Normalized difference vegetation index forecasting using a regularized layer recurrent neural network

Arthur Stepchenko
Faculty of Computer Science and Information Technology
Riga Technical University
Riga, Latvia
arturs1312@gmail.com

Abstract— **In this paper predictions of the normalized difference vegetation index (NDVI) are discussed. Time series of Earth observation based estimates of vegetation inform about changes in vegetation. NDVI is an important parameter 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, drought monitoring and forest fire detection. Artificial Neural Networks (ANN's) are computational models and universal approximators, which are widely used for nonlinear, non-stationary and dynamical process modeling and forecasting. A layer recurrent neural network (LRN) is used in this paper to make one-step-ahead prediction of the NDVI time series.**

Keywords- layer recurrent neural networks, normalized difference vegetation index, time series forecasting

I. INTRODUCTION

Human activities reflect on ecosystems, including the natural vegetation cover. Vegetation cover change is important factor that reflect on ecosystem condition and function. A change of vegetation cover may have long-term influence on sustainable food production, freshwater and forest resources, the climate and human welfare. Monitoring and forecasting changes occurring in vegetation cover at periodic intervals is very important to providing information about the stability of vegetation.

The use of satellite-based remote sensing data as a cost-effective technique has been widely applied to develop land cover coverages over large geographic regions. Vegetation cover is an important part of land cover. Change detection has become an outspread application of remotely sensed data because of repetitive wide coverage, short revisit intervals and good image quality. Change detection is the process of identifying differences in the state of an object or phenomenon by observing it at different times. The main precondition in using remote sensing data for vegetation change detection is that changes in land cover result in changes in radiance values and changes in radiance due to land cover change are large with respect to radiance change caused by others factors such as differences in atmospheric conditions, differences in soil moisture and differences in sun angles [1].

Vegetation indices calculated from satellite images can be used for monitoring temporal changes related to vegetation. Vegetation indices (VI's) are combinations of surface reflectance intended to take out a specific property of vegetation. Each of the VI's is designed to accent a specific vegetation property. Analyzing vegetation using remotely sensed data requires knowledge of the structure and function of vegetation and its reflectance properties. This knowledge enables linking together vegetative structures and their condition to their reflectance behavior in an ecological system of interest [2]. The normalized difference vegetation index (NDVI) is designed for estimating vegetation cover from the reflective bands of satellite data. The NDVI is an indicator, which numerically determines the amount of green vegetation. Past studies have demonstrated the potential of using NDVI to study vegetation dynamics. The NDVI index is defined as:

$$NDVI = (NIR - R)/(NIR + R), \quad (1)$$

where *NIR* represents the spectral reflectance in near infrared band and *R* represents red band in satellite images. Greener and dense vegetation has low red light reflectance and high near infrared reflectance, and therefore high NDVI values. The NDVI values are normalized between -1 and +1, where increasing positive values indicate increasing green vegetation, but low positive values and negative values indicate non-vegetated surface features such as water, barren land, rock, ice, snow, clouds or artificial materials [3]. The NDVI also has the ability to reduce external noise factors such as topographical effects and sun-angle variations.

Time series analysis of remotely sensed data has gained wide usability supported by availability of wide-coverage, high temporal satellite data. Univariate autoregressive integrated moving average (ARIMA) models are widely used for a univariate time series forecasting, also for the NDVI time series [4]. However, these models are parametric and are based on the assumption that the time series been forecasted are linear and stationary. The difficulty of forecasting arises from the imprescriptible non-linearity and non-stationarity in the NDVI time series. Many previous studies propose that non-linear machine learning approaches such as artificial neural network (ANN) models perform better than traditional time series linear models with minimum initial assumptions and high forecasting accuracy. In addition, ANN has also been shown to be effective in modeling and forecasting nonlinear time series with noise. Therefore, neural networks are used as an alternative to traditional statistical forecasting methods.

II. Study Area and Character of the Data

A. Study Area

Ventspils Municipality is located in the western part of Courland, Latvia with total area of 2472 km^2 (Fig. 1).

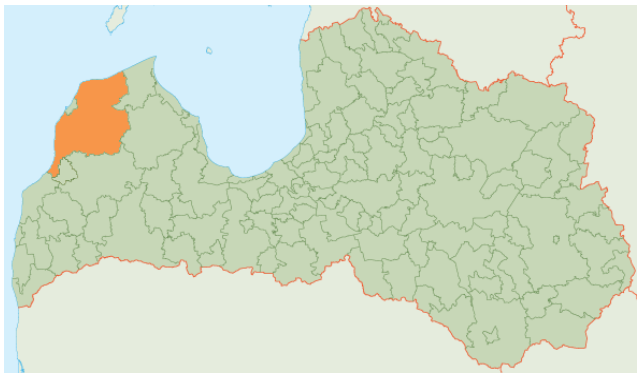

Figure 1. Location of the Ventspils Municipality.

Area by size 250 meters x 250 meters from Ventspils Municipality was selected as test site (Fig. 2).

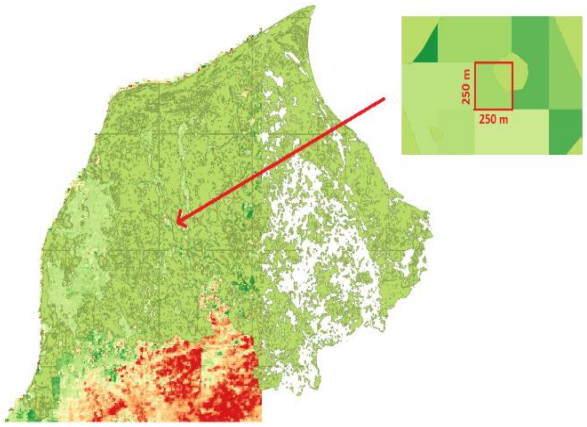

Figure 2. Test site in Ventspils Municipality.

Approximately half of the test area is covered by forests; the other half is covered by agricultural lands.

B. The NDVI Data Set

Multi-temporal 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 [5]. Used data are smoothed and gap-filled using the Whittaker smoothing algorithm with smoothing parameter $\lambda=15$ and two filtering iterations [6]. Iterative filtering was used, because undetected clouds and poor atmospheric conditions decrease the observed NDVI values.

The NDVI data set consists of 814 smoothed NDVI images that obtained every 7 days over 15 years. NDVI values of these images were obtained for corresponding test site and used as NDVI time series (Fig. 3).

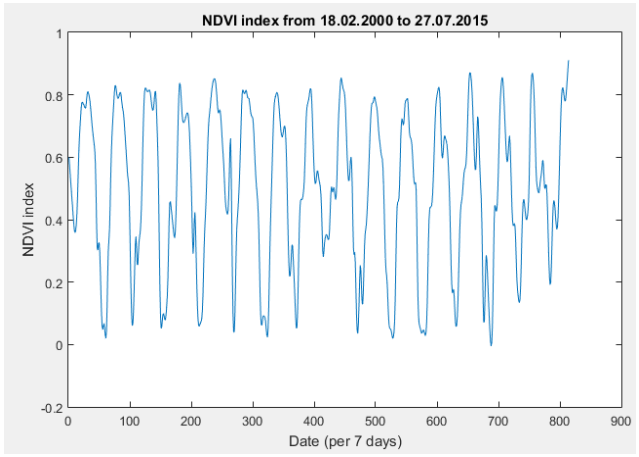

Figure 3. Smoothed NDVI time series from 18.02.2000 to 27.07.2015.

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. NDVI trends are not always monotonic but can change. A positive trend can change for example into a negative one and reversely.

III. Artificial Neural Networks

Artificial neural networks (ANN's) are a form of artificial intelligence, which are trying mimic the function of real neurons found in the human brain [7]. ANN's are one of the most accurate and widely used forecasting models that have used in forecasting social, economic, business, engineering, foreign exchange, stock problems and other. Structure of artificial neural networks make them valuable for a forecasting task with good accuracy.

As opposed to the traditional model-based empirical and statistical methods such as regression and Box-Jenkins approaches, which need prior knowledge about the nature of the relationships between the data, artificial neural networks are self-adaptive methods that learn from data and there about the problem only few a priori assumptions are needed [8].

Neural networks learn from examples and can find functional relationships among the data even if relationships are unknown or the physical meaning is the difficult [7]. Therefore, ANN's are well suited for problems whose solutions require knowledge that is difficult to specify but for which there are enough data or observations.

Artificial neural networks can generalize. After learning the input data (a sample or pattern), ANN's can often correctly processing the early unseen sample even if the sample data are noisy. 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. It was proved that a neural network can approximate any continuous function with any accuracy [8].

For a time series forecasting problem, a training patterns consists of a history data with fixed number of observations. If time series contains N observations $y_1, y_2, \ldots, y_N$, then using an ANN with n input nodes, we have N-n training patterns than can be used for short-term forecasting – one value ahead. The first training pattern will be contain $y_1, y_2, \ldots, y_n$ as inputs and y_{n+1} as the output. The second training pattern will contain $y_2, y_3, \ldots, y_{n+1}$ as inputs and y_{n+2} as the output. The last training pattern will be contain $y_{N-n}, y_{N-n+1}, \ldots, y_{N-1}$ inputs and y_N as the output. Then pattern $y_{N-n+1}, y_{N-n+2}, \ldots, y_N$ will be used to get forecasting value y_{N+1}. The ANN performs the following unknown function mapping:

$$y_{t+1} = f\left(y_t, y_{t-1}, \ldots, y_{t-p}\right), \qquad (2)$$

where y_t is the observation at time t [8].

ANN's structure include input data and artificial neurons that are known as „units". The multilayer perceptron include an input layer, an output layer and one or more intermediate layers called hidden layers. The size and nature of the data set affect the number of hidden layers and neurons within each layer. Usually ANN's with one or two hidden layers perform better than neural networks with the large number of hidden layers.

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. If weight value is zero then there is no connection between two neurons and if weight value is negative then relationship between two neurons is a prohibitive. An individual processing element receives weighted inputs from previous layers, which are summed in each node using a combination function, and a bias neuron, which is connected to every hidden or output unit, is added.

The result of this combined summation is passed through a transfer function to produce the nodal output of the processing element, which is weighted and passed to processing element in the next layer [7]. The combination function and transfer function together constitute the activation function. In the majority of cases input layer neurons do not have an activation function, as their role is to transfer the inputs to the hidden layer. The most widely used activation function for the output layer is the linear function as non-linear activation function may introduce distortion to the predicated output. The sigmoid (logistic), exponential (hyperbolic) tangent, quadratic or linear functions are often used as the hidden layer transfer function. The relationship between the output – predicted value (y_t) and the inputs – past observations of the time series ($y_{t-1}, \ldots, y_{t-p}$) is given by:

$$y_t = w_0 + \sum_{j=1}^{q} w_j f\left(w_{0,j} + \sum_{i=1}^{p} w_{i,j} y_{t-i}\right) + \varepsilon_t, \qquad (3)$$

where w_j are weights between hidden and output layer, $w_{i,j}$ are weights between input and hidden layer, f is an activation function, q is the number of hidden nodes, p is the number of input nodes, ε_t is random error at time t.

A. A Layer Recurrent Neural Network

A recurrent neural network (RNN) is a class of artificial neural networks where connections between units form a directed cycle. This creates an internal state of the network, which allows it to exhibit dynamic temporal behavior [9]. Recurrent neural networks can use their internal memory to process arbitrary sequences of inputs. Therefore, recurrent neural networks are powerful sequence learners.

The layer recurrent network (LRN) is a dynamic recurrent neural network that was developed using earlier introduced neural network by Elman [10]. The layer recurrent neural network has feedback loops at every layer, except the output layer. Feedback connection in the layer recurrent neural network is connection from the outputs of neurons in the hidden layer to neurons in the context layer that store the delayed hidden layer outputs. The most important advantage of the LRN is a robust feature extraction ability cause context layer store useful information about data points in past.

The LRN generalizes the Elman network to have an arbitrary number of layers and to have arbitrary transfer functions in each layer. The LRN can be trained using exact versions of standard backpropagation algorithm [11]. The original Elman network was trained using an approximation to the backpropagation algorithm.

B. Levenberg-Marqardt Backpropogation Algorithm with Bayesian Regularization

A neural network is trained with patterns that consists of input and target pairs. The process of training a neural network includes tuning the values of the weights and biases of the network to optimize network performance, as defined by the network performance function. The problem of neural network learning can be seen as a function optimization problem, where the best network parameters (weights and biases) need to be obtained in order to minimize the network's global error. Traditionally gradient descent backpropogation algorithm is used for neural network training. Gradient descent backpropagation algorithm updates weights and biases in the direction of the negative gradient of the performance function. There are some major disadvantages of gradient descent approach, one of them is stucking into local minima, and another problem is very slow convergence of the learning algorithm. Regarding to this issues, there are some more methods available to use in aid of standard back-propagation learning, such as the Levenberg-Marqardt backpropagation algorithm with Bayesian regularization.

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 [12]. It minimizes a combination of squared errors and weights, and then determines the correct combination to produce a network that generalizes well. The process is called Bayesian regularization.

The Levenberg-Marqardt backpropagation algorithm is quasi-Newton method that use an approximate Hessian matrix. The Hessian matrix is matrix that contains second-order partial derivatives of the neural network errors with respect to the

weights and biases [13]. The Levenberg–Marquardt algorithm introduces approximation to Hessian matrix *H*:

$$H \approx J^T J + mI \,, \tag{4}$$

where *J* is the Jacobian matrix, I is the identity matrix and *m* is combination coefficient, that is always positive. Jacobian matrix contains first-order partial derivatives of the neural network errors with respect to the weights and biases. It is *N-by-n* matrix, where *N* is the number of the neural network training patterns and *n* is the number of weight and bias variables. The elements on the main diagonal of the approximated Hessian matrix always will be larger than zero. Therefore, with approximation (4) matrix *H* is always invertible. Gradient of the performance function with respect to the weights and biases can be computed:

$$g = J^T e \,, \tag{5}$$

where *e* is a vector of the neural network errors. The length of the gradient vector is equal to the number of all weights and biases in the network. Error vector *e* of the length equal to the number of patterns can be computed:

$$e = d - o, \tag{6}$$

where *d* is the desired (observed) output vector and *o* is the actual (predicted) output vector. Levenberg-Marquardt back-propagation is used to calculate changes in weight and bias variables:

$$dw = -H^{-1} g \,, \tag{7}$$

where *dw* is a vector of changes in weight and bias variables [12]. The Levenberg–Marquardt optimization technique is more powerful than the conventional gradient descent techniques. However, the Levenberg-Marquardt is very sensitive to the initial network weighs. In addition, it does not consider outliers in the data, what may lead to overfitting. Overfitting appear when a statistical model describes random error or noise instead of the true function. To avoid those situations, a regularization can be used.

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 [14]:

$$MSE_{reg} = \gamma \frac{1}{\mathrm{N}} \sum_{i=1}^{N} (e_i)^2 + (1-\gamma) \frac{1}{n} \sum_{j=1}^{n} w_j^2 \,, \tag{8}$$

where γ is the performance ratio. Weight and bias variables *w* are updated with the following Newton-like update:

$$w(k+1) = w(k) + dw, \tag{9}$$

where *k* is a training iteration (epoch) index. Minimizing performance function (8) 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, because the Bayesian regularized artificial neural networks calculates and trains on a number of effective network parameters or weights, effectively turning off those that are not relevant. This effective number is usually considerably smaller than the number of weights in a standard fully connected back-propagation neural net. With regularization, neural network should be able to sufficiently represent the true function.

IV. Experimental Procedure

The aim of this experiment is to investigate the capability and accuracy of layer recurrent neural networks in the NDVI time series forecasting.

The data set was divided into three sets, training, validation and testing data set by 70/15/15 principle, namely, 70% of the NDVI data (a total of 568 observations) were used as a training data set, 15% of the NDVI data (a total of 122 observations) were used as a validation data set and the remaining 15% of the NDVI data (a total of 122 observations) were used as a testing data set.

LRN model used in this study was trained by Levenberg-Marqardt backpropagation algorithm with the Bayesian regularization. Neural network's weights and biases were initialized with random numbers in [-1,1]. The number of network's hidden layers was one. The hyperbolic tangent function and a linear function are used as activation functions for the hidden and output layers, respectively. The number of epochs that are used to train was set to 1000. As the number of hidden neurons is an important factor that determining the forecasting accuracy, is required to find an optimal value, but there is currently no theory to determine how many nodes in the hidden layer are optimal. Alike optimal number of input values (e.g. past values of the NDVI time series) need to be found. The optimal complexity of LRN model, that is, the number of input values and the number of hidden nodes, was determined by a trial-and-error approach. In the present study, the number of input values was progressively increased from 1 to 10 and the number of hidden nodes was progressively increased from 1 to 20.

In order to improve neural netwok generalization ability early stopping technique was used. When the network begins to overfit the data, the global error on the validation set typically begins to rise. When the validation error increased for a 10 epochs in a row, the training was stopped, and the weights and biases at the minimum of the validation error were used. It is often useful to examine the network response in more detail. A linear regression analysis between the network response and the corresponding targets was used in order to improve forecasting accuracy. This neural network's configuration was determined experimentally as giving the best results. A program code was written in MATLAB enviroment.

As performance criteria there were chosen the square root of the mean of the square of all of the errors (RMSE), mean absolute percentage error (MAPE), directional symmetry (DS) and the adjusted coefficient of multiple determination (R_{adj}^2).

The square root of the mean of the square of all of the errors (RMSE) is a measure of the differences between values predicted by a model and the values actually observed and is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}, \quad (10)$$

where, $\hat{y}_i$ – forecasted value, y_i – observed value, N – number of observations.

The MAPE (mean absolute percentage error) measures the size of the error in percentage terms and is given by:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| 100. \quad (11)$$

Directional symmetry (DS) is a statistical measure of a model's performance in forecasting the direction of change, positive or negative, of a time series from one period to the next and is given by:

$$DS = \frac{100}{N-1}\sum_{i=2}^{N} d_i, \quad (12)$$

where,

$$d_i = \begin{cases} 1, if\ (y_i - y_{i-1})(\hat{y}_i - \hat{y}_{i-1}) \geq 0 \\ 0, else \end{cases}. \quad (13)$$

Directional symmetry statistic gives the percentage of events in which the sign of the change in value from one period to the next is the same for both the actual and forecasted time series.

The adjusted coefficient of multiple determination (R_{adj}^2) shows how well a regression model fits the data and it lying within a range from [0,1]. A perfect fit would result in an R_{adj}^2 value of one, a very good fit near one, and a very poor fit at zero. The formula used for R_{adj}^2 is given by:

$$R_{adj}^2 = 1 - \left(\frac{N-1}{N-p}\right)\frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}, \quad (14)$$

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

V. RESULTS

In the several experiments were found, that optimal number of input data is eight past values of the NDVI time series and optimal number of hidden nodes is two. Optimal LRN topology is shown in Fig. 4.

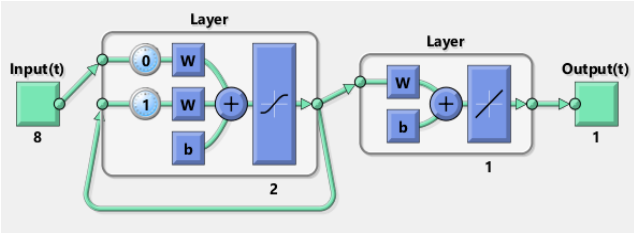

Figure 4. Optimal LRN topology.

LRN convergence for best model was obtained after 61 epochs (Fig. 5).

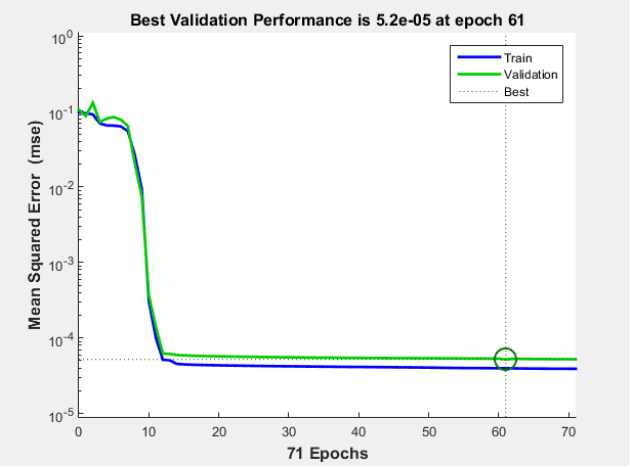

Figure 5. LRN convergence.

Tab. I shows the performance of the best LRN model on the NDVI data set.

TABLE I. FORECASTING PERFORMANCE

Data set	RMSE	MAPE	DS	R_{adj}^2
Training	0.006305	0.021837%	95.559503%	0.999371
Validation	0.013919	0.093256%	95.833333%	0.997181
Testing	0.006088	0.011930%	93.333333%	0.999019

The RMSE and MAPE errors were smallest on testing data set, directional symmetry was best on validation data set and the adjusted coefficient of multiple determination was best on training data set. These results showed a good performance of a regularized layer recurrent neural network because the train set errors, the validation set errors and the test set errors have similar characteristics, and it does not appear that any significance over fitting has occurred. Actual and predicted values of the NDVI time series on training data set is shown in Fig. 6.

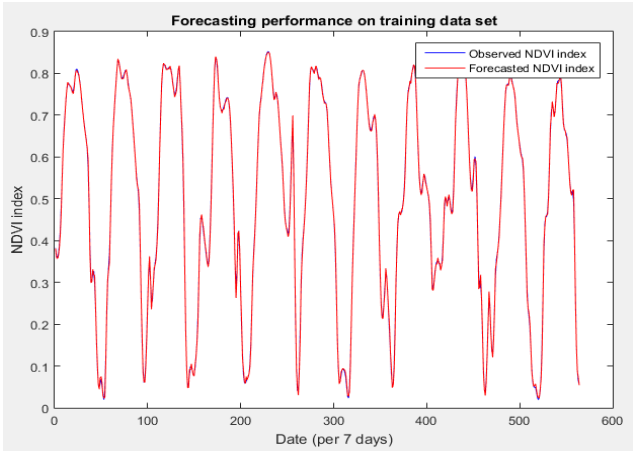

Figure 6. Forecasting performance on training data set.

Actual and predicted values of the NDVI time series on validation data is shown in Fig. 7.

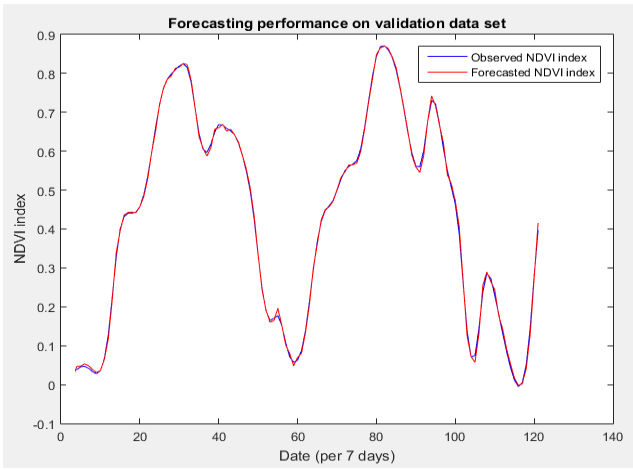

Figure 7. Forecasting performance on validation data set.

Actual and predicted values of the NDVI time series on testing data is shown in Fig. 8.

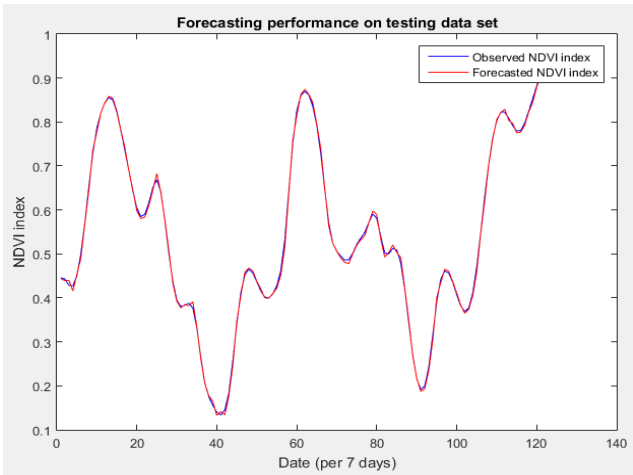

Figure 8. Forecasting performance on testing data set.

From this figures, it can be observed that using the optimal LRN model there is a small deviation between the actual and predicted time series.

VI. Summary And Conclusions

In this paper one-step-ahead predictions of the normalized difference vegetation index (NDVI) is obtained using a layer recurrent neural network (LRN). The presence of recurrent feedback in neural network is a positive factor in forecasting of NDVI time series. This is evidently because the recurrent neural network has a "deeper memory" than other classes of neural networks. The study concludes that the forecasting abilities of a regularized LRN provides a potentially very useful method for the NDVI time series forecasting.

References

[1] M. M. Badamasi, S. A. Yelwa, M. A. AbdulRahim and S. S. Noma, "NDVI threshold classification and change detection of vegetation cover at the Falgore Game Reserve in Kano State, Nigeria," Sokoto Journal of the Social Sciences, vol. 2, no. 2, pp. 174-194.

[2] N. B. Duy and T. T. H. Giang, "Study on vegetation indices selection and changing detection thresholds selection in Land cover change detection assessment using change vector analysis," presented at International Environmental Modelling and Software Society (iEMSs), Sixth Biennial Meeting, Leipzig, Germany, 2012.

[3] E. Sahebjalal and K. Dashtekian, "Analysis of land use-land covers changes using normalized difference vegetation index (NDVI) differencing and classification methods," African Journal of Agricultural Research, vol. 8, no. 37, pp. 4614-4622, September 26, 2013.

[4] A. F. Manso, C. Quintano and O. F. Manso, "Forecast of NDVI in coniferous areas using temporal ARIMA analysis and climatic data at a regional scale," International Journal of Remote Sensing, vol. 32, no. 6, pp. 1595-1617, March 2011.

[5] University of Natural Resources and Life Sciences, Vienna. *Data service platform for MODIS Vegetation Indices time series processing at BOKU, Vienna*. Available at: http://ivfl-info.boku.ac.at/.

[6] F. Vuolo, M. Mattiuzzi, A. Klisch and C. Atzberger, "Data service platform for MODIS Vegetation Indices time series processing at BOKU Vienna: current status and future perspectives," Proc. SPIE 2012, vol. 8538A, pp. 1–11.

[7] A. Shabri and R. Samsudin, "Daily crude oil price forecasting using hybridizing wavelet and artificial neural network model," Mathematical Problems in Engineering, vol. 2014, article ID 201402, July 2014.

[8] G. Zhang, B. E. Patuwo and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art," International Journal of Forecasting, vol. 14, no. 1, pp. 35-62, March 1998.

[9] Y. Yuan, "Image-based gesture recognition with support vector machines, " University of Delaware Newark, DE, USA, 2008.

[10] J. L. Elman, "'Finding structure in time," *Cognitive Science*, vol. 14, pp. 179-211, 1990.

[11] F. H. Nordin, F. H. Nagi and A. A. Z. Abidin, "Comparison study of computational parameter values between LRN and NARX in identifying nonlinear systems," Turkish Journal of Electrical Engineering & Computer Sciences, vol. 21, pp. 1151-1165, 2013.

[12] O. Kişia and E. Uncuoglub, "Comparison of three back-propagation training algorithms for two case studies," *Indian Journal of Engineering & Material Sciences*, vol .12, pp. 434-442, October 2005.

[13] M. Jadhav, S. L. Nalbalwar and A. A. Ghatol, "Artificial neural network models based cardiac arrhythmia disease diagnosis from ECG signal data," *International Journal of Computer*, vol. 44, no. 15, April 2012.

[14] M. Fernandez, J. Caballero, L. Fernandez and A. Sarai, "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, pp. 269-289, 2011.