

## Short-term Power Demand Forecasting using the Differential Polynomial Neural Network

Ladislav Zjavka\*

*IT4innovations, VŠB-Technical University of Ostrava, 17. listopadu 15/2172, 708 33 Ostrava  
Ostrava, Czech Republic  
lzjavka@gmail.com*

Received 7 January 2014

Accepted 15 October 2014

### Abstract

Power demand forecasting is important for economically efficient operation and effective control of power systems and enables to plan the load of generating unit. The purpose of the short-term electricity demand forecasting is to forecast in advance the system load, represented by the sum of all consumers load at the same time. A precise load forecasting is required to avoid high generation cost and the spinning reserve capacity. Under-prediction of the demands leads to an insufficient reserve capacity preparation and can threaten the system stability, on the other hand, over-prediction leads to an unnecessarily large reserve that leads to a high cost preparations. Differential polynomial neural network is a new neural network type, which forms and resolves an unknown general partial differential equation of an approximation of a searched function, described by data observations. It generates convergent sum series of relative polynomial derivative terms which can substitute for the ordinary differential equation, describing 1-parametric function time-series. A new method of the short-term power demand forecasting, based on similarity relations of several subsequent day progress cycles at the same time points is presented and tested on 2 datasets. Comparisons were done with the artificial neural network using the same prediction method.

*Keywords:* power demand prediction, week and day load cycle, differential polynomial neural network, sum relative derivative term, ordinary differential equation composition

### 1. Introduction

The potential benefits of an energy demand prediction are obvious useful in automatic power dispatch, load scheduling and energy control. Forecasting methods in general might be classified into 2 types, the time-series models, in which the load is a function of its past observed values and causal models, which the load is a function of some exogenous factors, especially weather and social variables, both model types can be combined. Day-ahead load forecasting techniques can involve autoregressive integrated moving average (ARIMA) [1], chaotic dynamic non-linear models with evolutionary hybrid computation [2], exponential smoothing and interval time-series [3] or fuzzy logic, characterized by

generalizing classical two-valued logic for reasoning under non-linear and uncertain conditions [4][5]. Regression analysis is considered as a conventional way in power demand predictions. This statistical method used to estimate the parameter values of a function, in which the function predicts the value of a response variable in terms of the value of other variables [6]. Linear regression models can decompose the load into basic and weather dependent components, which are attached some physical interpretation with an exactly described behavior [7]. A real data 1-parametric function time-series progress is difficult to predict using deterministic methods as weather conditions as well as other extraneous factors can by far influence it. Artificial neural network (ANN) can define simple and

---

\*Typeset names in 10 pt Times Roman, uppercase. Use the footnote to indicate the present or permanent address of the author.

robust models, which exact solution is problematic or impossible to get using standard regression techniques. The power demand model might be trained with similarity data relations of several corresponding days of previous weeks, as the daily power load cycles of each following week are of a similar progress. After that the prediction is formed with respect to power values of a few last successive days with the same denomination in the current week. The model should be updated to take into account a dynamic character of the problem.

$$y = a_0 + \sum_{i=1}^m a_i x_i + \sum_{i=1}^m \sum_{j=1}^m a_{ij} x_i x_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m a_{ijk} x_i x_j x_k + \dots \quad (1)$$

$m$  – number of input variables  $X(x_1, x_2, \dots, x_m)$   
 $A(a_1, a_2, \dots, a_m), \dots$  – vectors of parameters

Differential polynomial neural network (D-PNN) is a new neural network type designed by the author, which results from the GMDH (Group Method of Data Handling, created by a Ukrainian scientist Aleksey Ivakhnenko in 1968, when the back-propagation technique was not known yet [8]). GMDH forms a multi-layer polynomial neural network (PNN) in successive steps, adding one layer a time. It is possible to express a general connection between input and output variables by means of the Volterra functional series, a discrete analogue of which is the Kolmogorov-Gabor polynomial (1). This polynomial can approximate any stationary random sequence of observations and can be computed by either adaptive methods or system of Gaussian normal equations. GMDH decomposes the complexity of the process into many simpler relationships each described by low order polynomials (2) for every pair of the input values  $x_i, x_j$ . Typical GMDH network maps a vector input  $x$  to a scalar output  $y$ , which is an estimate of the true function  $f(x) = y^f$  [9].

$$y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \quad (2)$$

It is possible to use just a simple linear form of the eq.(2), which enables only a linear regression, i.e. a linear dependence and pattern [10] identification but it could be applied also to linear function approximation [11]. There is also possible to use several extended or modified forms of the GMDH polynomial (bicubic, triquadratic, ...) [12], which might improve some applications however this study applied only the general form of the polynomial (2), which generally yields the best results. D-PNN can combine the PNN functionality with some mathematical techniques of differential equation (DE) substitutions. Its models lie on the

boundary of neural networks and exact computational techniques. D-PNN forms and resolves an unknown general DE description of an approximation of a searched function. It produces sum series of fractional polynomial derivative terms, which can substitute for the DE, decomposing the system model into many partial derivative specifications of data relations. In contrast with the common neural network functionality, each neuron (i.e. derivative term) can take part directly in the total network output calculation, which is generated by the sum of all the active neuron output values.

## 2. Short-term energy load forecasting

Short-term electric energy estimations of a future demand are needful for the planning of generating electricity of regional grid systems and operating power systems. An overall generation plan requires a forecast of total generation requirements and also peak demands. The inaccuracy in a forecast means that load matching is not optimized and consequently the generation and transmission systems are not being operated in an efficient manner. In order to guarantee a regular supply, it is necessary to keep a reserve. Depending on how fast it is available, it is called spinning reserve or cold reserve. Over-estimating the future load results in an unused spinning reserve, being “burnt” for nothing. An unexpected supply to the international energy network is usually not welcome. Under-estimating the future load is equally detrimental, because of high starting costs of cold reserves, buying at the last minute from other suppliers is obviously too expensive. Cooperation on the electricity grid requires from all providers to foresee the demands within a sufficient accuracy [1]. The nature of parameters that affect this problem includes many uncertainties. The accuracy of a dispatching system is influenced by various conditional input parameters (weather, time, historical data and random disturbances), which can a prediction model involve. ANN is able to model the non-linear nature of dynamic processes, reproduce an empirical relationship between some inputs and one or more outputs. It is applied for such purpose regarding to its approximation capability of any continuous nonlinear function with arbitrary accuracy that offer an effective alternative to more traditional statistical techniques. The load at a given hour is dependent not only on the load at the previous hour, but also on the load at the same hour on

the previous day, and on the load at the same hour on the day with the same denomination in the previous week. There are also many important exogenous variables that should be considered, especially weather-related variables [13].

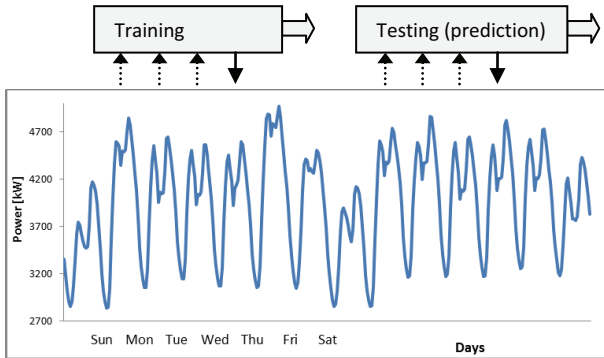


Fig. 1. Typical winter week cycles of the power demand (January) [A].

Generally the load pattern on normal weekdays remains almost constant with small random variations from varying industrial activities and weather conditions, so the load values for normal days are functions of the short-term historical data and weather parameters. The load on Mondays and Fridays is different from that on other weekdays due to pick-up loads on Monday mornings, and evening loads on Fridays similar to the weekend. The load pattern on Saturdays is different from rest of the weekdays as the peak load takes a dip. The shape of the load curve on Sundays is similar to that on holidays, with the peak load considerably decreasing [4]. To allow for all these week cycle regularities it is necessary to consider not only time-series of past days but also to heed to the previous week(s) day power patterns of the same denominations.

$$y_t(d) = f(o_t(d-1), o_t(d-2), \dots, o_t(d-n)) \quad (3)$$

$n = \text{number of inputs} \quad t = t_1, t_2, \dots, t_k$   
 $o_t(d) = \text{observation in time } t \text{ and day } d$

The proposed method keynote of the power load forecasting, using only 1-parametric function historical time-series, is to train a neural network model with actual daily cycle similarity relations of previous weeks, concerning several consecutive days with the same denomination, foregoing the 24-hour prediction. Power values of the 3 consequent days in previous weeks at the same time points form the input vector while the following day 24-hour shifted series define desired

network outputs of the training data set (3). After training the network can estimate the following day power progress, using 3 input vector variables of 24-hour shifted time-series of the same time stamps of the current week last 3 days (Fig.1.). The model does not allow for weather or other disturbing effects, as these are not at disposal in most cases. The power demand day cycle progress is more variable in winter than summer months, which is influenced largely by using heating systems and temperature conditions [A].

### 3. General sum differential equation composition

The D-PNN decomposes and substitutes for a general sum partial differential equation (4), in which an exact definition is not known in advance and which can generally describe a system model, with sum derivative terms. The searched function  $u$  may be expressed in the form of sum series (6), consisting of series arising from derivative sum convergent term series (6) in the case of 2 input variables. The study replaces the general partial DE (4) with fractional multi-parametric polynomial terms (9), which define relative derivative dependent changes over some combinations of input variables of a DE solution [10].

$$a + bu + \sum_{i=1}^n c_i \frac{\partial u}{\partial x_i} + \sum_{i=1}^n \sum_{j=1}^n d_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \dots = 0 \quad u = \sum_{k=1}^{\infty} u_k \quad (4,5)$$

$a, b, c(c_1, c_2, \dots, c_n), d(d_{11}, d_{12}, \dots), \dots$  - parameters  
 $n$  - number of variables  $\mathbf{x}(x_1, x_2, \dots, x_n)$  - input variables  
 $u(\mathbf{x})$  - unknown function of  $n$ -variable vector

$$\left( \sum \frac{\partial u_k}{\partial x_1}, \sum \frac{\partial u_k}{\partial x_2}, \sum \frac{\partial^2 u_k}{\partial x_1^2}, \sum \frac{\partial^2 u_k}{\partial x_1 \partial x_2}, \sum \frac{\partial^2 u_k}{\partial x_2^2} \right) \quad (6)$$

The similarity theory is based on the hypothesis functional relationships exist among the non-dimensional parameters describing a physical system. The Buckingham  $PI$  theorem is the fundamental of dimensional analysis, removing extraneous information from a problem by forming dimensionless groups. It states if the eq. (7) is the only relationship among the  $q_i$  and if it holds for any arbitrary choice of the units in which  $q_1, q_2, \dots, q_n$  are measured, then (7) can be written in the form (8) [14].

$$\phi(q_1, q_2, \dots, q_n) = 0 \quad (7)$$

$$\phi(\pi_1, \pi_2, \dots, \pi_n) = 0 \quad (8)$$

$\pi_1, \pi_2, \dots, \pi_n$  are independent dimensionless products of the  $q$ 's

The method of integral analogues, which is a part of the similarity model analysis, applying various formal adaptations of a DE [15], provides an essential principle for the partial polynomial DE (4) term substitution. It replaces mathematical operators and symbols in a DE by the ratio of corresponding variables. Derivatives are replaced by their integral analogues, i.e. derivative operators are removed and along with all operators are replaced by analogue or proportion signs in equations.

$$u_i = \frac{(a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2 + \dots)^{1/n}}{b_0 + b_1x_1 + \dots} = \frac{\partial^m f(x_1, \dots, x_n)}{\partial x_1 \partial x_2 \dots \partial x_m} \quad (9)$$

$n$  – combination degree of a complete polynomial of  $n$ -variables  
 $m$  – combination degree of a denominator polynomial

The complete input polynomials (2) can replace the partial functions  $u_k$  of partial sum derivative terms (6), while the combination polynomials of denominators represent the alterative derivative parts (9). The root function of a neuron's numerator decreases a combination degree of the input polynomial of a term, in order to get the dimensionless values. In the case of time-series data observations an ordinary differential equation (10) is formed with only time derivatives, which describe 1-parametric function time-step relations using partial DE terms, analogous to the general partial DE substitution (4).

$$a + bf + \sum_{i=1}^m c_i \frac{df(t, x_i)}{dt} + \dots + \sum_{i=1}^m cc_i \frac{df^2(t, x_i)}{dt^2} + \dots = 0 \quad (10)$$

$f(t, x)$  – function of time  $t$  and independent observations  $x(x_1, \dots, x_m)$

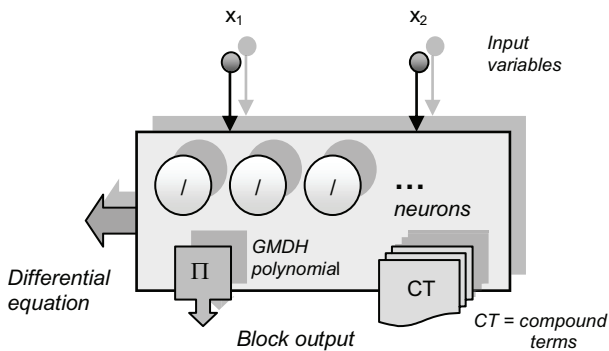


Fig. 2. D-PNN block involves simple and compound neurons (DE terms).

Blocks of the D-PNN (Fig.2.) consist of neurons having the same inputs, one for each fractional polynomial derivative combination, so each neuron is considered a summation DE term (5). Each block contains a single output GMDH polynomial (2), without derivative part.

Neurons do not affect the block output but can participate directly in the total network output sum calculation of a DE composition. Each block has 1 and neuron 2 vectors of adjustable parameters  $a$ , then  $a, b$ .

$$F\left(x_1, x_2, u, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, \frac{\partial^2 u}{\partial x_2^2}\right) = 0 \quad (11)$$

where  $F(x_i, x_2, u, p, q, r, s, t)$  is a function of 8 variables

While using 2 input variables the 2<sup>nd</sup> order partial DE (11) is formed, which involves all derivative terms with respect to the GMDH polynomial variables (2). Each block of the D-PNN forms 5 corresponding simple neurons (DE terms) with respect to 2 single variables  $x_1, x_2$  (12), 2 square variables  $x_1^2, x_2^2$  (13) and 1 combination term  $x_1x_2$  (14). This type of the 2<sup>nd</sup> order partial DE is most often used to model physical or natural system non-linearities (11).

$$y_1 = \frac{\partial f(x_1, x_2)}{\partial x_1} = w_1 \frac{(a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2)^{1/2}}{1.5 \cdot (b_0 + b_1x_1)} \quad (12)$$

$$y_3 = \frac{\partial^2 f(x_1, x_2)}{\partial x_2^2} = w_3 \frac{a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2}{2.7 \cdot (b_0 + b_1x_2 + b_2x_2^2)} \quad (13)$$

$$y_5 = \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} = w_5 \frac{a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2}{2.3 \cdot (b_0 + b_1x_1 + b_2x_1^2 + b_3x_1x_2)} \quad (14)$$

#### 4. Multi-layer differential polynomial neural network

Multi-layer networks form composite polynomial functions (Fig.3.); the previous layers produce internal functions (15), which substitute for the next hidden layer input variables of neuron and block polynomials to form external functions (16). Compound terms (CT), i.e. composite function derivatives with respect to the variables of previous layer blocks, are calculated according to the partial derivation rules (17)(18).

$$y_i = \varphi_i(X) = \varphi_i(x_1, x_2, \dots, x_n) \quad i=1, \dots, m \quad (15)$$

$$F(x_1, x_2, \dots, x_n) = f(\phi_1(X), \phi_2(X), \dots, \phi_m(X)) \quad (16)$$

$$\left. \begin{aligned} \frac{\partial F}{\partial x_1} &= \frac{\partial f}{\partial z_1} \cdot \frac{\partial \phi_1}{\partial x_1} + \frac{\partial f}{\partial z_2} \cdot \frac{\partial \phi_2}{\partial x_1} + \dots + \frac{\partial f}{\partial z_m} \cdot \frac{\partial \phi_m}{\partial x_1} \\ \frac{\partial F}{\partial x_2} &= \frac{\partial f}{\partial z_1} \cdot \frac{\partial \phi_1}{\partial x_2} + \frac{\partial f}{\partial z_2} \cdot \frac{\partial \phi_2}{\partial x_2} + \dots + \frac{\partial f}{\partial z_m} \cdot \frac{\partial \phi_m}{\partial x_2} \end{aligned} \right\} \quad (17)$$

$$\left. \begin{aligned} \frac{\partial F}{\partial x_n} &= \frac{\partial f}{\partial z_1} \cdot \frac{\partial \phi_1}{\partial x_n} + \frac{\partial f}{\partial z_2} \cdot \frac{\partial \phi_2}{\partial x_n} + \dots + \frac{\partial f}{\partial z_m} \cdot \frac{\partial \phi_m}{\partial x_n} \\ \frac{\partial F}{\partial x_k} &= \sum_{i=1}^m \frac{\partial f(z_1, z_2, \dots, z_m)}{\partial z_i} \cdot \frac{\partial \phi_i(X)}{\partial x_k} \quad k=1, \dots, n \end{aligned} \right\} \quad (18)$$

The blocks of the 2<sup>nd</sup> and following hidden layers (Fig. 3) are additionally extended with compound terms (neurons), which form composite function derivatives with respect to the output and input variables of the back-connected previous layer blocks. For example, the 1<sup>st</sup> block of the last (3<sup>rd</sup>) hidden layer forms linear compound neurons (19)(20). The amount of neurons of blocks, which involve composite functions, doubles in each previous back-connected layer. As the couples of input variables of the internal functions  $\phi_1(x_1, x_2)$  and  $\phi_2(x_3, x_4)$  (16) can differ from each other, the partial derivations are calculated separately in respect of each individual block variables. This way the sums (17)(18) consist of only 1 derivative term, which represents a single neuron. The back-calculation of the composite function derivatives is well done by a recursive algorithm in the network tree-like structure [10].

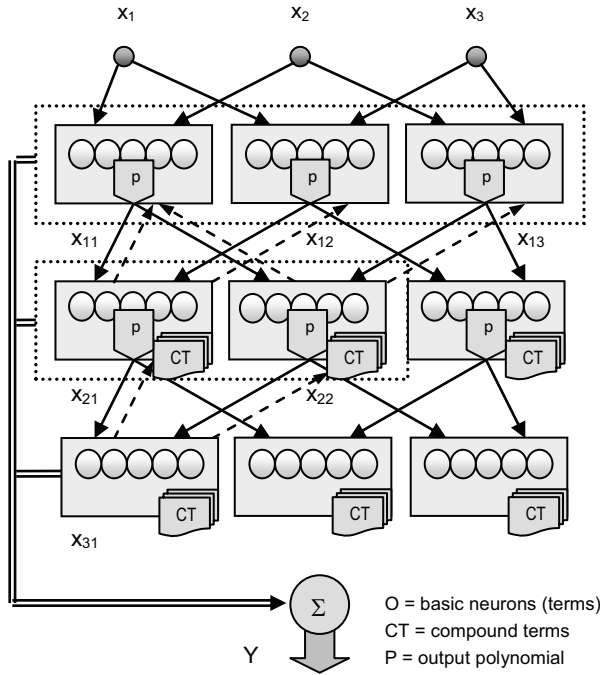


Fig. 3. Backward connections of the 3rd layer 1st block (dashed lines) in the 3-variable multi-layer D-PNN

$$y_2 = \frac{\partial f(x_{21}, x_{22})}{\partial x_{11}} = w_2 \frac{(a_0 + a_1 x_{21} + a_2 x_{22} + a_3 x_{21} x_{22} + a_4 x_{21}^2 + a_5 x_{22}^2)^{\frac{1}{2}}}{1.6 \cdot x_{22}} \cdot \frac{(x_{21})^{\frac{1}{2}}}{1.5 \cdot (b_0 + b_1 x_{11})} \quad (19)$$

$$y_3 = \frac{\partial f(x_{21}, x_{22})}{\partial x_1} = w_3 \frac{(a_0 + a_1 x_{21} + a_2 x_{22} + a_3 x_{21} x_{22} + a_4 x_{21}^2 + a_5 x_{22}^2)^{\frac{1}{2}}}{1.6 \cdot x_{22}} \cdot \frac{(x_{21})^{\frac{1}{2}}}{1.5 \cdot (b_0 + b_1 x_{11})} \quad (20)$$

The compound square and combination terms, which are also calculated according to the composite function derivation rules (21)(22)(23), consist of several sum items. The combination items, which both formulas involve (22)(23), are formed with respect to the variables of both back connected previous layer blocks together.

$$F(x_1, x_2) = f(u, v) = f[\varphi(x_1, x_2), \psi(x_1, x_2)] \quad (21)$$

$$\frac{\partial^2 F}{\partial x_1^2} = \frac{\partial^2 f}{\partial u^2} \left( \frac{\partial \varphi}{\partial x_1} \right)^2 + 2 \frac{\partial^2 f}{\partial u \partial v} \frac{\partial \varphi}{\partial x_1} \cdot \frac{\partial \psi}{\partial x_1} + \frac{\partial^2 f}{\partial v^2} \left( \frac{\partial \psi}{\partial x_1} \right)^2 + \frac{\partial f}{\partial u} \cdot \frac{\partial^2 \varphi}{\partial x_1^2} + \frac{\partial f}{\partial v} \cdot \frac{\partial^2 \psi}{\partial x_1^2} \quad (22)$$

$$\frac{\partial^2 F}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial u^2} \cdot \frac{\partial \varphi}{\partial x_1} \cdot \frac{\partial \varphi}{\partial x_2} + \frac{\partial^2 f}{\partial u \partial v} \left( \frac{\partial \psi}{\partial x_1} \frac{\partial \varphi}{\partial x_2} + \frac{\partial \varphi}{\partial x_1} \frac{\partial \psi}{\partial x_2} \right) + \frac{\partial^2 f}{\partial v^2} \cdot \frac{\partial \psi}{\partial x_1} \cdot \frac{\partial \psi}{\partial x_2} + \frac{\partial f}{\partial u} \cdot \frac{\partial^2 \varphi}{\partial x_1 \partial x_2} + \frac{\partial f}{\partial v} \cdot \frac{\partial^2 \psi}{\partial x_1 \partial x_2} \quad (23)$$

The denominator coefficients (19)(20) balance a length variety of the derivative polynomials in respect of the constant length of the complete numerator polynomial (2). The number of hidden layers in the network should coincide with the total number of input variables in order to enable the D-PNN to form all the possible combination derivative terms of a sum DE solution. D-PNN is trained with only a small set of input-output data samples, in a similar way to the GMDH algorithm [9].

$$Y = \frac{\sum_{i=1}^k y_i}{k} \quad k = \text{number of active neurons (DE terms)} \quad (24)$$

Only some of all the potential DE terms (neurons) may participate in the DE composition, even though they have an adjustable term weight  $w_i$ . A proper neuron combination, which can substitute for a DE solution, is not able to accept the disturbing effect of the rest of the neurons (which may compose other solutions) on the optimization of the parameters. The selection of a fit neuron combination is the principal part of a DE composition and it can apply the simulated annealing (SA) method [16] combined with a standard genetic algorithm [17] in the initial composing phase. The total output  $Y$  of D-PNN is the arithmetical mean of all the active neuron output values (24) so as to prevent a

changeable number of neurons (of a combination) from influencing the total network output value.

$$E = \sqrt{\frac{\sum_{i=1}^M (y_i^d - y_i)^2}{M}} \rightarrow \min \quad \begin{matrix} y_i^d = \text{desired output} \\ y_i = \text{estimated output for the } i^{\text{th}} \text{ training vector} \end{matrix} \quad (25)$$

The error function  $E$ , calculated using the root mean square error (RMSE) method (25), requires a minimization in respect of the polynomial parameters of vectors. The parameter optimization might be performed by means of an evolution algorithm [18] in conjunction with the gradient steepest descent (GSD) method supplied with sufficient random mutations so as to prevent from to be trapped to a local error depression. GSD iterative process requires an error function to be continuous (differentiable) and cannot become negative values. Parameter updates result from the partial derivatives of polynomial DE terms with respect to the single polynomial parameters  $a, b$  [9].

**5. Power demand model experiments**

The presented prediction method, based on daily similarity relations of the power demand cycles, applied the D-PNN (Fig.3.) model trained with 1-parametric historical time-series [A] to form following day hourly estimations. 3 con-sequent day power values (foregoing the forecasting day denomination) of several previous weeks at the same time points form the network input vector; the 24-hour shifted time-series define desired scalar outputs in the training data set. After training the network can predict the following day power series in respect of the current week last 3 day values in a uniform time (Fig.4.- Fig.9.). The training set can consist of 2-6 previous week periods; the testing (prediction) applies the current week input data. A power demand daily period consists typically of 2 peak-hours, morning (midday) and evening peak consumptions (Fig.4.). Most models can obtain a very good approximation (Fig.4.), some are less accurate (Fig.5) or few even difficult to be formed, which are likely influenced by a rude break in weather conditions.

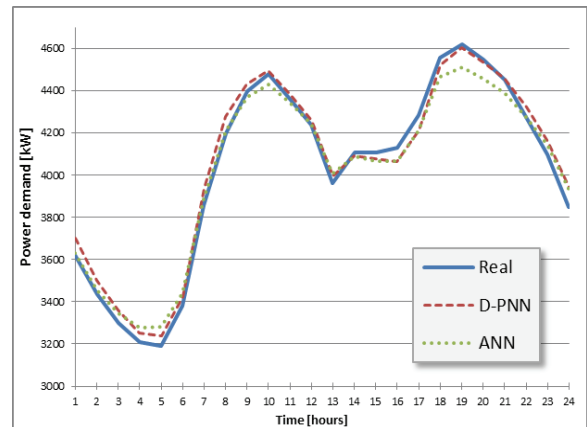


Fig. 4. TEPCO : 1.2.2012 (Wednesday). D-PNN<sub>RMSE</sub> = 51.5, ANN<sub>RMSE</sub> = 56.8

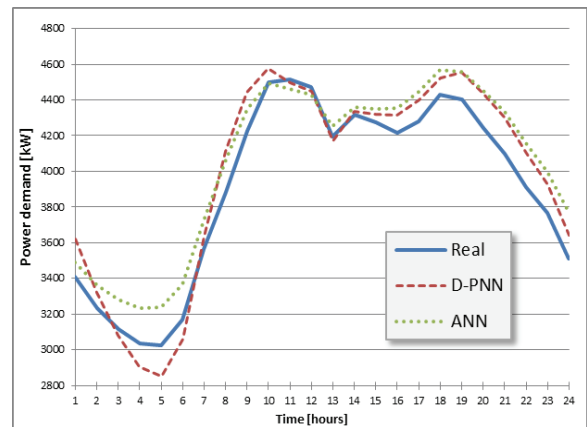


Fig. 5. TEPCO : 7.2.2013 (Tuesday). D-PNN<sub>RMSE</sub> = 134.9, ANN<sub>RMSE</sub> = 161.6

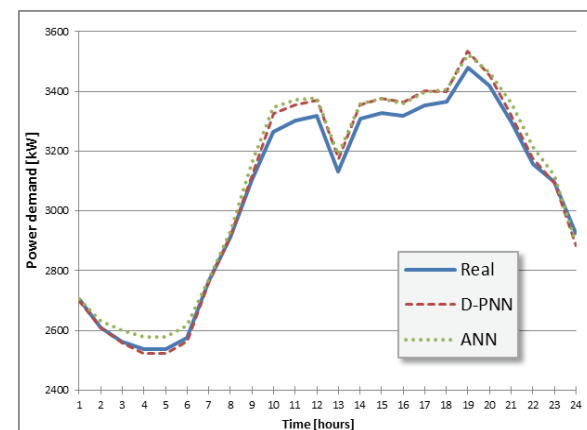


Fig. 6. TEPCO : 13.4.2012 (Friday). D-PNN<sub>RMSE</sub> = 34.8, ANN<sub>RMSE</sub> = 46.1



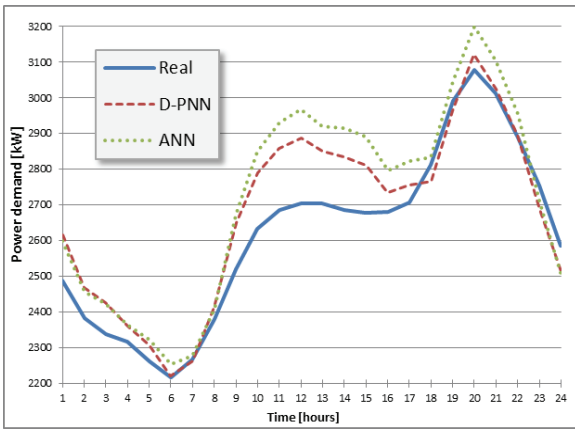


Fig. 7. TEPCO : 13.5.2012 (Sunday).  $D-PNN_{RMSE} = 96.5$ ,  $ANN_{RMSE} = 135.9$

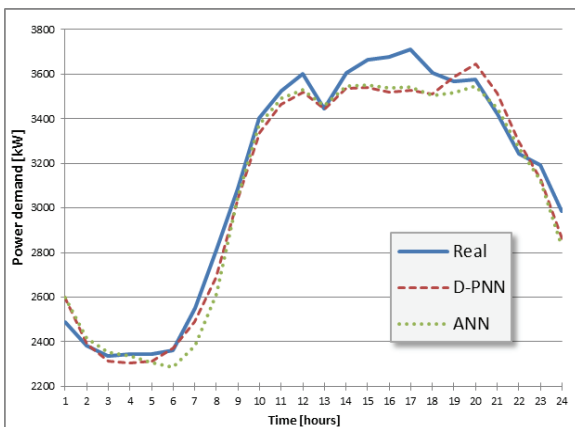


Fig. 8. TEPCO : 2.7.2012 (Monday).  $D-PNN_{RMSE} = 85.7$ ,  $ANN_{RMSE} = 92.2$

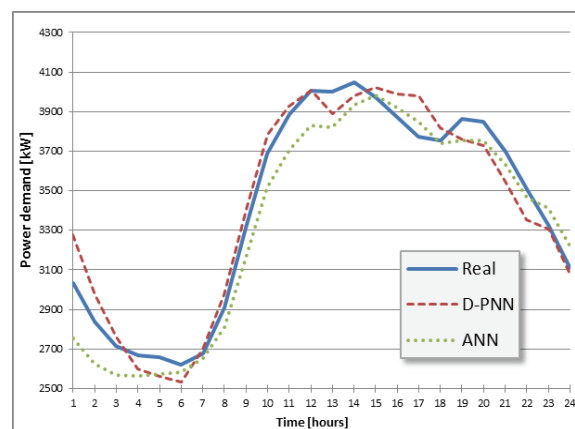


Fig. 9. TEPCO : 11.8.2012 (Saturday).  $D-PNN_{RMSE} = 107.9$ ,  $ANN_{RMSE} = 127.1$

Comparisons were done with 1 or 2-layer ANN consisting of around 15 neurons in each hidden layer, using the sigmoidal activation function and the standard back-propagation algorithm. The ANN forecast models using the same similarity prediction method as the D-PNN, get mostly with any worst results (Fig.10.). The ANN (less commonly D-PNN) adjustment can appear heavily time-consuming in some day predictions.

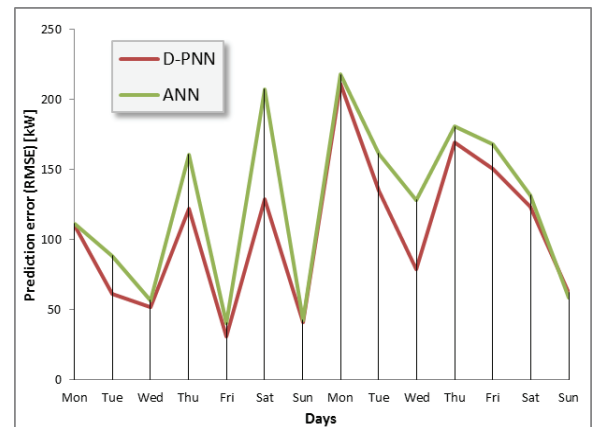


Fig. 10. 2-week daily predictions (30.1.2012 to 12.2.2012), Total average  $D-PNN_{RMSE} = 105.43$ ,  $ANN_{RMSE} = 125.29$  [A].

Fig.11 and Fig.12 show the convergence stability of the 2 presented models (in Fig.4. and Fig.6) in several runs of the evolutionary training process using the SA of the neuron selection in conjunction with the GSD polynomial parameter modifications and random mutations. Each iteration cycle can randomly involve the neuron combination change (only the first 2000 iterations), stochastic GSD parameter adjustment or mutation. Each hundred iteration period can take about several seconds. The training errors of the all above presented models (Fig.4 - Fig.9) decrease gradually but the parallel testing shows different goings. The evolution process commonly falls into a global testing error minimum at the beginning and after it converges to a more or less increased value (Fig.11.); less commonly it can converge all along the training process (Fig.12.). That's why the training requires as a rule to be broken in a test error minimum and this moment can indicate a parallel backward testing of the model for the last not trained week or there is necessary to reserve a testing interval (4-6 hours) at the end of the training data

set. The ANN model training using the standard back-propagation algorithm with momentum results in an analogous test error progress. Sundays and Saturdays power models result largely in higher inaccuracies (Fig.7), which are likely induced by more variable demands of weekends and holidays.

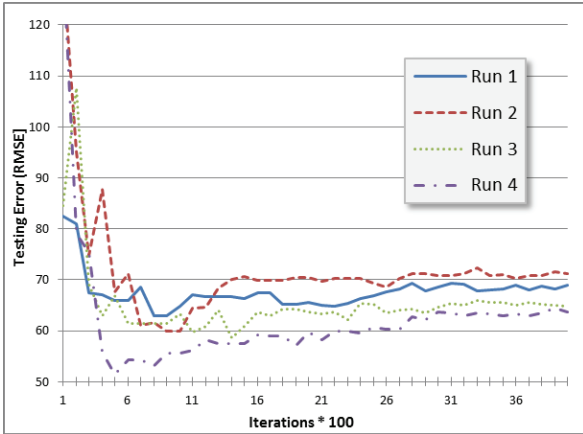


Fig. 11. The testing RMSE convergence during the evolution of the prediction model shown in Fig.4.

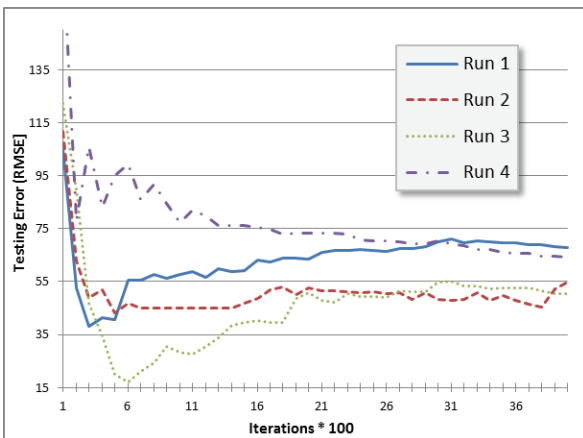


Fig. 12. The testing RMSE convergence during the evolution of the prediction model shown in Fig.6.

The 2<sup>nd</sup> data set [B] is characterized by more complicated power time-rundowns, its daily prediction models (Fig.13.- Fig.17.) get with any worst results, some of them can fairly succeed (Fig.14.) but most result in average error estimations and some few week periods can embody higher inaccuracies, evoked likely by any more exogenous factors or broken weather conditions. The applied power demand 1-parametric data are characterized by a similar progress form, which each of

the successive day cycles takes and which depends mainly on the day denomination in a week. Their form changes progressively (weather or other factors can temporarily influence and deform it), undergoing a gradual advance in the shape from week to week. Thus several successive power day cycles in a week keep their information (time-point relations) in more or less the same character and come through mild changes with some disturbances. Fig.4-Fig.9 and Fig.13-Fig.17 show the gradual daily cycle form progress from winter to summer months.

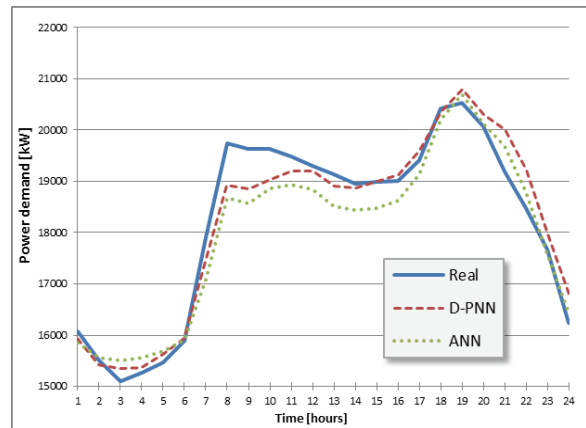


Fig. 13. IESO : 3.2.2012 (Friday).  $D-PNN_{RMSE} = 404.2$ ,  $ANN_{RMSE} = 501.9$

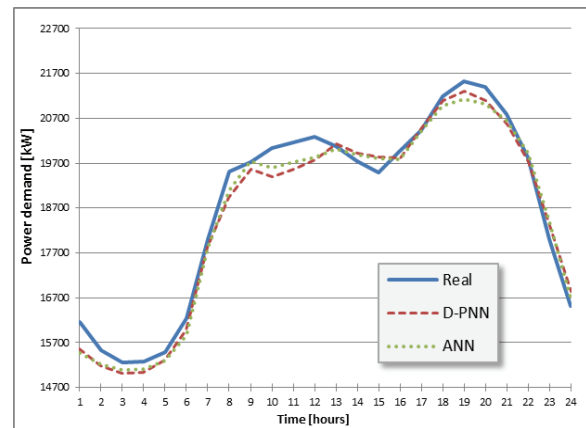


Fig. 14. IESO : 1.3.2012 (Thursday).  $D-PNN_{RMSE} = 335.0$ ,  $ANN_{RMSE} = 312.1$



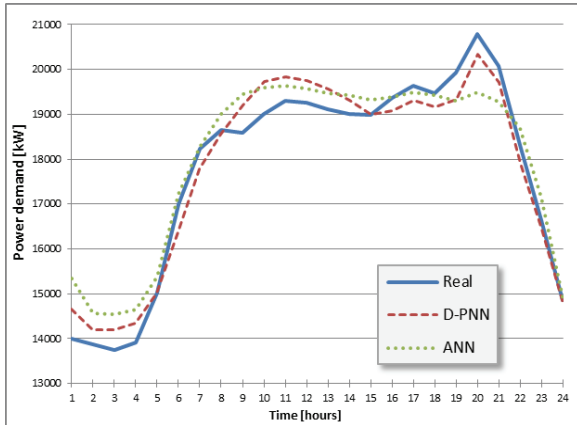


Fig. 15. IESO : 23.4.2012 (Monday).  $D-PNN_{RMSE} = 430.1$ ,  $ANN_{RMSE} = 598.6$

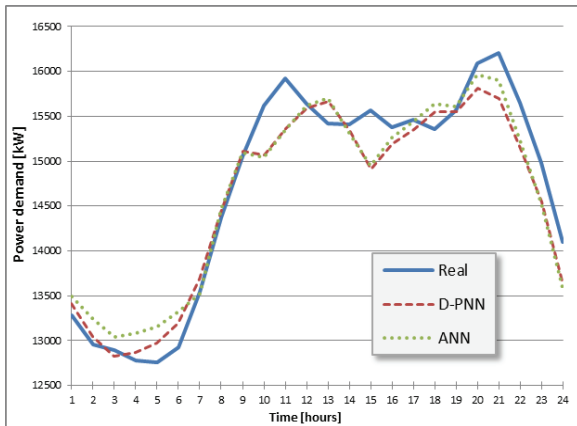


Fig. 16. IESO : 13.5.2012 (Sunday).  $D-PNN_{RMSE} = 317.4$ ,  $ANN_{RMSE} = 326.1$

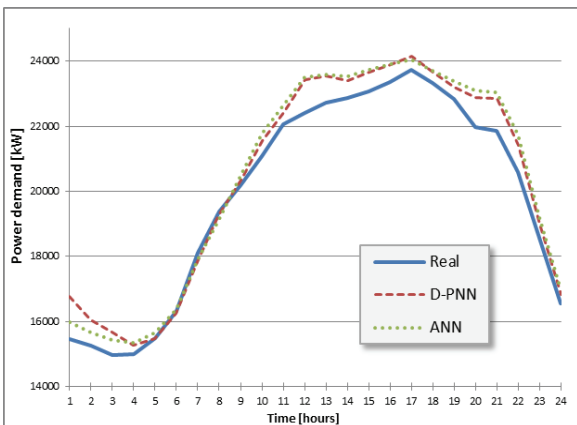


Fig. 17. IESO : 10.7.2012 (Tuesday).  $D-PNN_{RMSE} = 623.8$ ,  $ANN_{RMSE} = 651.2$

The daily prediction model samples (Fig.4-Fig.17) were selected with respect to the variability of testing errors between working days and holidays and many extraneous factors, which can influence considerably some results. Non-considered factors might induce an unstable power load and sudden error increases, making forecasting hard but these irregularities occur only occasionally. The real model could be backward tested for the corresponding days of last not trained week. 3 training week periods seem to define an optimal learning scheme; it is better to apply less or more week-cycles in some cases. The valid number of training week periods could be estimated in practice with respect to previous day(s) best testing results, as the progress of subsequent daily cycles is of a quite uniform character. There is necessary to consider also legal holidays, which corresponding week Sundays (or working days) might replace in the learning scheme.

## 6. Conclusions

The presented method of daily power demand predictions is based on similarity relations of subsequent day progress cycles at the same time points in past week periods. The power history affinities of daily cycles, foregoing to the current week next day denomination, primarily influence the model solution accuracy, as no other considerable effects were taken into account, especially weather conditions. 3-day time interval series of several previous weeks define a learning scheme of the neural network model, which after training can apply the current week last 3-day power series to estimate following 24-hour progress. An optimal forecast model could combine the presented daily similarity relation method with the standard time-series prediction and apply also some possible input exogenous factors, including especially weather conditions if available. These can lower the appearance of unexpected prediction errors. The study used 3 input vector variables of a new neural network type called the D-PNN, which extends the GMDH polynomial neural network complete structure. 1-parametric function time-series of data observations are described by an ordinary differential equation, which is formed and substituted with sum series of polynomial derivative terms. The D-PNN advantages should become more evident in the case of modeling of systems (e.g. including weather conditions) described by differential equations, which

are too complex to be solved using standard soft computation techniques. The increasing number of input factors, which can impact the power demand progress, might significantly influence the effect of the D-PNN application. D-PNN's relative data non-linear regression is contrary to the common soft-computing method approach, which applications are subjected to a fixed interval of absolute values. Its operating principles differ by far from other common neural network techniques. The experimental results proved the D-PNN can define comparable models, despite the fact the network structure, the selective and adjustment algorithms have not been completed yet.

### Acknowledgements

The article has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 funded by Structural Funds of the European Union and state budget of the Czech Republic and in the framework of the project Opportunity for young researchers, reg. no. CZ.1.07/2.3.00/30.0016, supported by Operational Programme Education for Competitiveness and co-financed by the European Social Fund and the state budget of the Czech Republic.

### References

- [1] G.A Darbellay and M. Slama. Forecasting the short-term demand for electricity. *International Journal of Forecasting*, 16:71–83, 2000.
  - [2] C. Unsuhay-Vila, A.C Zambroni, J.W. Marangon-Lima, and P.P. Balestrassi. Electricity demand and spot price forecasting using evolutionary computation combined with chaotic nonlinear dynamic model. *Electrical Power and Energy Systems*, 32:108–1116, 2010.
  - [3] K. Garcia-Ascanio and C. Mate. Electric power demand forecasting using interval time series. *Energy Policy*, 38:715–725, 2010.
  - [4] M.R. Khan and A. Abraham. Short term load forecasting models in czech republic using soft computing paradigms. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 7(4):172–179, 2006.
  - [5] R. Mamlook, O. Badran, and E. Abdulhadi. A fuzzy inference model for short-term load forecasting. *Energy Policy*, 37:1239–1248, 2009.
  - [6] S. Pandian, K. Duraiswamy, C. Rajan, and N. Kanagaraj. Fuzzy approach for short term load forecasting. *Electric Power Systems Research*, 67(6):541–548, 2006.
  - [7] O. Hyde and P.F. Hodnett. An adaptable automated procedure for short-term electricity load forecasting. *IEEE Transactions On Power Systems*, 12(1):84–93, 1997.
  - [8] A.G. Ivakhnenko. Polynomial theory of complex systems. *IEEE Transactions on systems*, 1(4), 1971.
  - [9] Nikolay Y. Nikolaev and Hitoshi Iba. Polynomial harmonic GMDH learning networks for time series modeling. *Neural Networks*, 16:1527–1540, 2003.
  - [10] L. Zjavka. Recognition of generalized patterns by a differential polynomial neural network. *Engineering, Technology & Applied Science Research*, 2(1):167–172, 2012.
  - [11] L. Zjavka. Approximation of multi-parametric functions using the differential polynomial neural network. *Mathematical Sciences*, 33(7):1–7, July 2013.
  - [12] S.-K. Oh, W. Pedrycz, and B.-J. Park. Polynomial neural networks architecture: analysis and design. *Computers & Electrical Engineering*, 29:703–725, 2003.
  - [13] H.S. Hippert, C.E. Pedreira, and R.C. Souza. Neural networks for short-term load forecasting- a review and evaluation. *IEEE Transactions On Power Systems*, 16(1):1239–1248, 2001.
  - [14] D. Randall. *Dimensional Analysis, Scale Analysis, and Similarity Theories*. September 2012.
  - [15] K.L. Chan and W. Y. Chau. Mathematical theory of reduction of physical parameters and similarity analysis. *International Journal of Theoretical Physics*, 18:835–844, November 1979.
  - [16] D. Bertsimas and J. Tsitsiklis. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.
  - [17] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
  - [18] S. Das, A. Abraham, and A. Konar. Particle swarm optimization and differential evolution algorithms. In *Studies in Computational Intelligence*, volume 116, pages 1–38, Berlin, 2008. Springer-Verlag.
- [A] TEPCO Tokyo Electric Power Company – past electricity demand data,  
<http://www.tepco.co.jp/en/forecast/html/download-e.html>
- [B] IESO Independent Electricity System Operator, Power to Ontario, Market data – Hourly demands 2002-2012,  
<http://www.ieso.ca/imoweb/marketdata/marketData.asp>