

Experimental Analysis of Forecasting Solar Irradiance with Echo State Networks and Simulating Annealing

Sebastián Basterrech^(✉)

Department of Computer Science, Faculty of Electrical Engineering
and Computer Science, VŠB–Technical University of Ostrava,
Ostrava, Czech Republic

`Sebastian.Basterrech.Tiscordio@vsb.cz`

Abstract. The solar energy is a well alternative for covering the high electrical demand, and it starts to be integrated into the energetic grid infrastructure. High forecast accuracy can help in the management of industrial strategies. We present an approach that combines the potential of a Neural Network named *Echo State Networks (ESN)* and a well-known optimisation technique named *Simulating Annealing (SA)*. We use the SA technique for selecting the meteorological variables relevant in the forecasting task and the ESN as forecasting model. We present the results evaluating our approach on a public dataset.

Keywords: Solar irradiance · Echo State Networks · Simulating Annealing · Forecasting · Time-series problems

1 Introduction

Solar energy has received significant attention during last years because is an alternative of renewable resource that can help for reducing the carbon emissions, and it can be used for covering a relevant part of the growing demand of electrical energy. To have accurate solar irradiance predictions help to integrate the energy into the grid, as well as to avoid congestions. Besides, high forecast accuracy helps to mitigate the negative impacts of instable energy sources. In this paper, we present a procedure for forecasting the solar power irradiance using the history of the irradiance and other several meteorological variables. The approach is based on a widely applied metaheuristic technique named *Simulating Annealing (SA)*, which is used for selecting the most significant input features, and the forecasting is done using the *Echo State Networks (ESN)* model. An ESN is a Recurrent Neural Network often used for solving temporal learning problems. We have two main goals in our article, one consists in defining a group of meteorological variables that impact on the solar power. The second one consists in evaluating the accuracy of Echo State Networks for forecasting solar irradiance using the previous information about the solar irradiance and a group

of external meteorological variables, such as: wind characteristics, air temperature, etc. Related works of forecasting solar power irradiance has been presented during the last years. Some approaches have been based on classic statistical methods [1], Neural Networks [1–3], and other machine learning techniques have been also studied [4–6]. We evaluate our approach using a well-known public dataset [7], and we present the results for predicting the solar irradiance with a forecasting horizon of three days.

The article is organised as follows. In the next section we define the problem of forecasting a time-series and we present a background on the SA metaheuristic and the ESN model. Section 3 introduces our methodology. Section 4 is divided in two parts. First part describes the data set and second part presents the experimental results. The article ends with an outlook and conclusions.

2 Background

In this section we start by formalising the problem of forecasting time-series data. Next, we present a background of the methods used in this article: Echo State Networks and Simulating Annealing.

2.1 Formalization of the Problem

The goal of forecasting a time-series is to predict or estimate future events or trends using the information concerning the past. Given a time-series of real observations $y(1), y(2), \dots, y(t)$ the problem of forecasting a time-series consists in computing a learning tool $\varphi(\cdot, \mathbf{w})$ with parameters \mathbf{w} that predicts (better as possible) the value of $y(t + \tau)$ with $\tau > 0$ using the precedent points $y(t), y(t - 1), \dots$. The accuracy of $\varphi(\cdot)$ is assessed using an average over all distances between the target $y(t + \tau)$ and the predicted value that we denoted by $\hat{y}(t + \tau)$. This problem is generalised when we have a set of external features $\mathbf{a}(t)$ in a multidimensional space. In this case the forecast of $y(t + \tau)$ ($\tau > 0$) is given using the information of $\mathbf{a}(t), \mathbf{a}(t - 1), \dots, y(t), y(t - 1), \dots$. The parameters of $\varphi(\cdot, \mathbf{w})$ are computed such that an error measure in an arbitrary range of time $[1, T]$ is minimised, here we consider the widely used *Mean Squared Errors (MSE)*:

$$MSE = \frac{1}{T} \sum_{t=1}^T (\hat{y}(t) - y(t))^2. \quad (1)$$

2.2 Simulating Annealing Method

A popular optimisation technique is *Simulating Annealing (SA)*, which is used for continuous and combinatorial optimisation problems on multi-dimensional spaces [8]. The technique is inspired from the thermodynamical process wherein liquids freeze and crystallise or metals cool and anneal. The goal consists in optimising an objective function that in this context is named *energy function*.

The procedure is iterative and stochastic, at each step the method tests as feature solution a random point on the searching space. We replace a current solution \mathbf{s}^{curr} (a point on the large space) by a randomly selected *nearby solution* \mathbf{s}^{new} that is chosen with a probability p . A *nearby solution* \mathbf{s}^{new} is a solution that has a Hamming distance with the current solution \mathbf{s}^{curr} less than or equal to d , for an arbitrary d value. The method has a global parameter called *temperature* (T) that decreases in the number of iterations until some arbitrary *frozen* condition T^{end} (following the metal annealing analogy). The model is given by the following selection rule

$$p = \min\{\exp(-(E(\mathbf{s}^{\text{new}}) - E(\mathbf{s}^{\text{curr}}))/kT), 1\}, \quad (2)$$

where k is a constant and p is a probability of selecting a new solution. This rule gives to the model the capacity for exploring new regions that is done jumping from a local minimum to other regions on the searching space. The algorithm has the following input parameters: an initial temperature $T^{(0)}$, a cooling schedule ρ in $[0, 1]$, and a stop condition T^{end} , in next section we specify how we set those parameters.

2.3 Echo State Neural Networks

A Recurrent Neural Network (RNN) is a bio-inspired dynamical system used for solving temporal learning problems. The recurrences allow to the network to learn complex dynamics and to model systems that evolve in time. Besides, the model has been also successfully applied for solving any type of supervised learning problems. Despite the potential of the RNN for solving supervised tasks, they have been seldom applied in real-world applications, due to the fact that often can be hard to set-up the network parameters. First-order methods (optimisation techniques based on the gradient information) have been appropriated for training feedforward networks, although they can fail in the case of recurrent networks [9]. An alternative of the RNN has been introduced at the beginning of the 2000s with the name of *Echo State Networks (ESN)* [10]. The technique uses the power of RNNs for memorising temporal data and overcomes the drawbacks of training the weights of RNNs, without introducing additional inconveniences. For that reasons, the model is a good alternative for tackling temporal learning tasks.

The network has three layers connected in a forward schema. The first layer typically process the input patterns. The second layer contains recursive connections, and its role is memorising the temporal structure of the patterns and expanding their geometrical information from the input layer in a higher dimensional space. The third layer generates a linear combination of the expansion created by the second layer. The ESN has circuits only in the second layer, which is named *reservoir*. A main characteristic of the model is that the training algorithm only focuses in adjusting a subset of weights, only the weights to the third layer are adjusted. All the rest connections (input and reservoir weights) are random initialized following some algebraic conditions and they are fixed

during the learning process. As a consequence, the learning algorithm is fast and robust because it consists in training the parameters of a linear regression. The literature about ESN is very rich, and we can find several applications of the model that show the well performance of ESN for solving temporal learning tasks [11].

We follow by specifying the notation, let N_a , N_x and N_o be the number of input, reservoir and output neurons, respectively. The parameters of the model are the weight matrices, let \mathbf{w}^{in} be a $N_x \times N_a$ matrix collecting input-reservoir weights, let \mathbf{w}^{r} be a $N_x \times N_x$ matrix collecting hidden-hidden weights, and let \mathbf{w}^{out} a $N_o \times (N_a + N_x)$ matrix with the parameters from input and projected space to the output space. The reservoir is characterized by a multidimensional state $\mathbf{x} = (x_1, \dots, x_{N_x})$ given by:

$$x_m(t) = \psi\left(w_{m0}^{\text{in}} + \sum_{i=1}^{N_a} w_{mi}^{\text{in}} a_i(t) + \sum_{i=1}^{N_x} w_{mi}^{\text{r}} x_i(t-1)\right), \quad (3)$$

for all $m \in [1, N_x]$ where $\psi(\cdot)$ is the hyperbolic tangent function ($\tanh(\cdot)$). Let $\mathbf{y}(t)$ be the prediction N_o -dimensional vector of the model at time t , which is computed by a linear regression:

$$y_s(t) = w_{s0}^{\text{out}} + \sum_{i=1}^{N_a} w_{si}^{\text{out}} a_i(t) + \sum_{i=1}^{N_x} w_{si}^{\text{out}} x_i(t), \quad \forall s \in [1, N_o]. \quad (4)$$

In our experimental results we use a generalisation of the canonical ESN that computes the reservoir state as follows: firstly, we compute a temporarily vector state \mathbf{x}' using the expression (3). Secondly we compute the state given by:

$$x_m(t) = (1 - \alpha)x'_m(t) + \alpha x_m(t-1), \quad (5)$$

where the parameter α is called leaky rate and is used for controlling the reservoir state update.

The ESN model has the following global parameters that impact in the model performance: the size of the reservoir (given by the number of reservoir neurons), the input scaling factor (a weighting factor of the input patterns), the spectral radius of the reservoir matrix, the density and topology of the reservoir matrix [11, 12]. The reservoir size impacts in the linear separability of the data, there is a tradeoff between the large of the reservoir and overfitting. In our experiments, the training data is normalised. We consider the input scaling factor equal to 1, therefore all the input patterns have equal relevance. The spectral radius controls the stability of the reservoir state and impacts in the memory capacity of the model. An important property of the model is that the stability of the dynamical system $\mathbf{x}(t)$ only depends of the reservoir weight matrix \mathbf{w}^{r} [11], . The stability is controlled by the spectral radius of \mathbf{w}^{r} , that we denote by $\rho(\mathbf{w}^{\text{r}})$, if $\rho(\mathbf{w}^{\text{r}}) < 1$ the stability of the ESN can be ensured [11]. An usual practice consists in scaling the initial reservoir, in order to control the spectral radius, the scaling procedure is as follows: $\mathbf{w}^{\text{r}} \leftarrow (\beta/\rho(\mathbf{w}^{\text{r}}))\mathbf{w}^{\text{r}}$, where β is a constant in $(0, 1]$. The sparsity of the reservoir matrix is often set on 20 % non-zero values.

3 Methodology

The section is divided in two parts, the first one present the procedure applied for setting the global parameters of the ESN. The second part contains the used methodology of this article.

3.1 Setting of the Global ESN Parameters

We begin by finding the *best* global parameters of the ESN model. We arbitrary select three parts of the solar global irradiance time-series. The selection was made considering the different trends of the 2015. A first part A has a grown increasing trend in an arbitrary range of time $[a_1, a_2]$ (days in February and March), a second part B doesn't present any evident trend in $[b_1, b_2]$ (days of May and June), and a third part C has a downward trend in $[c_1, c_2]$ (days of October and November). For each period A , B and C we compute ESNs with different global parameters ($N_x, \rho(\mathbf{w}^r), \alpha$), and we evaluate their accuracy using the MSE (as an averaged error of the three parts). The global parameters of the ESN are computed using the model for forecasting three days ahead. We forecast the solar power using only information of the past of the solar power series, in other words we don't use any other meteorological variables. The evaluated ESN parameter values are defined in a regular spaced-grid points in the following intervals: $\alpha \in [0.5, \dots, 0.9]$, $N_x \in [30, 35, \dots, 120, 125]$ and $\rho(\mathbf{w}^r) \in [0.1, 0.15, \dots, 0.95]$. Let N_x^* , ρ^* and α^* be the *best* global parameters of an ESN according our empirical evaluations. A remark, the evaluations for reservoir matrices with $\rho(\mathbf{w}^r) > 0.55$ were in some cases unstable. This means that the accuracy presented a large variance, therefore we analyse only the results for $\rho(\mathbf{w}^r) \leq 0.55$.

3.2 Feature Selection Using SA Method

We apply the SA method for automatically selecting other meteorological variables for forecasting the solar irradiance. We assume that several external variables impact in the solar irradiance, such as: air temperature, humidity, wind characteristics, etc. Therefore, we use SA as feature selection tool for defining a set of meteorological variables. The selection can not be done in reasonable time using a brute-force strategy or a greedy method due to the large number of variables (in our experiments, we are using more than 20 variables). The procedure for using SA is as follows. Without loss of generality we enumerate the input features by $\{1, \dots, N\}$, where N is the number of meteorological variables including the solar power. As a consequence, the searching space is $\{0, 1\}^N$, where the solutions have the form $\mathbf{s} = [s_1, s_2, \dots, s_N]$ where $s_i = 0$ represents that the input feature i is omitted as input of the ESN, and $s_i = 1$ represents that the variable i is an input of the model. For each combination we evaluate the accuracy of an ESN with parameters N_x^* , ρ^* and α^* , the objective is to find $\mathbf{s} \in \{0, 1\}^N$ such that the MSE is minimized. In the SA method, given a current solution \mathbf{s}^{curr} we must select a *nearby* solution of \mathbf{s}^{curr} that we denote by \mathbf{s}^{new} . In this step, we random select a set \mathcal{D} of d integer values in $[1, N]$. Next, we define

the nearby solution \mathbf{s}^{new} as $s_j^{\text{new}} = s_j^{\text{curr}}$ for all $j \notin \mathcal{D}$ and $s_j^{\text{new}} = s_j^{\text{curr}} + 1 \bmod 2$ for all $j \in \mathcal{D}$ where mod is the module function.

Our main goal is developing a device for predicting future values in a period Δt using information until a current time. Therefore, given an explanatory variable $\mathbf{a}(t)$ and the target $y(t)$ until time t , we predict the solar irradiance value at time $t + 1$ ($\hat{y}(t + 1)$), we use $\mathbf{a}(t)$, $y(t)$ and $\hat{y}(t + 1)$ for predicting $\hat{y}(t + 2)$, $\mathbf{a}(t)$, $y(t)$, $\hat{y}(t + 1)$ and $\hat{y}(t + 2)$ for predicting $\hat{y}(t + 3)$, and so on. We assume that after a period Δt , we are able to have new measured values for the explanatory variables (\mathbf{a}). In other words, we use also other meteorological variables, for instance temperature, at time t for predicting the solar irradiance at time $t + \Delta t$, and so on. We divide the time-series in two parts. The first part (named training) is used for finding the best configuration of the input features and the best global ESN parameters. The second part (named validation) is used for evaluating the adjusted model. We use the fitted model for predicting the values on the validation time-series, and the predicted values of power solar as well as the other meteorological variables are used as input patterns for predicting new values. We set Δt with the value of three days. All codes for data processing have been developed in Matlab (Mathworks Inc. Natick, Ma, USA).

4 Experimental Results

The first part of this section contains a description of the data, the second one presents our experimental results.

4.1 Data Description

We use the meteorological data provided by the National Renewable Energy Laboratory and Solar Technology Acceleration Center (SolarTAC) [7]. The collected data corresponds to the period started in January 1, 2015 till December 5, 2015. The temporal precision of the data is 1 min. The output variable is the global irradiance given by the *Global Horizontal Irradiance* in W/m^2 , the input features are: Air Temperature, Wind Chill Temp, Dew Point Temp, Relative Humidity, Wind Speed, Pk Wind Speed, SDev Wind Speed, Wind Direction, Wind Dir at Pk WS, SDev Wind Direction, Station Pressure, Precipitation, Accumulated Precipitation, Zenith Angle, Azimuth Angle, Airmass, CMP22 Temp, CR1000 Temp, CR1000 Battery, and CR1000 Process Time. More information about those variables and the used protocol for collecting the data see is available in [7]. The preprocessing of the data consisted in changing the temporal precision from 1 min to 10 min. Instead of using the variable information each minute, we consider the data each 10 min. The time-series data has 50232 points in this period. All the variables were normalised in $[0, 1]$. Figure 1 presents the three periods used for setting the parameters of the ESN model. Due to the fact that SA is a metaheuristic technique we evaluate our approach of different 30 experiment trials. For each one, we start the SA method by randomly selecting the half part of the input features.

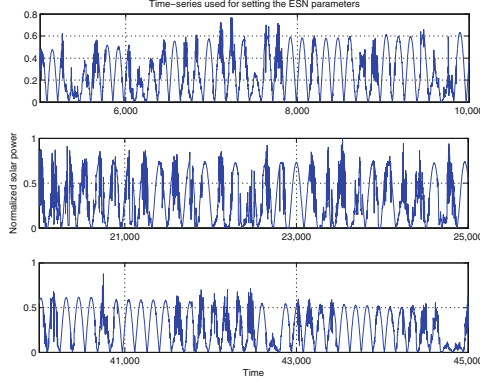


Fig. 1. Training data used for finding the *best* ESN global parameters. The first graphic covers the period since Feb. 3 till Mar. 10, the second graphic covers the period since May 18 till Jun 22, and the third graphics covers the period since Oct. 4 till Nov. 8.

4.2 Results Analysis

As example, we present in Fig. 2 the accuracy of the model when two leaky parameters (α in expression (5)) are evaluated. The graphic shows the MSE on the validation data (three days ahead) for different size of the reservoir and spectral radius. We can see that models with large reservoir size can provoke overfitting on the training data, as a consequence they can have low accuracy for modelling the validation data. According to the results, we set the parameters as follows: $\alpha^* = 0.8$, $N_x^* = 40$ and $\rho^* = 0.25$. A large leaky parameter (0.8) means that a better accuracy is reached when the reservoir state is gradually updated, that is weighting only with 0.2 the new information at each step given by expression (3). Figure 3 illustrates how SA improves the model by selecting a better configuration of input features. The vertical axis shows the $\log(MSE)$ and the horizontal axis represents the first 80 iterations. The different curves of Fig. 3 represent different SA experiments. We can see for all cases how the error decreases with the number of iterations. We set the Hamming distance between the current solution and the near solution with $d = 3$. The maximum number of iterations of the SA was 400. A remark, in the SA algorithm we guarantee that the solar power data is always an input feature of the ESN model. Figure 4 presents the evolution of the number of input features by the model over the first 400 iterations. For a better visibility we present as example only 5 random selected SA trials. For instance, the blue curve of Fig. 4 shows how at the iteration 51 of the SA method, the best ESN solution has 12 input features, and at the next step (iteration 52) the best solution has only 6 input features. Table 1 presents the performance on the validation dataset for forecasting 3 days ahead, according to different number of iterations of the SA method. The first column is the number of iterations, the second column shows the best reached accuracy among the 30 SA trials. The next two columns are the mean and the variance of the MSE among the 30 SA experiments. In addition the table shows the number

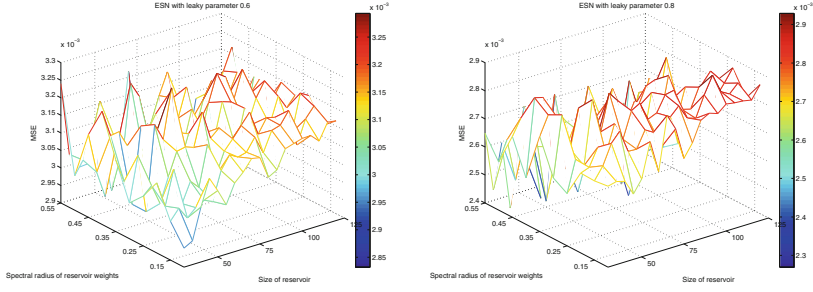


Fig. 2. Sensitivity analysis of the ESN parameters. Example of the accuracy on the validation data reached by two ESNs with leaky rate 0.6 and 0.8 and parameters $N_x \in [30, 125]$ and $\rho(\mathbf{w}^r) \in [0.1, 0.55]$.

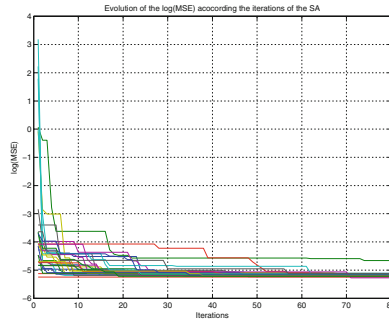


Fig. 3. Evolution of the model accuracy ($\log(\text{MSE})$) over the first 80 iterations of the SA algorithm. Each curve represents the evolution for different initial points of the SA technique

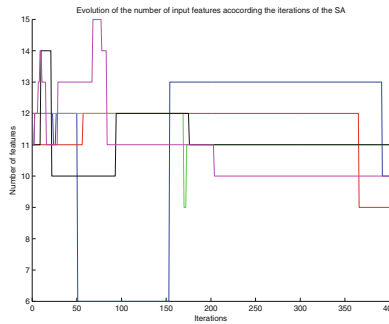


Fig. 4. Example of the evolution of the number of variables used by the SA method in the first 400 iterations.

Table 1. Accuracy of the proposed method when is forecasted three days ahead. The accuracy is presented according the number of iterations in SA algorithm. The columns 2, 3 and 4 are presented using scientific notation.

Iteration	Min (10^{-4})	Mean (10^{-4})	Var (10^{-8})	Number of features
50	5.2612	6.1036	1.0421	12
100	5.1195	5.7287	0.2834	10
150	4.9987	5.5482	0.0891	12
200	4.9986	5.4859	0.08349	10

of input features used by the best configuration at the iterations 50, 100, 150, 200. The lowest reached MSE was 4.9986633×10^{-4} computed using free running prediction over three days. The best combination of input features reached with 400 iterations of SA was composed by the variables: global horizontal irradiance, air temperature, wind chill temp, dew point temp, relative humidity, Pk wind speed, standard deviation of wind speed, accumulated precipitation, Zenith angle, Azimuth angle, CR1000 Temp, and CR1000 Process Time. For more information about those variables see [7].

5 Conclusions and Future Work

We present a procedure for forecasting the solar power irradiance using several external meteorological variables. The approach uses the well-known metaheuristic technique *Simulating Annealing* (SA) for selecting the most significant input features, as well as a specific type of Recurrent Neural Network named *Echo State Networks* (ESN) for forecasting the time-series. We evaluate the proposed method over a real meteorological dataset provided by the Solar Technology Acceleration Center (SolarTAC), Colorado, USA. The SA technique automatically finds a good combination of meteorological variables, which affect the solar power estimation. We consider that we obtain promising results for a forecasting horizon of three days. We are interested in the near future to analyse the group of meteorological variables computed by SA, as well as to extend the period used for training the network model.

Acknowledgement. This work was supported by Grant of SGS No. SP2016/97, VŠB-Technical University of Ostrava, Czech Republic.

References

1. Pedro, H.T.C., Coimbra, C.F.M.: Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy* **86**(7), 2017–2028 (2012)
2. Basterrech, S., Prokop, L., Buriánek, T., Misak, S.: Optimal design of neural tree for solar power prediction. In: 15th International Scientific Conference on Electric Power Engineering (EPE), Proceedings of the 2014, pp. 273–278, May 2014

3. Basterrech, S., Zjavka, L., Prokop, L., Misak, S.: Irradiance prediction using echo state queueing networks and differential polynomial neural networks. In: 13th International Conference on Intelligent Systems Design and Applications (ISDA), 2013, pp. 271–276, December 2013
4. Diagne, M., David, M., Lauret, P., Boland, J., Schmutz, N.: Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renew. Sustain. Energy Rev.* **27**, 65–76 (2013)
5. Letendre, S., Makhoun, M., Taylor, M.: Predicting solar power production: irradiance forecasting models, applications and future prospects. Technical report, Solar Electric Power Association, Washington, DC, USA, March 2014. www.solarelectricpower.org
6. Pelland, S., Remund, J., Kleissl, J., Oozeki, T., De Brabandere, K.: Photovoltaic, solar forecasting: state of art. Technical Report IEA-PVPS T14-01: 2013, International Energy Agency Photovoltaic Power Systems Programme (2013). <http://www.iea-pvps.org>
7. Andreas, A., Wilcox, S.: Aurora, Colorado (data). Technical Report DA-5500-56491, Solar Technology Acceleration Center (SolarTAC), Colorado, USA (2011). doi:[10.5439/1052224](https://doi.org/10.5439/1052224)
8. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C++: The Art of Scientific Computing. Cambridge University Press, Cambridge (2002)
9. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks* **5**(2), 157–166 (1994)
10. Jaeger, H.: The “echo state” approach to analysing and training recurrent neural networks. Technical report 148, German National Research Center for Information Technology (2001)
11. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009)
12. Butcher, J.B., Verstraeten, D., Schrauwen, B., Day, C.R., Haycock, P.W.: Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural Networks* **38**, 76–89 (2013)

Artificial Intelligence and Soft Computing
15th International Conference, ICAISC 2016, Zakopane,
Poland, June 12-16, 2016, Proceedings, Part I
Rutkowski, L.; Korytkowski, M.; Scherer, R.;
Tadeusiewicz, R.; Zadeh, L.A.; Zurada, J.M. (Eds.)
2016, XXIV, 786 p. 234 illus., Softcover
ISBN: 978-3-319-39377-3