

Evolutionary Prediction of Photovoltaic Power Plant Energy Production

Pavel Krömer
VŠB - Technical University of
Ostrava
17.listopadu 15
Ostrava, Czech Republic
pavel.kromer@vsb.cz

Lukáš Prokop
VŠB - Technical University of
Ostrava
17.listopadu 15
Ostrava, Czech Republic
lukas.prokop@vsb.cz

Václav Snášel
VŠB - Technical University of
Ostrava
17.listopadu 15
Ostrava, Czech Republic
vaclav.snasel@vsb.cz

Stanislav Mišák
VŠB - Technical University of
Ostrava
17.listopadu 15
Ostrava, Czech Republic
stanislav.misak@vsb.cz

Jan Platoš
VŠB - Technical University of
Ostrava
17.listopadu 15
Ostrava, Czech Republic
jan.platos@vsb.cz

Ajith Abraham
VŠB - Technical University of
Ostrava
17.listopadu 15
Ostrava, Czech Republic
ajith.abraham@ieee.org

ABSTRACT

This paper presents an application of genetic programming to the evolution of fuzzy predictors based on fuzzy information retrieval. The fuzzy predictors are used to estimate the output of a Photovoltaic Power Plant (PVPP). The PVPPs are energy sources with an unstable production of electrical energy. It is necessary to back up the energy produced by the PVPPs for stable electric network operations. An optimal value of backup power can be set with advanced prediction models that can contribute to the robustness of the electric network within the framework of an intelligent power grid. This work extends previous research on evolutionary design of fuzzy PVPP output predictors by the evaluation of the method on a larger data set describing the operations of a real PVPP.

Categories and Subject Descriptors

I.5.4 [Pattern Recognition]: Applications

Keywords

Genetic programming, fuzzy rules, prediction

1. INTRODUCTION

Renewable energy sources are becoming a significant part of the current energy mix. The unstable production of renewable energy sources including photovoltaic, wind, and other weather-dependent power plants puts increased demands on grid management, power transmission and the power grid as a whole. The power grid has to 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 too demanding (both technically and economically), 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 [11].

Nowadays, the energy 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 can contain power plants with unstable energy production whose output heavily depends on the meteorological conditions at given time and 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.

The 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 stable sources of electrical energy will be able to balance the production of unstable energy sources and satisfy the demand for electricity by all customers. Otherwise, the power grid can become unstable and unreliable.

Fuzzy sets and fuzzy logic can be used for soft classification of data. In contrast to crisp classification, which leads to crisp decisions about the data, the fuzzy classification allows a more sensitive data analysis [2]. Fuzzy decision trees and if-then rules are an example of efficient, transparent, and easily interpretable fuzzy classifiers [2, 25].

Genetic programming is a powerful machine learning technique from the wide family of evolutionary algorithms. In contrast to the traditional evolutionary algorithms, it can be

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
Copyright 2012 ACM 978-1-4503-1178-6/12/07 ...\$10.00.

used to evolve complex hierarchical tree-like structures and symbolic expressions. It has been used to evolve Lisp S-expressions, mathematical functions, and general symbolic expressions including the crisp and fuzzy decision trees. Previously, the genetic programming has been used to infer search queries describing fuzzy sets of relevance ranked documents in an information retrieval system [19].

The query evolution can be used for general data mining. The extended Boolean queries (i.e. weighted Boolean search expressions) can be interpreted as symbolic fuzzy rules that describe fuzzy subset of a data set by the 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 fuzzy rule evolution can be used to find 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 in a collective decision making of an ensemble of data classification methods.

In this work, we evolve a fuzzy predictor in the form of a fuzzy rule inspired by the area of fuzzy information retrieval. The same concept was successfully used for data classification in [18, 23, 15, 17]. When compared to the more complex fuzzy classifier systems, it can be seen as a sole fuzzy rule that maps data features onto a real output value from the interval $[0, 1]$. This study uses an improved fuzzy rule to process data as an ordered (time-like) series of records. The extended fuzzy rule is used to estimate the amount of energy produced by a real world PVPP. This paper is extending the initial research presented in [16]. In this study, the fuzzy rules are evolved over a more comprehensive data set describing longer term operations of a real PVPP in the Czech Republic.

The rest of this paper is organized in the following way: first, we summarize selected principles of fuzzy information retrieval as the background of presented prediction method. Section 3 describes in detail the application of the genetic programming to the fuzzy rule evolution and its customization to time series data processing. Finally, section 4 presents computational experiments and in section 5 we draw the conclusions.

2. FUZZY INFORMATION RETRIEVAL

The fuzzy rule evolution is based on information retrieval query optimization [7, 8, 19, 22]. The area of information retrieval (IR) is a branch of computer science dealing with storage, maintenance, and searching in large volumes of unstructured data [5]. It defines and studies IR systems and models, i.e. formal definitions of document representation, query language, and document-query matching mechanisms.

2.1 Extended Boolean IR model

The fuzzy rules are inspired by the extended Boolean IR model, which is based on fuzzy set theory [27] and fuzzy logic [5, 14]. This section provides details on the basic concepts of Boolean IR because they are fundamental for fuzzy predictor construction, evaluation, and evolution.

In the extended Boolean IR model, the documents are modeled as fuzzy sets of indexed terms. In a document, each indexed term has a weight from the range $[0, 1]$ expressing

the degree of significance of the term in the representation of the document within the collection. Many different weighting approaches can be used to assign weights to index terms, e.g. the $tf \cdot idf_t$ term weighting scheme [21].

A formal description of a collection of documents in the extended Boolean IR model is shown in (1) and (2), where d_i represents i -th document and t_{ij} j -th index term in the i -th document. The entire document collection can be represented by the matrix D .

$$d_i = (t_{i1}, t_{i2}, \dots, t_{im}), \forall t_{ij} \in [0, 1] \quad (1)$$

$$D = \begin{pmatrix} t_{11} & t_{12} & \dots & t_{1m} \\ t_{21} & t_{22} & \dots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \dots & t_{nm} \end{pmatrix} \quad (2)$$

The query language in the extended Boolean IR model improved with the possibility to assign weights to query terms and operators. Query term weights represent different level of importance of search terms within the search expression and operator weights are usually used to soften the impact of the aggregation operators on query evaluation [5, 14].

Consider $F(d, t)$ to be the weight of term t in document d and Q to be the set of user queries; then the weight of the term t in query q is denoted by $a(q, t)$ satisfying $a : Q \times T \rightarrow [0, 1]$. To evaluate the atomic query of one term representing a single search criterion the function $g : [0, 1] \times [0, 1] \rightarrow [0, 1]$ will be used. The value of $g(F(d, t), a)$ is in information retrieval called the *retrieval status value* (RSV). For the RSV evaluation, the interpretation of the query term weight a is crucial. The three most commonly used query term weight interpretations understand the weight as a relative importance weight, minimum document term weight threshold, and an ideal document term weight [5, 14].

The formulas for the RSV evaluation for the importance weight interpretation and threshold interpretation are shown in (3) and (4) respectively [5, 14]. $P(a)$ and $Q(a)$ in Equation 4 are coefficients used to define the shape of the threshold curve. An example of $P(a)$ and $Q(a)$ can be $P(a) = \frac{1+a}{2}$ and $Q(a) = \frac{1-a^2}{4}$. The RSV from (4) is illustrated in Figure 1. For the threshold interpretation, an atomic query containing term t of the weight a is a request to retrieve documents having $F(d, t)$ equal or greater to a . The documents satisfying this condition will be rated with high RSV and documents having $F(d, t)$ smaller than a will be rated with a small RSV.

$$g(F(d, t), a) = \begin{cases} \min(a, F(d, t)) & \text{for } or \text{ operator} \\ \max(1 - a, F(d, t)) & \text{for } and \text{ op.} \end{cases} \quad (3)$$

$$g(F(d, t), a) = \begin{cases} P(a) \frac{F(d, t)}{a} & \text{for } F(d, t) < a \\ P(a) + Q(a) \frac{F(d, t) - a}{1 - a} & \text{for } F(d, t) \geq a \end{cases} \quad (4)$$

The query term weight a can be seen as an ideal document term weight. In that case, the RSV will be evaluated according to (5), expressing the distance between $F(d, t)$ and a in a symmetric manner. This means that a document with a lower term weight will be rated with the same RSV as a document with a higher term weight, considering the same differences. An asymmetric version of (5) is shown in (6).

$$g(F(d, t), a) = e^{K \cdot (F(d, t) - a)^2} \quad (5)$$

$$g(F(d, t), a) = \begin{cases} e^{K \cdot (F(d, t) - a)^2} & \text{for } F(d, t) < a \\ P(a) + Q(a) \frac{F(d, t) - a}{1 - a} & \text{for } F(d, t) \geq a \end{cases} \quad (6)$$

The operators *and*, *or*, and *not* can be evaluated with the

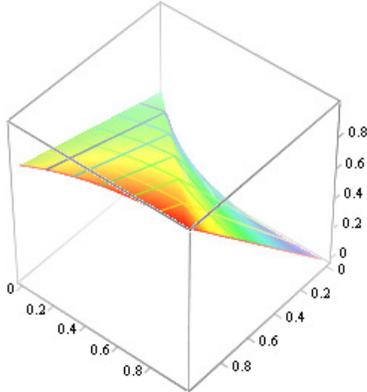


Figure 1: $g(F(d, t), a)$ according to (4).

help of the fuzzy set operations. The fuzzy set operations are an extension of crisp set operations developed for fuzzy sets [27]. A characteristic function fully defines a fuzzy set and hence the fuzzy set operations are defined using the characteristic functions [10]. In [27], Lotfi Zadeh defined basic methods for the evaluation of the complement, union, and intersection of fuzzy sets. Besides these standard (Zadeh's) fuzzy set operations, whole classes of formulas for defining the complements, intersections, and unions on fuzzy sets were proposed. Their common aim is to soften the effects of the fuzzy set operations.

For the fuzzy rule evaluation, we use the threshold query term weight interpretation and the standard t-norm (8) and t-conorm (s-norm)(9) for the implementation of *and* and *or* operators. The fuzzy complement for the evaluation of the *not* operator used in this work is shown (7).

$$c(x) = 1 - x \quad (7)$$

$$t(x, y) = \min(x, y) \quad (8)$$

$$s(x, y) = \max(x, y) \quad (9)$$

However, the use of any other t-norm and t-conorm pairs is possible.

2.2 IR evaluation

The effectiveness of an information retrieval system can be evaluated using the measures precision P and recall R . Precision corresponds to the probability of retrieved document to be relevant and recall can be seen as the probability of retrieving a relevant document.

Precision and recall in the extended Boolean IR model can be defined using the Σ -count $\|A\|$ [28]:

$$\rho(X|Y) = \begin{cases} \frac{\|X \cap Y\|}{\|Y\|} & \|Y\| \neq 0 \\ 1 & \|Y\| = 0 \end{cases} \quad (10)$$

$$P = \rho(REL|RET) \quad (11)$$

$$R = \rho(RET|REL) \quad (12)$$

where REL stands for the fuzzy set of all relevant documents, RET for the fuzzy set of all retrieved documents,

and $\|A\|$ is the Σ -count, i.e. the sum of the values of characteristic function μ_A for all members of the fuzzy set A [28]:

$$\|A\| = \sum_{x \in A} \mu_A(x) \quad (13)$$

For an easier IR effectiveness evaluation, measures combining precision and recall into one scalar value were developed. The F-score F is among the most used scalar combinations of P and R :

$$F = \frac{(1 + \beta^2)PR}{\beta^2P + R} \quad (14)$$

In IR, the β parameter evaluates the preference of R over P in the formula (i.e. $\beta = 2$ would put twice as much importance R) [24].

The index matrix D can be seen as a general data matrix with m rows (data samples) and n columns (data features). The evaluation of extended Boolean query over the document collection generates an ordering of the documents (i.e. it assigns a real value from the range $[0, 1]$ to each document). The ordering can be also interpreted as a fuzzy set of documents. If we leave out the IR terminology, we can call the extended Boolean query a fuzzy rule and use it to describe fuzzy sets or fuzzy subsets of data by its features.

3. FUZZY RULES EVOLVED BY GENETIC PROGRAMMING

The genetic programming (GP) is an extension to the genetic algorithms enabling work with hierarchical, often tree-like, chromosomes with an uneven length [12, 13]. The GP was introduced as a tool for automatic evolution of computer programs and it represented a step towards adaptable computers that could solve problems without being explicitly programmed [12, 1]. The GP can be used to find and optimize solutions in the field of machine learning and other domains that can formulate its solutions by the means of structured symbolic expressions. The GP enables an efficient evolution of such a symbolic expressions with well-defined syntax and grammar.

The GP has been previously used for the optimization of extended Boolean queries [7, 8, 19, 22]. It was shown that the GP was able to optimize search queries so that they described a set of relevant documents. In the fuzzy information retrieval, the relevant documents formed a fuzzy subset of the universe of all documents and extended Boolean queries describing such a fuzzy subset of all documents were evolved. The IR measure F-Score (14) was used as a fitness function. This study uses the framework for genetic evolution of fuzzy queries to evolve fuzzy rules for PVPP output prediction.

3.1 Evolutionary fuzzy classifier design

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. [3]. Multi-objective evolutionary algorithms were used for the evolution of linguistic fuzzy rule-based classification systems in the work of Cordon et al. [4]. Another multi-objective evolutionary approach to the evolution of fuzzy rule-based systems was proposed by Ishibuchi

and Nojima [9]. 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. [26] used genetic algorithms to integrate fuzzy rule sets and membership functions learned from various information sources. In [6], Freischlad et al. used an evolutionary algorithm to generate fuzzy rules for knowledge representation. Zhou and Khotanzad [29] 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 [20].

3.2 Fuzzy rules for time series data analysis

The fuzzy rules use similar data structures, basic concepts, and operations as the fuzzy IR 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. Such a general real valued matrix 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.

3.3 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 (*feature1:0.5*), past feature node is defined by feature name, index of previous record, and weight

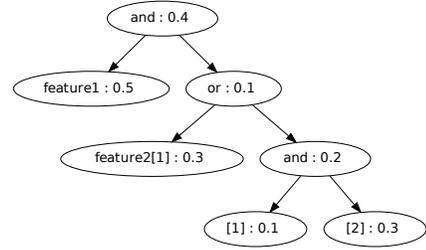


Figure 2: Tree form of a fuzzy predictor

(*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 Figure 2.

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 analyze the stream of records as an ordered sequence similar to a time series.

4. EXPERIMENTS

We have used the fuzzy predictor for a PVPP output estimation. We have recorded the volumes of electrical energy produced by a PVPP located in the Czech Republic and the values of solar radiation in the same location. The values were recorded in 10 minute intervals between November 2010 and April 2011. The original data set contained 21515 records. After initial screening were removed records from days with irregular operations of the PVPP i.e. when the facility was disconnected from the grid but sensors were active. The remaining 20513 records were divided in two halves. The first part containing 10257 records was used as the training data set for predictor evolution. The second part containing 10256 records was used as testing data set. In an independent experiment, a separate predictor was developed for the full data set. The parameters of the GP used to search for fuzzy predictor are summarized in Table 1. They were set on the basis of initial experiments and past experience. Due to the stochastic nature of genetic programming, the evolution of fuzzy predictor was executed 10 times. The average training time for the full data set was 2 seconds on a commodity laptop with an Intel(R) Core(TM) i5 CPU at 2.40GHz.

The average prediction error for the full data set, for the training data set, and for the test data set is shown in Table 3. The table also shows the average absolute deviation of average prediction error for all experimental runs. The average absolute deviation was small, i.e. the GP was able to find similar fuzzy predictors in all independent runs. The prediction error is in all cases lower than 2 percent 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 evolve predictors when the in-

formation about the entire period is available. We note that the environment characteristics that affect the operations of the PVPP (e.g. the solar elevation angle) change during the year. The training error shows how the predictor 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 predict the electric power output of the PVPP in the period covered by the test set.

The best predictor found by the GP is shown in Figure 3. 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]*).

Table 1: GP parameters.

Parameter	Value	Note
Population size	100	
Crossover probability P_C	0.8	
Mutation probability P_M	0.2	see Table 2
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 $\beta = 1$

Table 2: Random rule generation an mutation probabilities.

(a) Probabilities of generating random fuzzy rule nodes.

Event	Prob.
Generate term	0.5
Generate <i>and</i>	0.24
Generate <i>or</i>	0.24
Generate <i>not</i>	0.02

(b) Probabilities of mutation operations.

Event	Prob.
Mutate node weight	0.5
Insert or delete <i>not</i>	0.1
Replace with another node or delete <i>not</i>	0.32
Replace with random branch	0.08

Table 3: Average PVPP output prediction error.

Data set	Average prediction error (W)	Average absolute deviation (W)
Full	13589.4	200.31
Training	10001.4	5.4
Test	18313.2	34.62

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 Figure 4. 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 the test data set are shown in Figure 5. 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

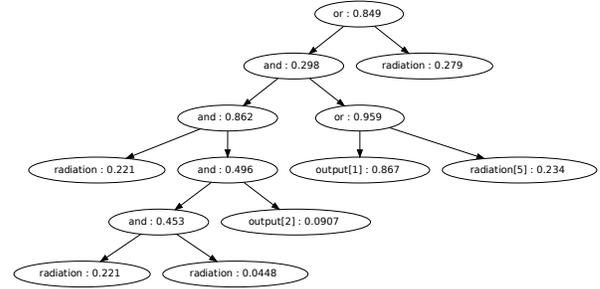
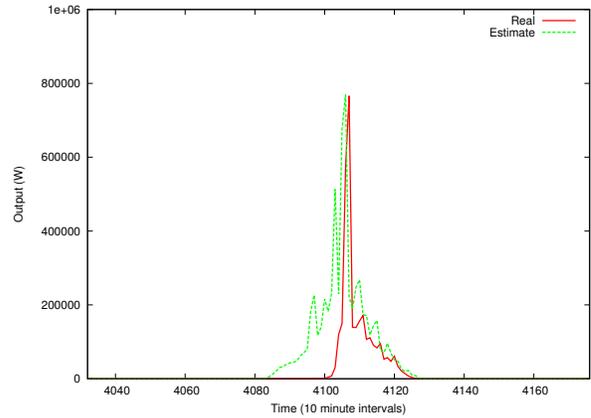
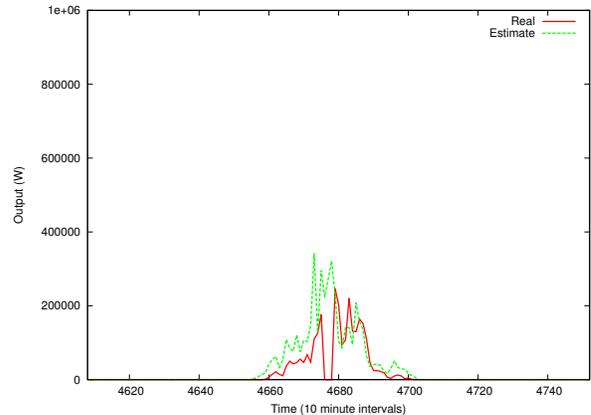


Figure 3: Best fuzzy predictor for PVPP output estimation.



(a) An example of error in the training data



(b) Different type of error in the training data

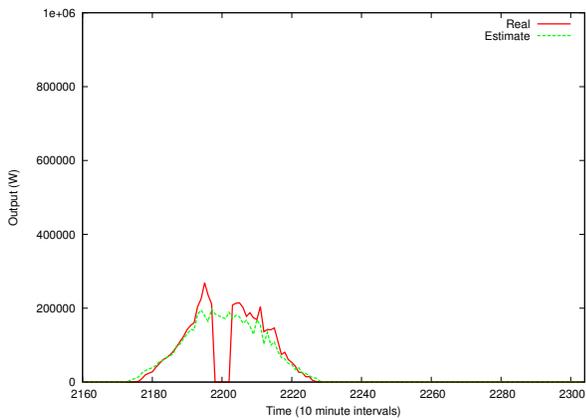
Figure 4: Examples of anomalies in the training data set.

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.

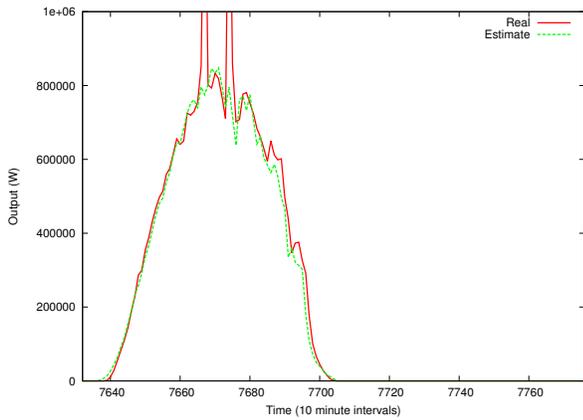
The fuzzy predictors managed to estimate the PVPP output very well for some days and less precisely for some other days. An example of good predictions is shown in Figure 6 and an example of less accurate predictions is shown in Figure 7.

5. CONCLUSIONS

We have used the genetic programming for the evolution

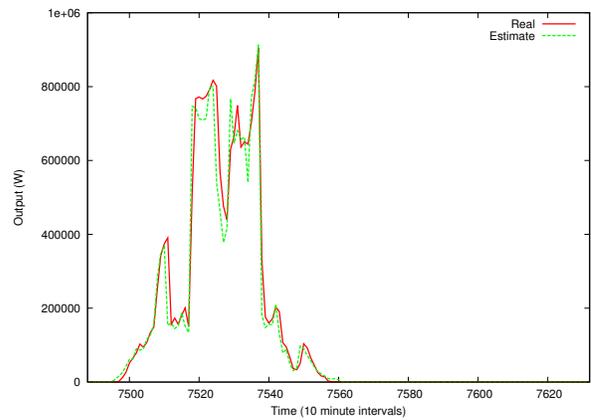


(a) An example of error in the test data

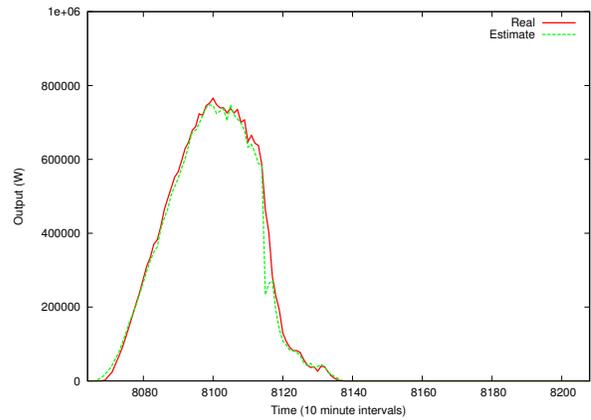


(b) Another type of error in the test data

Figure 5: Examples of anomalies in the test data set.



(a) A day with good prediction

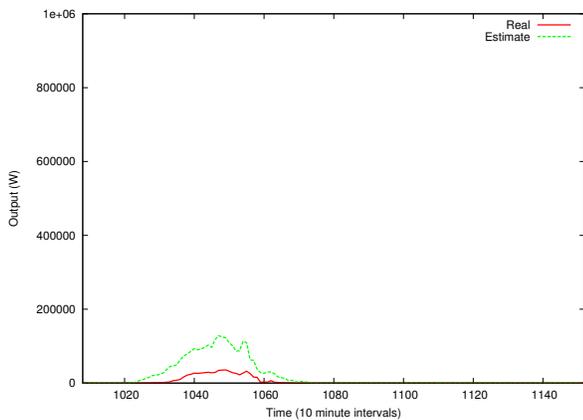


(b) Another example of good prediction

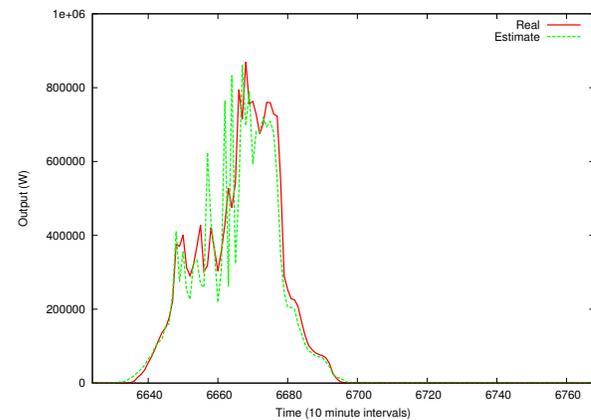
Figure 6: Examples of days with good prediction.

of fuzzy rules to estimate the output of a PVPP. An experiment with a real world photovoltaic power plant was conducted and a predictor based on the data describing more than three months of the operations of a PVPP was generated. 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 of smart electrical networks. Moreover, the development of custom predictors tailored to the needs of specific PVPPs is appealing because every PVPP



(a) A day with bad prediction



(b) Another example of bad prediction

Figure 7: Examples of days with bad prediction.

is unique (because of e.g. the solar panel technology, configuration, age, location, geographical setting, and so on). The work presented in this paper can be extended in many ways. The data set should be cleared of all records that do not describe regular operations of the PVPP. More inputs (e.g. wind speed, cloud coverage, humidity) should be considered for the estimation and prediction of then PVPP output. Next, the evolution of separate predictors for different levels of production of the PVPP should be considered because the need required precision of the prediction is different for different level of produced energy. Finally, data describing longer period of operations of the PVPP should be considered for better prediction.

Acknowledgement

This work was supported by the Ministry of Industry and Trade of the Czech Republic, under the grant no. FR-TI1/420 and by SGS, VSB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/58.

6. REFERENCES

- [1] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Chapman & Hall/CRC, 2009.
- [2] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. 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.
- [3] O. Cordon. 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):894 – 913, 2011.
- [4] O. Cordon, M. J. del Jesus, and F. Herrera. Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. *International Journal of Intelligent Systems*, 13:1025–1053, 1998.
- [5] F. Crestani and G. Pasi. Soft information retrieval: Applications of fuzzy set theory and neural networks. In N. Kasabov and R. Kozma, editors, *Neuro-Fuzzy Techniques for Intelligent Information Systems*, pages 287–315. Springer Verlag, Heidelberg, DE, 1999.
- [6] M. Freischlad, M. Schnellenbach-Held, and T. Pullmann. Evolutionary generation of implicative fuzzy rules for design knowledge representation. In I. F. C. Smith, editor, *EG-ICE*, volume 4200 of *Lecture Notes in Computer Science*, pages 222–229. Springer, 2006.
- [7] D. Húsek, S. S. J. Owais, V. Snášel, and P. Krömer. Boolean queries optimization by genetic programming. *Neural Network World*, pages 359–409, 2005.
- [8] D. Húsek, V. Snášel, R. Neruda, S. S. J. Owais, and P. Krömer. Boolean queries optimization by genetic programming. *WSEAS Transactions on Information Science and Applications*, 3(1):15–20, 2006.
- [9] H. Ishibuchi and Y. Nojima. Multiobjective formulations of fuzzy rule-based classification system design. In E. Montseny and P. Sobrevilla, editors, *EUSFLAT Conf.*, pages 285–290. Universidad Polytécnica de Catalunya, 2005.
- [10] J. Jantzen. *Tutorial On Fuzzy Logic*. Technical Report 98-E-868 (logic), Technical University of Denmark, Dept. of Automation, 1998.
- [11] P. Kačor, S. Mišák, and L. 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*, pages 1053–1054. DAAAM International Vienna, Vienna, October 2010.
- [12] J. Koza. *Genetic 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.
- [13] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
- [14] D. H. Kraft, F. E. Petry, B. P. Buckles, and T. Sadasivan. Genetic Algorithms for Query Optimization in Information Retrieval: Relevance Feedback. In E. Sanchez, T. Shibata, and L. Zadeh, editors, *Genetic Algorithms and Fuzzy Logic Systems*, Singapore, 1997. World Scientific.
- [15] P. Kromer, J. Platos, V. Snasel, and A. 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 number CFP1061C-CDR, ISBN978-1-4244-7408-0.
- [16] P. Krömer, J. Platoš, V. Snášel, A. Abraham, L. Prokop, and S. Mišák. Genetically evolved fuzzy predictor for photovoltaic power output estimation. In *2011 Third International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pages 41 – 46. IEEE, 2011.
- [17] P. Krömer, V. Snášel, and J. Platoš. Learning patterns from data by an evolutionary-fuzzy approach. In E. Corchado, V. Snášel, J. Sedano, A. Hassanien, J. Calvo, and D. Slezák, editors, *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*, volume 87 of *Advances in Intelligent and Soft Computing*, pages 127–135. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-19644-7-14.
- [18] P. Krömer, V. Snášel, J. Platoš, and A. Abraham. Evolving fuzzy classifier for data mining - an information retrieval approach. In Ā. Herrero, E. Corchado, C. Redondo, and Ā. Alonso, editors, *Computational Intelligence in Security for Information Systems 2010*, volume 85 of *Advances in Intelligent and Soft Computing*, pages 25–32. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-16626-6-3.
- [19] S. Owais, P. Krömer, V. Snášel, D. Husek, and R. Neruda. Implementing GP on optimizing both boolean and extended boolean queries in IR and fuzzy IR systems with respect to the users profiles. In G. G. Yen, L. Wang, P. Bonissone, and S. M. Lucas, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 5648–5654, Vancouver, BC, Canada, 6-21 July 2006. IEEE Computer Society.

- [20] R. Prado, S. Garcia-Galán, J. Exposito, and A. Yuste. Knowledge acquisition in fuzzy-rule-based systems with particle-swarm optimization. *Fuzzy Systems, IEEE Transactions on*, 18(6):1083–1097, dec. 2010.
- [21] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):pp. 513–523, 1988.
- [22] V. Snasel, A. Abraham, S. Owais, J. Platos, and P. Kromer. *Emergent Web Intelligence: Advanced Information Retrieval*, chapter User Profiles Modeling in Information Retrieval Systems, pages 169–198. Advanced Information and Knowledge Processing. Springer London, 2010. ISBN 978-1-84996-073-1 (Print) 978-1-84996-074-8 (Online), DOI10.1007/978-1-84996-074-8-7.
- [23] V. Snásel, P. Krömer, J. Platos, and A. Abraham. The evolution of fuzzy classifier for data mining with applications. In K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakraborty, S. Das, J. Dutta, S. K. Gupta, A. Jain, V. Aggarwal, J. Branke, S. J. Louis, and K. C. Tan, editors, *SEAL*, volume 6457 of *Lecture Notes in Computer Science*, pages 349–358. Springer, 2010.
- [24] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.
- [25] A. Verikas, J. Guzaitis, A. Gelzinis, and M. Bacauskiene. A general framework for designing a fuzzy rule-based classifier. *Knowledge and Information Systems*, pages 1–19. 10.1007/s10115-010-0340-x.
- [26] C.-H. Wang, T.-P. Hong, and S.-S. Tseng. Integrating membership functions and fuzzy rule sets from multiple knowledge sources. *Fuzzy Sets Syst.*, 112:141–154, May 2000.
- [27] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:pp. 338–353, 1965.
- [28] L. A. Zadeh. *Empirical Semantics*, volume 1 of *Quantitative Semantics, Vol. 12*, chapter Test-score semantics dor natural languages and meaning representation via Pruf, pages 281–349. Studienverlag Brockmeyer, Bochum, 1981.
- [29] E. Zhou and A. Khotanzad. Fuzzy classifier design using genetic algorithms. *Pattern Recogn.*, 40:3401–3414, December 2007.