Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 104 (2017) 578 – 585

Procedia Computer Science

ICTE 2016, December 2016, Riga, Latvia

# Nonlinear, Non-stationary and Seasonal Time Series Forecasting Using Different Methods Coupled with Data Preprocessing

Arthur Stepchenko[a,*], Jurij Chizhov[a], Ludmila Aleksejeva[a], Juri Tolujew[b]

[a]*Institute of Information Technology, Riga Technical University, 2 Daugavgrivas Str., Riga, LV-1048, Latvia*
[b]*Fraunhofer Institute for Factory Operation and Automation, Universitatsplatz 2, D-39106, Magdeburg, Germany*

**Abstract**

Time series forecasting is important in several applied domains because it facilitates decision-making in this domains. Commonly, statistical methods such as regression analysis and Markov chains, or artificial intelligent methods such as artificial neural networks (ANN) are used in forecasting tasks. In this paper different time series forecasting methods were compared using the normalized difference vegetation index (NDVI) time series forecasting. NDVI is a nonlinear, non-stationary and seasonal time series used for short-term vegetation forecasting and management of various problems, such as prediction of spread of forest fire and forest disease. In order to reduce input data set dimensionality and improve predictability, stepwise regression analysis and principal component analysis (PCA) were used as data pre-processing techniques. For comparing the obtained performance for the different methods, several performance criteria commonly used in forecasting statistical evaluation were calculated.


Peer-review under responsibility of organizing committee of the scientific committee of the international conference; ICTE 2016

*Keywords:* Artificial neural networks; Markov chains, Principal component analysis; Ridge regression; Stepwise regression

## 1. Introduction

The modelling and forecasting of time series is an important mathematical problem with many applications and it is important in many industries to make a better development and decision-making. The main aim of time series modelling is to collect and study the past observations of a time series to develop an appropriate model, which

* Corresponding author. Tel.: +371-26307683.
*E-mail address:* arturs.stepcenko@edu.rtu.lv


Peer-review under responsibility of organizing committee of the scientific committee of the international conference; ICTE 2016
doi:10.1016/j.procs.2017.01.175

describes the characteristics of the time series[1]. This model is then used to forecast future values for the time series. A time series is a sequential set of data points, measured typically over successive times. It is mathematically defined as a set of vectors $x(t)$, $t = 0,1,2,...$ where $t$ represents the time elapsed[1]. The measurements in a time series are arranged in a proper chronological order.

Real-world systems exhibit mostly nonlinear and nonstationary behaviours. A nonlinear time series is a signal coming from a nonlinear dynamic process. In other words, it is a partial solution of a nonlinear stochastic differential (or difference) equation[2]. Most real-world nonlinear dynamic systems also operate under transient (i.e., nonstationary) conditions, therefore nonstationary time series have time-changing statistics. At present, there are many objects described by time series containing unknown behavior trends, seasonal components, stochastic and random components, which significantly complicate acquisition of an effective predictive model.

The normalized difference vegetation index (NDVI) is a nonlinear, nonstationary and seasonal time series, which are developed for estimating vegetation cover from the reflective bands of satellite images. The NDVI is an indicator that quantifies the amount of green vegetation. The NDVI index is defined as:

$$NDVI = (NIR - R)/(NIR + R) \tag{1}$$

where *NIR* represents the spectral reflectance in near-infrared band and *R* represents reflectance in red light band [3]. The NDVI real values, by definition, would be between −1 and +1. The NDVI index is an important variable for vegetation forecasting and management of various problems, such as climate change monitoring, energy usage monitoring, managing the consumption of natural resources, agricultural productivity monitoring and drought monitoring and forest fire detection.

## 2. Data pre-processing techniques

Data pre-processing is an important step in the time series forecasting that prepares raw data for further processing. Data pre-processing steps include cleaning, normalization, feature extraction and feature selection.

### *2.1. Phase space reconstruction*

The fundamental starting point of many approaches in nonlinear data analysis is the construction of a phase space portrait of the considered system. A phase space (also called state space or lag space) of a dynamical system is a space in which all possible states of a system are represented, where each possible state is corresponding to one unique point in the multidimensional phase space.

The phase space of a dynamical system can be reconstructed using time-delayed versions of the original signal [4]. This new state space is commonly referred to in the literature as a reconstructed phase space (RPS). From the original time series $Y$ with length $N$:

$$Y = \{y(1), y(2),..., y(N)\}, \tag{2}$$

*i-th* state vector (or delay vector) of embedding dimension $m$ and time delay $\tau$ can be obtained by:

$$S_i = [y(t_i), y(t_i + \tau), y(t_i + 2\tau),..., y(t_i + (m-1)\tau)]. \tag{3}$$

Reconstructed phase space *PhS* then is obtained by[11]:

$$PhS = \begin{bmatrix} y(1) & y(1+\tau) & y(1+2\tau) & \dots & y(1+(m-1)\tau) \\ y(2) & y(2+\tau) & y(2+2\tau) & \dots & y(2+(m-1)\tau) \\ \dots & & & & \dots \\ y(M) & y(M+\tau) & y(M+2\tau) & \dots & y(M+(m-1)\tau) \end{bmatrix}, \quad (4)$$

where $M$ is the number of points (i.e. states) in reconstructed phase space. Therefore, a sequence of scalar time series measurements in time is converted into a sequence of $m$-dimensional state vectors.

### *2.2. Stepwise regression*

Stepwise regression is a sequential feature selection method that is used in the exploratory stages of model building to identify a useful subset of predictors. Initial stepwise regression model includes a single independent variable that has the largest absolute t-test value. T-test is used in order to determine if two sets of data are significantly different from each other. In the next step, a second variable is added and a new model is created. If the t-test values with the new model are better than the first model, the new model is kept and a third variable is added. If the new model performs worse (i.e. none of the absolute t-test values are significant) compared to the first one, the first variable is discarded, the second variable is kept and the next model is created that contains the second and the third variable[5]. This procedure repeats until all two variable combinations are tested, the best performing two-variable combination is selected as the final model before a third variable is added. The process ends when all significant variables are included in the model.

### *2.3. Principal component analysis*

Principal component analysis (PCA) is a statistical feature extraction method that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The main purposes of a principal component analysis are the analysis of data to identify patterns and finding patterns to reduce the dimensions of the dataset with minimal loss of information. The first step in the PCA algorithm is to normalize the components using z-score normalization so that they have zero mean and unity variance. Then, an orthogonalization method is used to compute the principal components of the normalized components[6]. The principal components are orthogonal because they are the eigenvectors ($e_1$,$e_2$,…,$e_d$) of the sample covariance matrix, which is symmetric. Each of those eigenvectors is associated with an eigenvalue ($l_1$,$l_2$,…,$l_d$). We might be interested in keeping only those eigenvectors with the much larger eigenvalues, since they contain more information about our data distribution. Eigenvalues that are close to zero are less informative, and corresponding eigenvectors can be removed from the dataset.

## 3. Time series forecasting techniques

In this paper, three time series forecasting techniques are discussed. These techniques are discrete time, continuous state *m-th* order Markov chains, ridge regression and layer recurrent neural networks.

### *3.1. Discrete time, continuous state m-th order Markov chains*

A Markov chain is a stochastic process $X = \{X_n; n = 0, 1, \dots\}$ that operates sequentially, transitioning from one state to another on a state space[7]. A Markov chain consists of a state space $S$, which is a set of values that the chain can take and a transition operator that defines the probability of moving from one state to another. A Markov chain can have a discrete or continuous (i.e., uncountable) state space that exists in the real numbers.

First-order Markov chain next state probability depends only on the current state. Higher-order Markov chain is a random process, in which the next state probability depends not only on the current, but also on the sequence of several previous states (history)[8]. The amount of states in history is the order of the Markov chain.

*3.2. Multilinear ridge regression by regularized least-squares*

Ridge regression is a regression technique used to attempt to solve some of the problems of ordinary least-squares by imposing a penalty on the side of the coefficients. Ridge regression solution in matrix notation is obtained by:

$$\hat{\beta} = \left(X^T X + \lambda I\right)^{-1} X^T Y \ , \tag{5}$$

where $X$ is matrix of predictors, $Y$ is vector of observed responses, $\lambda$ is the ridge parameter (penalty term) and $I$ is the identity matrix [9]. Small positive values of $\lambda$ improve the conditioning of the problem and reduce the variance of the estimates. While biased, the reduced variance of ridge estimates often results in a smaller mean square error when compared to ordinary least-squares estimates.

*3.3. Artificial neural networks*

Artificial neural networks (ANN's) are a form of artificial intelligence, which are trying to mimic the function of real neurons found in the human brain. Artificial neural networks are self-adaptive methods that learn from data and can find functional relationships among the data even if relationships are unknown or the physical meaning is the difficult[10]. Neural networks are less sensitive to error term assumptions and they can tolerate noise and chaotic components better than most other methods. Artificial neural networks also are universal function approximators.

The scalar weights along with the network architecture store the knowledge of a trained network and determine the strength of the connections between interconnected neurons. The Levenberg-Marqardt backpropagation algorithm with Bayesian regularization is a neural network training function that updates the weight and bias values according to Levenberg-Marquardt optimization. It minimizes a combination of squared errors and weights, and then determines the correct combination to produce a network that generalizes well. The objective of neural network training is to reduce the global error determined by performance function. The following performance (cost) function is used for Bayesian regularization[12]:

$$MSE_{reg} = \gamma \frac{1}{N} \sum_{i=1}^{N} (e_i)^2 + (1-\gamma) \frac{1}{n} \sum_{j=1}^{n} w_j^2 \ , \tag{6}$$

where $\gamma$ is the performance ratio, $e$ is the error vector between observed and predicted values, $w$ is the weight and bias variable vector. Minimizing performance function (6) will cause the network to have smaller weights and biases, and this will force the network response to be smoother and less likely to overfit.

A recurrent neural network (RNN) is a class of artificial neural networks where connections between units form a directed cycle. This allows it to exhibit dynamic temporal behaviour; therefore, recurrent neural networks are powerful sequence learners. The layer recurrent neural network (LRNN) is a dynamic recurrent neural network that has feedback loops at every layer, except the output layer. The most important advantage of the LRNN is a robust feature extraction ability cause context layer store useful information about data points in past.

## 4. Experimental analysis and results

The objective of this experiment is to present a comparison about the three approaches for the NDVI time series forecasting: Markov chains, ridge regression analysis and layer-recurrent neural networks where each of them is combined with data pre-processing methods.

### *4.1. Data set*

Multi-temporal, smoothed NDVI composite data obtained from MODIS Terra (NASA research satellite) with spatial resolution 250 m and produced on 7-day intervals were used in this study. Data are obtained from data service platform for MODIS Vegetation Indices time series processing.

One pixel from these images was chosen as test site. The NDVI data set for corresponding pixel consists of 814 observations that obtained every 7 days over 15 years (see Fig. 1).

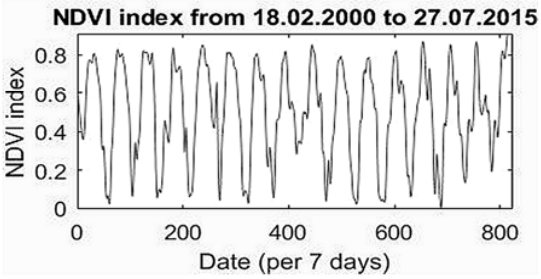


Fig. 1. Smoothed NDVI time series from 18.02.2000 to 27.07.2015.

The NDVI time series data provide a seasonal trajectory – time series show obvious seasonal oscillations, which correspond to the vegetation phenological cycles where maximum NDVI values are observed between May and August. Variations in the NDVI values are seen to be -0.0050 to 0.9109 units, mean value is 0.4965 and standard deviation is 0.2492.

### *4.2. Data pre-processing procedure*

Initially, phase space was reconstructed from observed NDVI time series with embedding dimension 100 and time delay 1. Then z-score normalization was applied to each column (feature) of reconstructed phase space, and initial input data set was created.

Stepwise regression was applied to this initial input data set in order to reduce input data dimensionality and improve predictability of the forecasting techniques used in this study. The maximum p-value to add a feature in the model was set to 0.05, and the minimum p-value to remove a feature from the model was set to 0.05. The data set was divided into training set and test set, 70% of observations (500 observations) were used for training set and 15% of observations (107 observations) were used for validation. Last 15% of data (test set) were not used in this stage. After this stage, reduced input data set that minimizes root mean squared error was created.

Finally, the principal component analysis method was applied to the reduced input data set and linearly uncorrelated data set was obtained that contained 13 features. The obtained data set is then used as input data set for ridge regression and artificial neural networks. For Markov chains, pre-processing procedure was a little different.

### *4.3. Case I: Markov chains*

In our constructed Markov chain, each state is equal to real number that time series observation can take. However, we used *m-th* order Markov chain, and therefore last *m* states or last *m* time series observations were used in order to forecast the next state. Geometrically, the current state vector $S_N$ is a point in the *m*-dimensional phase space. Neighbouring points in this space represent similar state vectors. It can be assumed that similar conditions produce a similar probability distribution. By $\Phi_\varepsilon(S_N)$ is marked a neighbourhood of small diameter ε around the vector $S_N$.

In the experiment with discrete time, continuous state *m-th* order Markov chains, data pre-processing was performed parallel with searching for optimal diameter that minimizes root mean squared error. Here, Euclidean distance was used as a diameter. The searching interval was [0.01, 1], and using cross-validation it was found that

optimal diameter ε = 0.06, but optimal number of features after applying stepwise regression and the PCA algorithm was two. Therefore, Markov chain input data set dimensionality was two, and second order Markov chain was used in this study.

The number of vectors $S_k$ in this neighborhood taken from the past values of the time series, $k<N$, is marked as $|\Phi_\varepsilon(S_N)|$. For these vectors the future values $S_{k+1}(y(t_{k+1}+(m-1)\tau))$ were examined and their number in the interval is marked as $N(v)$. The optimal prediction is given by the first moment of conditional probability calculated by:

$$E[v] \approx \frac{1}{\left|\phi_\varepsilon(S_N)\right|} \sum_{k \in \phi_\varepsilon(S_N)} S_{k+1}(y(t_{k+1} + (m-1)\tau)) \,. \tag{7}$$

The first moment of conditional probability is an ensemble average value of current state $S_N$ possible future values. It minimizes the root mean squared error in maximum likelihood manner.

### *4.4. Case II: ridge regression*

In case of experiment with ridge regression as input data set a data set was used that was obtained after pre-processing procedure, and the only parameter optimized was regularization or ridge parameter $\lambda$. Optimal $\lambda$ that minimizes root mean squared error was searched using cross-validation in [-1,1]. It was found that optimal $\lambda$ was zero, and therefore equation (5) was reduced to ordinary least squares.

### *4.5. Case III: artificial neural networks*

The layer recurrent neural network model used in this study was trained by Levenberg-Marqardt backpropagation algorithm with the Bayesian regularization. Neural network weights and biases were initialized with small random numbers in [-0.1,0.1]. The number of network hidden layers was one. The hyperbolic tangent function and a linear function were used as activation functions for the hidden and output layers, respectively. The number of epochs that are used to train was set to 10,000. As the number of hidden neurons is an important factor determining the forecasting accuracy, it is required to find an optimal value, but there is currently no theory to determine how many nodes in the hidden layer are optimal. The optimal complexity of LRNN model, that is, the number of hidden nodes, was determined by a trial-and-error approach. In the present study, the number of hidden nodes was searched in[1,13], i.e., between one and input data dimensionality.

### *4.6. Results*

To evaluate forecast accuracy as well as to compare among different models fitted to a time series, we have used the four error performance measures - the square root mean squared error (RMSE), mean absolute percentage error (MAPE), directional symmetry (DS) and the adjusted coefficient of multiple determination ($R_{adj}^2$). The root mean squared error is a measure of the differences between the values predicted by a model and the values actually observed, and is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N}} \,, \tag{8}$$

where, $\hat{y}_i$ – forecasted value, $y_i$ – observed value, $N$ – number of observations. The mean absolute percentage error measures the size of the error in percentage terms and is set by:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|100 \ . \tag{9}$$

Directional symmetry is a statistical measure of a model performance in forecasting the direction of change, positive or negative, of a time series from one period to the next:

$$DS = \frac{100}{N-1}\sum_{i=2}^{N}\begin{cases}1, if\ (y_i - y_{i-1})(\hat{y}_i - \hat{y}_{i-1}) \geq 0\\ 0, else\end{cases}. \tag{10}$$

The adjusted coefficient of multiple determination shows how well a regression model fits the data and is given by:

$$R_{adj}{}^2 = 1 - \left(\frac{N-1}{N-p}\right)\frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}, \tag{11}$$

where $p$ is the number of model input parameters (features) and $\bar{y}$ is the mean of the observed values.

Validation set was used in order to find optimal parameters for each forecasting technique, but test set was used to check obtained models on new data that earlier were not used. Table 1 presents the results of forecasting by discrete time, continuous state second order Markov chain, ridge regression and the layer recurrent neural network.

Table 1. Forecasting results.

| | RMSE | | MAPE | | DS | | $R_{adj}{}^2$ | |
|---|---|---|---|---|---|---|---|---|
| | Validation | Test | Validation | Test | Validation | Test | Validation | Test |
| Markov chain | 0.0291 | 0.0277 | 27.2361 | 4.9372 | 80.9917 | 87.6033 | 0.9883 | 0.9803 |
| Ridge regression | 0.0071 | 0.0057 | 4.3829 | 1.0914 | 97.1962 | 91.4285 | 0.9991 | 0.9990 |
| The LRNN | 0.0059 | 0.0056 | 3.2300 | 1.0896 | 95.7627 | 94.9579 | 0.9982 | 0.9973 |

From Table 1 it can be seen that the best method to forecast the NDVI index is the layer recurrent neural network, because the RMSE and MAPE errors of this method are lower than those of other methods and DS statistics are higher than DS statistics of other methods; while the ridge regression outperforms discrete time, continuous state second order Markov chain.

## 5. Conclusion

In this paper, one-step-ahead predictions of the normalized difference vegetation index (NDVI) are obtained using a layer recurrent neural network, ridge regression and discrete time, continuous state second order Markov chain, and a comparison is made between accuracy of these methods. Phase space reconstruction, stepwise regression and principal component analysis were used as data pre-processing techniques. The layer recurrent neural network outperforms both other methods. This is evident because the layer recurrent neural network has a "deeper memory" than other methods in this study and is a more powerful sequence learner. The study concludes that the forecasting abilities of a regularized LRNN in combination with phase space reconstruction, stepwise regression and the principal component analysis provide a potentially very useful system for the NDVI time series forecasting.

## References

1. Sarwar U, Muhammad MB, Karim ZAA. Time Series Method for Machine Performance Prediction Using Condition Monitoring Data. In: *Proceedings of a 1st International Conference on Computer, Communications, and Control Technology.* IEEE; 2014. p. 394-399.
2. Fan J, Yao Q. *Nonlinear Time Series.* Vol. 2, New York: Springer; 2002.

3. Sahebjalal E, Dashtekian K. Analysis of land use-land covers changes using normalized difference vegetation index (NDVI) differencing and classification methods. *African J. Agricultural Research.* Vol. 8 (37); 2013. pp. 4614-4622.
4. Klikova B, Raidl A. Reconstruction of Phase Space of Dynamical Systems Using Method of Time Delay. In: *WDS'11 Proceedings of Contributed Papers*: *Part III*. Prague: Matfyzpress; 2011. p. 83-87.
5. Templ M, Kowarik A, Filzmoser P. Iterative stepwise regression imputation using standard and robust methods. *J. of Computational Statistics and Data Analysis*. Vol. 55 (10); 2011. p. 2793-2806.
6. Saleh JM, Hoyle BS. Improved Neural Network Performance Using Principal Component Analysis on Matlab. *Int. J. of The Computer, the Internet and Management.* Vol. 16 (2); 2008. p. 1-8.
7. Watthayu W. Loopy Belief Propagation: Bayesian Networks for Multi-Criteria Decision Making (MCDM). *Int. J. of Hybrid Information Technology*. Vol. 2 (2); 2009. p. 141-152.
8. Soloviev V, Saptsin V, Chabenko D. Markov Chains Application To The Financial-Economic Time Series Prediction. *Computer Modelling and New Technologies.* Vol. 14 (3); 2011. p. 16 – 20.
9. Marquardt DW, Snee R. Ridge Regression in Practice. *The American Statistician*. Vol. 29 (1); 1975. p. 3-20.
10. Shabri A, Samsudin R. Daily crude oil price forecasting using hybridizing wavelet and artificial neural network model. *Mathematical Problems in Engineering;* 2014. p. 1-10.
11. Zhang G, Patuwo BE, Hu MY. Forecasting with artificial neural networks: the state of the art. *Int. J. of Forecasting*. Vol. 14 (1); 1998. p. 35-62.
12. Fernandez M, Caballero J, Fernandez L, Sarai A. Genetic algorithm optimization in drug design QSAR: Bayesian-regularized genetic neural networks (BRGNN) and genetic algorithm-optimized support vectors machines (GA-SVM). *Mol Divers*. Vol. 15; 2011. p. 269-289.

Arthur Stepchenko is a third-year Doctoral Student majoring in Information Technology at Riga Technical University (RTU). He received his Mg. sc. comp. degree in Computer Science from Ventspils University College (VUC) in 2013. He is a Research Assistant at the Department of Space Technology of the Engineering Research Institute "Ventspils International Radio Astronomy Centre", Ventspils University College (VIRAC). His research interests include machine learning, discrete signal processing, programming and digital image processing. Contact him at arturs.stepcenko@edu.rtu.lv.

Jurij Chizhov is currently a Lecturer and Leading Researcher at the Department of Modelling and Simulation, Riga Technical University (RTU). He received his Dr. sc. ing. from Riga Technical University in 2012. His major research interests include a number of techniques related to computational intelligence, in particular, cluster analysis, ontology building, evolutionary algorithms, reinforcement learning, artificial neural network. Now he focuses on the learned artificial neural network simplification and translating to a set of rules. Contact him at jurijs.cizovs@rtu.lv.

Ludmila Aleksejeva received her Dr. sc. ing. degree from Riga Technical University (RTU) in 1998. She is an Associate Professor at the Department of Modelling and Simulation, Riga Technical University. Her research interests include decision-making techniques and decision support system design principles, as well as data mining methods and tasks, and especially collaboration and cooperation of the mentioned techniques. She has more than 60 academic and scientific publications. Contact her at ludmila.aleksejeva_1@rtu.lv.

Juri Tolujew is a project manager in the Department for Logistic Systems at the Fraunhofer Institute for Factory Operation and Automation IFF in Magdeburg, Germany. He received a Ph.D. degree in automation engineering in 1976 from the University of Riga. He also received a habil. degree in computer science from the University of Magdeburg in 2001. His research interests include the simulation-based analysis of production and logistics systems using decision-making procedures. Contact him at juri.tolujew@iff.fraunhofer.de.