International Journal of Innovative Computing, Information and Control
Volume **11**, Number **3**, June **2015**

ICIC International ©2015 ISSN 1349-4198
pp. 1021–1031

OPTIMIZED ECHO STATE NETWORKS FOR DROUGHT MODELING BASED ON SATELLITE DATA

Mahdi Jalili[1,2] and Amir Mohammadinezhad[1]

[1]Department of Computer Engineering
Sharif University of Technology
Azadi Ave., Tehran 11365, Iran
mjalili@sharif.edu

[2]School of Electrical and Computer Engineering
RMIT University
Melbourne, VIC 3001, Australia

Received May 2014; revised September 2014

Abstract. *Remotely sensed data obtained through satellite imaging is a useful tool for modeling environmental phenomena such as drought. In this manuscript, we apply optimized echo state networks to model and predict drought severity based on satellite images. To this end, a model is constructed in which the satellite-based vegetation index is fed as an input and drought severity index is obtained as output. We use a Kronecker-based approach to reduce the number of parameters of echo state networks to be optimized (i.e., the internal weights of reservoir). A number of evolutionary algorithms are used to optimize the parameters, of Differential Evolution results in the best performance as compared to genetic algorithms and simulated annealing. The proposed model also outperforms neural network models including multilayer perceptrons, radial basis function networks and support vector machines.*

Keywords: Drought modeling, Prediction, Artificial neural networks, Optimization, Evolutionary algorithms, NDVI, SPI

1. **Introduction.** Drought is a severe environmental condition affecting many parts of the world including Iran. Often, shortage in the precipitation over a period of time results in meteorological drought. There is a number of metrics to measure severity of meteorological drought, of which standardized precipitation index (SPI) is the most widely used one [1]. It is obtained as difference of precipitation from the mean value over a specified period. In order to have SPI-maps for a region, one should have sufficient precipitation stations, which is always not the case in many countries. One of the major problems that the relevant policy making bodings face regarding drought management is the availability of data. The cleaned meteorological and agricultural data is often made available after a long time that cannot be effectively used for monitoring purposes. Another problem is the spatial resolution of the data; the distribution of the stations is not homogenous across different parts of the country and there are not enough stations in some parts. This calls for remotely sensed data that are widely available for almost all regions of interest. Satellite-based images have been successfully used for monitoring (and predicting) environmental phenomena in general [2-4], and meteorological drought in particular [5-10].

Availability of satellite images over rather long periods and their high spatial resolution (as compared to meteorological data) make them a suitable tool for drought modeling. A number of scholars have successfully modeled drought-related indices (e.g., SPI) through satellite-based vegetation features such as normalized difference vegetation index (NDVI),

vegetation condition index (VCI) and temperature condition index (TCI). NDVI has shown significant correlations with SPI values, suggesting it as an effective measure for monitoring vegetation condition [11,12]. We have recently reported an extensive study on a nation-wide drought modeling based on NOAA-AVHRR data in Iran [5]. We constructed neural network models (multilayer perceptrons (MLP), support vector machines (SVM) and radial basis function (RBF) networks) and successfully predicted drought conditions based on satellite data [5]; i.e., the model takes NDVI as inputs and outputs SPI value. In this manuscript, we apply echo state networks (ESN) to model drought conditions.

ESN have been recently developed as a class of recurrent neural networks [13-15], which have shown promising performance in various modeling and classification applications. A major difficulty in training an ESN is to find the optimal parameters of its reservoir. In this manuscript, we use an innovative method based on Kronecker product [16], thus dramatically reducing the number of parameters to be optimized. We then use some evolutionary optimization methods including genetic algorithm (GA), simulated annealing (SA) and differential evolution (DE) in order to optimize the reservoir parameters. Our experiments show that DE results in the best performance as compared to GA and SA. We also show that the optimized ESN outperforms MLP, RBF and SVM in terms of prediction accuracy.

Here we indeed aim at building a reliable prediction tool for drought-related policy makers in Iran. Helping public policy making is an important issue in sustainability applications. Policy makers have a major role in minimizing the risks of a natural hazard like drought condition. In order to help the policy makers to have access to real-time drought conditions (and future forecast), one can use models developed in this work. The model developed in this work is given the satellite data as inputs and results in the drought conditions as the output. In the application (after training), it only needs the satellite data (which is available on a real-time basis) to predict the drought conditions. Therefore, the reaction time to a specific condition can be faster than the case when only meteorological data are used. In this work we show that ESN outperforms previously developed models for this purpose [5].

2. **Data.**

Iran is located in the Middle East covering an area of 1.64 million km^2 and extends between latitudes 25° and 40°N and longitudes 44° and 63°E. It is surrounded by two mountain ranges, namely Zagros to the west/north west and Alborz to the north, both with important role in the spatial and temporal distribution of precipitation over the country. These mountains avoid Mediterranean moisture bearing systems by crossing through the region to the east. Iran's climate ranges from arid or semiarid to subtropical along the Caspian coast and the northern forests. On the northern edge of the country (the Caspian coastal plain), temperature rarely falls below freezing and the area is humid in most of the time. Annual precipitation is 680 mm in the eastern part of the plain and more than 1700 mm in the western part. To the west, areas in the Zagros basin experience lower temperatures, severe winters with below zero average daily temperatures and heavy snowfall. The eastern and central basins are arid with less than 200 mm of rain and have occasional desert. The coastal plains of the Persian Gulf and Gulf of Oman in southern Iran have mild winters and very humid and hot summers. The annual precipitation ranges from 135 to 355 mm.

2.1. **Meteorological data.**

In order to obtain SPI time series, we use monthly rainfall and temperature data gathered for four stations in different parts of Iran: Abadan in south west, Bushehr in south, Gorgan in north and Mashhad in north east of Iran. These cities belong to different clime conditions and can be considered as a representative of

TABLE 1. Correspondence between SPI values and drought severity conditions

SPI Values	Drought Category
≤ -2.0	Extremely Dry
–2.0 to –1.5	Moderately Dry
–1.5 to –1.0	Dry
–1.0 to 1.0	Neutral
1.0 to 1.5	Wet
1.5 to 2.0	Moderately Wet
$2.0 \leq$	Extremely Wet

Iran's general climate condition. The precipitation data used in this work are for the years between 1978 and 2008. The data were checked statistically to ensure their reliability and consistency. We use standardized precipitation index (SPI) to quantify drought conditions. SPI is one of the widely used metrics for drought modeling and forecasting [17-20]. SPI not only captures information about the amount of rainfall, but also compares it with a long-term mean. It describes how precipitation is more (or less) than normal in a period of time in an observation point. For regions where parameters such as soil moisture are also available, other proper metrics can also be used to measure drought conditions. Table 1 shows the correspondence between SPI values and drought conditions.

2.2. **Satellite data.** In this work, we obtained the vegetation index from NOAA-AVH RR. As a vegetation index, we considered NDVI, which has been previously applied for drought modeling [11,12,21,22]. NDVI is calculated as [23]

$$NDVI = \frac{b_{NIR} - b_{RED}}{b_{NIR} + b_{RED}}, \tag{1}$$

where b_{NIR} and b_{RED} are the reflectance in the NIR and red bands, respectively.

The NOAA-AVHRR images have a spatial resolution of 1.1 km at best. This decreases with increase in the view angle off-nadir. The maximum off-nadir resolution along track is 2.4 km and across track is 6.9 km. Iran has a receiving station for downloading NOAA-AVHRR data allowing us to have access to near-nadir images. We applied standard procedures for radiometric and geometric correction on the images. Then, in order to remove cloud effect, a number of cloud detection methods were applied and their performance was compared. We finally used a threshold test based on [24], in order to mask the cloud effects. In order to obtain the NDVI value for each station, first, the pixel in which the station is located was found. Then, the NDVI values of that pixel and those in its first neighborhood were calculated. Finally, the NDVI value of the station was obtained by averaging over these 9 values. The stations were chosen such that there was no major building structure and agricultural lands in their regions (there were only some and averaging over neighboring pixels minimizes their unwanted effects). This strategy has also been previously used for the same purpose [5].

3. **Neural Network Based Drought Prediction.** Here we use neural network modeling tools for the drought prediction task. The SPI values are categorized into 7 classes of drought conditions (Table 1), which are indeed taken into account as the models' output. In other words, the model solves a classification task with NDVI values as the input and drought classes as the output. Various architectures of artificial neural networks (ANNs) have been applied to real-world applications such as classification, modeling, filtering, smoothing, optimization, prediction, and function approximation [25-27]. Recurrent neural networks (RNNs) represent a class of neural networks that are used to model systems

with dynamical behavior [27]. The data that is used in this work represent a dynamical system, i.e., the input and out data is time series. To model time series data, static networks cannot be used and one should use RNNs.

There are two general approaches in modeling dynamical systems using RNNs [27]. One approach is to use a dynamic network in which each neuron has a specific dynamics; all state space based models are of this type. The other approach is to use a static network (i.e., composed of neurons without dynamics) and to transfer the dynamics to the data. In this approach, the relationship between the output and the input (and its past) is transferred to the data. Since the network is static, the standard learning rules can be used in order to find the model parameters. There is a number of such classifiers that have been frequently used in the literature for classification of dynamic data. Examples include Multilayer Perceptron (MLP), Radial Basis Function (RBF) and Support Vector Machines (SVM). In order to perform input-output mapping using MLP, RBF and SVM, one can use Auto Regressive Moving Average (ARMA) model. In this model, the general relation between SPI (output) and NDVI (input) is as

$$\mathrm{SPI}(t+p) = f\left(NDVI(t), NDVI(t-1), \ldots, NDVI(t-q)\right), \tag{2}$$

where $f(\cdot)$ is the function to be found by the neural network models. Indeed, in this approach, in order to predict the SPI in a specific month (let us suppose $p = 1$ and $q = 6$), the NDVI values in the past 6 months are given to the model as inputs, which means that the model has 6 inputs and 1 output. Then, standard learning rules are used in order to optimize parameters of MLP, RBF and SVM. The results reported in this work are for one step ahead prediction (i.e., $p = 1$). Usually, by increasing q, higher prediction accuracy can be achieved; however, this also makes the model more complex. Thus, one should find the smallest q that gives satisfactory performance. Here, we found that $q = 12$ gives acceptable results in our data. This means that SPI in a specific month is related to the NDVI values in the past 12 months.

MLP has a simple feedforward structure in which the neurons are placed in distinct computational layers [27]. The signals are directly fed from the input layer to the hidden layers where the computations are performed. Error between the output of the network and the desired output is calculated at the output layer, and backpropagated to the previous layers. This way, the free parameters of the model can be calculated using efficient algorithms such as error backpropagation rule. In this learning rule, the weight of connection between two neurons is a function of the output of the presynaptic neuron and local gradient of the postsynaptic neurons. The local gradients are computed based on backpropagation of error signals. RBF is another paradigm to learn input-out mapping. There is only a single hidden layer in RBF, and each neuron in this layer has a radial basis function (often chosen as Gaussian function). The neurons in the hidden layer receive the input, and produce an output that is a function of the Euclidian distance between the input and the center of their basis function. Finally, the neurons in the last layer obtain the output by linearly integrating the outputs of the hidden layer. SVM is another approach that has been frequently used in various classification tasks. It is designed such that the distance between the data points in each class from the classifying boundary is maximized. In this work, we use SVMs with Gaussian kernels.

4. **Echo State Networks.**

RNNs often have expensive computational cost and slow convergence, which limit their applications in large-scale systems [28]. In order to bypass expensive computational load of RNNs and propose a simpler yet effective modeling tool, Jeager introduced a new model, known as echo state networks (ESNs) [13-15]. An ESN is composed of a discrete-time RNN and a linear readout output layer, which maps the

reservoir states to the desired output. ESNs are different from other RNNs in that a recurrent network is often randomly created and remains unchanged in an ESN. This random architecture is called a reservoir. The reservoir is passively excited by the input signal and maintains a nonlinear transformation of the input history in its state. The desired output signal is generated as a linear combination of the neuron's signals from the input-excited reservoir. In its simplest form, only the readout weights (i.e., the weights from the reservoir to the output) are trained based on linear regression models. A schematic overview of this network is given in Figure 1. By considering $u(n)$ as input units, $x(n)$ as internal (reservoir) units and $y(n)$ as output units, the neurons' states and the readout output are updated as

$$\boldsymbol{x}(n+1) = f\left(\boldsymbol{W}^{in}\boldsymbol{u}(n+1) + \boldsymbol{W}\boldsymbol{x}(n) + \boldsymbol{W}^{fb}\boldsymbol{y}(n)\right), \tag{3}$$

$$\boldsymbol{y}(n+1) = f^{out}\left(\boldsymbol{W}^{out}\boldsymbol{x}(n+1)\right), \tag{4}$$

where f and f^{out} are appropriate activation functions. In this work, we set both of them as $tanh(\cdot)$. $\boldsymbol{W}^{in}$, $\boldsymbol{W}$, $\boldsymbol{W}^{fb}$ and $\boldsymbol{W}^{out}$ denote input, reservoir, feedback and output weights, respectively. In most cases, only $\boldsymbol{W}^{out}$ is trained and the other weight vectors are kept unchanged (they are often set randomly). A good way to train the output weights is through a recursive least-squares (RLS) learning rule, which is a well-known learning algorithm in neural networks. In order to update the output weights using an RLS learning rule, one performs

$$\boldsymbol{W}^{out}(n+1) = \boldsymbol{W}^{out}(n) - e(n+1)\boldsymbol{P}(n+1)\boldsymbol{x}(n+1), \tag{5}$$

where n denotes the learning step, e is the output error (i.e., the difference between the output produced by the network and the desired output) and $\boldsymbol{P}(n+1)$ is a matrix that is updated at the same time as the weights according to the rule, as

$$\boldsymbol{P}(n+1) = \boldsymbol{P}(n) - \frac{\boldsymbol{P}(n)\boldsymbol{x}(n+1)\boldsymbol{x}^{\boldsymbol{T}}(n+1)\boldsymbol{P}(n)}{1+\boldsymbol{x}^{\boldsymbol{T}}(n+1)\boldsymbol{P}(n)\boldsymbol{x}(n+1)}. \tag{6}$$

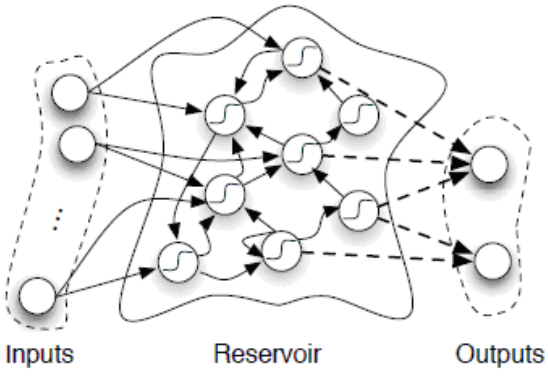

FIGURE 1. Architecture of the echo state network. In this work, we use Kronecker product to construct the reservoir.

5. **Optimization of the Reservoir Weights in ESNs.** ESNs (or equivalently reservoir computing (RC) methods) have been successfully applied to many real-world applications. Although they result in superior performance as compared to classic RNNs, their performance is sensitive to the choice of weight parameters in the reservoir. A usual strategy to choose the internal weights of the reservoir is to set them randomly in a range (often $[-0.5, 0.5]$) and further use a scaling parameter in order to fulfill the fading memory effect (i.e., in order to guarantee asymptotic stability) [14]. It has been shown that the internal

structure of the reservoir has a great influence on the prediction ability of ESNs [29]. In order to find a good reservoir with optimal parameters, a number of works have been introduced to optimize the internal weights [16,30-34].

The performance of the network, like other neural network models, is assessed on the error produced by test dataset. In this work, we considered normalized root mean square error (NRMSE) criterion for the optimization task. NRMSE is calculated as

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{n_{test}} (y_{test}(n) - O^{desired}(n))^2}{\sigma^2 n_{test}}}, \tag{7}$$

where n_{test} is the number of test samples, y_{test} and $O^{desired}$ are the output of the network and the real (desired) output respectively, and σ^2 is the variance of the real output.

5.1. **Kronecker product to optimize the parameters.** The main drawback of many reservoir optimization methods [32-34] is that they either use too many parameters, e.g., optimization of all weights of reservoir, or only a few effective parameters such as the spectral radius and sparsity of the reservoir. When only spectral radius and sparsity of the reservoir are optimized, the performance of the system cannot be significantly improved, and to gain more, one has to optimize the internal weights as well. However, the number of internal weights is often huge, calling for strategies to effectively reduce them. We have recently introduced a method to optimize reservoir parameters [16]. The method is based on Kronecker product, which is explained in the following. The parameters used for the optimization process are the internal weights of the reservoir ($\boldsymbol{W}$). The proposed method dramatically reduces the number of parameters to be optimized. To this end, we used Kronecker product which is an operation on two matrices of arbitrary size, resulting in a new matrix with a bigger size. If A is an $m \times n$ and B is a $p \times q$ matrices, then the Kronecker product $B \otimes A$ is an $mp \times nq$ block matrix that is defined as

$$B \otimes A = \begin{bmatrix} b_{11}A & \cdots & b_{1n}A \\ \vdots & \ddots & \vdots \\ b_{n1}A & \cdots & b_{nn}A \end{bmatrix}, \tag{8}$$

where b_{ij} is the (i, j) element of matrix B. In this work, we picked the kernels (A and B) from a normal distribution. In the proposed method, the weight matrix of the reservoir is constructed based on Kronecker products as

$$\boldsymbol{W} = \rho_{\max} B_{q\times q} \otimes A_{p\times p}^{\otimes m}, \tag{9}$$

where A and B are arbitrary matrices, $\rho_{\max}$ is a constant coefficient and $A_{p\times p}^{\otimes m}$ denotes the m-fold product of the kernel by itself, i.e., the m-th Kronecker power of $A_{p\times p}$. Here, we considered a reservoir with 256 neurons; thus the weight matrix $\boldsymbol{W}$ having 65536 elements that should be optimized. Instead of optimizing all these 65536 parameters which can be computationally too expensive, we used a 4×4 kernel with only 16 elements and generated the 4-th Kronecker power to obtain the 256×256 weight matrix. This significantly reduces the number of parameters to be optimized, since only the entries of the kernel are optimized. In our previous work, we used a simple steepest descent method (through numerically approximating the gradients) for the optimization [16]. Here, we use sophisticated evolutionary optimization algorithms that work much better than the gradient-based steepest descent algorithm.

5.2. **Evolutionary optimization algorithms.** We used three evolutionary optimization algorithms including simulated annealing (SA), genetic algorithms (GA), and differential evolution (DE). SA is a common and effective optimization algorithm for nonlinear

problems. In order to perform the optimization task, first, a 4×4 kernel (with 16 elements) was considered with elements picked up from a normal distribution. Then, the reservoir was created based on Equation (9) with $\rho_{\max} = 0.04$ in order to calculate the cost function (i.e., NRMSE on the test dataset). In the next step, the kernel weights were changed and the new cost function was calculated. If new cost function (C_{new}) was lower than its values in the previous step ($C_{previous}$), the change was accepted; otherwise, the change was accepted with probability $\exp[(C_{new} - C_{previous})/T]$, where T is temperature. In this work, we initially set $T = 20$, and then, decreased it with a rate equal to 0.8 up to $T = 10^{-8}$ from which it is kept unchanged.

GA is a direct, parallel and statistical method for general exploration and optimization. A GA usually has three main operators: selection, crossover and mutation. Here, we first created the initial population consisting of 20 random kernels for which the cost function was calculated considering $\rho_{\max} = 0.04$. Then, at each step, two kernels with the best cost value (i.e., the lowest NRMSE) were selected and two new kernels (children) were created based on the conventional crossover rule. A random mutation was also applied to these new chromosomes with the probability of 0.05. Then, they were replaced the worst kernels in the population based on their cost values. As another optimization technique, we used DE, which is a direct and parallel exploration method similar to GA. In DE optimization algorithm, the initial population was created with 20 random kernels. The mutant vector was created for each chromosome x of the population as

$$v_i = x_{r_1} + F(x_{r_2} - x_{r_3}), \tag{10}$$

where r_1, r_2 and r_3 are random indices, mutually different and F is a positive value chosen as 0.9. Then, in order to increase the diversity of the perturbed parameter vector, crossover was applied through forming the trial vector as

$$u_{ji} = \left\{ \begin{array}{ll} v_{ji} & \text{if } (randb(j) \leq CR) \text{ or } j = rnbr(i) \\ x_{ji} & \text{if } (randb(j) > CR) \text{ and } j \neq rnbr(i) \end{array} \right\}, \tag{11}$$

where $randb(j)$ is the j-th evaluation of a uniform random number generator with outcome $\in [0, 1]$, CR is the crossover constant $\in [0, 1]$ which is set to 0.8, and $rnbr(i)$ is a random chosen index which ensures that u gets at least one parameter from v. At the end of the optimization step, the trial vector u_i is compared to the target vector x and it was replaced, if the trial vector has a smaller cost function.

6. **Results and Discussion.** In order to assess the performance of the proposed ESN, we compared its performance with a number of classic classification methods including MLP, RBF and SVM. These methods have been previously applied for predicting drought based on vegetation indexes, of which MLP and SVM have been found outperforming RBF [5]. In this work, we aimed at predicting the drought conditions or class (i.e., extremely dry, moderately dry, dry, neutral, wet, moderately wet, extremely wet), based on the definitions given in Table 1. The NDVI and SPI values were calculated for all periods for which the data was available (from 1989-2008). We extracted monthly values for NDVI and SPI, resulting in 240-valued time series for this 20-year period. We performed the classification task for three years 2006, 2007, and 2008 (on a monthly basis). The networks have a single input that is the NDVI value and one output that is the drought condition corresponding to the SPI value. In order to model (and predict) the drought class in a specific month of a year, all time series before that month were calculated as training and validation (80% as training and 20% as validation), while that particular month was treated as a test data. The accuracy of prediction for each month is either 0 or 1; 0 when

the class is predicted wrongly, and 1 when it is correctly predicted. Finally, the yearly prediction accuracy is obtained as

$$\text{Accuracy} = C/12, \tag{12}$$

where C indicates the number of months with correct classification.

We first compared the performance of different optimization algorithms (SA, GA, and DE). All these three optimization algorithms have by far better performance as compared to gradient-based steepest descent, as proposed in [16]. Figure 2 shows the accuracy of drought prediction for different optimization methods. As it is seen, except for a single case (Gorgan station in 2008), DE exhibited better or equal accuracy than the other two optimization algorithms. In average, SA achieved prediction accuracy of %60.33, GA an accuracy of %64, and DE an accuracy of %74.25. A reason for superior performance of DE over GA and SA is the way its mutation and crossover works. In the mutation step, the mutant vector, which is created with three other vectors, performs the local searches and causes the algorithm to move to the local minimum. In the crossover and recombination step, the trial vector, which is compared to the target vector to avoid local minima, performs the global searches.

As the next experiment, we compared the performance of ESN with DE optimization with that of classic modeling tools including MLP, RBF, and SVM. We optimized the structure of these models. We considered an MLP with 2 hidden layers each with 10 and 15 neurons. We first considered a single hidden layer for MLP; however, we could not obtain satisfactory performance by increasing the number of neurons in the hidden layer. Therefore, we used an MLP with two layers and determined the number of neurons

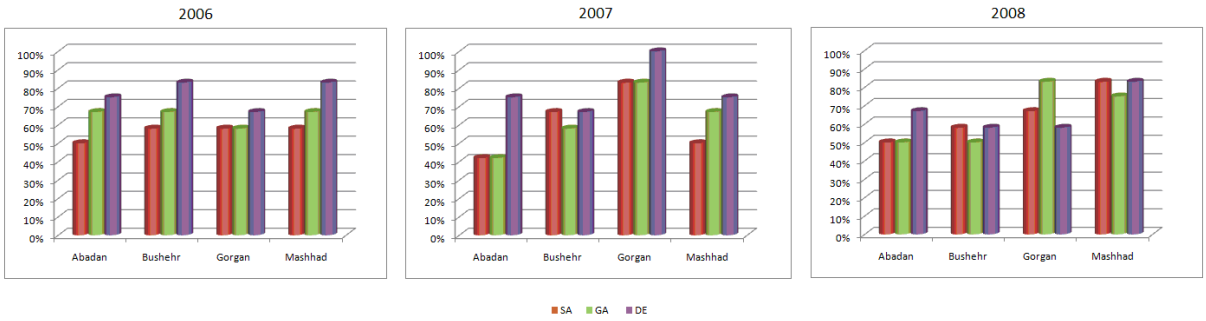

FIGURE 2. Accuracy of SA, GA and DE optimization on ESNs with input as NDVI and output as drought condition. The results are for all months in years 2006, 2007 and 2008.

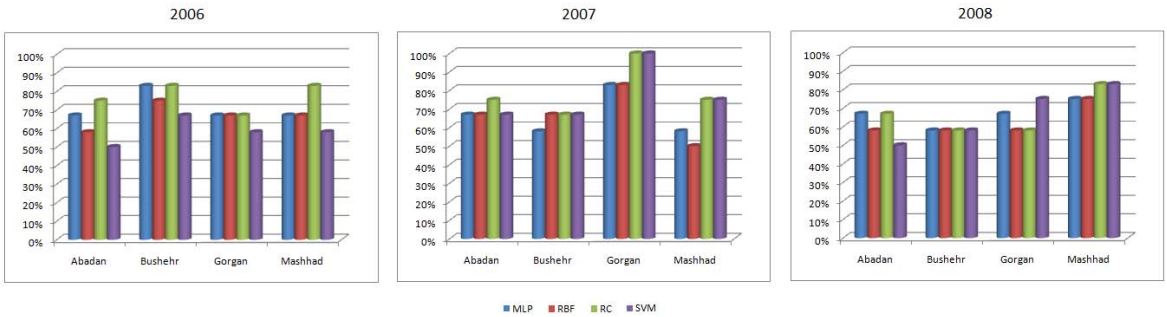

FIGURE 3. Accuracy of MLP, RBF, SVM and ESN (denoted as reservoir computing (RC) method) with differential evolution optimization of reservoir parameters

by try and error (increasing the number of neurons until a satisfactory performance is obtained). All neurons were identical with $tanh(\cdot)$ activation function. Furthermore, backpropagation learning algorithm was used in order to perform the optimization task to find MLP parameters. For RBF networks, we adopted multivariate Gaussian functions as basis function with 20 neurons in the hidden layer; we first started from small number of neurons and increased the number until satisfactory performance is obtained. In order to construct the SVM, we considered SVM with Gaussian basis function.

Figure 3 compares the performance of reservoir computing (RC) methods, i.e., ESNs, with that of RBF, MLP, and SVM. As it is seen, expect for one case (Gorgan station in 2008), the proposed ESN model showed better or equal performance than the other three models. The other three models did not show the same behavior across all years. For 2006, the second top performer was MLP, followed by RBF and SVM. In 2007, ESN and SVM were the top performers with better or equal performance than MLP and RBF. SVM showed better performance than ESN in one of the stations (Gorgan) in 2008, while ESN was the top performer for the other three stations.

As discussed in Section 3, there are two techniques to use neural network models for modeling time series data: using dynamic networks or using static networks with memory taps introduced into the data. Unlike MLP, RBF and SVM, the ESNs are composed of dynamical neurons (i.e., state space based model), and they are capable of extracting the input-output dynamic relationship without introducing the memory taps into the input data.

7. **Conclusion.** Unavailability of real-time meteorological and agricultural data for many regions of Iran makes it difficult, if not impossible, to efficiently monitor drought conditions in the country. However, real-time availability of satellite imaging data as well as their long-term time series, makes it a useful tool for them. In this manuscript, we applied ESNs to build a model for predicting drought conditions based on satellite imaging. Drought condition was quantified through SPI index that could be obtained using long-term precipitation data. NOAA-AVHRR images were used to extract NDVI time series, measuring land vegetation. The model took NDVI as input and outputted SPI class (there were seven SPI classes for different drought conditions). One of the major challenges in ESNs is that there are many internal reservoir weights to be optimized. In this work, we used an approach based on Kronecker product to dramatically reduce the size of the parameter space. We then employed an RLS algorithm for optimizing the readout weights of the ESN and evolutionary algorithms for finding the optimal internal weights of the reservoir (i.e., the original parameters before applying the Kronecker product). Differential evolution optimization technique showed better performance as compared to genetic algorithms and simulated annealing. We also compared the performance of optimized ESN with that of classic classifiers including MLP, RBF, and SVM. The proposed ESN outperformed these classifiers.

Acknowledgments. The authors would like to thank R. Farshi, J. Gharibshah, H. Tajari, M. Ghavami, and M. Beheshtifar for their help in processing the data. This work was partially supported by Australian Research Council project DE140100620.

REFERENCES

[1] T. B. McKee, N. J. Doesken and J. Kleist, The relationship of drought frequency and duration to time scales, *The 8th Conference on Applied Climatology*, Anaheim, CA, USA, 1993.

[2] A. H. Weerts, J. Schellekens and F. S. Weiland, Real-time geospatial data handling and forecasting: Examples from delft-FEWS forecasting platform/system, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.3, pp.386-394, 2010.

[3] L. Pulvirenti, N. Pierdicca, M. Chini and L. Guerriero, Monitoring flood evolution in vegetated areas using COSMO-SkyMed data: The tuscany 2009 case study, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.6, pp.1807-1816, 2013.

[4] J. L. Case, F. J. LaFontaine, J. R. Bell, G. J. Jedlovec, S. V. Kumar and C. D. Peters-Lidard, A real-time MODIS vegetation product for land surface and numerical weather prediction models, *IEEE Trans. Geoscience and Remote Sensing*, 2013.

[5] M. Jalili, J. Gharibshah, S. Ghavami, M. Beheshtifar and R. Farshi, Nation-wide prediction of drought conditions in Iran based on remote sensing data, *IEEE Trans. Computers*, vol.63, 2013.

[6] C. J. Tucker and B. J. Choudhury, Satellite remote sensing of drought conditions, *Remote Sensing of Environment*, vol.23, pp.243-251, 1987.

[7] C. M. Rulinda, A. Dilo, W. Bijker and A. Steina, Characterising and quantifying vegetative drought in East Africa using fuzzy modelling and NDVI data, *Journal of Arid Environments*, vol.78, pp.169-178, 2012.

[8] A. Mohammadinezhad and M. Jalili, Optimization of echo state networks for drought prediction based on remote sensing data, *The Industrial Electronics and Applications*, Melbourne, Australia, 2013.

[9] L. Zhou, J. Wu, J. Zhang, S. Leng, M. Liu, J. Zhang, L. Zhao, F. Zhang and Y. Shi, The integrated surface drought index (ISDI) as an indicator for agricultural drought monitoring: Theory, validation, and application in mid-eastern China, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.6, pp.1254-1262, 2013.

[10] L. Feng, C. Hu and X. Chen, Satellites capture the drought severity around China's largest freshwater lake, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.5, pp.1266-1271, 2012.

[11] X. Song, G. Saito, M. Kodama and H. Sawada, Early detection system of drought in east Asia using NDVI from NOAA AVHRR data, *International Journal of Remote Sensing*, vol.25, pp.3105-3111, 2004.

[12] L. Ji and A. J. Peters, Assessing vegetation response to drought in the northern great plains using vegetation and drought indices, *Remote Sensing of Environment*, vol.87, pp.85-98, 2003.

[13] H. Jaeger, Adaptive nonlinear system identification with echo state networks, *The Advances in Neural Information Processing Systems*, 2002.

[14] M. LukoševičIus and H. Jaeger, Reservoir computing approaches to recurrent neural network training, *Computer Science Review*, vol.3, pp.127-149, 2009.

[15] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science*, vol.304, pp.78-80, 2004.

[16] A. A. Rad, M. Hasler and M. Jalili, Reservoir optimization in recurrent neural networks using properties of Kronecker product, *Logic Journal of the IGPL*, vol.18, pp.670-685, 2010.

[17] T. B. McKee, N. J. Doeskin and J. Kieist, The relationship of drought frequency and duration to time scales, *Proc. of the 8th Conference on Applied Climatology*, Boston, Massachusetts, pp.179-184, 1993.

[18] A. C. Costa, Local patterns and trends of the standard precipitation index in southern Portugal (1940-1999), *Advances in Geosciences*, vol.30, pp.11-16, 2011.

[19] A. Belayneh and J. Adamowski, Standard precipitation index drought forecasting using neural networks, wavelet neural networks, and support vector regression, *Applied Computational Intelligence and Soft Computing*, vol.2012, p.6, 2012.

[20] S. Khan, H. F. Gabriel and T. Rana, Standard precipitation index to track drought and assess impact of rainfall on watertables in irrigation areas, *Irrigation and Drainage Systems*, vol.22, pp.159-177, 2008.

[21] P. Omute, R. Corner and J. L. Awange, The use of NDVI and its derivatives for monitoring lake Victoria's water level and drought conditions, *Water Resources Management*, vol.6, pp.1591-1613, 2012.

[22] F. Tonini, G. J. Lasinio and H. H. Hochmair, Mapping return levels of absolute NDVI variations for the assessment of drought risk in Ethiopia, *International Journal of Applied Earth Observation and Geoinformation*, vol.18, pp.564-572, 2012.

[23] J. W. Rouse, R. H. Haas, J. A. Schell, D. W. Deering and J. C. Harlan, Monitoring the vernal advancement and retrogradation (greenwave effect) of natural vegetation, *Greenbelt*, Maryland, 1974.

[24] L. Giglio, J. Kendall and C. Tucker, Remote sensing of fires with the TRMM VIRS, *Remote Sensing of Environment*, pp.203-207, 2003.

[25] L. O. Chua and L. Yang, Cellular neural networks: Applications, *IEEE Trans. Circuits and Systems*, vol.35, pp.1273-1290, 1988.
[26] G. P. Zhang, Neural networks for classification: A survey, *IEEE Trans. Systems, Man, and Cybernetics C: Applications and Reviews*, vol.30, pp.451-462, 2000.
[27] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice Hall, 1999.
[28] B. Hammer, B. Schrauwen and J. J. Steil, Recent advances in efficient learning of recurrent networks, *Proc. the European Symposium on Artificial Neural Networks*, pp.213-216, 2009.
[29] Q. Song and Z. Feng, Effects of connectivity structure of complex echo state network on its prediction performance for nonlinear time series, *Neurocomputing*, vol.73, pp.2177-2185, 2010.
[30] A. Rodan and P. Tino, Minimum complexity echo state network, *IEEE Trans. Neural Networks*, vol.22, pp.131-144, 2011.
[31] A. F. Krause, V. Dürr, B. Bläsing and T. Schack, Evolutionary optimization of echo state networks: Multiple motor pattern learning, *The Artificial Neural Networks and Intelligent Information Processing*, 2010.
[32] K. Ishu, T. van der Zant, V. Becanovic and P. Ploger, Identification of motion with echo state network, *OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04*, pp.1205-1210, 2004.
[33] K. Bush and B. Tsendjav, Improving the richness of echo state features using next ascent local search, *Proc. of the Artificial Neural Networks in Engineering Conference*, pp.227-232, 2005.
[34] F. Jiang, H. Berry and M. Schoenauer, Supervised and evolutionary learning of echo state networks, *Parallel Problem Solving from Nature*, pp.215-224, 2008.