

# Wind Speed NWP Local Revisions Using a Polynomial Decomposition of the General Partial Differential Equation

Ladislav Zjavka<sup>1</sup>(✉) and Václav Snášel<sup>2</sup>

<sup>1</sup> ENET Centre, VŠB-Technical University of Ostrava, Ostrava, Czech Republic  
ladislav.zjavka@vsb.cz

<sup>2</sup> Faculty of Electrical Engineering and Computer Science,  
Department of Computer Science, VŠB-Technical University of Ostrava,  
Ostrava, Czech Republic  
vaclav.snasel@vsb.cz

**Abstract.** Precise daily weather forecasts are necessary for the utilization of renewable energy sources and their penetration into grid systems. Standard meteorological statistical post-processing methods relate local observations with numerical predictions to eliminate systematic forecast errors. Neural networks, trained with the last historical series, can model the current weather frame to refine a target forecast for specific local conditions and reduce random prediction errors. Their daily correction models can process numerical prediction model outcomes of the same data types (instead of the unknown data) to recalculate 24-hour wind speed forecast series. Global numerical weather models succeed generally in forecasting upper air patterns but are too crude to account for local variations in surface weather. Long-term complex forecast systems, which simulate the dynamics of the complete atmosphere in several layers, cannot exactly detail local conditions near the ground, determined by the terrain relief, structure, landscape character, pattern and other factors. Extended polynomial networks can decompose and solve general linear partial differential equations, being able to model properly unknown dynamic systems. In all the network nodes are produced series of relative polynomial derivative terms, which convergent sum combinations can directly define and substitute for the general differential equation to model an uncertain system target function. The proposed local forecast correction procedure using adaptive derivative regression model can improve numerical daily wind speed forecasts in the majority of days.

**Keywords:** Polynomial neural network · General partial differential equation substitution · Wind speed local correction model · Daily forecast revision

## 1 Introduction

Global forecast systems are based on the numerical integration of differential equation systems, which can describe atmospheric processes on account of meteorological observations. Statistical methods, which apply historical time-series to predict several

hours ahead, are excellent at forecasting idiosyncrasies in local weather [4]. Model output statistics (MOS) post-process first outputs of complex Numerical Weather Prediction (NWP) models, able to forecast large-scale weather patterns, using regression equations to reduce systematic NWP errors and clarify surface weather details [7]. Adaptive methods can improve conventional statistical corrections, eliminating also random forecast errors of NWP models, induced due to uncertain initial conditions and data computational limitations. However their applications are limited to some cases of a limited number of input data for specific model resolutions [8]. Artificial Neural Networks (ANN) can approximate any continuous nonlinear function that offers an effective alternative to more traditional regression techniques. Polynomial Neural Networks (PNN) can adapt some mathematical principles of the Partial Differential Equation (PDE) substitution to decompose and solve the general linear PDE, able to describe properly the local weather dynamics [10]. Extended PNNs, trained for local actual weather conditions to model relevant data fluctuant relations in few last days, can process numerical model outcomes of the same data types (replacing the unknown data) to refine target forecast series for specific local situation features.

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

$n$  – number of input variables  $\mathbf{x}(x_1, x_2, \dots, x_n)$

$\mathbf{a}(a_1, a_2, \dots, a_m), \dots$  – vectors of parameters

Differential polynomial neural network (D-PNN) is a new neural network type, which adopts some principles of the GMDH (Group Method of Data Handling) theory, created by a Ukrainian scientist Aleksey Ivakhnenko in 1968. The GMDH decomposes the complexity of the Kolmogorov-Gabor general polynomial (1), which can describe connections between input and output system variables, into a lot of simpler relationships each described by low order polynomials (2) for every pair of input variables. The GMDH network polynomials can approximate any stationary random sequence of observations and can be computed by either adaptive methods or system of Gaussian normal equations. A typical PNN maps a vector input  $\mathbf{x}$  to a scalar output  $Y$ , which is an estimate of the searched true function [6]. The PNN is a flexible architecture, whose structure is developed through learning. The number of layers of the PNN is not fixed in advance but becomes dynamically meaning that this self-organizing network grows over the trained period.

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

D-PNN defines and solves the general PDE to model an unknown searched function, producing sum series of fractional substitution derivative terms in all the PNN layer nodes. The D-PNN extends the basic PNN structure functionality to decompose the general PDE analogous to the GMDH does the general connection polynomial (1). In contrast with the ANN functionality, each neuron (i.e. substitution PDE term) can be directly included in the total network output, calculated as the sum of selected (active)

neuron values [10]. The D-PNN application merits become evident in the modeling of uncertain dynamic systems (including weather variables), which PDE can preferably describe and which are too complex requiring mass of input data to be solved by standard regression or soft-computing methods [9].

## 2 Short-Term Wind Speed Forecasting

One of the main problems of the wind speed forecasting are wind frequent continuous fluctuations. The NWP models smoothing and averaging the orographic and landscape characteristics lead to the weak representation of local effects on the airflow, emphasized by possible errors in the initial and lateral boundary conditions [6]. Only rough 24–48 h prognoses are usually provided by a meso-scale meteorological model resulting from a NWP global system, which does not take into account local obstacles and terrain asperity that can influence the wind speed (power) in a great measure. The wind power is primarily a result of the current wind-speed, much less affected by other conditions, e.g. the unstable wind direction, speed change, dusts or turbine operating temperature. Wind flow prediction inaccuracies become 3-times larger when the wind-speed is converted into the power through the characteristic curve of a wind turbine. The potential benefits of forecasting the wind energy production are obviously useful in the power control and load scheduling. Wind or power forecast models usually apply lagged values of the average data measurements together with some relevant meteorological variables. Forecasting methods can be generally divided into two main groups based on:

- physics of the atmosphere is applied in NWP models
- statistical consideration takes into account historical time-series only

Statistical methods can apply ANNs, wavelet or classical time-series analysis [5] combined with neuro-fuzzy or ARIMA models. The spatial correlation takes relationships of the wind speed at different sites into account and time-series of the predicted point and its neighboring sites are employed in prediction. Forecasting models developed for one location usually do not match the other site due to variety of reasons like change in terrain, different wind speed patterns and atmospheric factors. The physical method has advantages in long-term prediction, while the statistical method does well in short-term hourly forecasts. The iterative training can provide slightly different model results therefore the final solution is usually the average of several runs. Most of the existing methodologies show some drawbacks such as over-fitting or dependence on the particular local conditions. Hybrid methods combine different model unique features to overcome the single negative performance and try to improve the final forecast. These methods need as a rule a pre-processing of input data to show a better performance. The spatial averaging and temporal interpolation can derive specific prediction intervals to improve wind speed local forecasts. On-line forecast systems can monitor actual wind speed NWP errors and calculate several hour ahead intra-day corrections [2]. Most models combine time-series observations with NWP outputs, to relate measured data with the forecasts [3]. Adaptive post-processing methods require as a rule to reduce substantially the number of input variables,

necessary to model detailed weather data relations [8]. They may equal or exceed the MOS, starting from the given NWP that does not include local characteristics. The proposed revision methods process final NWP model outcomes, which were already corrected primarily for systematic forecast errors by several MOS and secondary data analysis models. The D-PNN forms daily correction models that can adopt corresponding NWP output data to recalculate the target 24-hour wind speed forecast series in consideration of the trained actual weather frame and local specifics.

### 3 The General PDE Decomposition and Substitution

The key idea of the D-PNN is to define and substitute for the general linear PDE, which exact form is not known in advance and which can describe any data relations, with sum series of selected relative polynomial derivative terms (3). The searched function  $u$ , which is possible to calculate as the sum of its derivative terms (+ bias) (3), may be expressed in the form of sum series (5), consisting of convergent series arisen from the competent partial derivatives (3) in the case of 2 input variables.

$$a + bu + \sum_{i=1}^n c_i \frac{\partial u}{\partial x_i} + \sum_{i=1}^n \sum_{j=1}^n d_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \dots = 0 \quad u = \sum_{k=1}^{\infty} u_k \quad (3)$$

$u(x_1, x_2, \dots, x_n)$  – unknown function of  $n$ -input variables  
 $a, B(b_1, b_2, \dots, b_n), C(c_{11}, c_{12}, \dots)$  – polynomial parameters

Substitution PDE terms (3) are built from the GMDH polynomials (2) according to the adapted Similarity Dimensional Analysis (SDA) that applies various formal adaptations of a PDE or data units to form dimensionless characteristic groups of variables. It replaces original mathematical operators and symbols of a PDE by the ratio of corresponding variables [1].

$$y_i = \frac{\left( a_0 + \sum_{i=1}^n a_i x_i + \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j + \dots \right)^{m/n}}{b_0 + b_1 x_1 + \dots} = \frac{\partial^m f(x_1, \dots, x_n)}{\partial x_1 \partial x_2 \dots \partial x_m} \quad (4)$$

$n$  – combination degree of a complete polynomial of  $n$ -variables  
 $m$  – combination degree of polynomial denominator  
 $f$  – polynomial substitution for the unknown function  $u$

The complete combination polynomials (2) of the numerators substitute for the partial unknown functions  $u_k$  of derivative term sum series  $u$  (5), while the reduced polynomials of denominators represent the alternative derivative parts (5) that result from the competent derivatives.

$$\left( \sum \frac{\partial u_k}{\partial x_1}, \sum \frac{\partial u_k}{\partial x_2}, \sum \frac{\partial^2 u_k}{\partial x_1^2}, \sum \frac{\partial^2 u_k}{\partial x_1 \partial x_2}, \sum \frac{\partial^2 u_k}{\partial x_2^2} \right) \quad (5)$$

The root function of the numerator (4), which takes the complete polynomials into competent combination degree, may or may not be applied. The general PDE (3) can be converted into an ordinary differential equation with only time derivatives (7) that describe 1-variable time-series. It is solved analogously using the same PDE multi-variable substitution terms (4) with time-series. Most models take a combined form of the ordinary-time and PDE solutions [10].

$$a + bs + \sum_{i=1}^m c_i \frac{ds(x_i)}{dt} + \sum_{i=1}^m \sum_{j=1}^m d_{ij} \frac{ds(x_i, x_j)}{dt} + \dots + \sum_{i=1}^m cc_i \frac{d^2s(x_i)}{dt^2} + \dots = 0 \quad (6)$$

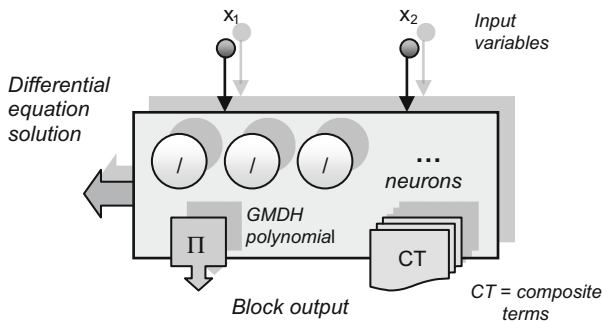
$s(\mathbf{x})$  – function of independent time-series observations  $\mathbf{x}(x_1, x_2, \dots, x_m)$

Blocks (extended nodes) of the D-PNN (Fig. 1.) form substitution sum DE terms (neurons in this context) with the same input variables, one for each fractional polynomial derivative combination (4). Each block contains a single output polynomial (2), without derivative part. Neurons do not affect the block output but can be directly included in the total network output sum calculation of a DE solution. Each block has  $l$  and neuron 2 vectors of adjustable parameters  $\mathbf{a}$ , and  $\mathbf{a}, \mathbf{b}$  respectively.

$$F\left(x_1, x_2, u, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, \frac{\partial^2 u}{\partial x_2^2}\right) = 0 \quad (7)$$

where  $F(x_1, x_2, u, p, q, r, s, t)$  is a function of 8 variables

While using 2 input variables the 2<sup>nd</sup> order partial DE may be expressed in the form (7), which involves derivative terms formed in respect of all the GMDH polynomial (2) variables. Each D-PNN block forms 5 corresponding simple derivative neurons in respect of single  $x_1, x_2$  (7) squared  $x_1^2, x_2^2$  (8) and combination  $x_1x_2$  (9) derivative



**Fig. 1.** D-PNN blocks forms simple ( $l$ ) and composite (CT) substitution PDE terms (neurons)

variables, which combination sum can directly solve and substitute for the 2<sup>nd</sup> order partial DE (6), most often used to model physical or natural system non-linearities.

$$y_1 = \frac{\partial f(x_1, x_2)}{\partial x_1} = w_1 \frac{(a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2)^{1/2}}{b_0 + b_1x_1} \quad (8)$$

$$y_4 = \frac{\partial^2 f(x_1, x_2)}{\partial x_2^2} = w_4 \frac{a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2}{b_0 + b_1x_2 + b_2x_2^2} \quad (9)$$

$$y_5 = \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} = w_5 \frac{a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2}{b_0 + b_1x_{11} + b_2x_{12} + b_3x_{11}x_{12}} \quad (10)$$

The Root Mean Squared Error (RMSE) is calculated for the polynomial parameter optimization and neuron combination selection (11).

$$RMSE = \sqrt{\frac{\sum_{i=1}^M (Y_i^d - Y_i)^2}{M}} \rightarrow \min \quad (11)$$

## 4 Backward Differential Polynomial Neural Network

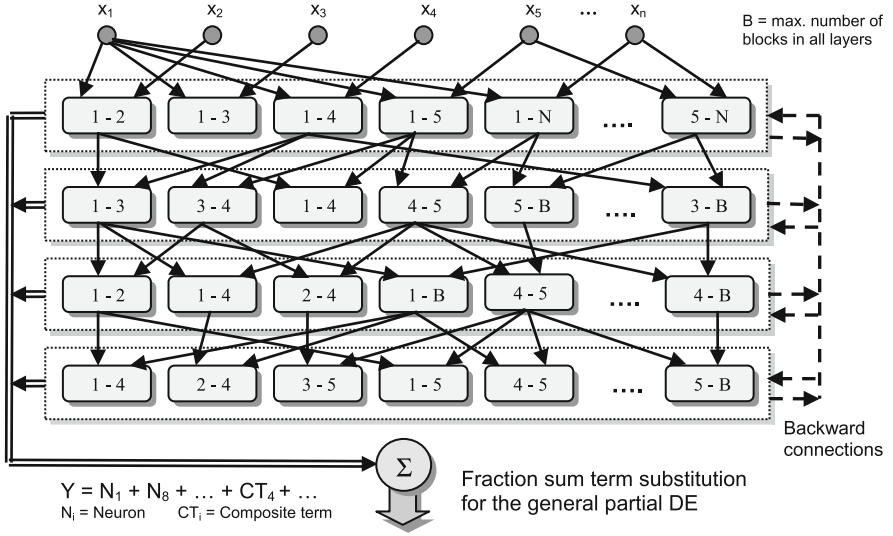
Multi-layer networks form composite functions (Fig. 2.). Composite Terms (CT), which substitute for the derivatives with respect to variables of previous layers, are calculated according to the composite function (12) partial derivation rules (13).

$$F(x_1, x_2, \dots, x_n) = f(z_1, z_2, \dots, z_m) = f(\phi_1(X), \phi_2(X), \dots, \phi_m(X)) \quad (12)$$

$$\frac{\partial F}{\partial x_k} = \sum_{i=1}^m \frac{\partial f(z_1, z_2, \dots, z_m)}{\partial z_i} \cdot \frac{\partial \phi_i(X)}{\partial x_k} \quad k = 1, \dots, n \quad (13)$$

Each D-PNN block can form 5 simple neurons (8)–(10) and the blocks of the 2<sup>nd</sup> and next hidden layers produce additional CTs using composite substitution derivatives with respect to the output and input variables of the back connected previous layers blocks, e.g. 3<sup>rd</sup> layer blocks can form linear CTs in respect of 2<sup>nd</sup> (14) and 1<sup>st</sup> layer (15). The number of block neurons, i.e. CTs that include composite function derivatives, doubles each previous back-connected layer. Thus the probability activations  $P_A$  of CTs, formed with respect to the previous layers block derivative input variables, must halve together with the increasing number of hidden layers they backward comprise in the network tree-like structure (Fig. 2) [10].

$$y_2 = \frac{\partial f(x_{21}, x_{22})}{\partial x_{11}} = w_2 \frac{(a_0 + a_1x_{21} + a_2x_{22} + a_3x_{21}x_{22} + a_4x_{21}^2 + a_5x_{22}^2)^{1/2}}{x_{22}} \cdot \frac{x_{21}}{b_0 + b_1x_{11}} \quad (14)$$



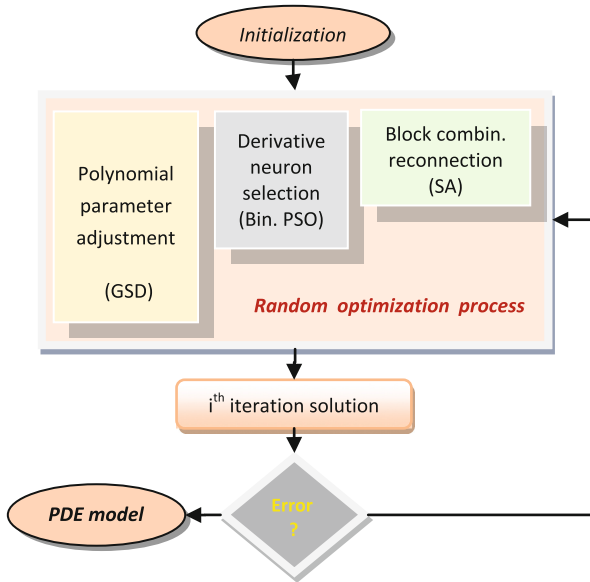
**Fig. 2.** N-variable D-PNN selects from 2-variable combination blocks in each hidden layer

$$\begin{aligned}
 y_3 &= \frac{\partial f(x_{21}, x_{22})}{\partial x_1} \\
 &= w_3 \frac{(a_0 + a_1 x_{21} + a_2 x_{22} + a_3 x_{21} x_{22} + a_4 x_{21}^2 + a_5 x_{22}^2)^{1/2} \cdot (x_{21})^{1/2}}{x_{22}} \cdot \frac{x_{11}}{x_{12} \cdot (b_0 + b_1 x_1)} \quad (15)
 \end{aligned}$$

The square and combination derivative terms are formed analogously. The D-PNN with an increased number of input variables must select from possible block nodes in each hidden layer (analogously to the GMDH combinatorial explosion) as the number of input combination couples grows exponentially in each next hidden layer. The D-PNN need not define all the possible PDE terms (using a deep multi-layer structure), e.g. 4–5 hidden layers can form an optimal model for 10-20 input variables (Fig. 3.).

$$Y = \frac{\sum_{i=1}^k y_i}{k} \quad k = \text{actual number of active neurons} \quad (16)$$

Only some of all the potential combination PDE terms (neurons) may be included in a PDE solution, in despite of they have the adjustable term weight ( $w_i$ ). A specific neurons combination, which forms a PDE solution, is not able to accept a disturbing effect of the rest of the neurons (possible to form other solutions) in the parameter optimization. The D-PNN total output  $Y$  is the arithmetic mean of active neurons output values so as to prevent their changeable number (in a combination) from influencing the total network output value (16) [10].



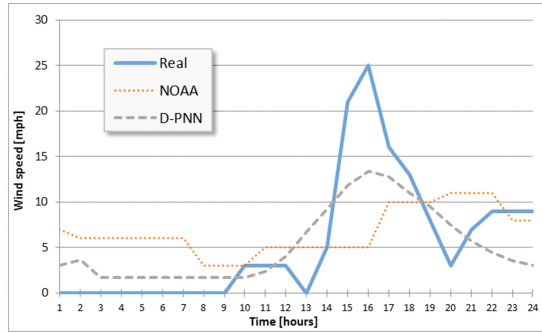
**Fig. 3.** 3-simultaneous separate processes of the structure and parameters optimization

The 2 simultaneous random gradually finishing processes of the optimal block 2-inputs and neurons combination selection are the principal initial phase of the D-PNN structure formations and PDE composition, performed simultaneously along with the continual polynomial parameters adjustment using the Gradient Steepest Descent (GSD) method (Fig. 4). The binary Particle Swarm Optimization (PSO), able to solve large combinatorial problems, may perform the optimal neurons selection. The Simulated Annealing (SA), which not only accepts changes that decrease the objective function (in a minimization problem) but also some changes that increase it, can improve the block inputs reconnection process.

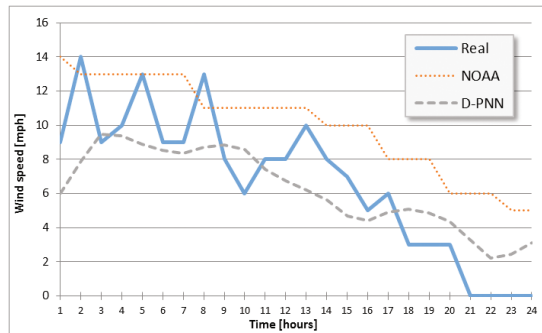
## 5 Wind Speed NWP Local Revisions

The National Oceanic and Atmospheric Administration (NOAA) provides the free National Weather Service (NWS), which includes among others tabular 4-day forecasts of the hourly average temperature, relative humidity and wind speed at a selected locality [11]. Daily wind speed correction models processed the last 3-input time-series (lags) of the above 3 local forecasted quantities, i.e. 9 input variables in total, in Helena, Montana [11] (Fig. 4, 5 and 6). The D-PNN was trained with corresponding hourly observation series (of the same data types) for the period of 2 to 6 days previously (48-144 data samples). The training data can be downloaded from NOAA complete free historical archives [12], which the Weather Underground (WU) shares [13], or current daily tabular NOAA observations [14] at many land-based stations.

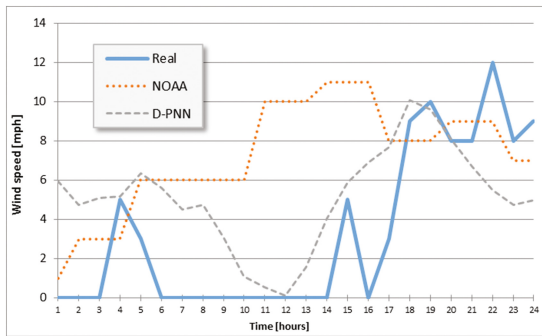




**Fig. 4.** 29.1.2014, Helena - RMSE: NOAA = 6.79, D-PNN = 4.31

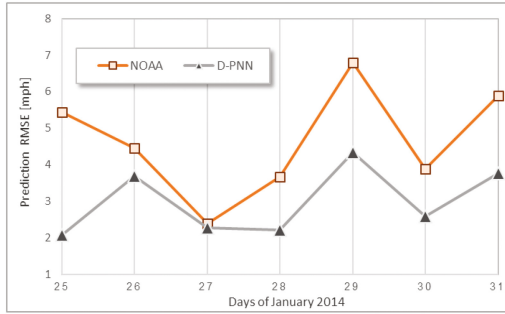


**Fig. 5.** 30.1.2014, Helena - RMSE: NOAA = 3.89, D-PNN = 2.59



**Fig. 6.** 31.1.2014, Helena - RMSE: NOAA = 5.90, D-PNN = 3.77

Settled weather periods of several days allow to train the D-PNN with historical observations, which actual fluctuant relevant data relations do not change essentially in the forecasted days (Fig. 6). It is infeasible to revise the NOAA wind speed NWP in sporadic days of an overnight break change in the weather however intervals of more or



**Fig. 7.** The week wind speed average prediction RMSE:  $NOAA = 4.65$ ,  $D-PNN = 2.99$ , Helena

less settled stable conditions tend to prevail (Fig. 7). The NOAA North American Meso-scale (NAM) forecast model may be also accurate enough to allow any successful revisions using the proposed procedure.

Figure 7 shows daily wind speed prediction RMSEs of the original NOAA and D-PNN minimal error correction models in a week period; the  $x$ -axis represents the ideal (real) approximation.

## 6 Conclusions

Meteorological conditions mostly do not change fundamentally within short time periods, which allow to form models that represent the forecasted day actual data relations. An appropriate NWP data analysis can detect overnight weather changes, followed by days, which out of date models show big prediction errors, to reject the failed revisions. Proposed method results are naturally bound to the accuracies of input NWP model outcomes, which are not completely valid and that enter correction models. The presented D-PNN output wind speed predictions were compared with the observations at the complete 24-hour forecasted interval, which are clearly not known in real-time, to select the best PDE correction solutions. It is necessary to estimate the D-PNN optimal daily training errors or test its models with the previous day several hours past forecasts and corresponding last observations to eliminate the weather dynamics and the NWP outputs inaccuracy in real predictions. The approximate training or testing parameters will any debase the presented minimal prediction errors however some improvements in the NWP are feasible according to the experiments.

**Acknowledgement.** This paper is supported by the following projects: LO1404: Sustainable development of ENET Centre; CZ.1.05/2.1.00/19.0389 Development of the ENET Centre research infrastructure; SP2017/159 Students Grant Competition and TACR TH01020426, Czech Republic.

## References

1. Chan, K., Chau, W.Y.: Mathematical theory of reduction of physical parameters and similarity analysis. *Int. J. Theoret. Phys.* **18**, 835–844 (1979)
2. Hirataa, Y., Yamadac, T., Takahashic, J., Aiharaa, K., Suzuki, H.: Online multi-step prediction for wind speeds and solar irradiation: evaluation of prediction errors. *Renewable Energy* **67**, 35–39 (2014)
3. Lei, M., Shiyang, L., Chuanwen, J., Hongling, L., Yan, Z.: A review on the forecasting of wind speed and generated power. *Renew. Sustain. Energy Rev.* **13**, 915–920 (2009)
4. Liu, H., Tian, H.-Q., Chen, C., Fei Li, Y.: A hybrid statistical method to predict wind speed and wind power. *Renewable Energy* **35**, 1857–1861 (2010)
5. Monfared, M., Rastegar, H., Kojabadi, H.: A new strategy for wind speed forecasting using artificial intelligent methods. *Renewable Energy* **34**, 845–848 (2009)
6. Nikolaev, N.Y., Iba, H.: *Adaptive Learning of Polynomial Networks. Genetic and Evolutionary Computation.* Springer, New York (2006)
7. Ranaboldo, M., Giebel, G., Codina, B.: Implementation of a model output statistics based on meteorological variable screening for short-term wind power forecast. *Wind Energy* **16**, 811–826 (2013)
8. Sweeney, C.P., Lynch, P., Nolan, P.: Reducing errors of wind speed forecasts by an optimal combination of post-processing methods. *Meteorol. Appl.* **20**, 32–40 (2013)
9. Zjavka, L.: Wind speed forecast correction models using polynomial neural networks. *Renewable Energy* **83**, 998–1006 (2015)
10. Zjavka, L., Snášel, V.: Constructing ordinary sum differential equations using polynomial networks. *Inf. Sci.* **281**, 462–477 (2014)
11. NAM forecasts, <http://forecast.weather.gov/MapClick.php?lat=46.60683&lon=-111.9828333&lg=english&&FcstType=digital>
12. NOAA National Climatic Data Center archives, [www.ncdc.noaa.gov/orders/qclcd/](http://www.ncdc.noaa.gov/orders/qclcd/)
13. WU history, [www.wunderground.com/history/airport/KGTF/2015/10/3/DailyHistory.html](http://www.wunderground.com/history/airport/KGTF/2015/10/3/DailyHistory.html)
14. NOAA tabular weather observations, [www.wrh.noaa.gov/mesowest/getobext.php?wfo=tx&sid=KGTF](http://www.wrh.noaa.gov/mesowest/getobext.php?wfo=tx&sid=KGTF)