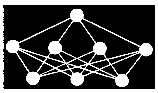


SUPERVISED LEARNING OF PHOTOVOLTAIC POWER PLANT OUTPUT PREDICTION MODELS

Lukáš Prokop[*], *Stanislav Mišák*[*], *Václav Snášel*[†‡], *Jan Platoš*[†‡], *Pavel Krömer*[†‡]

Abstract: This article presents an application of evolutionary fuzzy rules to the modeling and prediction of power output of a real-world Photovoltaic Power Plant (PVPP). The method is compared to artificial neural networks and support vector regression that were also used to build predictors in order to analyse a time-series like data describing the production of the PVPP. The models of the PVPP are created using different supervised machine learning methods in order to forecast the short-term output of the power plant and compare the accuracy of the prediction.

Key words: *Supervised learning, fuzzy rules, regression, artificial neural networks, support vector machines*

Received: July 4, 2013
Revised and accepted: September 10, 2013

1. Introduction

Fuzzy sets and fuzzy logic can be used for soft classification of data. In contrast with crisp classification, which leads to crisp decisions about the data, fuzzy classification allows a more sensitive data analysis [1]. Fuzzy decision trees and if-then rules are an example of efficient, transparent, and easily interpretable fuzzy classifiers [1, 29].

Genetic programming [16, 17] is a powerful machine learning technique from the wide family of evolutionary algorithms. In contrast with traditional evolutionary algorithms, it can be used to evolve complex hierarchical tree-like structures and

[*]Lukáš Prokop, Stanislav Mišák
Department of Electric Power Engineering, Faculty of Electrical Engineering and Computer Science, VŠB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic, E-mail: {lukas.prokop,stanislav.misak}@vsb.cz

[†]Václav Snášel, Jan Platoš, Pavel Krömer
Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VŠB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic, E-mail: {vaclav.snasel,jan.platos,pavel.kromer}@vsb.cz

[‡]Václav Snášel, Jan Platoš, Pavel Krömer
IT4Innovations, European Center for Excelence, 17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic, E-mail: {vaclav.snasel,jan.platos,pavel.kromer}@vsb.cz

symbolic expressions. It has been used to evolve Lisp S-expressions, mathematical functions, and general symbolic expressions including crisp and fuzzy decision trees. Recently, genetic programming has been used to infer search queries describing fuzzy sets of relevance ranked documents in an information retrieval system.

Query evolution can be used for general data mining [20]. The extended Boolean queries (i.e. weighted Boolean expressions) can be interpreted as symbolic fuzzy rules that describe a fuzzy subset of a data set by means of its features and combinations of features. Moreover, a fuzzy rule evolved using a training data set can be later used for an inexpensive analysis of new data to e.g. predict quality of products, detect harmful actions in a computer network, assign labels to data, and estimate the values of an output variable.

The artificial evolution of search expressions is interesting because genetic programming has shown a very good ability to find symbolic expressions in various problem domains. The general process of fuzzy rule evolution can be used to evolve custom rules for different data classes and various data sets with different properties and with different internal structure. The evolved fuzzy rules can be used as standalone data labeling tools or e.g. to participate on a collective decision of an ensemble of data classification methods.

This article presents an application of evolutionary fuzzy rules to the modeling and prediction of power output of a real-world Photovoltaic Power Plant (PVPP).

1.1 Challenges of (smart) power grid

A power grid must be operated with balanced energy levels. The electrical energy produced by energy sources within the network must be at the same time consumed by customers. The accumulation of reasonable quantities of electrical energy is currently still technically and financially too demanding, even though experimental systems are installed at prototype energy storage facilities and there are major research efforts to find advanced ways of accumulation of large quantities of electrical energy [24, 13].

Nowadays, the energetic balance is mostly achieved by the regulation of sources of electrical energy because the consumption is usually beyond grid operators control. The power grid consists of power plants with stable production of electrical energy such as coal, gas, and nuclear power plants. On the other hand, it contains power plants with unstable (stochastic) energy production whose output heavily depends on the meteorological conditions at given time and in given location. Examples of unstable energy sources include wind power plants and photovoltaic power plants. The amount of the electrical energy produced by such power plants changes with changing weather conditions significantly.

A power grid operator has to maintain a reliable, safe, and efficient operation of the electrical network. In order to meet this objective, the operator must be able to estimate the volume of electrical energy produced by unstable energy sources. In a power grid with a plenty of unstable energy sources, a reliable prediction is needed in order to ensure that the regulation of the stable sources will balance the production of intermittent energy sources and satisfy the demand for electricity. Otherwise, the power grid could became unstable and unreliable. The accommodation of renewable energy sources, robustness, self-healing and real-time

optimization capability are among the key attributes of the upcoming smart grids. The operation of next generation smart grids is directly linked to advanced optimization and prediction methods featuring computational intelligence [24].

In this work, we genetically evolve a fuzzy predictor in the form of a fuzzy rule inspired by the area of information retrieval. The same concept was successfully used for data classification in [23, 28, 19, 22] and initially for PVPP output estimation in [21]. When compared to the more complex fuzzy classifier systems, it can be seen as a sole symbolic fuzzy expression that maps data features onto an output value from the interval $[0, 1]$. The fuzzy rule is in this research enhanced by the ability to process data as an ordered (time-like) series of records and it is used to estimate the power output of a PVPP. The usefulness of this approach is illustrated on an experiment with a real world PVPP. To provide a comparison of the proposed method with selected traditional regression and prediction approaches, the PVPP output prediction by fuzzy rules is compared to prediction provided by a feed-forward artificial neural network (multilayer perceptron) and support vector regression.

2. Fuzzy rules for time series analysis evolved by genetic programming

The design of fuzzy classifiers and fuzzy rule-based systems has been successfully aided by the nature inspired methods in the recent years. In this section we summarize few examples of such an evolution or more generally examples of nature inspired fuzzy classifier design. For a comprehensive survey on the automated evolution of fuzzy classification tools see e.g. [4]. Multi-objective evolutionary algorithms were used for the evolution of linguistic fuzzy rule-based classification systems in the work of Cordón et al. [5]. Another multi-objective evolutionary approach to the evolution of fuzzy rule-based systems was proposed by Ishibuchi and Nojima [12]. They used a hybrid 2-stage approach that combined an initial heuristic stage to select fuzzy rules and evolutionary stage to optimize and tune the system.

Wang et al. [30] used genetic algorithms to integrate fuzzy rule sets and membership functions learned from various information sources. In [7], Freischlad et al. used an evolutionary algorithm to generate fuzzy rules for knowledge representation. Zhou and Khotanzad [31] used genetic algorithm to learn various parameters of fuzzy classification system from a training data set. The usage of another nature inspired method - the particle swarm optimization - to fuzzy classification system design was studied recently in [26]. In this study, we use genetic programming to find a fuzzy rule to forecast the output volumes of a PVPP. The operation of photovolatic panels has a time-series like behavior. Current output depends not only on momentary solar radiation but also on the state of the panel which can be determined from previous solar radiation and previous output. Therefore, the fuzzy rules were extended by time-series processing capabilities.

2.1 Time series analysis

Time series modeling and forecasting is a complex task with a number of applications in data analysis, planning, control, and optimization [3]. There are many

exact (hard) and stochastic (soft) methods for time series modeling, analysis, and prediction. The exact methods include state space models, growth curve models, ARIMA models, single equation models, vector AR and ARMA models, and econometric models [3, 25].

More recently, various computational intelligence methods such as different types of artificial neural networks, fuzzy logic, evolutionary computation, and many hybrid approaches have been used for time series analysis [25].

2.2 Fuzzy rules for time series data analysis

The fuzzy rules use similar data structures, basic concepts, and operations as the fuzzy Information retrieval (IR) [18] and they are applied to general data processing (i.e. classification, prediction, and so forth).

The data used by the fuzzy rule is a real valued matrix. Each row of the matrix corresponds to a single data record which is interpreted as a fuzzy set of features in IR language we can interpret a row of the matrix as a document and features as words weight [18]. Such a general real valued matrix $\boldsymbol{D}$ with m rows (data records) and n columns (data features) can be mapped to an IR index that describes a collection of documents.

The fuzzy predictor has the form of a weighted symbolic expression roughly corresponding to an extended Boolean query in the fuzzy IR analogy. The predictor consists of weighted feature names and weighted aggregation operators. The evaluation of such an expression assigns a real value from the range $[0, 1]$ to each data record. Such a valuation can be interpreted as an ordering or a fuzzy set over the data records.

2.3 Fuzzy rule structure

The fuzzy rule is a symbolic expression that can be parsed into a tree structure. The tree structure consists of nodes and leafs (i.e. terminal nodes). In the fuzzy rules for time series analysis, three types of terminal nodes are defined:

- *feature* node - which represents the name of a feature (a search term in the IR analogy). It defines a requirement on a particular feature in the currently processed data record.

- *past feature* node - which defines a requirement on certain feature in a previous data record. The index of the previous data record (current – 1, current – 2 etc.) is a parameter of the node.

- *past output* node – which puts a requirement on a previous output of the predictor. The index of the previous output (current – 1, current – 2) is a parameter of the node.

The last two node types allow the fuzzy predictor to take into account the order of the data samples, i.e. to see it as a complex time series rather than a simple valuation of unordered records in the data base. Consider the following example of the fuzzy predictor:

feature1:0.5 and:0.4 (feature2[1]:0.3 or:0.1 ([1]:0.1 and:0.2 [2]:0.3))

In the inline syntax, the feature node is defined by feature name and its weight (*feature1:0.1*), past feature node is defined by feature name, index of previous record, and weight (*feature2[1]:0.3*), and past output node is defined by the index of previous output and weight (*[1]:0.5*). The tree that corresponds to the example given above is shown in Fig. 1.

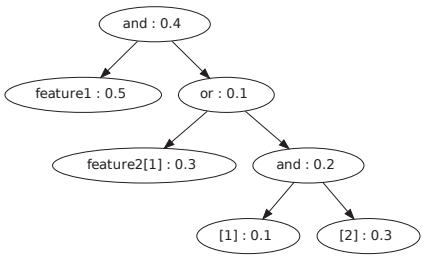


Fig. 1: Tree form of a fuzzy predictor

The operator nodes supported currently by the fuzzy predictor are *and*, *or*, and *not* node. Both nodes and leafs are weighted to soften the criteria they represent.

The fuzzy rule with past feature nodes and past output nodes can effectively express requirements on past feature values and past output values and therefore allow the predictor to analyse the stream of records as an ordered sequence similar to a time series.

2.4 Fuzzy rule evolution

The fuzzy rules are evolved in supervised manner by a straightforward application of genetic programming [20, 21]. Every fuzzy rule was represented by a tree-like chromosome. Genetic operators were applied to the nodes of the chromosomes. Crossover operator was implemented as a mutual exchange of randomly selected sub-trees of parent chromosomes. Mutation aimed to modify the chromosomes by pseudo-random arbitrary changes in order to broaden the coverage of the fitness landscape and prevent premature convergence. Mutation was implemented using a random combination of various modifications of chromosome structure including:

i) removal of a sub-tree at a randomly chosen node

ii) replacement of a randomly chosen node by a newly generated sub-tree

iii) replacement of node instruction by a compatible node instruction (i.e. a terminal can be replaced by another terminal, a function can be replaced by another function of the same arity)

The information retrieval measure F-Score [28, 20] was used as fitness function.

3. Traditional machine learning methods

To be able to compare the PVPP output forecast obtained by evolutionary fuzzy rules with other more traditional computational intelligence methods, two widely used supervised regression algorithms were trained and evaluated on the PVPP data set. Artificial neural networks were previously used for the prediction of power grid variables [9, 8] and support vector regression is known to be able to approximate linear and non-linear functions very well [14, 2, 10]. Moreover, they represent two different yet highly successful approaches to supervised learning from data.

In this section, we briefly outline the basic principles of simple feed-forward artificial neural networks and support vector regression.

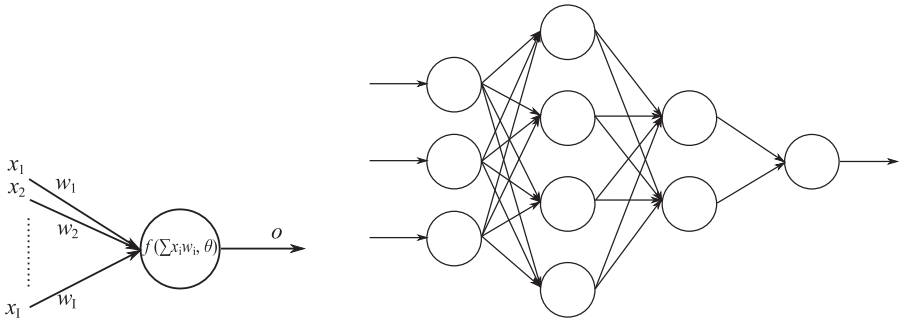
3.1 Multilayer perceptron

Artificial neural networks (ANNs) constitute a family of computing models based on the emulation of electrochemical processes observed in neural systems of living organisms [15, 6].

Various types of complex general-purpose artificial neural networks composed of simple computing units - artificial neurons - have been proposed. Artificial neurons emulate the behavior of biological neurons in terms of signal processing (aggregation, thresholding, modification, propagation etc.) and information storage (input weights, activation function parameters). A single artificial neuron (perceptron) represents a non-linear mapping $f : \mathbb{R}^I \to \mathbb{R}$ and it can be used to solve linearly separable problems. A schematic view of a perceptron is shown in Fig. 2a. In the figure, x_i represents input signal, w_i input connection weights, $f(\sum x_i w_i, \theta)$ activation function and o the output signal. The perceptron in Fig. 2a is a summation unit (i.e. it performs a weighted sum of input signals) and it can be used to solve linearly separable problems. The perceptron must be set-up (trained) before it can be used to process data. The training involves setting the input connection weights w_i and activation function parameters.

An ANN is a layered network of artificial neurons with certain topology [6]. It implements a non-linear mapping $f_{ann} : \mathbb{R}^I \to \mathbb{R}^K$ from I-dimensional input space to K-dimensional output space [6]. In contrast to single perceptron, ANN is able to solve problems that are not linearly separable and in general to provide an approximation of a function. ANNs have been used for countless applications in data mining, pattern recognition, data classification, control and so fort. Multilayer perceptron (MLP) is a basic type of multilayer feed-forward ANN [6] that consists of multiple fully-connected layers of perceptrons. The MLP consists of the input layer, one or more hidden layers, and the output layer (see Fig. 2b).

The ANN training methods include supervised, unsupervised, and reinforced learning. The well-known backpropagation (BP) algorithm is an example of supervised learning method based on gradient descent optimization of network parameters [6]. The BP training algorithm consists of a number of iterations in which the network (that was originally randomly initialized) first processes training patterns and computes the error for each pattern and second performs the backward propagation of error in which the weights are adjusted as a function of the error



(a) A perceptron. (b) An example of a 3-4-2-1 MLP.

Fig. 2: Perceptron and multilayer perceptron (feed-forward network).

signal. The training is finished when the terminating criteria (i.e. the maximum number of iterations was reached, training error was small enough) are met. Other parameters of the BP algorithm include the learning rate which is the size of each learning step and momentum that controls how the network avoids fluctuations when processing different training patterns during stochastic learning. For more details on the MLPs and the BP algorithm see e.g. [6, 15].

3.2 Support vector regression

Support vector regression (SVR) is an extension of support vector machines (SVM), a family of popular supervised machine learning tools based on statistical learning theory originally proposed by Vapnik [14, 2]. SVM were designed to find an optimal directed hyperplane separating two non-overlapping classes of data with the help of support vectors (i.e. the points in the data closest to the separating hyperplane) [2]. However, later extensions enabled the SVM to learn and classify multiple classes of data, overlapping classes, and noisy data by the introduction of slack variables ξ_i, ξ_i^* that enable soft-margin classifiers [10, 2].

The SVM uses a linear separating hyperplane to construct a classifier with maximum margin by the means of constrained non-linear optimization [14]. Data that is not lineary separable can be processed by the SVM with the help of kernel substitution, i.e. a translation of input data to a high-dimensional feature space where it might be lineary separable [11, 2]. The SVM combine both, success in practical applications and well-established theory.

The basic SVM for binary classification aims to learn a decision function [2]

$$f(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x} + b) \tag{1}$$

where $\cdot$ is dot product, $\mathbf{x}$ is the set of input data vectors (points) $x_1, x_2, \ldots, x_m$, $f(\mathbf{x})$ is the vector of corresponding labels $y_1, y_2, \ldots, y_m$, subject to $y_i = \pm 1$, $sign$ is the signum (sign) function, and $\mathbf{w}$ is vector of weights. In a geometrical representation, the hyperplanes $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$ are called canonical hyperplanes and the area between them margin band. Maximizing the margin (i.e. finding

optimal hyperplane) involves maximization of the function

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \tag{2}$$

subject to

$$\alpha_i \geq 0, \qquad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{3}$$

where K is a kernel used for mapping of input data to high-dimensional feature space (kernel substitution) and α_i, α_j are Lagrange multipliers. Bias b is given by [2]

$$b = -\frac{1}{2} \left[\max_{\{i|y_i=-1\}} \left(\sum_{j=1}^{m} \alpha_j y_j K(x_i, y_i) \right) + \min_{\{i|y_i=1\}} \left(\sum_{j=1}^{m} \alpha_j y_j K(x_i, y_i) \right) \right] \tag{4}$$

In contrast with SVM, SVR aims to learn the mapping of data to real-valued labels [2, 10, 27]. The ϵ-SVR algorithm aims to learn a function $f(x_i)$ that has at most ϵ deviation from corresponding y_i [27]

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{5}$$

and leads to maximization of linear or quadratic ϵ-insensitive loss function. The linear ϵ-insensitive loss function is given by [2]

$$\begin{aligned} W(a, a^*) = & \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) - \epsilon \sum_{i=1}^{m} (\alpha_i + \alpha_i^*) \\ & - \frac{1}{2} \sum_{i,j=1}^{m} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) \end{aligned} \tag{6}$$

subject to

$$\sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{m} \alpha_i^*, \quad \alpha_i, \alpha_i^* \in [0, C] \tag{7}$$

The ϵ-SVR can be visualised as a tube around hypothesis function which outlines training errors from valid training points.

4. Experiments

A series of computational experiments was conducted in order to evaluate the ability of evolutionary fuzzy rules to forecast the production of a photovoltaic power plant and to compare fuzzy rules to other well known regression methods. In order to do so, the volumes of electrical energy produced by a PVPP located in the North Moravia, Czech Republic and the values of solar radiation in the same

location were recorded. The value of electric energy is obtained by time integration of output power, the unit of electric energy is $W.h$. The unit for solar radiation is $W.m^{-2}$.The values were recorded in 10 minute intervals between November 2010 and April 2011. The data set contained 21515 records. After initial analysis, we removed records from days with irregular PVPP operations i.e. when the facility was disconnected from the grid but sensors were active. The remaining 20513 records were divided into two halves. The first part containing 10257 records was used as a training data set for predictor evolution and the second part containing 10256 records was used as test data set.

MLP and SVR were also used to build a prediction model from the data set in order to see how precise are the forecasts by evolutionary fuzzy rules in comparison with traditional machine learning methods. Both machine learning algorithms were executed in two different configurations selected on the basis of previous experiments and best practices. The parameters of the algorithms as well as Fuzzy rules (FR) parameters are summarized in Tab. I.

Algorithm	Parameters
FR	Fuzzy rules evolved with population size 100, crossover probability P_C 0.8, mutation probability P_M 0.2, mutation implementation as shown in Tab. II, generations limit 1000, past feature limit 20 (max 20 previous values of each feature will be considered), past output limit 20 (max 20 previous predicted values will be considered), fitness function F-Score with $\beta = 1$
MLP1	Three-layered 2-2-1 multilayer perceptron, backpropagation, 5000 training epochs, learning rate 0.3, momentum 0.2
MLP2	Same as MLP1, learning rate 0.03, momentum 0.02
SVR RBF	ϵ-SVR with radial basis function kernel $e^{-\gamma\|u-v\|^2}$, $\epsilon = 0.01$, cost $C = 1.0$, and $\gamma = 1$
SVR POLY	ϵ-SVR with polynomial kernel $(\gamma u'v+c_0)^d$, $\epsilon = 0.01$, cost $C = 1.0$, $\gamma = 1$, degree $d = 3$, and coefficient $c_0 = 0$

Tab. I: Algorithms and settings.

Event	Probability
Generate term	0.5
Generate op. AND	0.24
Generate op. OR	0.24
Generate op. NOT	0.02

(a) Probabilities of generating random fuzzy rule nodes.

Event	Probability
Mutate node weight	0.5
Insert or delete NOT node	0.1
Replace with another node or delete NOT node	0.32
Replace with random branch	0.08

(b) Probabilities of mutation operations.

Tab. II: Random rule generation an mutation probabilities.

4.1 Experimental evaluation

Power production forecast models were created by all aforementioned algorithms for full data set and training data set in independent runs. The average prediction error for the full data set, for the training data set, and for the test data set is shown in Tab. III. The prediction error is in all cases lower than or just slightly higher than 2% of the peak output of the PVPP, which is a good result, considering that only a single input variable (current solar radiation intensity), its past values and past estimates of the output variable were available. The error for the full data set illustrates the ability to create predictors when the information about the entire period is available. We note that the environment characteristics that affect the operations of the PVPP (e.g. sun elevation, wind speed, direction, and temperature) change during the year. The training error shows how the predictors managed to approximate the same data that was used for training and the test error shows how well could the predictor evolved using the training set forecast the electric power output of the PVPP in the period covered by the test data set (i.e. how well can it generalize). All investigated algorithms reached very similar

Data set	Average prediction error (W)				
	FR	MLP1	MLP2	SVR RBF	SVR POLY
Full	13589.4	*13552.4*	13712.2	14301.8	14330.1
Training	10001.4	*10000*	12254.2	10500.5	10552.4
Test	*18313.2*	22033.2	22811.4	20279.9	20181.6

Tab. III: Average PVPP output prediction error.

results. Moreover, when evaluating the prediction, we noted that both, the training data and the test data still contained anomalies. Typical errors in the training data set are shown in Fig. 3. The figures show a 24-hour window in the data set that roughly corresponds to one business day. The errors were caused by power meter malfunctions and problems with data acquisition. The examples of anomalies in

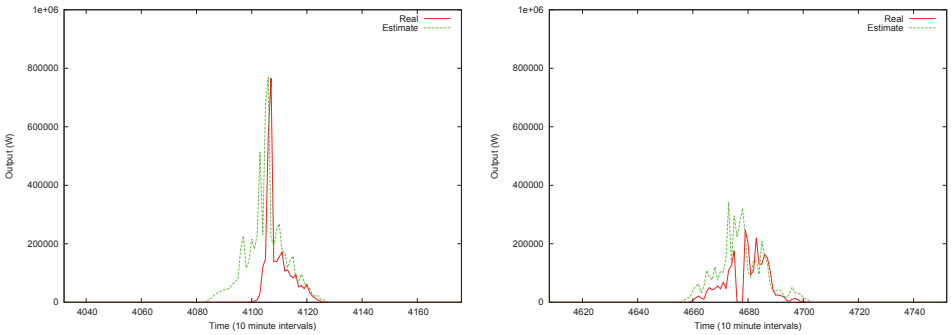


Fig. 3: Examples of anomalies in the training data set.

the test data set are shown in Fig. 4. We can see that the real output of the PVPP is sometimes zero when it should be non-zero (similar as the errors in the training data set). The PVPP output is in some cases greater than 1MW which is more than the possible peak output of this PVPP. We note that the anomalies affected both, the training process and the prediction error evaluation. Interestingly, all

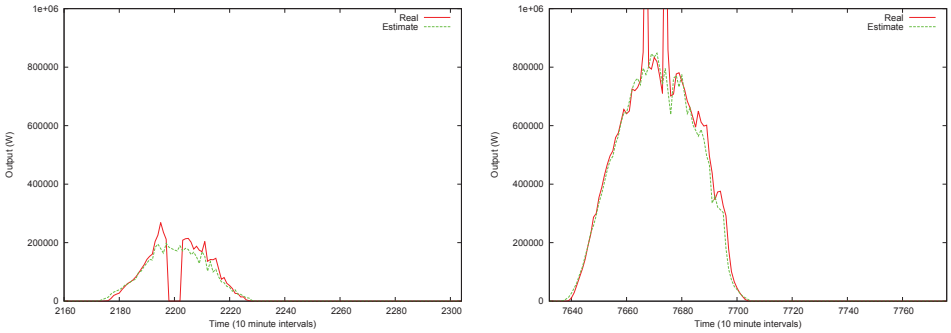


Fig. 4: Examples of anomalies in the test data set.

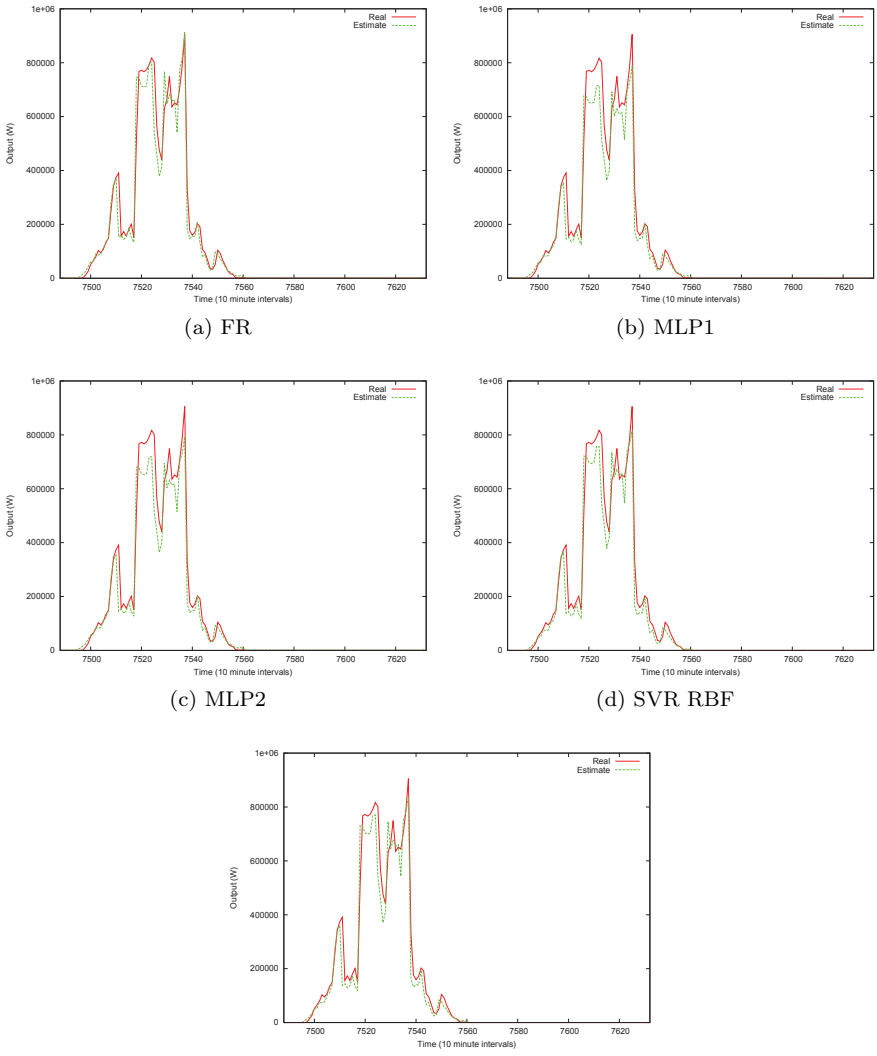
methods have managed to forecast PVPP output very well for some days and less precisely for some other days. Examples of good predictions are shown in Fig. 5 and Fig. 6 and examples of less accurate predictions are shown in Fig. 7 and 8.

The results suggest that the fuzzy rules are on a par with or better than (in generalization) multilayer perceptrons and support vector regression as used in this study. This opinion should be supported by comparison of standard deviation for the full data set, for the training data set and for the test data set presented in Tab. IV. An advantage of fuzzy rules is the symbolic nature of the model that can be used as a feedback for domain experts. The best fuzzy rule for PVPP output

Data set	Standard deviation of prediction error (W)				
	FR	MLP1	MLP2	SVR RBF	SVR POLY
Full	42630.36	42207.01	*41747.01*	42180.44	42114.95
Training	32403.14	31682.39	30978.77	31980.79	31840.87
Test	*49429.14*	51789.10	51095.95	49551.03	49629.67

Tab. IV: Standard deviation of average PVPP output prediction error.

forecasting found by GP is shown in Fig. 9. We can see that the algorithm took the advantage of both, the past feature node (*radiation[5]*) a the past output node (*output[2], output[1]*).



(a) FR (b) MLP1

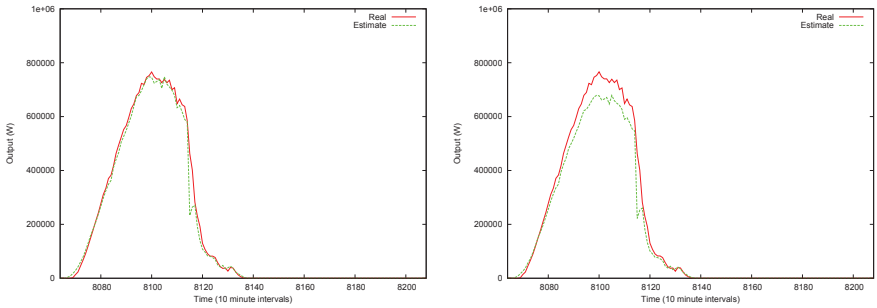
(c) MLP2 (d) SVR RBF

(e) SVR POLY

Fig. 5: Example of a day with good prediction.

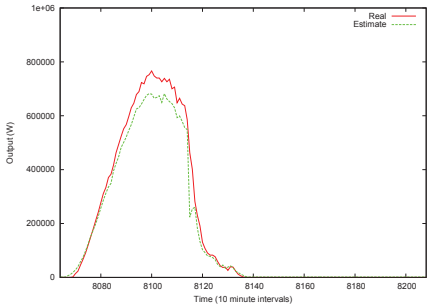
5. Conclusions

The fuzzy rules were used as predictors to estimate the output of a photovoltaic power plant. An experiment with a real-world PVPP was conducted and a fuzzy rule based on the data describing more than three months of the operations of a PVPP was evolved. The data contained only the information about the intensity of solar radiation in the location of the facility. Accurate predictions of the power output of PVPPs can be seen as a building block of intelligent power grids. It shows that soft computing and nature inspired algorithms can contribute to the creation

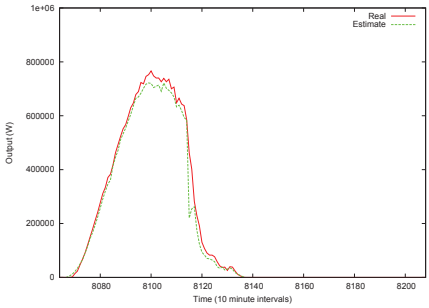


(a) FR

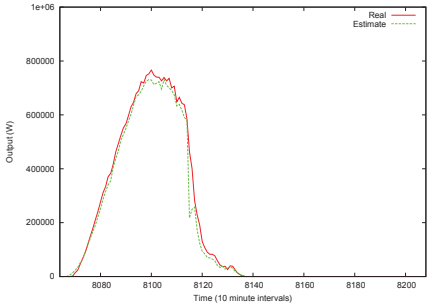
(b) MLP1



(c) MLP2



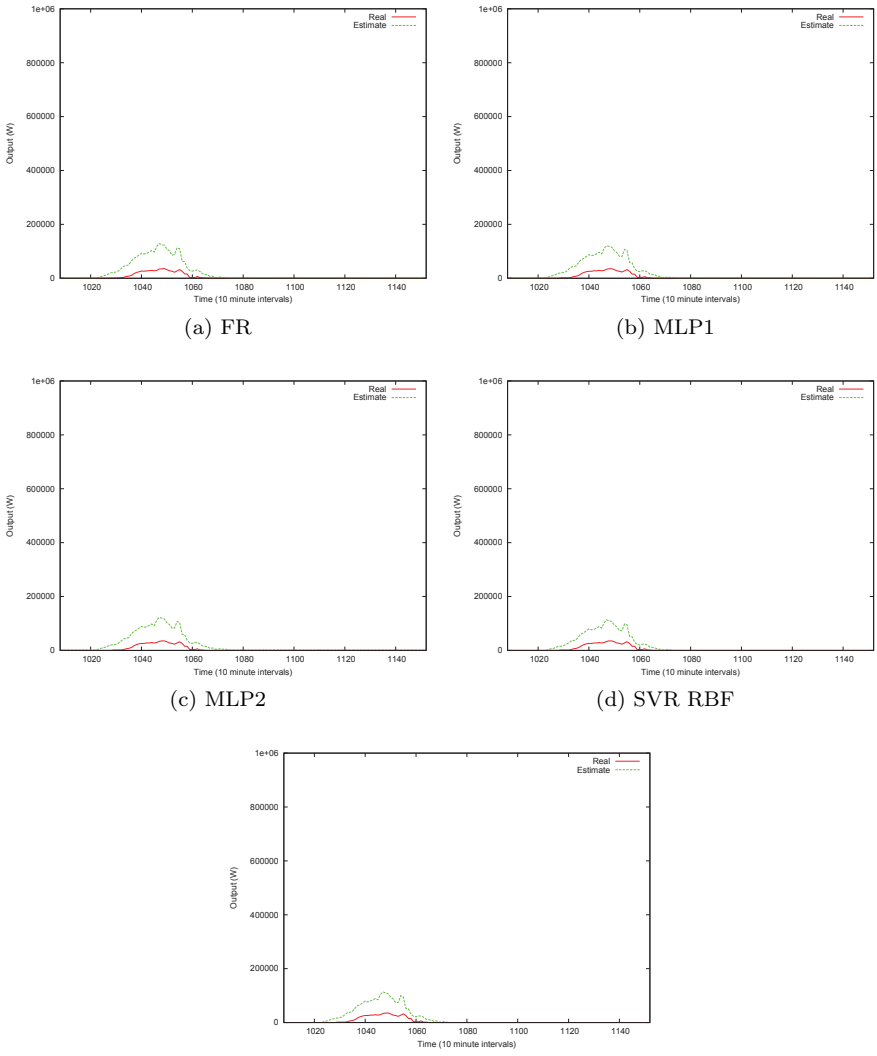
(d) SVR RBF



(e) SVR POLY

Fig. 6: Another day with good prediction.

of smart electrical networks. Moreover, the development of custom predictors tailored to the needs of specific PVPPs are appealing because every PVPP is unique (because e.g. solar panel technology, configuration, age, location, geographical setting, and so on) is very appealing. The accuracy of the predictions obtained by the fuzzy rule was compared to predictions obtained by artificial neural networks and support vector regression and the results were found competitive. The fact that all algorithms achieved better predictions for the same days and worse predictions for the same days suggest that all of them have discovered similar inner structure in the data set.



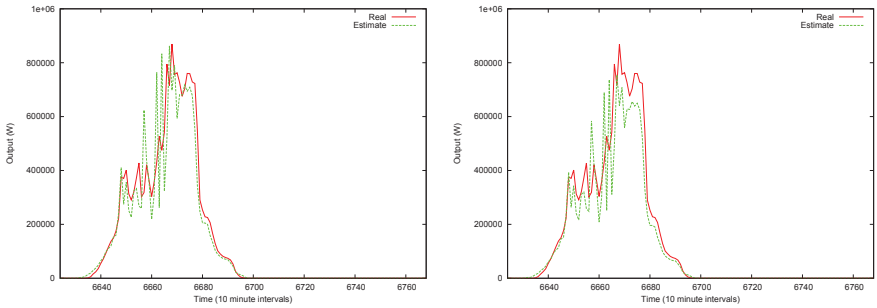
(a) FR (b) MLP1

(c) MLP2 (d) SVR RBF

(e) SVR POLY

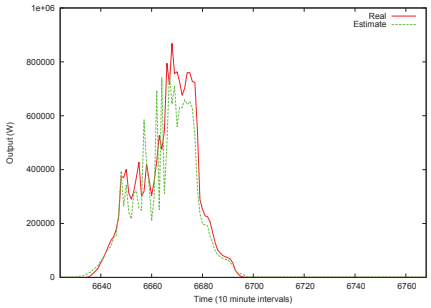
Fig. 7: Example of a day with bad prediction.

Another important factor, which outside the core of soft-computing methods influences the accuracy of the calculation, is also the type itself of the used silicon technology of the given solar power plant. For testing of the soft-computing methods a solar power plant with monocrystallic technologies was used. It is a technology which is able to process mainly the direct component of solar radiation, which means that in case of changeable cloudiness a considerable variation of the sum total of output of active power occurs, which can be as much as tens of percents compared with the installed output of active power. In case of usage of polycrystalline technology the variability of load flow is much lower, as this system

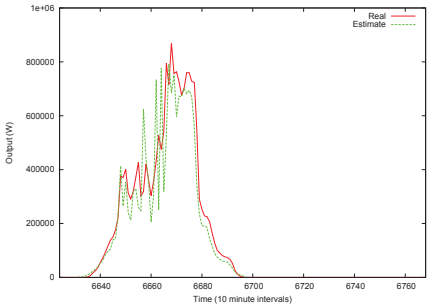


(a) FR

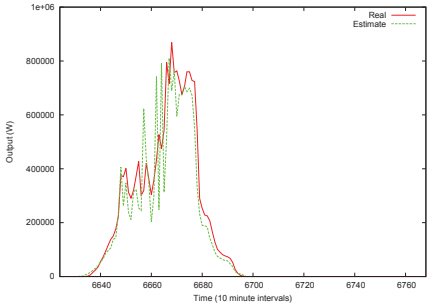
(b) MLP1



(c) MLP2



(d) SVR RBF



(e) SVR POLY

Fig. 8: Another day with bad prediction.

is able to process also the diffusive component of solar radiation. With changeable cloudiness there is a change of output of active power only in orders of several percents. If we then compared both the systems according to the criterion of the possibility of prediction of their output of energy, it is obvious that in case of application of soft-computing methods even higher accuracy of the prediction will be shown for the system solar power plant with lower dynamics of output of active power, i.e. for the system with polycrystalline technology.

The experiment presented in this paper can be extended in many ways. The data set should be cleared of all records that do not describe its regular operations.

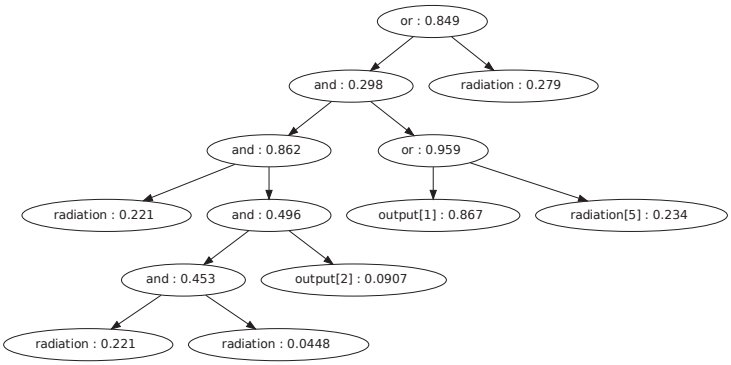


Fig. 9: Best fuzzy predictor for PVPP output estimation.

The data set can be more comprehensive. More inputs (e.g. wind speed, cloud coverage, humidity etc.) should be considered for the estimation and prediction of then PVPP output. Finally, data describing longer period of operations of the PVPP should be considered for better prediction. We will also compare presented approach to other classification and prediction techniques.

Acknowledgement

This work has been partly supported by the Czech Science Foundation under the project 102/09/1842 and by the Ministry of Education, Youth and Sports of the Czech Republic (ENET No. CZ.1.05/2.1.00/03.0069), project SP2013/68 and also by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

References

[1] James C. Bezdek, James Keller, Raghu Krisnapuram, Nikhil R. Pal: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing (The Handbooks of Fuzzy Sets). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[2] Colin Campbell, Yiming Ying: Learning with support vector machines. Synthesis Lectures on Artificial Intelligence and Machine Learning, 5(1), pp. 1–95, 2011.

[3] Christopher Chatfield: Time-Series Forecasting. Statistics (Chapman and Hall/CRC). Taylor & Francis, 2000.

[4] Oscar Cordón: A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. International Journal of Approximate Reasoning, 52(6), pp. 894 – 913, 2011.

[5] Oscar Cordón, María José del Jesus, Francisco Herrera: Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. International Journal of Intelligent Systems, 13, pp. 1025–1053, 1998.

[6] Andries Engelbrecht: Computational Intelligence: An Introduction, 2nd Edition. Wiley, New York, NY, USA, 2007.

[7] Mark Freischlad, Martina Schnellenbach-Held, Torben Pullmann: Evolutionary generation of implicative fuzzy rules for design knowledge representation. In Ian F. C. Smith, editor, EG-ICE, volume 4200 of Lecture Notes in Computer Science, pp. 222–229. Springer, 2006.

[8] L. L. Grant, G. K. Venayagamoorthy: Cellular multilayer perceptron for prediction of voltages in a power system. In Intelligent System Applications to Power Systems, 2009. ISAP'09. 15th International Conference on, pp. 1–6, nov. 2009.

[9] L. L. Grant, G. K. Venayagamoorthy: Voltage prediction using a cellular network. In Power and Energy Society General Meeting, 2010 IEEE, pages 1 –7, july 2010.

[10] Lutz H. Hamel: Knowledge Discovery with Support Vector Machines. Wiley-Interscience, New York, NY, USA, 2009.

[11] Ralf Herbrich: Learning Kernel Classifiers: Theory and Algorithms (Adaptive Computation and Machine Learning). The MIT Press, December 2001.

[12] Hisao Ishibuchi, Yusuke Nojima: Multiobjective formulations of fuzzy rule-based classification system design. In Eduard Montseny and Pilar Sobrevilla, editors, EUSFLAT Conf., pp. 285–290. Universidad Polytecnica de Catalunya, 2005.

[13] Petr Kačor, Stanislav Mišák, Lukáš Prokop: Optimization and redesign of vertical axis wind turbine for generator of independent source of energy. In: B. Katalinic, editor, Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, pp. 1053–1054. DAAAM International Vienna, Vienna, October 2010.

[14] Vojislav Kecman: Support vector machines an introduction. In Lipo Wang, editor, Support Vector Machines: Theory and Applications, volume 177 of Studies in Fuzziness and Soft Computing, pp. 605–605. Springer Berlin / Heidelberg, 2005. 10.1007/10984697_1.

[15] Amit Konar: Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain. CRC Press, Inc., Boca Raton, FL, USA, 2000.

[16] John R. Koza: phGenetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314, Dept. of Computer Science, Stanford University, 1990.

[17] John R. Koza: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA, 1992.

[18] Donald H. Kraft, Gloria Bordogna, Gabriella Pasi: Fuzzy Set Techniques in Information Retrieval. Springer, Fuzzy Sets in Approximate Reasoning and Information Systems The Handbooks of Fuzzy Sets Series Volume 5, 1999, pp. 469–510.

[19] Pavel Krömer, Jan Platoš, Václav Snášel, Ajith Abraham: Towards intrusion detection by information retrieval and genetic programming. In: 2010 Sixth International Conference on Information Assurance and Security (IAS 2010), pages 148–153, Atlanta, Georgia, USA, 8 2010. IEEE Catalog numberCFP1061C-CDR, ISBN978-1-4244-7408-0.

[20] Pavel Krömer, Jan Platoš, Václav Snášel, and Ajith Abraham. Fuzzy classification by evolutionary algorithms. In IEEE International Conference on Systems, Man, and Cybernetics, pp. 313–318. IEEE System, Man, and Cybernetics Society, 2011.

[21] Pavel Krömer, Jan Platoš, Václav Snášel, Ajith Abraham, Lukáš Prokop, Stanislav Mišák: Genetically evolved fuzzy predictor for photovoltaic power output estimation. In 2011 Third International Conference on Intelligent Networking and Collaborative Systems (INCoS), pp. 41–46. IEEE, 2011.

[22] Pavel Krömer, Václav Snásel, Jan Platoš: Learning patterns from data by an evolutionary-fuzzy approach. In Emilio Corchado, Václav Snásel, Javier Sedano, Aboul Hassanien, José Calvo, and Dominik Slezák, editors, Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011, volume 87 of Advances in Intelligent and Soft Computing, pp. 127–135. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-19644-7-14.

[23] Pavel Krömer, Václav Snášel, Jan Platoš, Ajith Abraham: Evolving fuzzy classifier for data mining - an information retrieval approach. In Álvaro Herrero, Emilio Corchado, Carlos Redondo, and Ángel Alonso, editors, Computational Intelligence in Security for Information Systems 2010, volume 85 of Advances in Intelligent and Soft Computing, pp. 25–32. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-16626-6-3.

[24] James Momoh: Smart Grid: Fundamentals of Design and Analysis. IEEE Press Series on Power Engineering. Wiley, 2012.

[25] Ajoy K. Palit, Dobrivoje Popovic: Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications (Advances in Industrial Control). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[26] R. P. Prado, S. Garcia-Galán, J. Exposito, A. J. Yuste: Knowledge acquisition in fuzzy-rule-based systems with particle-swarm optimization. Fuzzy Systems, IEEE Transactions on, 18(6), pp. 1083 –1097, dec. 2010.

[27] Alex J. Smola, Bernhard Schölkopf: A tutorial on support vector regression. Statistics and Computing, 14(3), pp. 199–222, August 2004.

[28] Václav Snásel, Pavel Krömer, Jan Platoš, Ajith Abraham: The evolution of fuzzy classifier for data mining with applications. In Kalyanmoy Deb, Arnab Bhattacharya, Nirupam Chakraborti, Partha Chakroborty, Swagatam Das, Joydeep Dutta, Santosh K. Gupta, Ashu Jain, Varun Aggarwal, Jürgen Branke, Sushil J. Louis, and Kay Chen Tan, editors, SEAL, volume 6457 of Lecture Notes in Computer Science, pp. 349–358. Springer, 2010.

[29] Antanas Verikas, Jonas Guzaitis, Adas Gelzinis, Marija Bacauskiene: A general framework for designing a fuzzy rule-based classifier. Knowledge and Information Systems, pp. 1–19. 10.1007/s10115-010-0340-x.

[30] Ching-Hung Wang, Tzung-Pei Hong, Shian-Shyong Tseng: Integrating membership functions and fuzzy rule sets from multiple knowledge sources. Fuzzy Sets Syst., 112, pp. 141–154, May 2000.

[31] Enwang Zhou, Alireza Khotanzad: Fuzzy classifier design using genetic algorithms. Pattern Recogn., 40, pp. 3401–3414, December 2007.