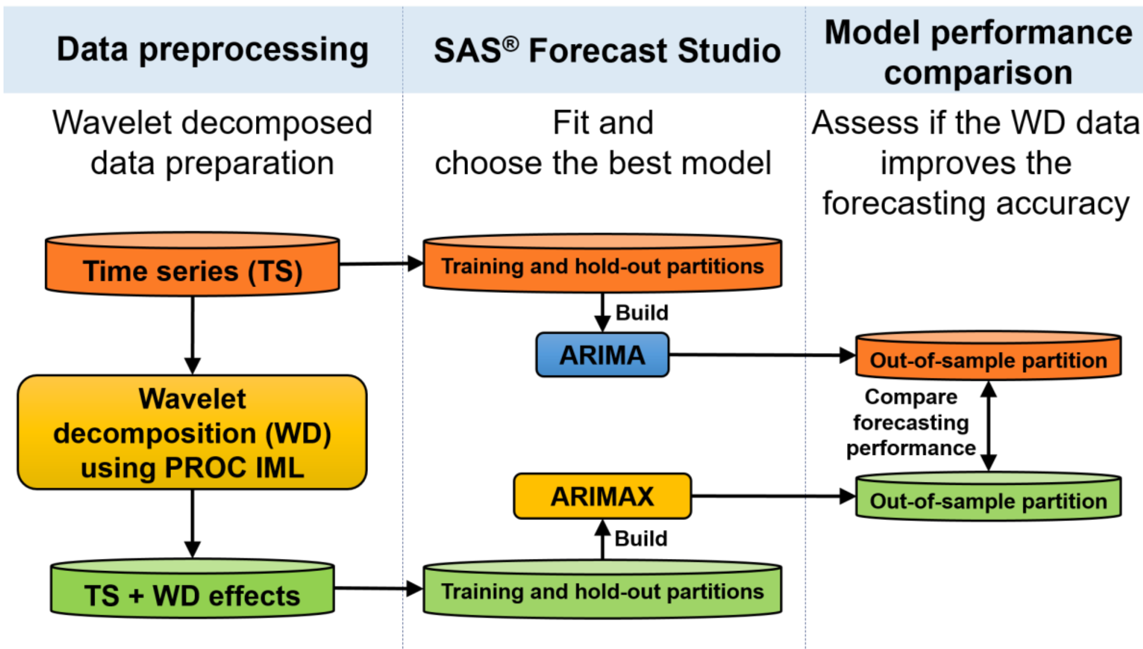

Figure 5. Research methodology

We also implement a data partitioning strategy to reflect a real-world performance of the proposed technique. A training partition is used for candidate models development. Then, the best model is chosen from its performances, mean absolute percentage error (MAPE) and adjusted R-squared, in a hold-out partition which is not initially included in the training partition. Finally, the out-of-sample partition is used for evaluating the real-world performances of each chosen model. The data partitioning scenarios used in this study are shown in Table 1 below:

Data	Training partition	Hold-out partition	Out-of-sample partition
Heart rate variability (HRV)	66,000 samples	3,000 samples	3,000 samples
Daily S&P 500 close values	4,960 samples	120 samples	120 samples
Wind speed	28,512 samples	1,008 samples	1,008 samples

Table 1. Data partitioning scenarios

We tried to have a training partition as long as possible to cover all the patterns embedded in the data. We also decide to have the length of the hold-out partition to be the same as the out-of-sample partition which is the desired forecasting horizon. The model performances are reported in the next section.

RESULTS

We compare ARIMA and ARIMAX-WD models using mean absolute percentage (MAPE) and adjusted R-squared in the out-of-sample data partition. The results are reported in three cases below:

Case I: Heart rate variability (HRV) with a forecasting horizon of 3,000 samples.

(note that the time information on x-axis of the plots does not reflect the real date or time due to limited options in SAS® Forecast Studio).

Model	Model architecture	In-sample		Out-of-sample	
		MAPE	adj. R-sq.	MAPE	adj. R-sq.
Multiplicative Seasonal ARIMA	Differencing: (0) P: (1,2,3,4) (10,20) Q: (1,2,3,4,5) (10,20)	0.59	99.68%	11.35	-44.69%
ARIMAX-WD	Differencing: (0) P: (1,2,3,4,5) (10) Q: (1,2,3,4,5) (10,20) Exogenous variable WD: (7,8,9,10,11,12)	0.0074	99.99%	4.99	74.62%

Table 2. Comparison of forecasting performances of the two models in out-of-sample HRV data

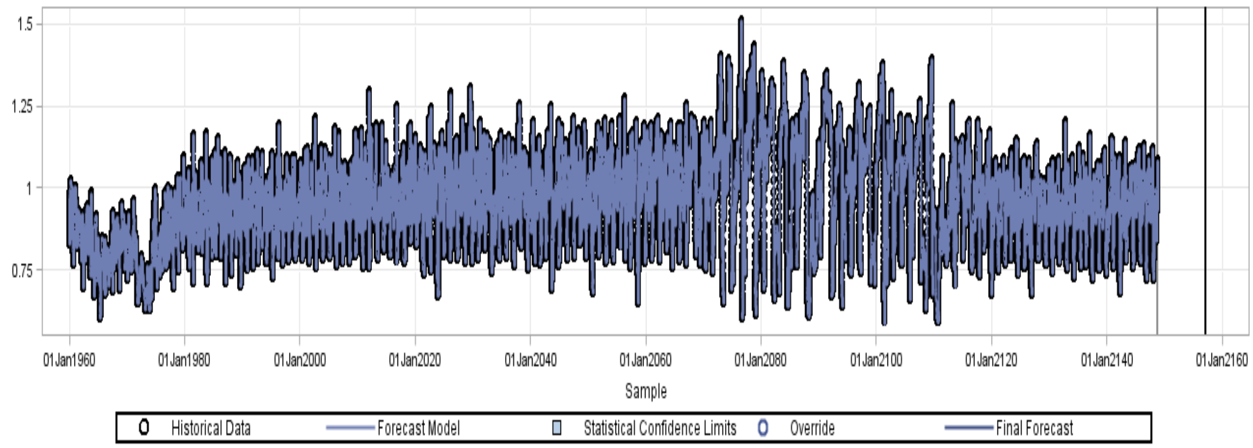

Figure 6. A plot between the actual and forecasted data using ARIMA model in HRV data (in-sample)

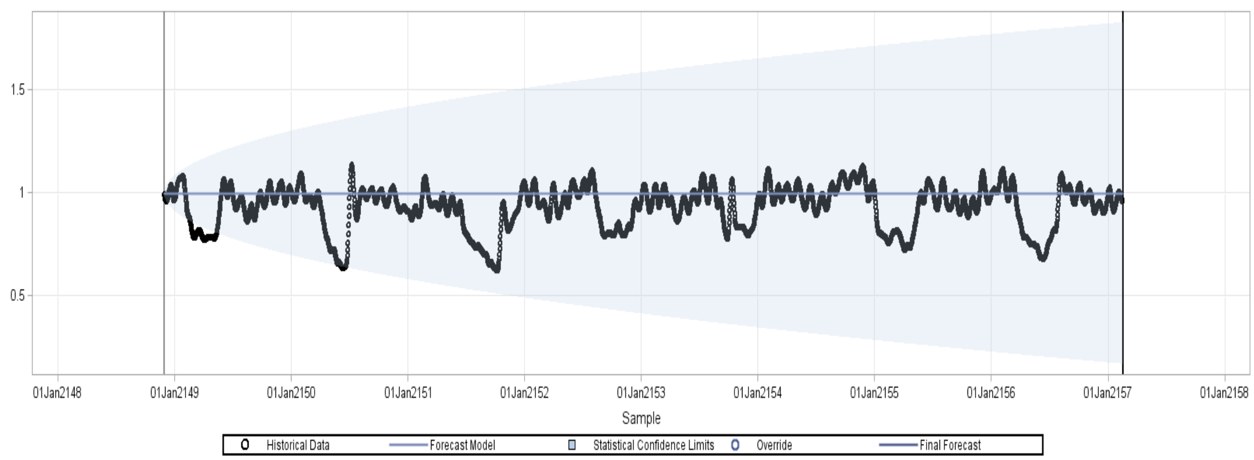

Figure 7. A plot between the actual and forecasted data using ARIMA model in HRV data (out-of-sample)

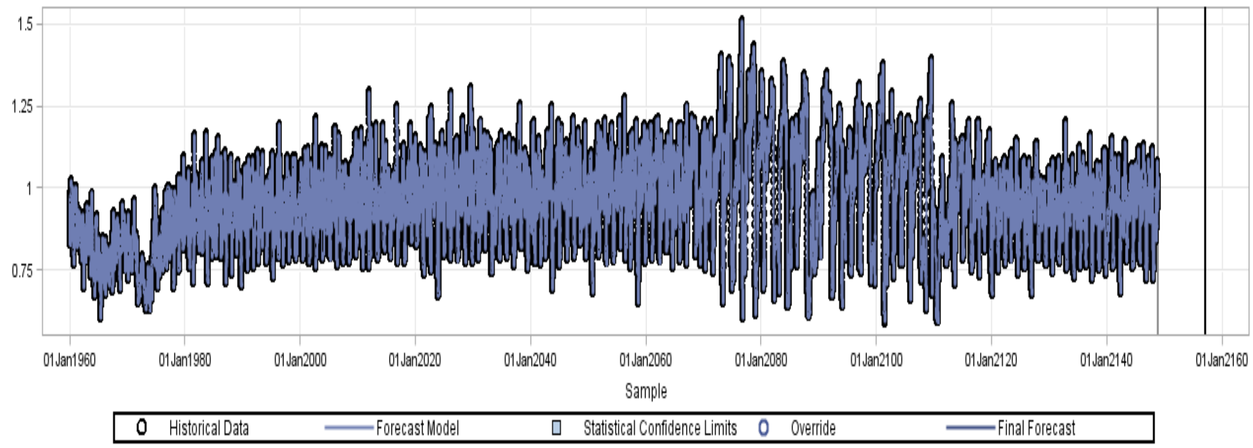

Figure 8. A plot between the actual and forecasted data using ARIMAX-WD model using in HRV data (in-sample)

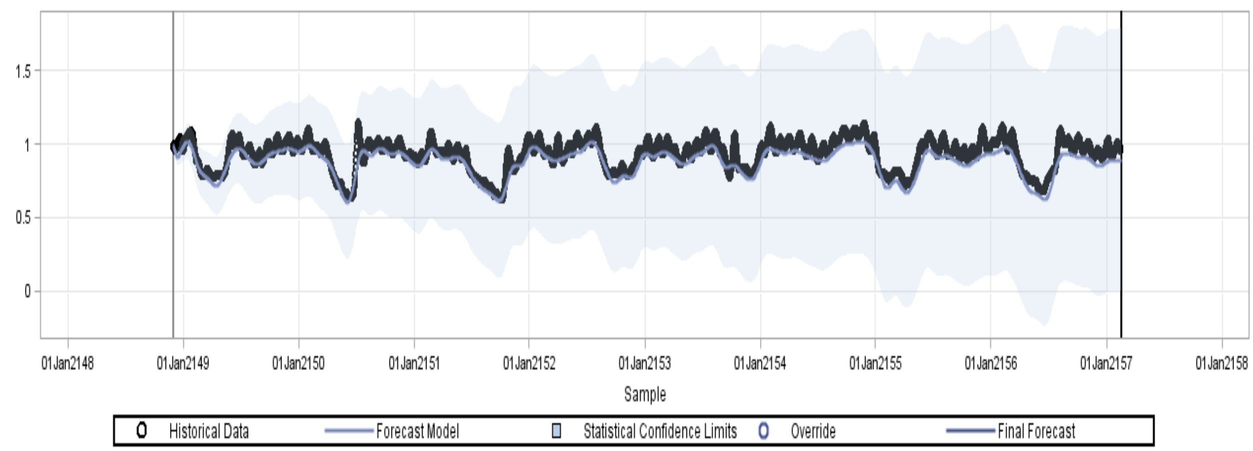

Figure 9. A plot between the actual and forecasted data using ARIMAX-WD model in HRV data (out-of-sample)

In HRV forecasting case (forecasting horizon: 3,000 samples), the results in Table 2 and Figure 6 and 8, suggest that both multiplicative seasonal ARIMA and ARIMAX-WD perform well in an in-sample data with low MAPE (0.59 and 0.0074) and high adjusted R-squared (99.68% and 99.99%). However, in out-of-sample data, the multiplicative seasonal ARIMA cannot keep up with a complex data pattern as seen from results in Table 2 (MAPE:11.35 and adj. R-sq:-44.69%) and Figure 7, whereas the ARIMAX-WD can forecast well with the reasonable results (MAPE:4.99 and adj. R-sq:74.62%) (see Figure 9).

Case II: Daily S&P500 close values with a forecasting horizon of 120 samples (120 days).

Model	Model architecture	In-sample		Out-of-sample	
		MAPE	adj. R-sq.	MAPE	adj. R-sq.
ARIMA	Differencing: (1) P: (1)	0.82	99.83%	2.76	-1.59%
ARIMAX-WD	Differencing: (1) P: (1) Exogenous variable WD: (3,4,5,6,7,8)	0.51	99.93%	0.90	65.52%

Table 3. Comparison of forecasting performances of the two models in out-of-sample daily S&P500 close value data

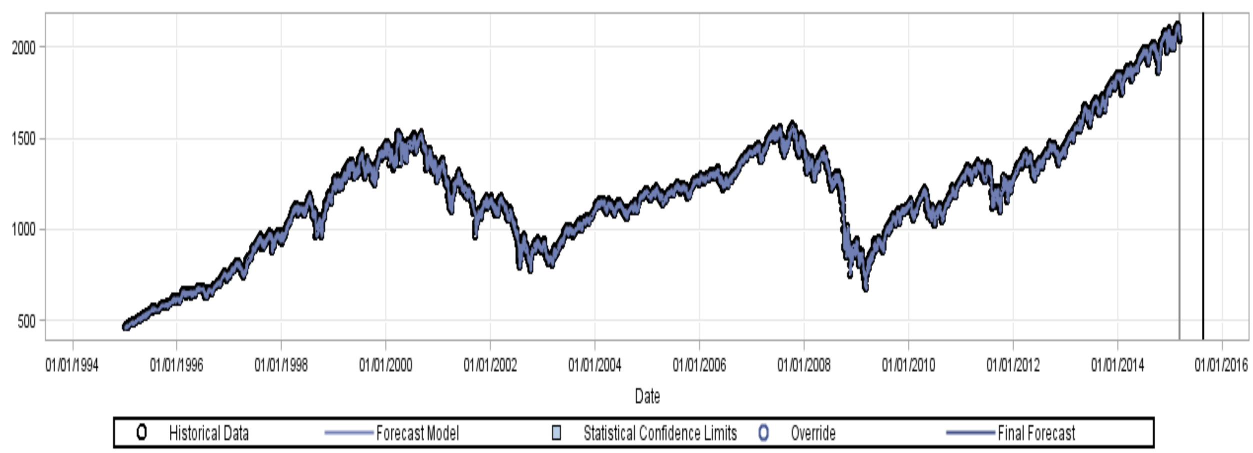

Figure 10. A plot between the actual and forecasted data using ARIMA model in daily S&P500 close value data (in-sample)

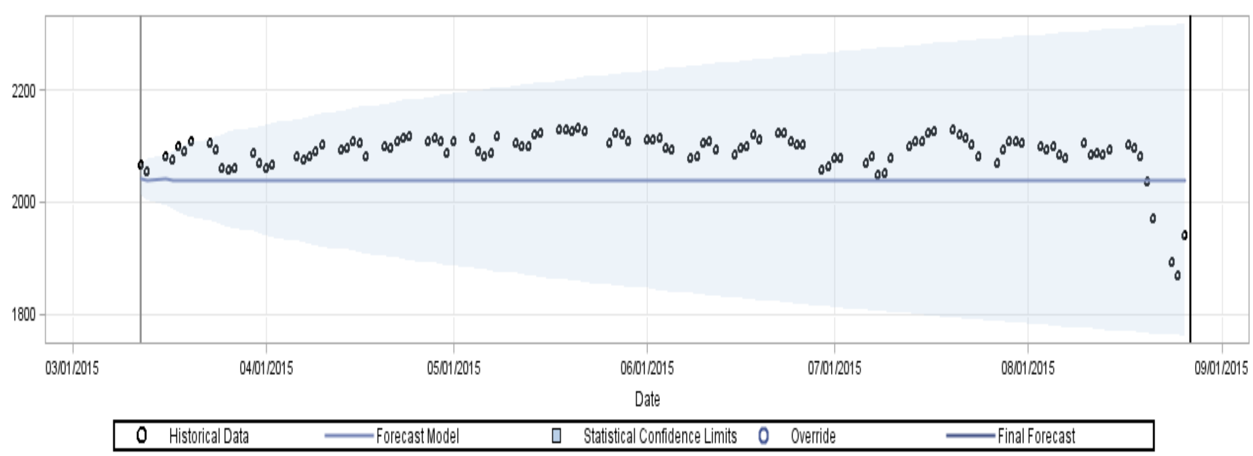

Figure 11. A plot between the actual and forecasted data using ARIMA model in daily S&P500 close value data (out-of-sample)

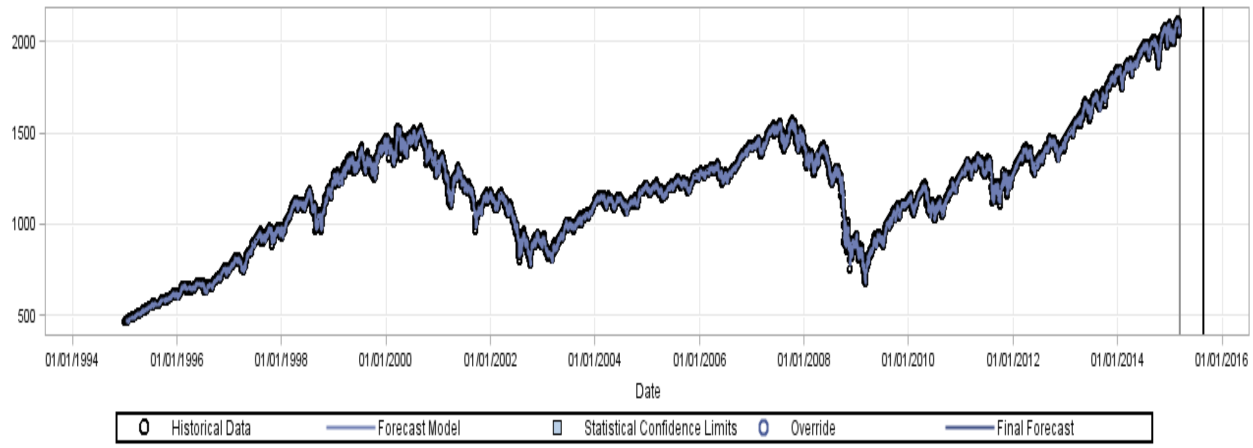

Figure 12. A plot between the actual and forecasted data using ARIMAX-WD model using in daily S&P500 close value data (in-sample)

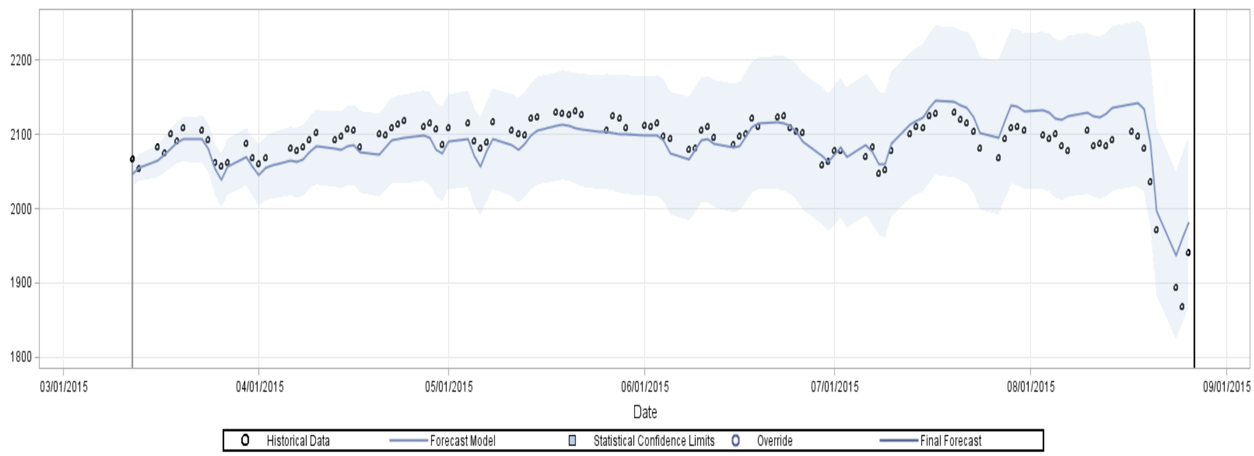

Figure 13. A plot between the actual and forecasted data using ARIMAX-WD model in daily S&P500 close value data (out-of-sample)

In daily S&P500 close value forecasting case (forecasting horizon: 120 samples), both models perform well in the in-sample data (see Figure 10 and 12) as seen from low MAPE and high adjusted R-squared in Table 3. Again, in the out-of-sample data partition, ARIMAX-WD model performs better with reasonable results (MAPE: 0.9, adj. R-sq.: 65.52%). Note that the model is also able to forecast a drastic fall and a recovery event in the very last part of the time-series shown in Figure 13.

Case III: Wind speed with a forecasting horizon of 1,008 samples (7 days).

(note that the time information on x-axis of the plots does not reflect the real date or time).

Model	Model architecture	In-sample		Out-of-sample	
		MAPE	**adj. R-sq.**	**MAPE**	**adj. R-sq.**
ARIMA	Differencing: (0) P: (1,2) Q: (1,2,3,4,5)	4.44	97.47%	40.30	11.30%
ARIMAX-WD	Differencing: (0) P: (1,2,3) Q: (1,2,3,4,5) Exogenous variable WD: (3,4,6,7,8,9,10,11,12,14)	2.57	99.18%	8.21	93.25%

Table 4. Comparison of forecasting performances of the two models in out-of-sample wind speed data

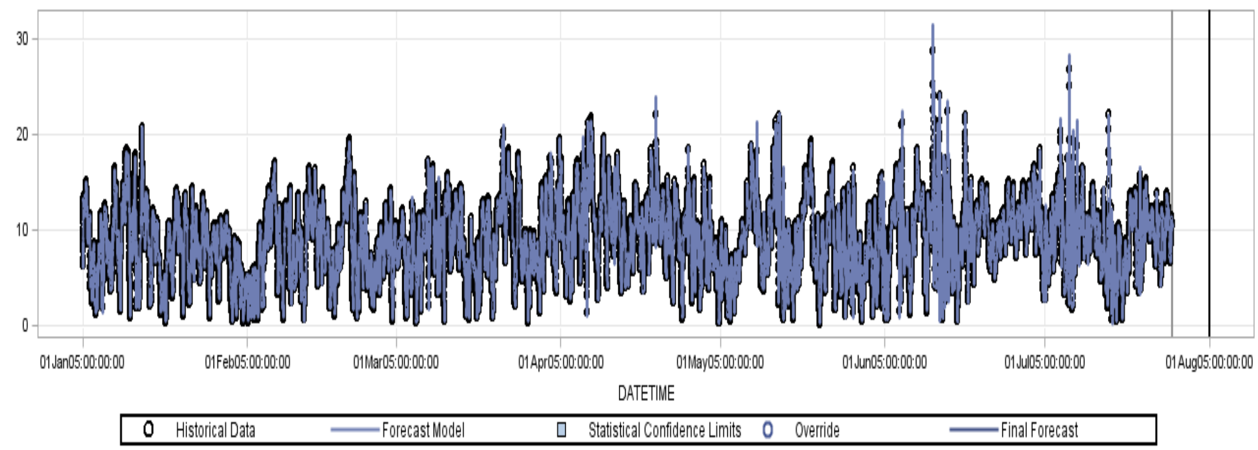

Figure 14. A plot between the actual and forecasted data using ARIMA model in wind speed data (in-sample)

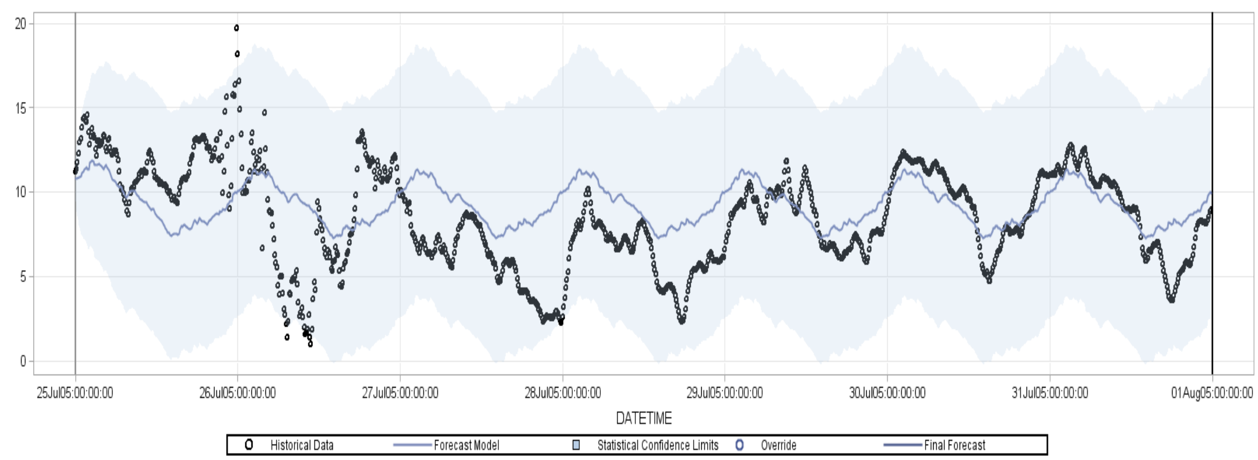

Figure 15. A plot between the actual and forecasted data using ARIMA model in wind speed data (out-of-sample)

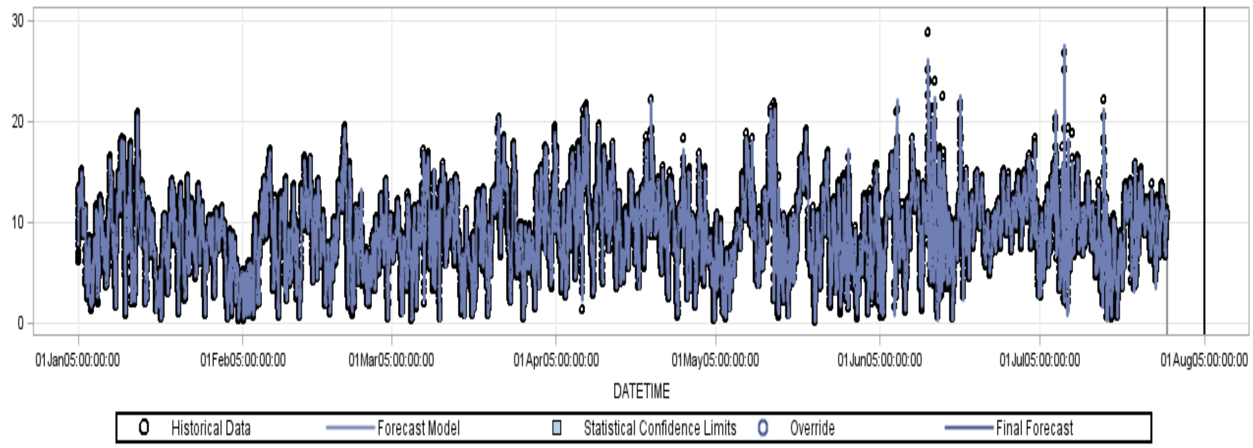

Figure 16. A plot between the actual and forecasted data using ARIMAX-WD model using in wind speed data (in-sample)

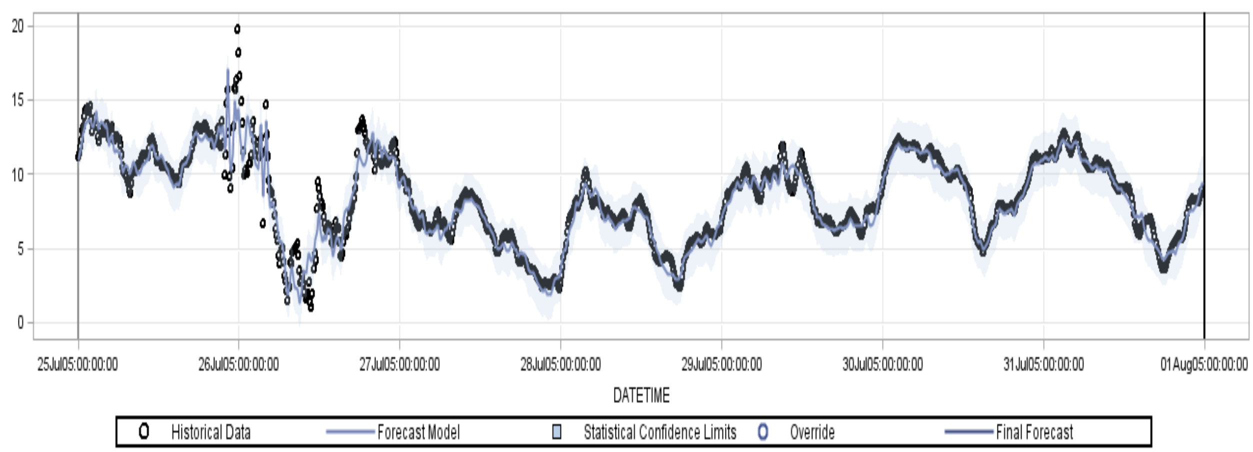

Figure 17. A plot between the actual and forecasted data using ARIMAX-WD model in wind speed data (out-of-sample)

In the wind speed forecasting case (forecasting horizon: 1,008 samples), although the ARIMA model performs well in the in-sample data partition (see Table 4 and Figure 14), it does poorly in the out-of-sample data partition (MAPE: 40.30, adj. R-sq.: 11.30%) (see Figure 15). However, the ARIMAX-WD still performs reasonably well in out-of-sample data partition with MAPE: 8.21 and adj. R-sq.: 93.25% (see Figure 17).

CONCLUSION

To be able to forecast accurately is an important task in all business domains. ARIMA model is one of the recognized time-series forecasting methods because of its solid foundation, and straightforward interpretation. Although there are many extensions available for this model such as seasonality adjustment methods, the model may not perform well to capture nonlinear patterns embedded in the data because of the ground of a linearity assumption.

To solve the problem, we proposed the method using a wavelet transformation to decompose a nonlinear time-series into less complex time-series based on their dominant frequencies. Then, use these time-series as explanatory variables in ARIMAX model. We demonstrate the usefulness of this method by comparison of the proposed model's performances to the performances of the ARIMA model (including seasonal adjustment method) in nonlinear data from several fields. The results suggest that the proposed

ARIMAX model using wavelet decomposed signals as explanatory variables (ARIMAX-WD) performs well in a long-term and nonlinear time-series forecasting application. Moreover, the wavelet decomposed time-series resulted from a decomposition also give insightful information about the underlying dynamics of the original time-series.

REFERENCES

1. Shim, I., J. Soraghan, and W. Siew, *Detection of Pd Utilizing Digital Signal Processing Methods. Part 3: Open-Loop Noise Reduction.* Electrical Insulation Magazine, IEEE, 2001. **17**(1): p. 6-13.
2. Chow, S.-M., E. Ferrer, and F. Hsieh, *Statistical Methods for Modeling Human Dynamics: An Interdisciplinary Dialogue*. 2012: Taylor & Francis
3. Mallat, S.G., *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1989. **11**(7): p. 674-693.
4. Gao, R.X. and R. Yan, *Wavelets: Theory and Applications for Manufacturing*. 2010: Springer Science & Business Media
5. SAS Institute Inc, *SAS/IML® 12.1 User's Guide*. 2012, Cary, NC: SAS Publishing.https://support.sas.com/documentation/cdl/en/imlug/65547/PDF/default/imlug.pdf
6. Box, G.E., G.M. Jenkins, and G. Reinsel, *Forecasting and Control.* Time-series Analysis, 1970. **3**: p. 75.
7. Andrews, B., M. Dean, R. Swain, and C. Cole, *Building Arima and Arimax Models for Predicting Long-Term Disability Benefit Application Rates in the Public/Private Sectors* 2013, Society of Actuaries.
8. SAS Institute Inc, *Sas High-Performance Forecasting 2.3: User's Guide*. 2007: SAS Publishing.https://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_913/hp_ug_10367.pdf
9. SAS Institute Inc, *Sas/Ets® 13.2 User's Guide. The Arima Procedure*. 2014: SAS Publishing.https://support.sas.com/documentation/onlinedoc/ets/132/arima.pdf
10. Laboratory, N.R.E., *Eastern Wind Dataset*. 2015: http://www.nrel.gov/electricity/transmission/eastern_wind_dataset.html.

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Goutam Chakraborty, Professor, Department of Marketing and founder of SAS and OSU Data Mining Certificate program, Oklahoma State University and my academic advisor, Dr. Sunderesh Heragu, Head and Regents Professor, Department of Industrial Engineering and Management, Oklahoma State University, for their support throughout the research.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Woranat Wongdhamma
Industrial Engineering and Management, Oklahoma State University
woranat@okstate.edu