Al-Messabi, N., Goh, C., El-Amin, I., and Li, Y. (2014) Heuristically enhanced dynamic neural networks for structurally improving photovoltaic power forecasting. In: 2014 International Joint Conference on Neural Networks (IJCNN), 6-11 Jul 2014, Beijing, China.

Heuristically Enhanced Dynamic Neural Networks for Structurally Improving Photovoltaic Power Forecasting

Naji Al-Messabi*, Cindy Goh*, Ibrahim El-Amin**, Yun Li*
* School of Engineering, University of Glasgow, Glasgow G12 8 QQ, U.K.,
(n.al-messabi.1@research.gla.ac.uk, Cindy.Goh@glasgow.ac.uk, Yun.Li@glasgow.ac.uk)
** Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia,
(imelamin@kfupm.edu.sa)

Abstract— **Among renewable generators, photovoltaics (PV) is showing an increasing suitability and a lowering cost. However, integration of renewable energy sources possesses many challenges, as the intermittency of these non-conventional sources often requires generation forecast, planning and optimal management. There exists scope to improve present PV yield forecasting models and methods. For example, the popular dynamic neural network modelling method suffers from the lack of a selection mechanism for an optimal network structure. This paper develops an enhanced network for short-term forecasting of PV power yield, termed a 'focused time-delay neural network' (FTDNN). The problem of optimizing the FTDNN structure is reduced to optimizing the number of delay steps and the number of neurons in the hidden layer alone and this problem is conveniently solved through heuristics. Two such algorithms, a genetic algorithm and particle swarm optimization (PSO) have been tested and both prove efficient and can improve the forecasting accuracy of the dynamic network. Given the success of the PSO in solving this discontinuous structural optimization problem, it is expected that PSO offers potential in optimizing both the structure and parameters of a forecasting model.**

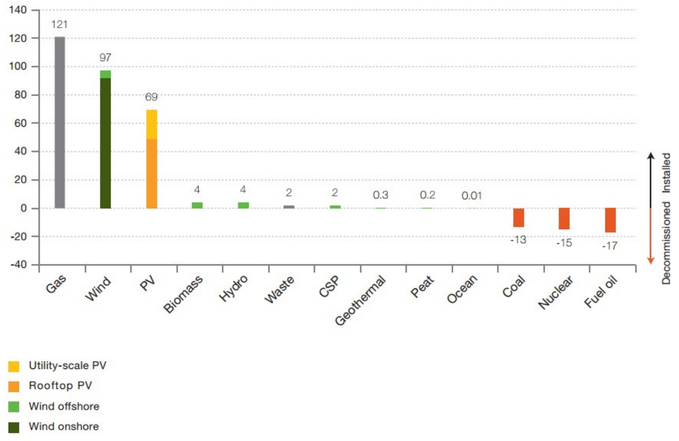
I. INTRODUCTION

Photovoltaic (PV) energy is now positioned amongst the top three new power generation means installed in Europe, as shown in Figure 1, and is expected to remain so [1]. Power from PV sources provides a number of benefits over other renewable energy sources (RES). It can be supplied locally to loads, reducing the cost of transmission lines and associated power losses. Furthermore, advances in technology and large scale manufacturing have led to the decline in PV cost at a steady rate [2]. Despite a high capital setup cost, the operation and maintenance costs of PV are almost zero [3].

However, like other RES, PV sources pose a number of integration challenges such as the impact on voltage profile [4], impact on operational costs of the grid [5], regulation and load-following requirements [6], and other issues investigated in numerous research papers [7, 8]. Knowing in advance an expected yield from PV sources will aid in tackling these challenges, which includes proper planning of available generation sources and providing insight into impact of PVs on the power network. However, the forecasting task requires non-primitive techniques, as power yield from PVs is intermittent in nature. The intermittent and non-linear characteristics of PV data is due to an interplay of various factors such as the variability in sunrise and the amount of sunshine, sudden changes in atmospheric conditions, cloud movements and dust [9]. The PV power data can thus be viewed as consisting of two parts: the deterministic and the stochastic parts [23]. The former represents the mathematical equations of irradiance that depend on location, sun's position, and equations of PV cells, whilst the latter represents the sudden atmospheric changes such as dust, clouds, and wind blow.

Various mathematical models that catch physics of PVs are possible but are inaccurate or impractical for large systems [10]. Data-driven models based on statistics or artificial-intelligence are gaining popularity, as they provide the advantage of simplicity and usability. Owing to the capability to handle nonlinearity and time-series data and absence of requirement for transformation to stationary data [15], Dynamic Neural Networks (DNNs) have been studied for PV forecasting, such as the 'Focused Time-Delay Neural Networks' (FTDNN) and the 'Distributed Time-Delay Neural Networks' (DTDNN) [12]. Advantages and disadvantages of existing models are briefly reviewed in Section 2.

This paper focuses on FTDNNs without exogenous inputs to cover very short-term forecasting and to improve the existing methods for forecasting PV data. Section 3 presents a framework of the FTDNN. This is followed by the development of a methodology to optimize the structure and parameters of the FTDNN in Section 4. Results and validations are presented in Section 5 and Conclusions are drawn in Section 6.



Source: EPIA, ESTELA, EWEA, Platts

Fig. 1. Net generation capacity (in GW) added in the EU 27 from 2000 to 2012 [1].

II. SOLAR PV POWER FORECASTING

There exist various forecasting models proposed for PV systems [17-37]. The simplest ones are naïve or persistence models where next value of power is assumed to be same as the previous step. Such models are usually taken as reference models in forecasting studies [16, 17]. Persistence can be written as follows:

$$\hat{x}_{t+k} = x_t \tag{1}$$

where $\hat{x}_{t+k}$ is the predicted value of the variable x for the next k steps and x_t is the current value at temporal instant t.
Another version of the persistence model (called Diurnal-persistence model) can be written as follows [17]:

$$\hat{x}_{t+k} = x_{t+k} - c \tag{2}$$

The constant c = 24 or 48 and this implies that the value of the predicted variable k steps ahead is equal to its value at the same time point of the previous day (24 hours back) or two days before (48 hours back) respectively.
Another common method is the 'clear-box' model based on physical principles, such as the double diode model [18], the simplified single diode model (SSDM), and further SSDMs in a descending order of complexity. A higher complexity can provide better accuracy on the expense of increased computational burden and unsuitability for real-time applications [19]. An SSDM that gives a good compromise between simplicity and accuracy [20] is shown in Figure 2.

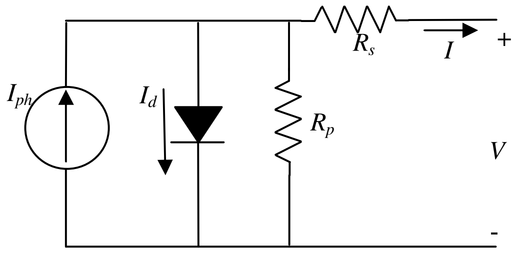


Fig. 2. PV cell/array Single Diode Simplified model

The following equations describe the relation between the current and voltage output of the PV cell/array:

$$I = I_{ph} - I_d - \frac{V+R_s I}{R_p} \tag{3}$$

$$I_d = I_o(e^{\left(\frac{V+R_s.I}{aV_t}\right)} - 1) \tag{4}$$

where I is the output current of the cell in amperes, V is the solar cell voltage in volts, I_{ph} is the photocurrent in amperes, I_d is the Shockley diode equation, I_o is the reverse saturation or leakage current of the diode, $V_t = kT/q$ is the thermal voltage of the array, q is the electron charge ($1.60217646 \times 10^{-19}$ C), k is the Boltzman constant ($1.3806503 \times 10^{-23}$ J/K), T is the temperature of the cell in kelvin, and a is the ideality factor constant. More details of these equations can be found in [19, 20]. To calculate power yield, values for I and V are usually computed using numerical methods [18, 20]. The mathematical approach is usually tedious especially when applied to large or widely spread PV systems [10].

Another way to model solar data is to use statistical methods. Regression models can be used where power value is expressed as a regression of previous power values, irradiance, and temperature [21]. Statistical approaches adopt classical time-series forecasting methods that assume data to be stationary. Therefore, different approaches are used to confront the non-stationarity of solar data such as differencing the original data [22] and normalizing against clear sky model [17]. Auto-regressive (AR), AR with exogenous input (ARX), and AR with integrated moving average (ARIMA) [17, 23] are some of the famous statistical models used in solar PV forecasting. As the parameters in these models usually do not represent a physical phenomenon or quantity, such models are often referred to as 'back-box' models or functional approximates.

The artificial neural network (NN) is another example of these models and is gaining popularity in PV forecasting owing to their modularity in handling non-linear models. There are various structures of NN, but they can be categorized into two static and dynamic types. Static NNs have been used to predict PV power [24] and irradiance [25] trained by standard back-propagation or heuristic algorithms [26].

Dynamic NNs has the ability to accommodate time-series data [14] and are therefore studied for PV forecasting in this paper. Nonlinear Autoregressive Neural Network with External input (NARX) [10], Recurrent NN (Elman type model) [27], and Distributed time delay NN [28] are some of the dynamic NNs investigated for PV forecasting applications.

III. FOCUSED TIME-DELAY NEURAL NETWORK

Time-delays in dynamic NNs can either be 'distributed' in different layers or 'focused' in the input layer only as illustrated in Figure 3 with one input variable. This structure has been found to be more suitable for PV forecasting applications [12]. The network shows *m* delayed inputs fed to the input layer, where Z^{-di} is a unit delay operator that yields $u(t-d_i)$ when it operates on a given input $u(t)$, and the connections between *m* inputs and p_1 neurons in the input layer are given weights that are lumped to a matrix W_1 called the input weight matrix. Similar matrices exist between hidden layers and between final layer and output layer. Furthermore, the output of each neuron is offset by a value called bias that is lumped into a vector b_n for a given n^{th} layer. The activation function chosen for input and hidden layers is the tansigmoid function. The output activation function $f_o(.)$ is usually a purelin function which is a linear summation of the outputs of the output layer neurons.

An example of applications of an FTDNN can be found in [29] and in the previous work [12]. Comparing with distributed or other NNs, the advantage of the FTDNN is lightweight, does not require more inputs to work, offers a simple structure of one hidden layer only, and is sufficiently generic to adapt to changes in input data quickly.

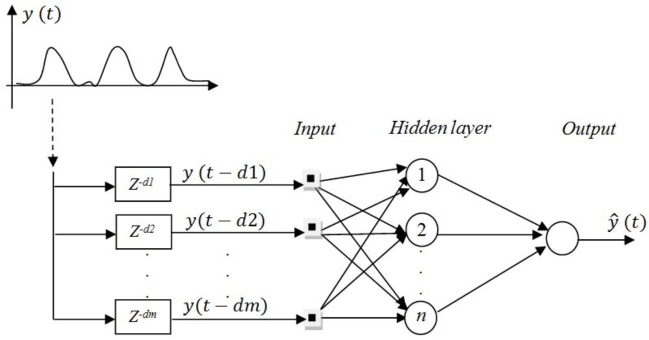


Fig. 3. Focused Time-Delay NN configuration

IV. Network and Parameter Training Methods

In this work, the PV power is taken as a time-series data. The forecasting is handled by an FTDNN where future values are predicted based only on historical data. The advantage of the FTDNN proposed in our previous work [12] is that optimizing the structure is reduced to optimizing the number of delays and the number of neurons in the hidden layer. In any NN applications, choosing the number of neurons will be necessary for maximizing performance.

Here, the FTDNN is optimized by a heuristic algorithm. Once the structure is set, calculating the weights and biases can be performed easily using a back-propagation (BP) algorithm as they are efficient in handling a large size weights [11]. The Levenberg-Marquardt (LM) version of BP [13] is used here as it was shown faster in the previous work [12]. Within the optimization loop, the FTDNN is trained using LM and the root-mean square error (RMSE) of testing is used as an objective function. The overall structure of the method is shown in Figure 4.

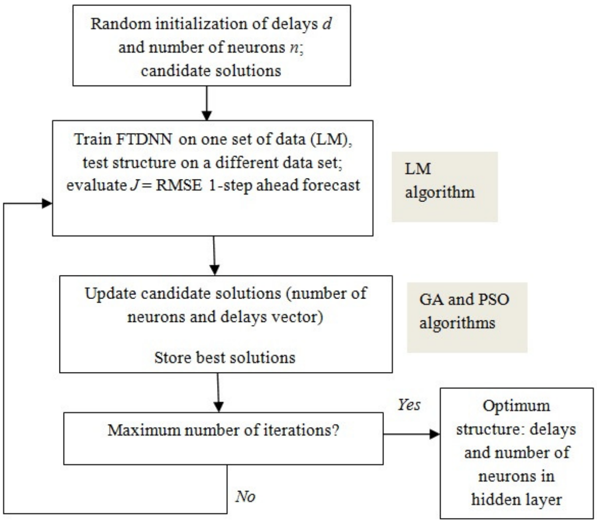

Random initialization of delays d and number of neurons n; candidate solutions
Train FTDNN on one set of data (LM), test structure on a different data set; evaluate J = RMSE 1-step ahead forecast
LM algorithm
Update candidate solutions (number of neurons and delays vector)
Store best solutions
GA and PSO algorithms
Maximum number of iterations?
Yes
No
Optimum structure: delays and number of neurons in hidden layer

Fig. 4. Flow chart for optimizing structure of FTDNN

In this work, the solution space for the number of neurons is set between $n = 2$ and 20 neurons for the hidden layer. The number of hidden layers is fixed to one as such structure for MLP is adequate to approximate any given function, as stated by the universal approximation theorem [14]. The delays, d_i, are searched in a solution space between 1 and 30 corresponding to 1 and 30 steps back values respectively; maximum number of delays, *m*, is set at 10. The search for the best values of delays and neurons is done using Particle Swarm Optimization (PSO) and a Genetic Algorithm (GA) as they are powerful heuristic search methods without the need for the evaluation of derivatives.

Genetic Algorithms are powerful domain independent search technique, which have been successfully applied to various areas of power system [30]. A set of potential solutions, described as a population of individuals, are encoded as chromosomes. A new set of solutions, called offsprings, are created in a new generation (iteration) by crossing some of the strings of the current generation. This process is called crossover, which is repeated at every generation. Further, new characteristics are introduced to add diversity, by altering some of the strings of the offsprings randomly in mutation. The new offsprings replace some or all previous generation population depending on the reinsertion rate chosen by the user.

Particle Swarm Optimization [31] is inspired by social behaviour of bird flocking or fish schooling. It can be applied as follows:

Step 1: Initialize a population (array) of particles with random positions and velocities *v* on *d* dimension in the problem space. The particles are generated by randomly selecting a value with uniform probability over the d^{th} optimized search space $[x_d^{\min}, x_d^{\max}]$.

Step 2: For each particle *x*, evaluate the desired optimization fitness function, *J*, in *d* variables.

Step 3: Compare particles fitness evaluation with x_{pbest}, which is the particle with best local fitness value. If the current value is better than that of x_{pbest}, then set x_{pbest} equal to the current value and x_{pbest} locations equal to the current locations in d-dimensional space.

Step 4: Compare fitness evaluation with population overall previous best. If current value is better than x_{gbest}, the global best fitness value then reset x_{gbest} to the current particle's array index and value.

Step 5: Update the velocity v as follows:

$$\begin{aligned} v_{id}(k) &= w(k)v_{id}(k-1) + \\ &\varphi_1 .rand_1(x_{idpbest}(k-1) - x_{id}(k-1)) + \\ &\varphi_2 .rand_2(x_{idgbest}(k-1) - x_{id}(k-1)) \end{aligned} \qquad (6)$$

where, k is the number of iteration, i is the number of the particles that goes from 1 to n, d is the dimension of the variables, and $rand_{1,2}$ is a uniformly distributed random number in (0, 1), $\varphi_{1,2}$ are acceleration constants and are set, as recommended by investigators [31], equal to 2. The weight w is often decreased linearly from about 0.9 to 0.4 during the search process.

Step 6: Update position of the particles,

$$x_{id}(t) = v_{id}(t) + x_{id}(t-1) \qquad (7)$$

Step 7: Loop to 2, until a criterion is met, usually a good fitness value or a maximum number of iterations (generations) m is reached.

There are several feasibility issues that need to be addressed when using heuristic algorithms to select suitable values for delays and the number of neurons. Heuristic algorithms can find solutions with repeated entries in the solution vector; which is not feasible in such an application. For example, GA/PSO may find a solution such as D = [1 2 2 4 5..] where the second '2' is a repetition. This means that "$y(t\text{-}1)$, $y(t\text{-}2)$, $y(t\text{-}2)$, $y(t\text{-}4)$, .." are delayed inputs fed to the network. In order to circumvent this, repeated values are removed from the solutions in each iteration of the search. Hence, the afore-mentioned solution vector will be truncated to D = [1 2 4 5..]. Another point in "automating" the structure of FTDNN through heuristics is that the knowledge of the problem helps in choosing suitable range for solution space; values close to current time usually carry the useful information to predict next values and an excessive number of neurons should be avoided as unnecessary complexity could lead to over-fitting.

V. Results and Validation

The proposed FTDNN and its training method are tested on PV data collected from 1.76kW array installed at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia [12]. As presented in Figure 4, FTDNN is trained using on one data set (2-6 February 2010, 5 days) and tested on a different data set (7-10 February 2011, 4 days) for validation. The data was measured at a ten minutes interval where 144 data points were collected over a 24 hour period. The Root Mean Square Error (RMSE) for a one step ahead prediction is used as the fitness/objective function value for a given structure in the loop shown in Figure 4.

TABLE I
Comparison between PSO and a GA: Parameters and Optimization results

GA	PSO
Number of populations = 20 Cross-over probability = 1 Mutation rate = 0.001 Selection: Stochastic selection Maximum iterations = 100 Reinsertion rate =0.9	Number of particles= 20 $\varphi_{1,2}$=2 *Weight w: max = 0.9 min =0.4* Maximum iterations = 100
Optimization: *Jmin* = 0.0030kW *Time*=274mins.	Optimization: *Jmin* = 0.0025kW *Time*= 312 mins.
FTDNN optimum structure Number of neurons (hidden layer) = 15 Delays = [1 2 4 5 6 12 17 18 25 27]	FTDNN optimum structure Number of neurons (hidden layer) = 20 Delays = [1 3 4 5 6 8 9 13 16 19]

The RMSE is calculated as follows:

$$RMSE = \sqrt[2]{\frac{\sum_{i=1}^{i=n}({P_a}^i - {P_p}^i)^2}{n}} \qquad (8)$$

where P^i_a is the i^{th} actual output power, P^i_p is the i^{th} predicted power by network, and n is number of data points. For further verification and also for comparison of methodology, the structure of the FTDNN is optimized using both a GA and PSO. The parameters of the algorithms are given in Table I.

The two algorithms produced very similar networks at the end of iterations, although the PSO recommends more hidden neurons. The PSO offers outperformance in terms of reaching a better RMSE, J, and a higher convergence speed as compared in Figure 6. The ratios of population updates or objective function evaluations in a given generation of a GA and PSO are 90% (due to the combined effect of crossover and reinsertion) and 100%, respectively, Figure 7. We observed for PSO, a larger downward stepwise reduction in the RMSE at the beginning of the search, Figure 6, which lasted for approximately 50 generations (~generation 15 to 65) while in the case of the GA, less such "stagnation" was observed. Similar observation is noticed when comparing the fall of objective function value with progress in function evaluations, Figure 7; the stagnation of PSO lasted for about 1000 function evaluations although with an initial rapid fall. This observation provides an interesting confirmation that using the current set of parameters, the GA was better at exploring the search in a more global manner, while PSO had a more local focus. The PSO was able to find better solutions in fewer iterations and function evaluations. This could be due to that the structural optimization problem is less multi-modal and due to the difference in the genotype *creation and retention* mechanisms between the two algorithms.

This suggests that PSO's search capability can be further enhanced by borrowing GA diversification techniques: cross-over and mutation to overcome the observed "stagnation". On the other hand, GA's local search can be improved by using concepts of particles or swarms tracking and updating their best local and global best fitness values. A hybrid algorithm with these features can be investigated in future work.

The proposed method is then compared with intuitive persistence models of equations (1) and (2) that are usually used as benchmarks in industry and solar forecast studies [16, 17]. Furthermore, comparison is also made with the results of previous work [12], where the structure was chosen by trial and error with D = [1 2 3 4 5 6 7 8] and number of neurons in the hidden layer being 10. Table II summarizes the test results of different forecasting models where RMSEs are compared and improvement over intuitive model of equation (1) is indicated. Figure 8 compares the best models of Table II while Figure 9 compares the forecasts using PSO-FTDNN and previous work FTDNN where structure was chosen by trial and error; the two plots are separated for ease of visualization. The results presented show an advantage of using the FTDNN in forecasting power yield over existing models. It also shows that the structure of FTDNN influences the accuracy of forecasts and can be optimized. Such tests need further repeated runs and average results can be compared as both heuristic and quasi-netwon based algorithms start with random initializations.

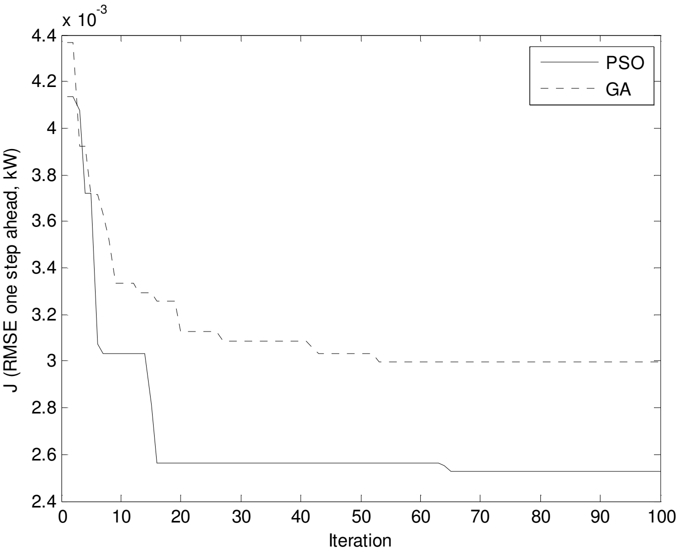


Fig. 6. Progress of structural optimization using PSO and GA: Iterations

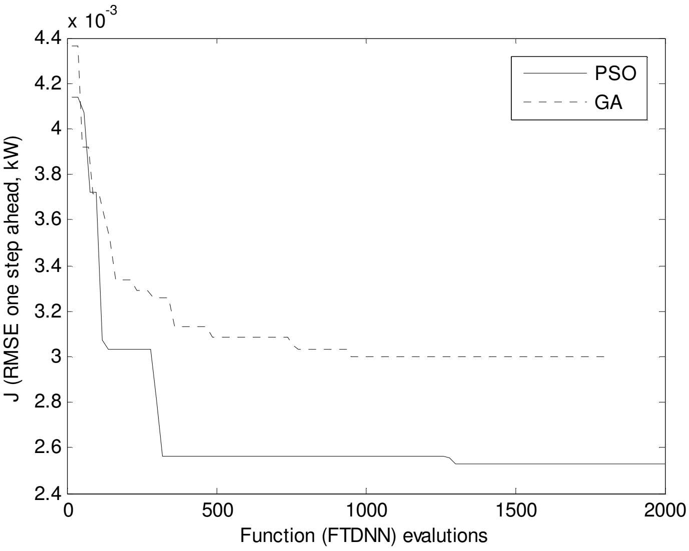


Fig. 7. Progress of structural optimization using PSO and GA: Function evaluations

TABLE II
COMPARISON BETWEEN GA/PSO-FTDNN, FTDNN, AND PERSISTENCE MODELS

Model	GA-FTDNN	PSO-FTDNN	FTDNN	Persist. 1 (eq. 1)	Persist. 2 (eq. 2)
RMSE, kW (% improvement over Persist. 1)	0.003 (+53%)	0.0025 (+61 %)	0.0044 (+31%)	0.0064	0.0264

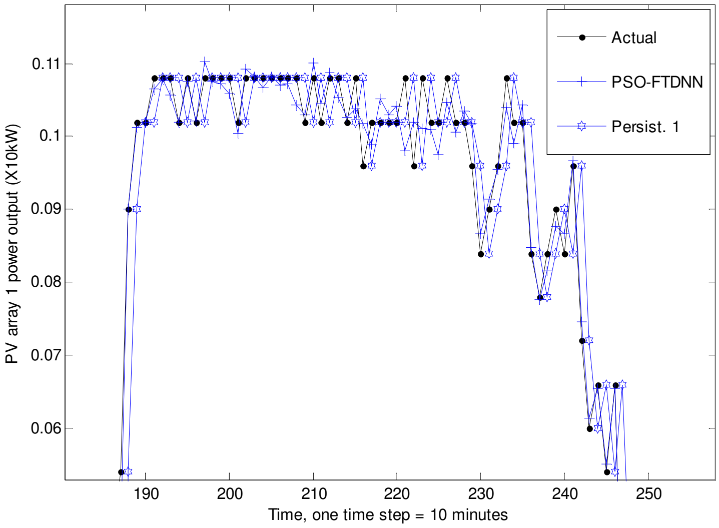


Fig. 8. Comparison between PSO-FTDNN, Persistence model 1, and actual data of 8 February 2011 PV; step ahead forecasts.

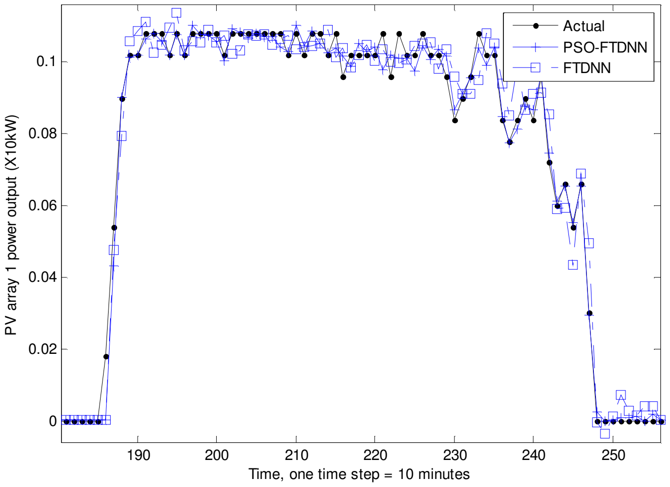


Fig. 9. Comparison between PSO-FTDNN, FTDNN, and actual data of 8 February 2011 PV step ahead forecasts.

.

VI. CONCLUSION

This paper has presented an enhanced artificial network for short-term forecasting of PV power yield. The difficulty of choosing an appropriate network structure has been tackled through a heuristic algorithm that improves the forecasting accuracy. Both PSO and a GA have been tested in this structure-optimization process and both have proven efficient with comparable results. Future work includes using the PSO algorithm alone for both structural and parametric optimization, as it is more efficient in numerical optimization than a GA and is able to tackle structure optimization which BP is unable to.

REFERENCES

[1] Global Market Outlook for photovoltaics 2013-2017, European Photovoltaic Industry Association (EPIA), May 2013.

[2] R. M. Swanson, “Photovoltaic Power Up”, Science, Vol. 324, pp. 891-892, 15 May 2009.

[3] IRENA reference: “Renewable Energy Technologies: Cost Analysis Series, Photovoltaics”, IRENA working paper, Volume 1: Power sector, Issue 4/5, June 2012.

[4] A. Woyte, V. V. Thong, R. Belmans, J. Nijs, “Voltage Fluctuations on Distribution Level Introduced by Photovoltaic Systems,” IEEE Trans. On Energy Conversion, Vol. 21, No. 1, pp. 202- 209, March 2006.

[5] M. Sandoval, S. Grijalva, "An assessment on the impacts of photovoltaic systems on operational costs of the grid? The case of the state of Georgia," 2013 IEEE PES Conference On Innovative Smart Grid Technologies Latin America (ISGT LA), pp.1-6, 15-17 April 2013.

[6] J. Ma, S. Lu, R. P. Hafen, P. V. Etingov, Y. V. Makarov, V. Chadliev, "The impact of solar photovoltaic generation on Balancing Requirements in the Southern Nevada system," Transmission and Distribution Conference and Exposition (T&D), 2012 IEEE PES, pp.1-9, 7-10 May 2012

[7] B. H. Chowdhury, "Optimizing the integration of photovoltaic systems with electric utilities," IEEE Transactions on Energy Conversion, vol.7, no.1, pp.72-78, Mar 1992.

[8] T. Stetz, F. Marten, M. Braun, "Improved Low Voltage Grid-Integration of Photovoltaic Systems in Germany," IEEE Transactions on Sustainable Energy, vol.4, no.2, pp.534-542, April 2013.

[9] L. Cibulka, M. Brown, L. Miller, A. V. Meier, “User Requirements and Research Needs for Renewable Generation Forecasting Tools that will Meet the Need of the CAISO and Utilities for 2020,” A White Paper Report Prepared by CIEE, September 2012.

[10] C. Tao, D. Shanxu, C. Changsong, "Forecasting power output for grid-connected photovoltaic power system without using solar radiation measurement," , 2010 2nd IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG), pp.773-777, 16-18 June 2010.

[11] Z.-G. Che, T.-A. Chiang, Z.-H. Che, “Feed-forward Neural Networks Training: A Comparison between Genetic Algorithm and Back-propagation Learning Algorithm,” International Journal of Innovative Computing Information and Control, Vol. 7, No. 10, pp. 5839-5850, October 2011.

[12] N. Al-Messabi, Yun Li, I. El-Amin, C. Goh, "Forecasting of photovoltaic power yield using dynamic neural networks," The 2012 International Joint Conference on Neural Networks (IJCNN), pp.1-5, 10-15 June 2012

[13] Hagan, M.T., and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, No. 6, 1999, pp. 989-993, 1994.

[14] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice-Hall, 1999.

[15] A. Sefetsos, A. H. Coonick, “Univariate and Multivariate Forecasting of Hourly Solar Radiation with Artificial Intelligence Techniques,” Solar Energy, Vol. 68, No. 2, pp. 169-178, 2000.

[16] Final Report: California Renewable Energy Forecasting, Resource Data and Mapping, Appendix A, Current State of Art In Solar Forecasting. [online] http://uc-ciee.org/downloads/appendixA.pdf.

[17] P. Bacher, H. Madsen, H. A. Nielsen, “Online short-term solar power forecasting,” Solar Energy, Vol. 83, pp. 1772-1783, 2009.

[18] J. A. Gow, C. D. Manning, “Development of a Photovoltaic Array Model for use in Power-Electronics Simulation Studies,” IEE proceedings on Electric Power Applications, Vol. 146, no. 2, pp. 193-200, March 1999.

[19] W. Xiao, M. G. J. Lind, W. G. Dunford, and A. Capel, “Real-time Identification of Optimal Operating Points in Photovoltaic Power Systems,” IEEE Transactions on Industrial Electronics, Vol. 53, No. 4, August 2006.

[20] M. G. Villalva, J. R. Gazoli, E. R. Filho, “Comprehensive Approach to Modeling and Simulation of Photovoltaic Arrays,” IEEE Transactions on Power Electronics, Vol. 24, No. 5, May 2009.

[21] B. Kroposki, K. Emery, D. Myers, and L. Mrig, “A Comparison of Photovoltaic Module Performance Evaluation Methodologies For Energy Ratings,” 24th IEEE Photovoltaic Conference, pp. 858-862, 1994.

[22] C. Craggs, E. M. Conway, N. M. Pearsall, “Statistical Investigation of the Optimal Averaging Time for Solar irradiance on Horizontal and Vertical Surfaces in the UK,” Solar Energy, Vol. 68, No. 2, pp. 179-187, 2000.

[23] B. H. Chowdhury, S. Rahman, "Is central station photovoltaic power dispatchable?," IEEE Transactions on Energy Conversion, Vol.3, No.4, pp.747-754, Dec. 1988.

[24] T. Hiyama and K. Kitabayashi, “Neural Network Based Estimation of Maximum Power Generation from PV Module using Environmental Information,” IEEE Trans. On Energy Conversion, Vol. 12, No. 3, 1997.

[25] A. Mellit, A. M. Pavan, “A 24-h Forecast of Solar Irradiance using Artificial Neural Network: Application for Performance Prediction of a Grid-connected PV plant at Trieste, Italy,” Solar Energy, Vol. 84, pp. 807-821, 2010.

[26] Caputo, D.; Grimaccia, F.; Mussetta, M.; Zich, R.E., "Photovoltaic plants predictive model by means of ANN trained by a hybrid evolutionary algorithm," The 2010 International Joint Conference on Neural Networks (IJCNN), pp.1-6, 18-23 July 2010.

[27] A. Yona, T. Senjyu, A. Y. Saber, T. Funabashi, H. Sekine, K. Chul-Hwan, "Application of Neural Network to One-Day-Ahead 24 hours Generating Power Forecasting for Photovoltaic System," International Conference on Intelligent Systems Applications to Power Systems , ISAP 2007, pp.1-6, 5-8 Nov. 2007.

[28] L. A. Fernandez-Jimenez, A. Munoz-Jimenez, A. Flaces, M, Mendoz-Villena, E. Garcia-Garrido, P. Lara-Santillan, E. Zorzano-Alba, J. Zorzano-Santamaria, “Short-term Power Forecasting System for Photovoltaic plants,” Renewable Energy, Vol. 44, pp. 311-317, 2012.

[29] M. Rawat, K. Rawat, F. Ghannouchi, “Adaptive Digital Predistortion of Wireless Power Amplifiers/Transmitters using Dynamic Real-Valued Focused Time-Delay Line Neural Networks,” IEEE Transactions on Microwave Theory and Techniques, vol. 58, No. 1, Jan. 2010.

[30] K. Y. Lee and F. F. Yang, “Optimal reactive power planning using evolutionary algorithms: A comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming,” IEEE Trans. on PWRS, vol. 13, no. 1, pp. 101–108, 1998.

[31] R. Eberhat and Y. Shi, Particle Swarm Optimization: Developments, applications, and resources, Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, 2001, pp. 81-86.