International Journal of Cognitive Informatics and Natural Intelligence
Volume 15 • Issue 2 • April-June 2021

Short-Term Wind Power Prediction Using Hybrid Auto Regressive Integrated Moving Average Model and Dynamic Particle Swarm Optimization

Pavan Kumar Singh, MNNIT, Allahabad, India
Nitin Singh, MNNIT, Allahabad, India
Richa Negi, MNNIT, Allahabad, India

ABSTRACT

With the upsurge in restructuring of the power markets, wind power has become one of the key factors in power generation in the smart grids and gained momentum in the recent years. The accurate wind power forecasting is highly desirable for reduction of the reserve capability, enhancement in penetration of the wind power, stability and economic operation of the power system. The time series models are extensively used for the wind power forecasting. The model estimation in the ARIMA model is usually accomplished by maximizing the log likelihood function and it requires to be re-estimated for any change in input value. This degrades the performance of the ARIMA model. In the proposed work, the model estimation of the ARIMA model is done using latest evolutionary algorithm (i.e., dynamic particle swarm optimization [DPSO]). The use of DPSO algorithm eliminates the need for re-estimation of the model coefficients for any change in input value and moreover, it improves the performance of ARIMA model. The performance of proposed DPSO-ARIMA model has been compared to the existing models.

KEYWORDS

ARIMA, Box-Jenkins Method, Dynamic Particle Swarm Optimization, Evolutionary Computing, Linear Time Series Model, Smart Grids, Swarm Intelligence, Wind Power Prediction

INTRODUCTION

The growth and utilization of renewable energy (RE) can make considerable contribution to the energy, environmental and economic policy especially in three interrelated areas. These areas are: (i) energy security (ii) decrease of CO_2 emission and other ecological impact of energy use, and (iii) economic growth (Chang, 2014). Many countries have increased the emphasis on the development of clean & green energy in order to counter the climatic changes and entrust energy security. The developing trend of the world energy is the provision of clean energy with low carbon footprints, and high energy efficiency. As the basis and premise of the low-carbon electricity, smart grid technology has rapidly developed in many countries in the recent years. Due to its high efficiency and no pollution, the wind power has emerged as one of the highly attractive technologies of renewable energy in smart grids (X. Zhao, Wang, & Li, 2011).

DOI: 10.4018/IJCINI.20210401.oa9

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Wind power generation plays an important role in the smart grid environment due to its availability in abundance and low carbon footprint. Wind power generation is dependent on the wind speed which is influenced by the geographical and environmental conditions of any particular region. It also varies with height, so the random character of wind is significant. As compared to the conventional generation, one of the major problems of wind power is its reliance on the unpredictability of the wind. The unpredicted variations of wind generation may enhance operating costs, which increases reserves necessities, and poses potential risks to reliability of the power system (Lei, Shiyan, Chuanwen, Hongling, & Yan, 2009; Z. Li et al., 2016) The intermittency of wind power is the major challenge in implementing the wind-energy as a reliable independent source of electric power supply.

Large-scale wind-penetration requires answers to a lot of problems such as competitive market designs, real-time grid operations, standards of interconnection, ancillary service requirements and costs, quality of power, capacity of transmission system and its future upgrades, stability and reliability of power system, optimal reductions in greenhouse gas emissions of entire power system (typically determined by optimal amount of wind penetration into system). The generation load balance is a well-known method for power balance objective in the integrated power systems, the mismatch in demand and generation may affect the stability of the power system, so, in order to overcome the problems of large wind integration and to match the power demand in order to maximize the profits the generating companies rely on the accurate wind power forecasting. When it comes to competitive electricity markets, accurate wind power forecast is always alluring for a variety of reasons. Firstly, appropriate incentives of attractive market price are offered on energy imbalance charges based on market price. Secondly, a correct forecast can help to develop well-functioning hour ahead or day-ahead markets (Madhiarasan & Deepa, 2016; Soman, Zareipour, Malik, & Mandal, 2010).

The wind forecasting can be widely classified on the basis of numerous aspects as shown in Figure 1. Based on time horizon (timescale), the wind forecasting is classified (Mao & Shaoshuai, 2016) with their applications as given in Table 1 (Chang, 2014; Soman et al., 2010).

For short term wind power forecasting, several approaches are well studied and implemented. (Aasim, Singh, & Mohapatra, 2019) proposed a novel hybrid repeated wavelet transform based ARIMA model which utilized the previous wind speed data to model wind speed for very short-term wind speed forecasting. A comparison of the proposed RWT-ARIMA model with the benchmark persistence model for very short-term wind speed forecasting, ARIMA model and WT-ARIMA model has been done for various time-scales of forecasting such as 1min, 3min, 5min, 7min and 10min. This comparison proves the superiority of the proposed RWT-ARIMA model over other models in very short-term wind speed forecasting. (Samson & Kainkwa, 2019) used Auto-Regression Integrated Moving Average (ARIMA) model to predict the wind power density for electricity generation at Makambako, Tanzania. (Du, Wang, Yang, & Niu, 2019) used a hybrid model based on complete ensemble empirical mode decomposition and an improved wavelet neural network to enhance prediction accuracy. The four 10-min wind power datasets collected from the Sotavento wind farm in Galicia, Spain are used to perform the multi-step wind power prediction. The proposed hybrid model possesses the lowest prediction error when compared with hybrid MOMFO-WNN with EMD, EEMD, VMD and CEEMD, and demonstrates that it is suitable for single-step and multi-step ahead wind power prediction.

(Handoyo, Efendi, Jie, & Widodo, 2017) implemented PSO for solving the maximum likelihood estimation (MLE) optimization problem for estimating the parameters of ARMA model, evaluated from a variety of inertia weights and population sizes and found that the PSO algorithm is efficient and has faster convergence rate. (Asadi, Tavakoli, & Hejazi, 2012) applied PSO to estimate the parameters of ARIMA model and predict the future of two different time series data and found it an effective method to improve forecasting accuracy. (Hadavandi, Ghanbari, & Abbasian-Naghneh, 2010) used PSO algorithm for parameter estimation of ARIMA for the gold price forecasting and found its ability to cope with the fluctuations of gold price time series, yielding good prediction accuracy in terms of MAE, outperforming other models. (H. Wang & Zhao, 2009) used PSO for parameter

Figure 1. Classification of wind forecasting based on different aspects



estimation of ARIMA model and observed more accurate (lower relative error) than the traditional moment estimation-based model through the experimental results of forecasting the consumer price index. (Chao-Ming Huang, Chi-Jen Huang, & Ming-Li Wang, 2005) applied PSO approach to identify ARMAX model for day ahead to week ahead hourly load forecasts and found that PSO has high-quality solution, superior convergence characteristics, and shorter computation time as compared to the other existing methods.

In a conventional statistical approach, wind power output behaviour is modeled as a time series. Autoregressive (AR), Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) models have been widely used for wind energy forecasting. The Artificial Neural Network (ANN) is also a widely used tool for short term wind power forecasting. However, the ANN based approach has very slow convergence during the training phase. On the other hand, statistical regressive models are computationally efficient. In the statistical approaches, the forecasting accuracy is highly dependent on the estimated model of the wind power output behaviour. Therefore, it is important to identify the estimated model parameters accurately. Different methods are widely used to estimate the model parameters, such as, ordinary Least Squares (LS) approach, Forward

Table 1. Timescale classification for wind forecasting

Timescale	Range	Applications
Ultra Short Term	Few Minutes to 1 hour ahead	• Electricity market clearing • Real-time grid operations • Regulation actions
Short Term	1 hour to several hours ahead	• Economic load dispatch planning • Load reasonable decisions • Operational security in electricity market
Medium Term	Several hours to 1 week ahead	• Unit commitment decisions • Reserve requirement decisions • Generator online/offline decisions
Long Term	1 week to 1 year of more ahead	• Operation Management • Optimal operating cost • Feasibility study for design of the wind farm • Maintenance scheduling to obtain optimal operating cost

Backward (FB) approach, Geometric Lattice (GL) approach and Yule-Walker (YW) approach, etc. As the wind power output has variable characteristics, the error function obtained from the estimated model may suffer from the problem of local minima. The time series modelling generally consists of four distinct stages: 1) identification of the model, i.e., identifying the autoregressive and moving average terms 2) estimating the coefficients of autoregressive and moving average terms, 3) validating the selected model and 4) forecasting (Ong, Huang, & Tzeng, 2005). The traditional model estimation of the ARIMA model is a complex process even if the simpler methods like moment estimation are considered. The accuracy of the ARIMA model is affected by the unpredictable results given by traditional model estimation methods. Moreover, the model coefficients need to be re-estimated for any change in the input values (Handoyo et al., 2017).

In the proposed work, the traditional moment estimation stage of ARIMA model is replaced with the DPSO algorithm. The main contribution of this paper is the combination of the techniques, i.e., ARIMA and dynamic particle swarm optimization (DPSO) for forecasting the wind power. Generally, in the evaluation of population based iterative optimization algorithms, convergence speed and local optima avoidance are being used as two main objectives. Since these objectives are contradictory to each other, it is difficult to attain both objectives simultaneously. Dynamic-PSO (DPSO) avoids problems of stagnation and getting trapped in local optima, without losing the fast convergence characteristic of PSO. Dynamic particle swarm optimization (DPSO) is a variant of particle swarm optimization (PSO) (Saxena et al., 2015), it meets the two prime objectives of the population based algorithm, i.e., the speed of the convergence and the mechanism for avoiding the local minima, which is difficult to achieve because of contradiction in both the objectives. The basic PSO (Eberhart & Kennedy, 1995) updates the velocity of the particles by using the global best which also influences the position of the particles and leads to faster convergence, due to which it becomes vulnerable to the problem of local optima especially in the case of multimodal problems (Jang-Ho Seo et al., 2006) (Liang, Qin, Suganthan, & Baskar, 2006). The dynamic PSO (DPSO) variant overcomes the problem of stagnation and local optima and at the same time maintains the fast convergence rate.

The DPSO focuses on identifying those particles which are unable to improve their personal best in predefined number of successive iteration. This indicates that particles are saturated and require external thrust to boost their power. DPSO provides thrust by heading particles towards potentially better unexplored regions which also add diversity too. At the same time when global best is not improving for predefined number of successive iteration due to get trapped in local optima, may mislead other particles by attracting towards it. This also requires some external push that send

trapped particle outside local optima position and mitigate its consequences. For rest of the situations DPSO follows the behavior of PSO and do not upset its fast convergence property (Saxena, Tripathi, Mishra, & Misra, 2015).

This paper has been organized into various sections which are as follows, the Box-Jenkins methodology for identifying and formulating the family of ARIMA model is explained in section 2. Section 3 introduces the ARIMA modeling procedure and section 4 describes the dynamic particle swarm optimization (DPSO) algorithm. Section 5 explains the proposed hybrid DPSO -ARIMA model. The performance evaluation parameters and results obtained using the proposed methodology has been discussed in section 6 and 7 respectively. Section 8 finally concludes the paper and presents the conclusion and future scope of the presented work.

BOX-JENKINS METHODOLOGY

The different phases in Box-Jenkins methodology are shown (Junior, Salomon, & Pamplona, 2014) in Figure 2. Generally, historical data (time series) based classical methods such as linear trend, exponential adjustment, moving average and nonlinear trend are used. For these methods to be applicable to the time series, the time series need to be stationary in nature, i.e. its mean as well as covariance between the periods of time series need to be constant. If the given time series is stationary then the, autoregressive (AR), moving average (MA) and autoregressive moving average (ARMA) methods are mostly used as they produce better results by increasing the forecasting accuracy. In case when the time series in non-stationary then auto-regressive, moving average and auto regressive moving average will not produce better results, in such cases the auto regressive integrated moving average (ARIMA) is used as the underlying non-stationary series is made stationary by differentiating it one or two times.

Autoregressive (AR) Model

The autoregressive (AR) model is a popular method for time series modelling. The least-squares technique is utilised in AR model estimation. In time-varying AR model estimation, autoregressive parameters are thought of as time-invariant for the appliance of least squares method. The AR model is generally used for analyzing the univariate time-series, which is extensively employed for estimation as well as forecasting. The output variable has linear relation with its own preceding data in an autoregressive model. Its p-order can be expressed as (1) (J. Wang, Liang, Che, & Sun, 2008), for a sample period of $t = [1, 2, ..., t]^{T}$.

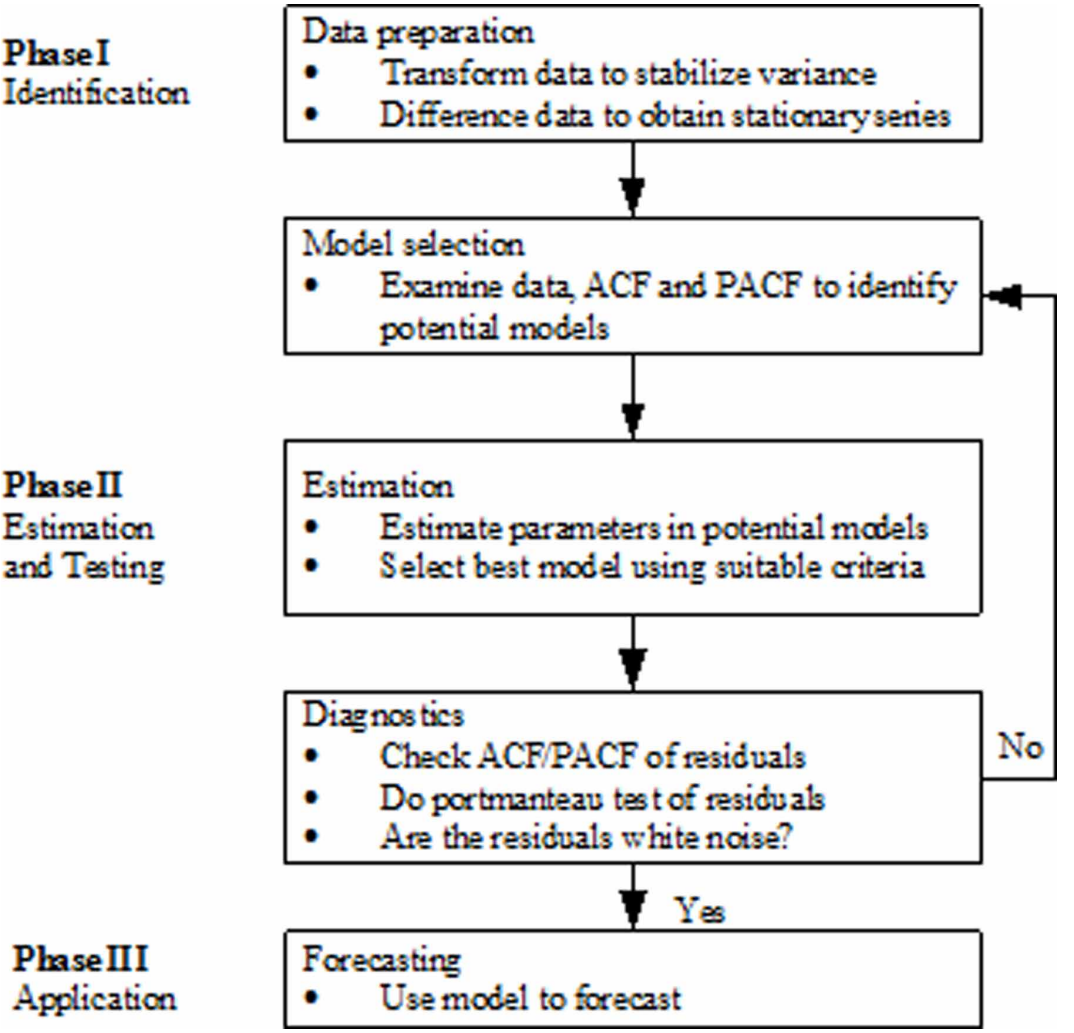
$$X_t = C + \sum_{i=1}^{p} \varnothing_i X_{t-i} + \mu_t \tag{1}$$

where, $\phi_1, \phi_2, \ldots, \phi_p$ and C denote the lag parameters and constant term of the model respectively; and ε_t being white Gaussian noise having mean as zero. Akaike's information criteria is used to choose the least appropriate lag order p, the least AIC value gives the best fitted model.

For a stationary time series having mean as zero $\left\{X_t,\ 0, \pm 1, \pm 2, \ldots\ldots\right\}$, the general form of AR model is given in (2) (J. Wang et al., 2008):

$$\varnothing\left(B\right) X_t = \mu_t \tag{2}$$

Figure 2. Box-Jenkins methodology



where $\varnothing\left(\mathrm{B}\right) = 1 - \varnothing_1\mathrm{B} - \varnothing_2\mathrm{B}^2 - \ldots\ldots\ldots - \varnothing_\mathrm{p}\mathrm{B}^\mathrm{p}$; B and $\varnothing_\mathrm{i}\left(\mathrm{i} = 1,2, \ldots\ldots\ldots,\mathrm{p}\right)$ denote the backward shift operator and p order of AR model parameters respectively; and μ_t is stationary white noise having zero mean. The mathematical expression for an AR model can also be given as (3) (J. Wang et al., 2008).

$$\mathrm{X}_\mathrm{t} = \varnothing_1\mathrm{X}_{\mathrm{t}-1} - \varnothing_2\mathrm{X}_{\mathrm{t}-2} - \ldots\ldots\ldots - \varnothing_\mathrm{p}\mathrm{X}_{\mathrm{t}-\mathrm{p}} - \mu_\mathrm{t} \tag{3}$$

Moving Average (MA) Model

For a stationary time series having mean as zero $\left\{\mathrm{X}_\mathrm{t},\ 0, \pm 1, \pm 2, \ldots\ldots\right\}$, the general form of MA model is given in (4) (J. Wang et al., 2008):

$$X_t = \theta(B)\mu_t \tag{4}$$

Where, $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots\ldots\ldots - \theta_q B^q$. B and $\theta_i (i = 1,2, \ldots\ldots\ldots, q)$ denote the backward shift operator and q order of MA parameters of the model respectively; μ_t stationary white Gaussian noise having mean as zero. Also, the MA model can be given as (5) (J. Wang et al., 2008).

$$\mathbf{X_t} = \boldsymbol{\mu_t} - \theta_1 \mathbf{X_{t-1}} - \theta_2 \mathbf{X_{t-2}} - \ldots\ldots\ldots - \theta_q \mathbf{X_{t-q}} \tag{5}$$

Autoregressive Moving Average (ARMA) Model

For a stationary time series having mean as zero $\{X_t, 0, \pm 1, \pm 2, \ldots\ldots\}$, the general form of ARMA model is given in (6) (J. Wang et al., 2008):

$$\varnothing(B) X_t = \theta(B)\mu_t \tag{6}$$

where $\varnothing(B) = 1 - \varnothing_1 B - \varnothing_2 B^2 - \ldots\ldots\ldots - \varnothing_p B^p$ and $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots\ldots\ldots - \theta_q B^q$. The ARMA model orders are p, q. The ARMA model parameters are $\varnothing_i (i = 1,2, \ldots\ldots\ldots, p)$ and $\theta_i (i = 1,2, \ldots\ldots\ldots, q)$. If q is equal to 0 in an ARMA (p,q) model, it becomes AR(p) model and MA(q) model when p equals to 0. Also, an ARMA model can be expressed as (7) (J. Wang et al., 2008):

$$X_t - \varnothing_1 X_{t-1} - \varnothing_2 X_{t-2} - \ldots - \varnothing_p X_{t-p} = \mu_t - \theta_1 X_{t-1} - \theta_2 X_{t-2} - \ldots - \theta_q X_{t-q} \tag{7}$$

Because of ARMA models being merely appropriate for stationary time series, the data set should be appropriately differentiated, in case the time series is non-stationary or has a unit root.

AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA) MODEL

ARIMA processes are a class of stochastic process used for evaluating the time series and has become one of the most popular approaches for forecasting. The application of time series models, i.e., ARIMA is made possible by the Box and Jenkins. The ARIMA model is formed by combining the autoregressive and moving average models proposed by Yule and Walker in the year 1927 and 1931 respectively. Unlike the autoregressive and moving average model the limitation of forecasting only stationary price series is not there with the ARIMA model. Any non-stationary time series which is given as input to the ARIMA model is made stationary series by doing the difference operation. The differenced stationary series is then predicted using the ARMA process.

The main requirement for ARIMA model is either the input series should be stationary or it should become stationary after performing one or more difference operations (Erdogdu, 2007). The model is represented as ARIMA (p, d, q) where p, d, and q represents the autoregressive lags, order of differencing and moving average lags respectively (Eldali, Hansen, Suryanarayanan, & Chong, 2016). The time series modelling generally consists of four distinct stages: 1) identification of the model, i.e., identifying the autoregressive and moving average terms 2) estimating the coefficients of

autoregressive and moving average terms, 3) validating the selected model and 4) forecasting (Ong et al., 2005). The various stages involved in modeling the ARIMA process are given in Figure 3.

An ARIMA (p,d,q) model is expressed in (8) (J. Wang et al., 2008), where the order of differencing is represented by symbol d. For a non-stationary time series, an order of differencing d is applied to generate a more stable variance.

$$\varnothing\left(\mathrm{B}\right)\left(1-\mathrm{B}\right)^{\mathrm{d}}\mathrm{X}_{\mathrm{t}} = \text{,}\left(\mathrm{B}\right)\mu_{\mathrm{t}} \tag{8}$$

where μ_{t} is independently and identically distributed white noise having zero mean as well as constant variance, d denotes degree of differencing required for making sure the stationarity of time series, B represents backward operator, generally defined as $\mathrm{B}^{\mathrm{k}}\mathrm{X}_{\mathrm{t}} = \mathrm{X}_{\mathrm{t-k}}$, $\varnothing$ and , denote the polynomials of B.

Model Identification

The model identification stage of the ARIMA (p, d, q) model determines the order of p, d, and q. i.e., number of autoregressive terms, order of differencing and moving average terms respectively. The order is mainly determined by analyzing the plots of autocorrelation function (ACF) and partial autocorrelation function (PACF) plotted against lag length (Erdogdu, 2007). The ACF and PACF at lag j, denoted as ρ_{j} and α_{jj} respectively, are shown in (9) and (10) (H. Wang & Zhao, 2009).

$$\hat{\rho}_{j} = \frac{\hat{\gamma}_{j}}{\hat{\gamma}_{0}} \tag{9}$$

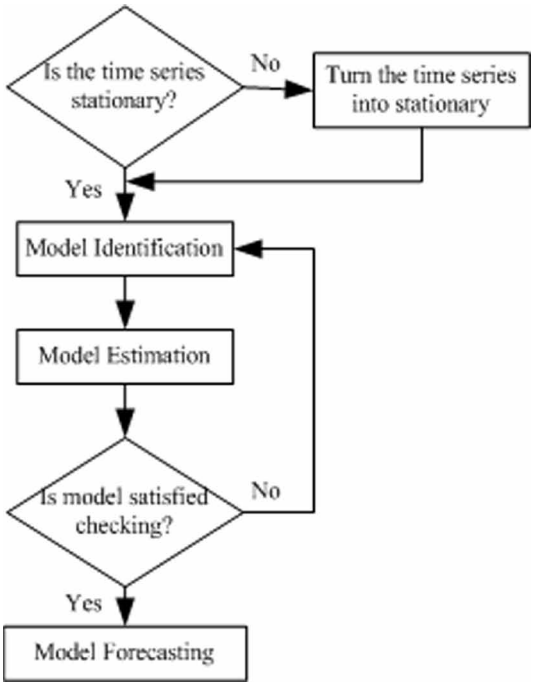
$$\left.\begin{array}{c} \hat{\alpha}_{11} = \hat{\rho}_{1} \\ \hat{\alpha}_{j+1,j+1} = \dfrac{\hat{\rho}_{j+1} - \sum_{k=1}^{j}\hat{\rho}_{j+1-k}\hat{\alpha}_{jk}}{1 - \sum_{k=1}^{j}\hat{\rho}_{k}\hat{\alpha}_{jk}} \\ \hat{\alpha}_{j+1,k} = \hat{\alpha}_{jk} - \hat{\alpha}_{j+1,j+1}.\hat{\alpha}_{j,j-k+1} \\ (k = 1,2,3...j) \end{array}\right\} \tag{10}$$

where, j = 1, 2,.....M. The differencing term that will make the underlying process stationary is determined by the analysing the coefficients of ACF. In order to achieve more stable variance the input price series is log transformed. The determination of parameter d converts the series into stationary series with zero mean.

Thus the ARIMA (p, d, q) process is reduced to ARMA (p, q) process after the determination of differencing terms. The ARMA (p, q) model in the generalized form can be expressed as (11) (H. Wang & Zhao, 2009).

$$\varphi(B)x_{t} = \theta(B)a_{t} \tag{11}$$

Figure 3. The stages of Building ARIMA model



where $\begin{cases}\varphi(B)=1-\varphi_1 B-\varphi_2 B^2-.....-\varphi_p B^p \\ \theta(B)=1-\theta_1 B-\theta_2 B^2-.....-\theta_p B^p\end{cases}$ where a_t is white noise with zero mean and $\varphi_i(i=1,2,....p)$ & $\theta_i(i=1,2,....p)$ are the coefficients of autoregressive and moving average part respectively. The model of ARMA model, i.e., the number of autoregressive terms and moving average terms are determined based on the autocorrelation and partial autocorrelation graphs. The significant coefficients of the partial auto correlation plot identify the autoregressive terms and number of moving average terms is identified based on the significant coefficients of autocorrelation graph.

Model Estimation

The model identification stage only determines the structure and order of the ARIMA (p, d, q) model, i.e., the order of the autoregressive (p) and moving average (q) terms. The coefficients of the autoregressive and moving average terms, i.e., $\varphi_i(i=1,2,....p)$ and $\theta_i(i=1,2,....q)$ respectively are estimated in the model estimation stage. Usually the moment estimation method is adopted to calculate the coefficients of the autoregressive and moving average terms. The Yule-Walker equation shown in (12) (H. Wang & Zhao, 2009) is generally used to estimate the $\{\varphi_i\}$.

$$\begin{bmatrix}\hat{\varphi}_1\\ \hat{\varphi}_2\\ \hat{\varphi}_3\\ \vdots\\ \hat{\varphi}_p\end{bmatrix}=\begin{bmatrix}\hat{\gamma}_q & \hat{\gamma}_{q-1} & \hat{\gamma}_{q-2} & \cdots & \hat{\gamma}_{q-p+1}\\ \hat{\gamma}_{q+1} & \hat{\gamma}_q & \hat{\gamma}_{q-1} & \cdots & \hat{\gamma}_{q-p+2}\\ \hat{\gamma}_{q+2} & \hat{\gamma}_{q+1} & \hat{\gamma}_q & \cdots & \hat{\gamma}_{q-p+3}\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ \hat{\gamma}_{q+p-1} & \hat{\gamma}_{q+p-2} & \hat{\gamma}_{q+p-3} & \cdots & \hat{\gamma}_q\end{bmatrix}^{-1}\begin{bmatrix}\hat{\gamma}_{q+1}\\ \hat{\gamma}_{q+2}\\ \hat{\gamma}_{q+3}\\ \vdots\\ \hat{\gamma}_{q+p}\end{bmatrix} \tag{12}$$

where $\hat{\gamma}_k(\overline{x}) = \sum_{j=0}^{p}\sum_{l=0}^{p}\hat{\varphi}_j\hat{\varphi}_l\hat{\gamma}_{k+l-j}, k = 0,1,....q$.

Then the parameters of the moving average terms i.e., $\{\theta_i\}$ and σ_ε^2 which is the variance of $\{\varepsilon_k\}$ can be estimated using (13) (H. Wang & Zhao, 2009).

$$\left.\begin{aligned}&\hat{\gamma}_0(\overline{x}) = \hat{\sigma}_\varepsilon^2(1+\hat{\theta}_1^2+.....+\hat{\theta}_q^2\\&\hat{\gamma}_k(\overline{x}) = \hat{\sigma}_\varepsilon^2(-\hat{\theta}_k+\hat{\theta}_{k+1}\hat{\theta}_1+.....+\hat{\theta}_q\hat{\theta}_{q-k})k=1,2,....,q\end{aligned}\right\} \tag{13}$$

where $\hat{\gamma}_k(\overline{x}) = \sum_{j=0}^{p}\sum_{l=0}^{p}\hat{\varphi}_j\hat{\varphi}_l\hat{\gamma}_{k+l-j}, k = 0,1,....q$.

After looking at the (12) and (13), it becomes clear that these equations can be solved easily for time series where the value of autoregressive and moving average terms is near to 1 but in most of the practical conditions the value of p and q becomes more than 1, in which case, the solution to (12) and (13) becomes difficult. This impacts the parameter estimation and eventually impacts the forecasting accuracy. It is proposed to replace the conventional model estimation stage by the evolutionary programming techniques to see the impact on the model estimation and forecasting accuracy of the ARIMA model.

Model Validation and Forecasting

After estimating the coefficients of the identified model the validation of the model is done in order to check and validate the assumptions taken in the model selection stage. The analysis of the residuals is done in this step to check whether the hypothesis of white noise assumed on the residuals holds true or not. After completing the stages of model selection, identification, estimation and validation the finalized model is used to predict the wind power (i.e., daily, 24 hour ahead or weekly, 168 hours ahead), the large forecasting horizon decreases the certainty of the forecasts.

In this paper, a well-known technique, i.e., ADF unit root tests, has been used for testing the stationarity of the time series under consideration, i.e, wind power. After the acceptance of underlying process as being stationary, the configuration of model is determined on the basis of ACF and PACF plots.

In the proposed method, AIC (Akaike's Information Criterion), as defined in (14) (Eldali et al., 2016), is used to determine p, q orders of the model. This is a metric to estimate the precision of the selected model and attains a substitution between fit and intricacy. Generally, the optimal model is determined with the minimum value of AIC.

$$\mathrm{AIC} = 2\mathrm{P} - 2\ \log\ \mathrm{L} \tag{14}$$

where P and $\log\ \mathrm{L}$ denote the number of model parameters and the maximum log-likelihood respectively. The p,d,q grouping of model order values gives the most appropriate model order leading to minimum AIC.

DYNAMIC PARTICLE SWARM OPTIMIZATION (DPSO) ALGORITHM

Particle Swarm Optimization (PSO) is one of the major paradigms of the computational swarm intelligence, which converges to the global optimal solution of a complex error surface and finds better solution compared with gradient search based stochastic time series techniques (Anwar & Mahmood, 2014). PSO is a heuristic method with advantages such as easy to implement, robust, reliable, and effective (Parsopoulos & Vrahatis, 2010). The particle swarm optimization algorithm

is utilized to determine the solution of an optimization problem and motivated from the conduction characteristics of the biological population (Patwal, Narang, & Garg, 2018). In the particle swarm optimization algorithm, every feasible result of an optimization problem can be considered as a point in n-dimensional search space, known as a particle. Each of the particles has a fitness value, which is found out by objective function. Every particle has a speed to make a decision for its flying direction and distance to search the solutions in the space after the present best particle (Garg, 2016). The fitness value of each individual is estimated by the optimization function, whose benefit as well as drawbacks can be determined by its fitness value in the search space. The result of the optimization problem is revealed by the position as well as velocity of each particle (Garg, 2019). The velocity of each particle can vary its position also. For comparing the performance of each particle, the fitness value of objective function is calculated (H. Zhao & Feng, 2014). The particle swarm optimization is also a meta-heuristic iterative algorithm like other computational evolutionary algorithms (genetic algorithm, differential evolution etc) (Garg, 2015). Generally, these algorithms need huge computational capability that inspires the researchers to build proficient optimization techniques (Saxena et al., 2015; H. Wang & Zhao, 2009b).

Generally in the evaluation of population based iterative optimization algorithms, convergence speed and local optima avoidance are being used as two main objectives. Since these objectives are contradictory to each other, it is difficult to attain both objectives simultaneously. As basic PSO uses the contribution of global best for updating velocity of particles, in turn influence their position too, lead to speedy convergence. But at the same time it is very much susceptible of getting trapped in local optima, specifically in complex multimodal problems, which results in premature convergence. Several PSO variants have been developed aiming to improve the performance of PSO by setting up balance between these two parameters. Majority of these variants maintain diversity by compromising convergence rate significantly.

The new variant of particle swarm optimization was proposed (Saxena et al., 2015) which overcomes the two prime objectives of the population based algorithm, i.e., the speed of the convergence and the method for avoiding the local minima, which is difficult to achieve because of contradiction in both the objectives. The basic PSO (Eberhart & Kennedy, 1995) updates the velocity of the particles by using the global best which also influences the position of the particles and leads to faster convergence, due to which it becomes vulnerable to the problem of local optima specially in the case of multimodal problems (Jang-Ho Seo et al., 2006; Liang, Qin, Suganthan, & Baskar, 2006). Various variants have already been proposed by the researchers to improve the performance of the PSO and provide solution to this problem by balancing the two parameters (Saxena et al., 2015; Shi & Eberhart, 1998).

The dynamic PSO (DPSO) variant overcomes the problem of stagnation and local optima and at the same time maintains the fast convergence rate. DPSO does it by keeping the track of the change in the personal best positions of the particles, the algorithm tracks time for which the position pBest is not updated and at the same time it also keeps a track of the iteration for which the gBest had not changed. If the pBest and gBest doesn't change for a predefined number of iterations then they are replaced by the best positions that they have attained previously.

The pBest and gBest are again monitored to see whether they are able to improve the targets or not if they improve the targets then the replacement is made permanent otherwise the old values are restored. This process is repeated until the optimal value is achieved or termination condition is reached. The terminology used in the DPSO algorithm is shown in Table 2.

The Dynamic-PSO algorithm (DPSO) procedure involves mainly three steps, i.e., (i) the identification and preservation of the potential region which are possibly unexplored, (ii) Restructuring pBest and gBest of the stagnated particles for whom the value of pBest and gBest has not change for a long time and (iii) monitoring the acceptability of the changes made in the values of pBest and gBest. The various steps involved in implementing the dynamic PSO algorithm for the given ARIMA model are given below (Saxena et al., 2015).

Table 2. Terminology used in Dynamic PSO algorithm

Symbol	Description
m=p+q	No. of particles in Swarm (population size)
X_i	Position vector of the i^{th} particle, i=1 to m
V_i	Velocity vector of the i^{th} particle, i=1 to m
$pBest_i$	Best position obtained by i^{th} particle
gBest	Global best position of the swarm
$pBest_count_i$	Counter when $pBest_i$ do not improve in successive generation for particle i
pBest_iter_thresh	Maximum number of iteration i^{th} particle wait for improvement in $pBest_i$
$pBest_temp_i$	Provisionally restructured vector of $pBest_i$ for particle I, i=1 to m
$pBest_chance_count_i$	Counter when $pBest_temp_i$ do not improve in successive generation for particle i
pBest_chance_max	Maximu number of iterations $pBest_temp_i$ will get chance to improve $pBest_i$ (constant)
gBest_h	Vector for historical values of gBest
gBest_h_maxcount	Maximum permissible number of elements in gbest_h (constant)
gBest_count	Counter when gBest do not improve in successive generation for particle i
gBest_iter_thresh	Maximum number of iterations group wait for improvement in gBest (constant)
gBest_temp	Provisionally restructured vector of $pBest_i$ for particle i=1 to m
gBest_chance_count	Counter when gBest_temp do not improve in gBest_chance_max successive iterations
gBest_chance_max	Maximum number of iterations gBest_temp will get chance to improve gBest (constant)

- Initialization
- Calculating the values of pBest, Fitness and gBest
- Updating the velocity and position of the particle
- Identification and preserving the regions with potential solutions
- Updating pBest and gBest
- Controlling the stagnation problem
- Restructuring pBest and gBest
- Checking acceptability of restructure pBest and gBest
- Checking for termination condition

The DPSO procedure is explained though flowchart shown in Figure 4 (Handoyo et al., 2017; Sankardoss & Geethanjali, 2017) and the pseudo code of the DPSO algorithm is given in Figure 5. Maximum number of gBest stored as history in gBest_h = 10*m, pBest_iter_thresh = gBest_iter_thresh = 5*m (dimension).

Step 1: Initialization

The velocity and position of the individual particle, i.e., $v_1^0, v_2^0, v_3^0, ..., v_m^0$ and $x_1^0, x_2^0, x_3^0, ..., x_m^0$ respectively is initialized randomly within search space where m = p+q, where p is the order of the autoregressive model and q is the order of moving average part of ARIMA model. The coefficients $\{\varphi_i\}$ and $\{\theta_j\}$ have p and q components respectively, i.e., the individual particle in the swarm will

be denoted as $\{\varphi,\theta\}$, the series is represented as $\hat{x}_t^{(\varphi,\theta)}$ where $\{\varphi,\theta\}$ are the coefficients of AR and MA terms respectively.

Counter for $pBest_count_i$ and $pBest_chance_count_i$ (for i = 1 to m) are initialized to 0 and -1 respectively, these counts keep a track of the successive generations for which $pBest_i$ and $pBest_temp_i$ has not improved. Similarly gBest_count and gBest_chance_count are set to 0 and -1 respectively, these counts keep a track of the number of generations for which gBest and gBest_temp has remain unchanged.

Step 2: pBest, gBest and Fitness Calculation

The mean square error (MSE) is taken as the objective function for calculating the optimal coefficients of the DPSO-ARIMA model, the mean square error (MSE) is calculated using (15) (H. Wang & Zhao, 2009).

$$MSE^{(\varphi,\theta)} = \frac{1}{N-d}\sum_{t=1}^{N-d}(x_t - \hat{x}_t^{(\varphi,\theta)})^2 \tag{15}$$

where N, is the number of data, x_t is the actual value of wind power and $\hat{x}_t^{(\varphi,\theta)}$, is the predicted value using (φ,θ) optimized value of the coefficients in DPSO-ARIMA model. The fitness of each particle is calculated, the value of initial position vector is assigned as $pBest_i$ (i = 1 to m) and best position produced as gBest.

Step 3: Updating Velocity and Position

The velocity and position vector, i.e., v_i and x_i (i = 1 to m) respectively of the individual particle is updated using the (16) and (17) (Liang et al., 2006) respectively.

$$v_i^{k+1} \Leftarrow w \times v_i^k + c_1\tau_1(pxbest_i - x_i^k) + c_2\tau_2(gxbest - x_i^k) \tag{16}$$

$$x_j^{k+1} = x_j^k + v_j^{k+1} \tag{17}$$

Step 4: Identify and Preserve Potential Regions

The potential regions most recently obtained in the form of gBest upto count gBest_h_maxcount are stored as historical global best position in gBest_h.

Step 5: Updating pBest and gBest

Fitness of the individual particle is computed again with the updated position, if the updated position of particle x_i is able to improve the personal best position of the particle $pBest_i$ (for i = 1 to m), it is updated and similarly if the current global best position gBest of the particle is better than previous it gets updated.

Step 6: Controlling Stagnation Problem

The count $pBest_count_i$ and $gBest_count_i$ (for i = 1 to m) is incremented by 1 in each generation and if the $pBest_i$ and gBest has not improved from the previous values they are reset to 0. If $pBest_count_i$ and or group gBest_count attains the threshold value for any particle i (for i = 1 to m), i.e., pBest_

iter_thresh and gBest_iter_thresh respectively, then pBest and gBest are restructured otherwise the termination condition of the algorithm is tested.

Step 7: Restructuring pBest and gBest

The particles whose pBest_counti and gBest_count has crossed the threshold limit, their pBesti and gBest are restructured with the new values from the history as pBest_tempi and gBest_temp respectively.

Step 8: Checking Acceptability of pBest and gBest

The particles whose personal best, i.e., pBesti is replaced with pBest_tempi are checked for the maximum number of iterations given as pBest_chance_max, if the value of pBesti further improves from the previous value then the replaced value is accepted as permanent otherwise old values are restored and replaced with a new pBest_tempi until a improved value of the pBesti is obtained. Similarly gBets replaced with gBest_temp is checked for a improved value then the previous gBest, if it is obtained then the change is taken as permanent otherwise original value is restored.

Step 9: Termination Condition

If the evaluation of the function has exceeded the maximum allowed evaluation number, then the algorithm terminates. Otherwise, the value of velocity and position vector is again updated and all the steps are repeated until the termination condition is reached.

In (16) and (17) τ_1 and τ_2 are uniformly distributed numbers within interval [0 1] that determines the impact of $pxbest_j$ and $gxbest$ on the velocity update formula. c_1 and c_2 are two constant terms namely "self-confidence" and "swarm-confidence", respectively. The computation of inertia weight w in a linearly decreasing method is enhanced to increase convergence speed and improve optimization quality successfully in algorithm. Together with iterative performance of algorithm, its inertia weight is reduced linearly. The linearly decreasing inertia weight is calculated as given in (18) (H. Wang & Zhao, 2009). Where w_{min} and w_{max} are the minimum and maximum inertia weights respectively; $iter_{max}$ and iter are the maximum and present iterations (H. Li, He, & Wen, 2015).

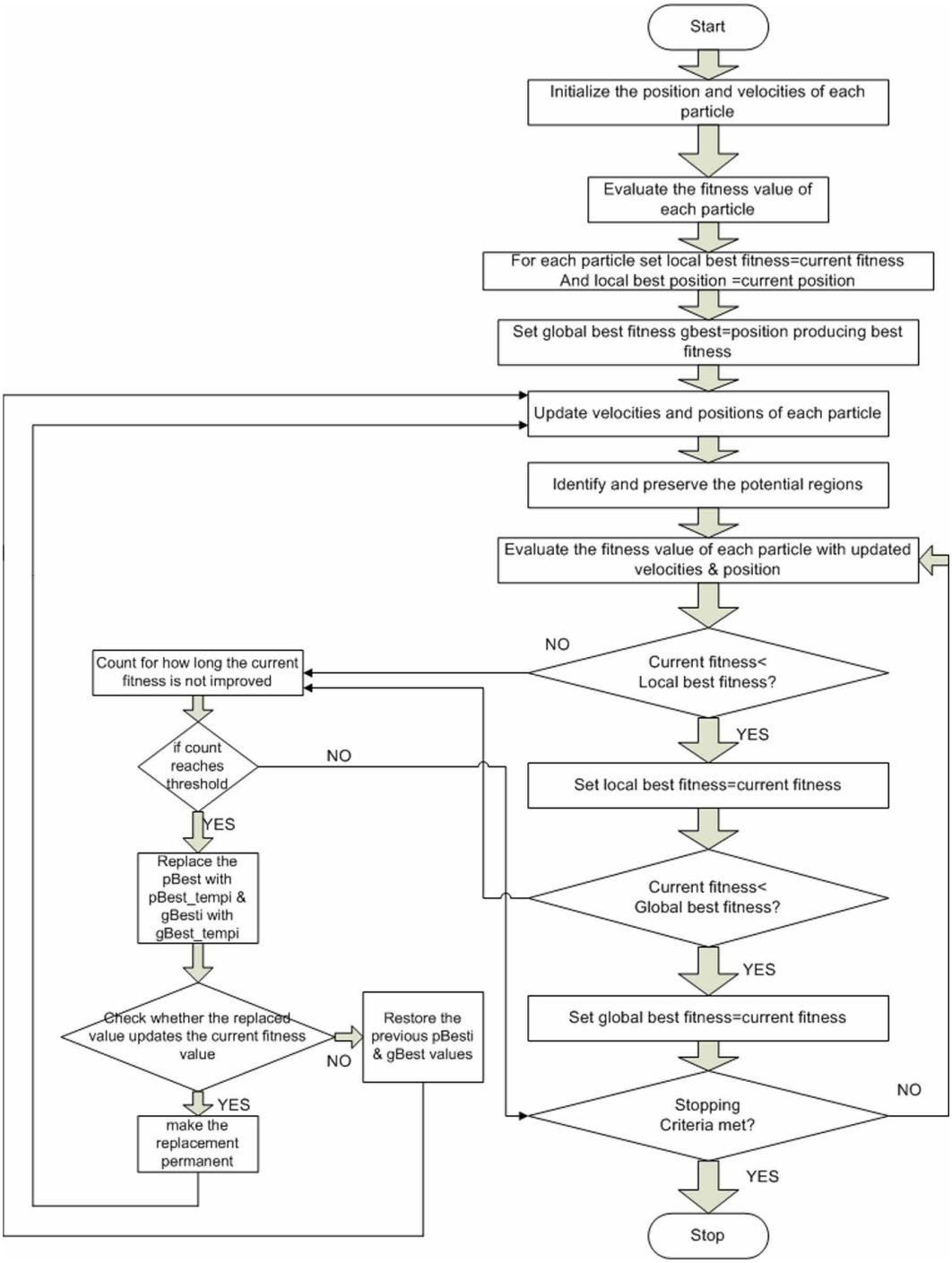
$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}} \right) iter \tag{18}$$

PROPOSED HYBRID DPSO-ARIMA MODEL

This paper proposed hybrid DPSO-ARIMA method consists of two main stages. Firstly, by applying Box Jenkins methodology, an ARIMA (p,d,q) model is built. For this, the average hourly data set of wind power is separated into estimation and validation data sets. The estimation and validation data sets have been employed to estimate parameters and evaluate the ARIMA model respectively. Secondly, the DPSO algorithm is utilized for estimating the ARIMA model parameters. The algorithm searches optimum values of all the parameters i.e. AR and MA coefficients in the range set to [-1, 1]. In this algorithm, mean square error (MSE) is employed as fitness function, and expressed as linear components, in terms of residuals and actual time series data. The flowchart for the method of building DPSO based ARIMA model is shown in Figure 6 (Alwee, Shamsuddin, & Sallehuddin, 2017; Asadi et al., 2012).

For identified ARIMA (p,d,q) model has p, auto regression (AR) terms and q, moving average terms (MA) for estimation. Once stationary time series is obtained by transforming data, d is

Figure 4. Flowchart of Dynamic PSO algorithm



determined. Now unknown p and q are estimated using DPSO algorithm implementation using objective or fitness function given in (15). The model parameters are iteratively optimized using the updated PSO algorithm with the objective of minimizing the mean squared error (MSE) between

the observed and predicted values. The values that the ARMA parameters and can assume have also been limited to ±1. It is apparent that having expert knowledge about specific variable domains when setting up the particle variable settings can increase the performance of the algorithm. Setting the DPSO particle variable settings is analogous to the procedure for setting up variable representations within a binary genetic algorithm; where one needs to choose the interval and a string length for each variable.

In the proposed DPSO-ARIMA technique the size of the swarm is taken as n=p+q coefficients (i.e. $\varphi_i \theta_j ; i = 1,2,..., p and j = 1,2.....,q$), where p is the number of autoregressive (AR) terms and moving average (MA) terms. The coefficients $\{\varphi_i\}$ and $\{\theta_j\}$ have p and q components respectively i.e. the form of individual particle can be denoted by $\{\varphi, \theta\}$, the series is represented as $\hat{x}_t^{(\varphi,\theta)}$ where $\{\varphi, \theta\}$ are the coefficients of AR and MA terms respectively.

PERFORMANCE EVALUATION

The evaluation of error measures on data is very vital in building the model for forecasting or tuning the parameters of the model. To evaluate and compare the forecasting accuracy of the proposed DPSO-ARIMA model with the persistence and ARIMA models, three criteria, i.e., mean absolute error (MAE), mean squared error (MSE) and mean absolute percentage error (MAPE) are applied. The formulae for calculating three different methods of evaluating forecast performance are listed as (19), (20) & (21) (Madhiarasan & Deepa, 2016):

$$\text{MAE} = \frac{1}{\text{N}} \sum_{\text{i=1}}^{\text{n}} \left| \text{P}_{\text{i}}^{\text{true}} - \text{P}_{\text{i}}^{\text{forecast}} \right| \tag{19}$$

$$\text{MSE} = \frac{1}{\text{N}} \sum_{\text{I=1}}^{\text{n}} \left(\text{P}_{\text{i}}^{\text{true}} - \text{P}_{\text{i}}^{\text{forecast}} \right)^2 \tag{20}$$

$$\text{MAPE} = \frac{1}{\text{N}} \sum_{\text{i=1}}^{\text{n}} \frac{\left(\left| \text{P}_{\text{i}}^{\text{true}} - \text{P}_{\text{i}}^{\text{forecast}} \right| \right)}{\text{P}_{\text{i mean}}^{\text{true}}} \text{ x } 100\% \tag{21}$$

where $\text{P}_{\text{i}}^{\text{true}}$ is the actual value and $\text{P}_{\text{i}}^{\text{forecast}}$ is the estimated forecasting value of the i^{th} test data, and N is the number of test data. The performance measure MAE has unit whereas MAPE is computed as percentage of the forecast error and the measured error. Depending on the feature of wind power forecast system, various evaluation methods have different effects. In case of uncertainty concerning to the quality of the performance evaluation criteria, the MAE should be chosen as a major evaluation criterion. It demonstrates to be more robust when dealing with great forecast errors. It is necessary to set up a superior precise and universal system for forecast error evaluation.

RESULTS AND DISCUSSION

In the presented work, the observation data with 5-minutes resolution for year 2012 has been obtained from a wind farm located at Aagar, South Dakota in USA. Averaging twelve data points, the average

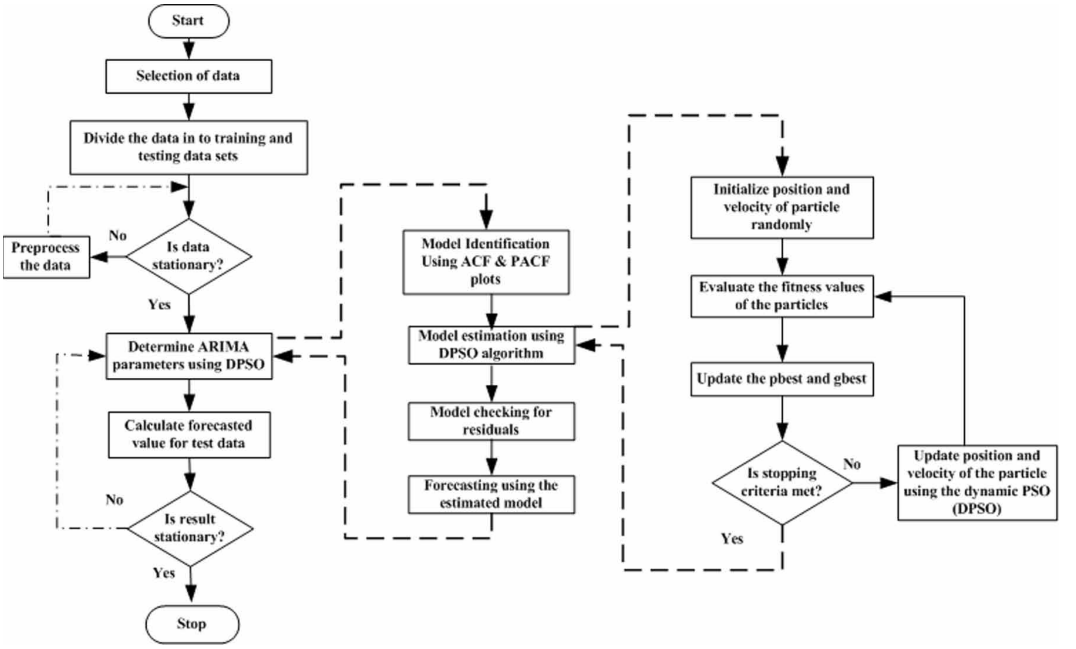
Figure 5. Pseudo-code of DPSO algorithm

```
for i = 1 to m
Randomly generate m initial particles and initial velocities which have p+q
components
Xi = random (Particles Position Initialization)
Vi = random (Particles Velocity Initialization)
pBest_counti = 0
pBest_chance_counti = -1
fitnessi = evaluate f(Xi)=MSE
pBesti = Xi
end for
gBest_count = 0
gBest_chance_count = -1
gBest = min (pBesti), i = 1 to M
gBest_h = gBest
while (func_e < func_max)
for i = 1 to m
if (pBest_counti == pBest_iter_thresh AND
pBest_chance_counti == -1)
pbest_oldi = pbesti
pBest_tempi = RESTRUCTpBest (pBesti)
pBesti = pBest_tempi
pBest_chance_counti = pBest_chance_max + 1
else
if (pBest_chance_counti > 0)
pBest_chance_counti = pBest_chance_counti -1
end if
end if
end for
if (gBest_count == gBest_iter_thresh AND
gBest_chance_count == -1)
gbest_old = gBest
gBest_temp = RESTRUCTgbest (gBest)
gBest = gBest_temp
gBest_chance_count = gBest_chance_max + 1
else
if (gBest_chance_count > 0)
gBest_chance_count = gBest_chance_count -1
end if
end if
for i = 1 to m
Vi = Vi + c1 * rand1 * ( pBesti - Xi ) + c2 * rand2 * (gBest - Xi)
Xi = Xi + Vi
fitness_tempi = evaluate f(Xi)
UPDATEpBest (pBesti, fitnessi, fitness_tempi)
end for
gBest_temp = min (pBesti)
UPDATEgBest(gBest , gBest_temp)
if (gBest < gbest_old)
UPDATEgBest_h(gBest))
end if
end while
end procedure
```

hourly data has been obtained from 01 Jan to 30 Apr 2012. Out of total 2880 observations of this wind power data, 80% and 20% have been utilized for estimation and validation of ARIMA model.

The average hourly wind power time series and its ACF/PACF plots are shown in Figure 7 and Figure 8. From the plots, it is clear that ACF decays slowly and PACF cuts off after two lags, which

Figure 6. Flowchart for building DPSO-ARIMA model



shows the non-stationary behavior of the wind power time series. Hence, data must be transformed by differencing to obtain stationary time series. The plot of first differenced time series is given in Figure 9, which shows zero mean and constant variance, i.e., stationary time series. In Figure 10, ACF and PACF plots show cut off after 2 and 1 significant lags respectively. This shows stationary behavior of time series, which is further confirmed with KPSS and ADF statistical tests, resulting to d as 1.

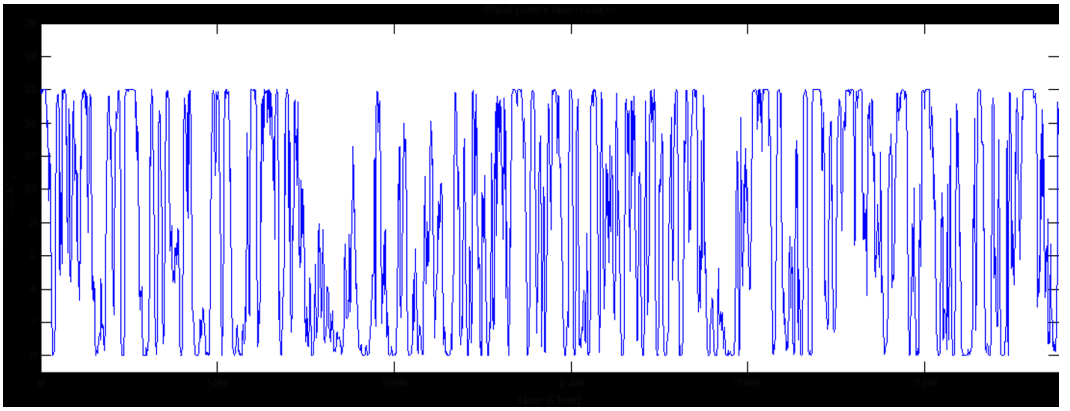
For the selection of optimal orders p, q of ARIMA (p,d,q) model, p and q are varied from 1 to 3. Among the nine models (ARIMA(1,1,1), ARIMA(1,1,2), ARIMA(1,1,3), ARIMA(2,1,1), ARIMA(2,1,2), ARIMA(2,1,3), ARIMA(3,1,1), ARIMA(3,1,2), ARIMA(3,1,3)), the appropriate model is chosen on the basis of Akaike information criterion (AIC) and Bayesian information criterion (BIC). It has been found that ARIMA (2,1,1) model gave minimum AIC /BIC value out of nine models. Hence, ARIMA (2,1,1) model has been validated and found satisfactory, which is confirmed from Figure 11, showing no significant lags in ACF and PACF plots of the residuals.

Further DPSO is used for estimating the parameters of ARIMA(2,1,1) model. There are three parameters (two AR and one MA terms) to be identified using the proposed hybrid DPSO based ARIMA model. The forecast equation for ARIMA (2,1,1) is given by (22) (Junior et al., 2014).

$$y_t = \mathrm{c} + \varnothing_1 \mathrm{y}_{\mathrm{t}-1} + \varnothing_2 \mathrm{y}_{\mathrm{t}-2} - ,_1 \epsilon_{\mathrm{t}-1} + \tag{22}$$

where c is constant term, model parameters Φ_1, Φ_2 and θ_1 are estimated via maximum likelihood. White noise terms ϵ_t are i.i.d. random variates which come from a normal distribution N (0, σ^2). The settings of parameters in DPSO algorithm are given in Table 3. The linearly decreasing inertia weight from w_{max} to w_{min} have typical value of 0.9 and 0.4 respectively, c1 and c2 are constant values typically in the range of 2 to 4. These constants are multiplied by r (a uniform random number between 0 and 1) and a measure of how far the particle is from its personal best and the global best particle so far. The values of Φ (AR coefficients) and θ (MA coefficients) were bounded between -1 and 1. The population size was varied from 10 to 50 to obtain stable convergence.

Figure 7. Average hourly wind power from 01Jan to 30 Apr 2012



Results show that DPSO converges faster and achieves a least MSE value after 10 iterations, converging towards the best one. The convergence characteristics of DPSO algorithm is shown in Figure 12, for optimum parameter estimation of ARIMA model. The value of optimal parameters obtained using DPSO algorithm for ARIMA (2,1,1) model are AR(1) = 0.8173, AR(2) = - 0.3265 and MA(1) = 0.8012 are given in Table 4. Figure 13 compares the performance of persistence, ARIMA (2,1,1) and DPSO-ARIMA (2,1,1) model for 24 hours ahead wind power forecasts respectively. The actual and forecasted values of each model along with the forecast errors are given in Table 5. The performance evaluation of proposed DPSO-ARIMA (2,1,1) model compared to that of the persistence and ARIMA (2,1,1) model are given in Table 6 in terms of MAE, MSE and MAPE for 12 and 24 hours ahead forecast of wind power. The superiority of the proposed method is evident from the result given in Table 6.

The modelling of persistence model, ARIMA (p, d, q) and DPSO-ARIMA (p, d, q) model and forecasting of wind power using these models has been done using MATLAB R2013a.

CONCLUSION AND FUTURE SCOPE

In this paper, ARIMA (p, d, q) model was successfully developed based on Box-Jenkins method for wind power forecasting. The parameters of the proposed hybrid DPSO-ARIMA model were estimated using dynamic particle swarm optimization algorithm instead of maximum likelihood estimation. The forecasting performance of ARIMA and DPSO-ARIMA models have been evaluated in terms of MAE, MSE and MAPE and compared with the persistence method. Dynamic particle swarm optimization algorithm overcomes the limitation of traditional estimation method, i.e., difficulty in implementation and need of re-estimation for slight change in the input value; this improves the forecasting performance of the ARIMA model. The dynamic particle swarm optimization algorithm also overcomes the problem of classical PSO getting stuck in the local optima by continuously monitoring the global and personal best values of the particle. The model estimation using evolutionary computing methods improves the overall prediction accuracy of ARIMA model. The simulation results demonstrate that the proposed DPSO-ARIMA (2,1,1) model has minimum value of MAE, MSE and MAPE as compared to ARIMA (2,1,1) and persistence model respectively. The results reveal the efficacy of the DPSO-ARIMA model as compared to the conventional ARIMA and persistence models. In future, the estimation of the ARIMA (p,d,q) model can be done using other evolutionary optimization algorithms or hybrid of the existing algorithms. Such an optimization would allow avoiding over-fitting of the model. The proposed hybrid model can also be used for forecasting stock markets, weather or any other commodity.

Figure 8. ACF and PACF plots

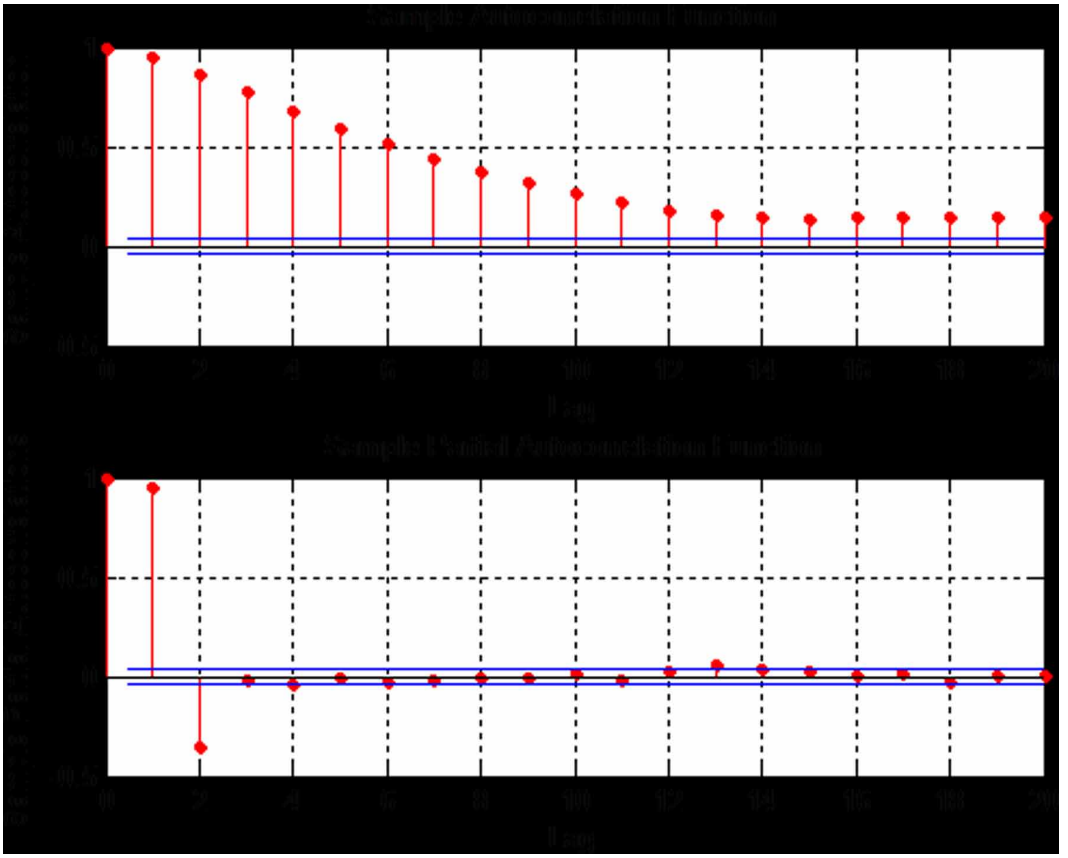


Figure 9. First differenced time series plot

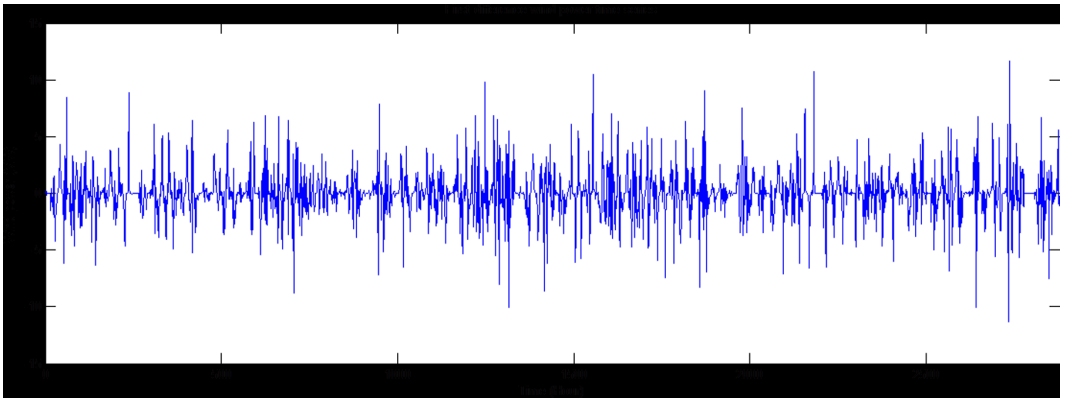


Figure 10. ACF and PACF plots of First differenced time series

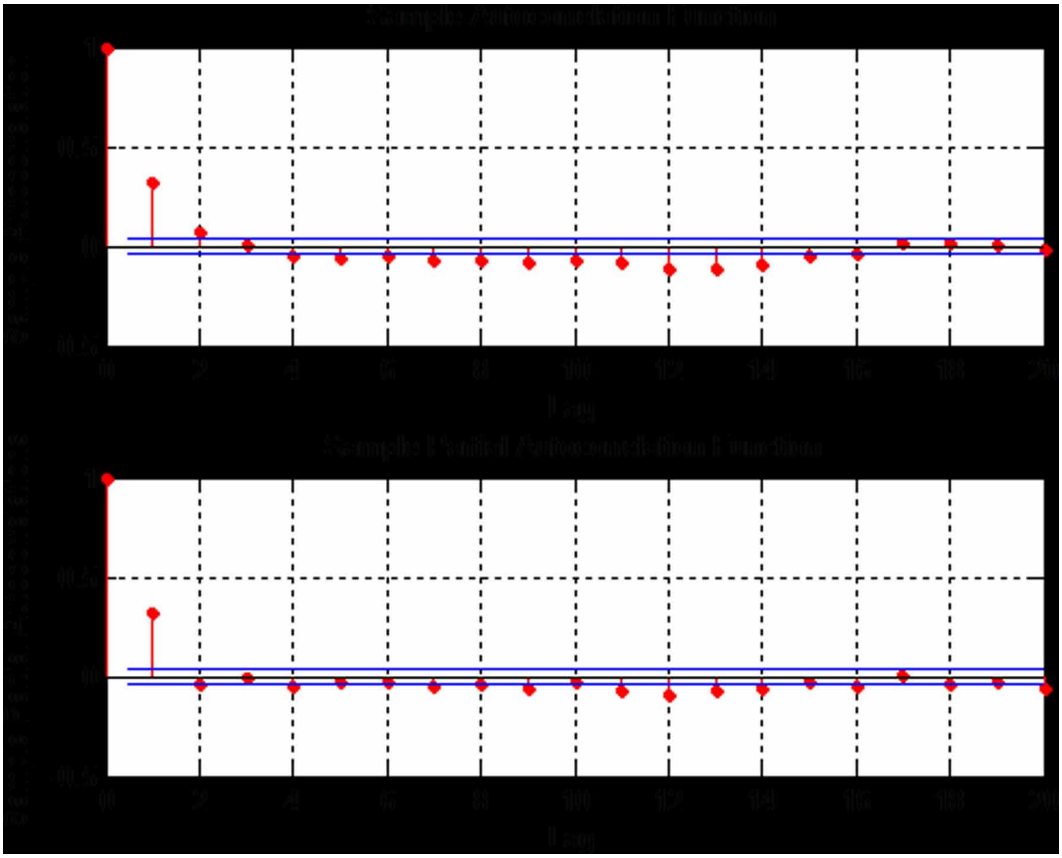


Figure 11. ARIMA(2,1,1) model validation

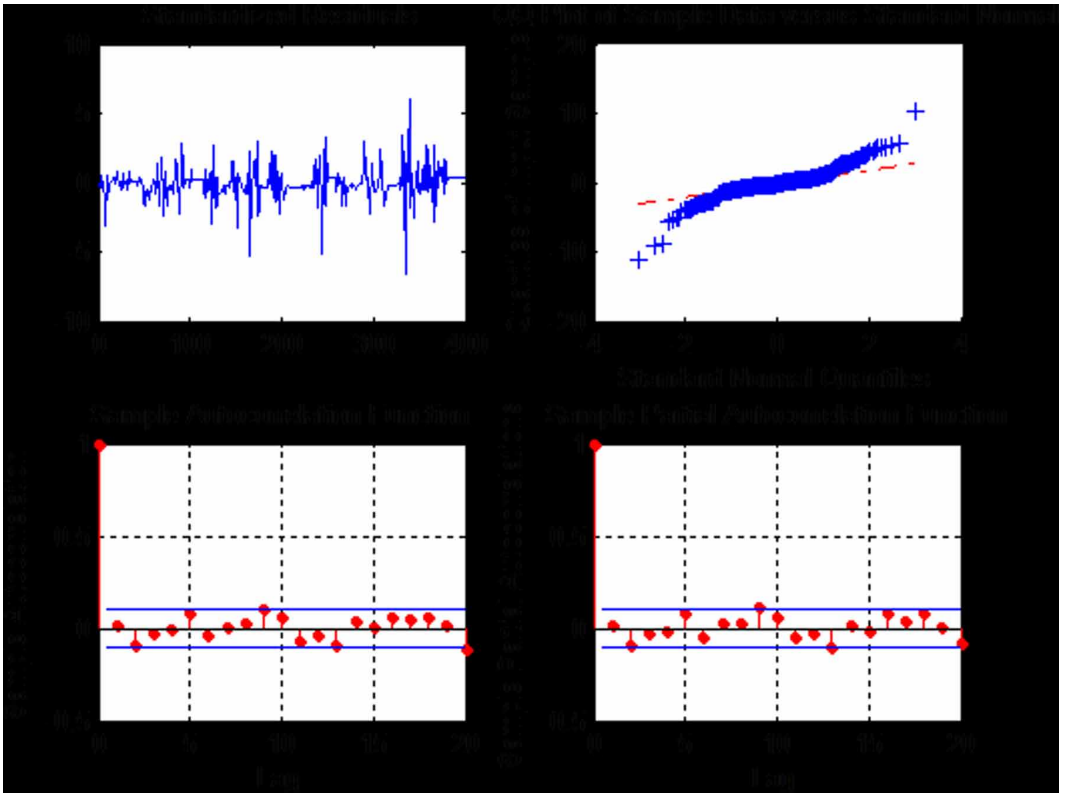


Figure 12. DPSO convergence characteristic

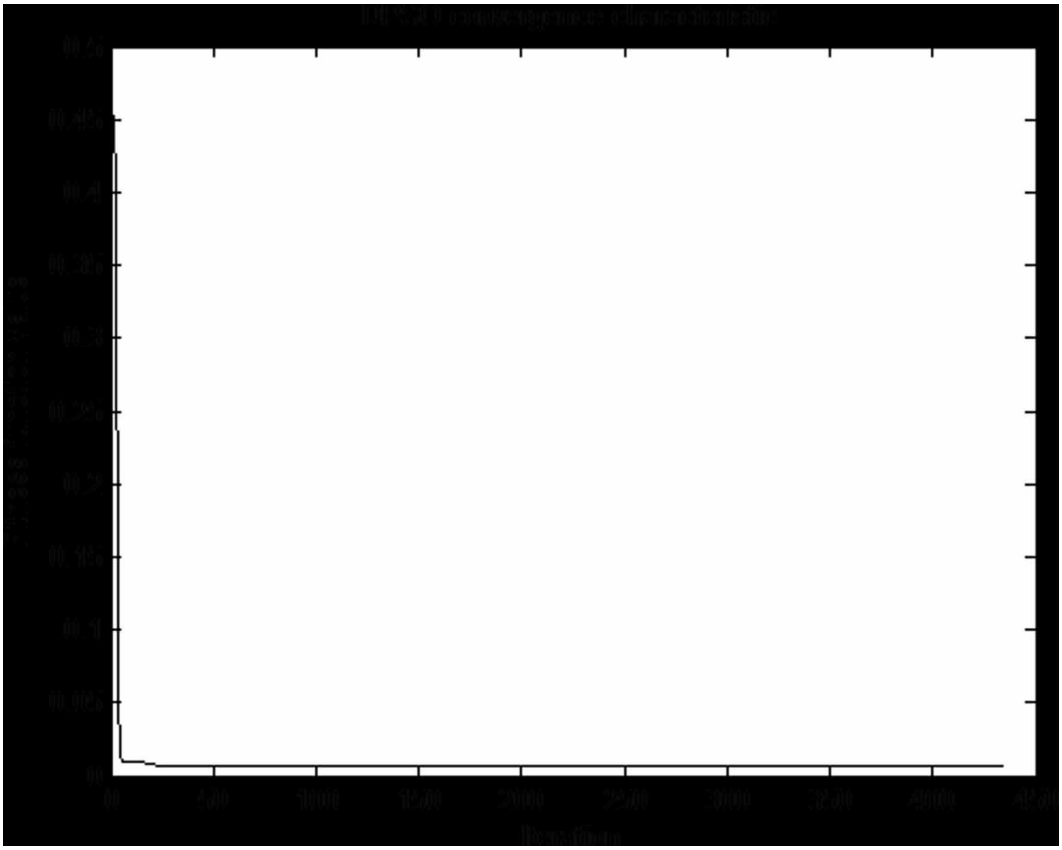


Figure 13. Actual and forecast plots of models

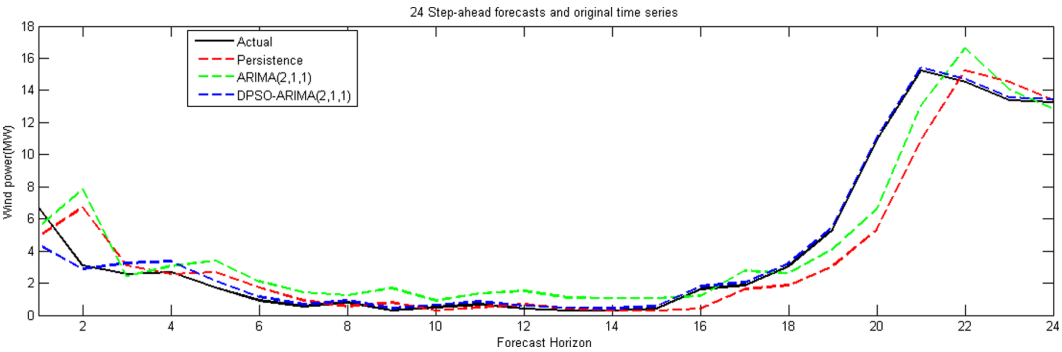


Table 3. DPSO parameters settings

Parameters	Setting
Dimension (no. of unknown variables)	3
Population (swarm) size	10 to 50
Maximum iterations	1000
Maximum runs	10
Lower and Upper bound of parameters	-1, 1
c1 and c2	2.0
Minimum inertia weight, w_{min}	0.4
Maximum inertia weight w_{max}	0.9

Table 4. Optimum parameters

Model	Parameters
ARIMA(2,1,1)	AR(1): 1.3, AR(2): - 0.3723, MA(1): - 0.9918
DPSO-ARIMA(2,1,1)	AR(1): 0.8173, AR(2): - 0.3265, MA(1): 0.8012

Table 5. Actual and forecast values for 24 hours ahead

Forecast Horizon	Actual	Persistence		ARIMA(2,1,1)		DPSO-ARIMA(2,1,1)	
		Forecast	Error	Forecast	Error	Forecast	Error
1	6.72133	4.89408	1.82725	5.46969	1.25164	5.39960	1.32173
2	3.10325	6.72133	-3.61808	4.84473	-1.74148	2.89310	0.21015
3	2.55467	3.10325	-0.54858	2.42189	0.13278	3.23040	-0.67573
4	2.66417	2.55467	0.10950	3.05664	-0.39247	3.37460	-0.71043
5	1.70442	2.66417	-0.95975	3.39986	-1.69544	2.15390	-0.44948
6	0.91825	1.70442	-0.78617	2.09736	-1.17911	1.14310	-0.22485
7	0.55983	0.91825	-0.35842	1.42282	-0.86299	0.68640	-0.12657
8	0.80475	0.55983	0.24492	1.24233	-0.43758	0.92420	-0.11945
9	0.29267	0.80475	-0.51208	1.69042	-1.39775	0.43850	-0.14583
10	0.48508	0.29267	0.19241	0.92192	-0.43684	0.65470	-0.16962
11	0.66150	0.48508	0.17642	1.35897	-0.69747	0.84200	-0.18050
12	0.39508	0.66150	-0.26642	1.51081	-1.11573	0.57670	-0.18162
13	0.29575	0.39508	-0.09933	1.08949	-0.79374	0.47470	-0.17895
14	0.27133	0.29575	-0.02442	1.05288	-0.78155	0.44780	-0.17647
15	0.39125	0.27133	0.11992	1.05156	-0.66031	0.56650	-0.17525
16	1.63008	0.39125	1.23883	1.21098	0.41910	1.80520	-0.17512
17	1.86292	1.63008	0.23284	2.78012	-0.91720	2.03830	-0.17538
18	3.07242	1.86292	1.20950	2.61393	0.45849	3.24810	-0.17568
19	5.33467	3.07242	2.26225	4.10323	1.23144	5.51040	-0.17573
20	10.90790	5.33467	5.57323	8.60384	2.30406	11.08400	-0.17610
21	15.23980	10.90790	4.33190	13.04210	2.19770	15.41600	-0.17620
22	14.51000	15.23980	-0.72980	16.61650	-2.10650	14.68600	-0.17600
23	13.39180	14.51000	-1.11820	14.03770	-0.64590	13.56700	-0.17520
24	13.26030	13.39180	-0.13150	12.85020	0.41010	13.43600	-0.17570

Table 6. Forecast performance evaluation of models

Model	12 hours ahead			24 hours ahead		
	MAE	MSE	MAPE	MAE	MSE	MAPE
Persistence	0.8000	1.5726	46.0101	1.1113	3.2788	26.3990
ARIMA(2,1,1)	0.9451	1.1540	54.3555	1.0111	1.3929	24.0192
DPSO-ARIMA(2,1,1)	0.3763	0.2626	21.6437	0.2762	0.1468	6.5600

REFERENCES

Aasim, S., Singh, S. N., & Mohapatra, A. (2019). Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. *International Journal of Renewable Energy*, *136*, 758–768. doi:10.1016/j.renene.2019.01.031

Alwee, R., Shamsuddin, S. M., & Sallehuddin, R. (2017). Swarm Optimized Grey SVR and ARIMA for Modeling of Larceny-Theft Rate with Economic Indicators. *International Journal of Computational Intelligence and Applications*, *16*(02), 1750008. doi:10.1142/S1469026817500080

Anwar, A., & Mahmood, A. N. (2014). Enhanced estimation of Autoregressive wind power prediction model using Constriction Factor Particle Swarm Optimization. *2014 9th IEEE Conference on Industrial Electronics and Applications*, 1136–1140. doi:10.1109/ICIEA.2014.6931336

Asadi, S., Tavakoli, A., & Hejazi, S. R. (2012). A new hybrid for improvement of auto-regressive integrated moving average models applying particle swarm optimization. *Expert Systems with Applications*, *39*(5), 5332–5337. doi:10.1016/j.eswa.2011.11.002

Chang, W.-Y. (2014). A Literature Review of Wind Forecasting Methods. *Journal of Power and Energy Engineering*, *02*(04), 161–168. doi:10.4236/jpee.2014.24023

Huang, C.-M., Huang, C.-J., & Wang, M.-L. (2005). A particle swarm optimization to identifying the ARMAX model for short-term load forecasting. *IEEE Transactions on Power Systems*, *20*(2), 1126–1133. doi:10.1109/TPWRS.2005.846106

Du, P., Wang, J., Yang, W., & Niu, T. (2019). A novel hybrid model for short-term wind power forecasting. *Applied Soft Computing*, *80*, 93–106. doi:10.1016/j.asoc.2019.03.035

Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1*, 39–43. Retrieved from http://www.ppgia.pucpr.br/~alceu/mestrado/aula3/PSO_2.pdf

Eldali, F., Hansen, T. M., Suryanarayanan, S., & Chong, E. K. P. (2016). Employing ARIMA models to improve wind power forecasts: A case study in ERCOT. *2016 North American Power Symposium (NAPS)*, 1–6. doi:10.1109/NAPS.2016.7747861

Erdogdu, E. (2007). Electricity demand analysis using cointegration and ARIMA modelling: A case study of Turkey. *International Journal of Energy Policy*, *35*(2), 1129–1146. doi:10.1016/j.enpol.2006.02.013

Garg, H. (2015). A Hybrid GA-GSA Algorithm for Optimizing the Performance of an Industrial System by Utilizing Uncertain Data. Handbook of Research on Artificial Intelligence Techniques and Algorithms, 620–654. doi:10.4018/978-1-4666-7258-1.ch020

Garg, H. (2016). A hybrid PSO-GA algorithm for constrained optimization problems. *Journal of Applied Mathematics and Computation*, *274*, 292–305. doi:10.1016/j.amc.2015.11.001

Garg, H. (2019). A hybrid GSA-GA algorithm for constrained optimization problems. *Journal of Information Science*, *478*, 499–523. doi:10.1016/j.ins.2018.11.041

Hadavandi, E., Ghanbari, A., & Abbasian-Naghneh, S. (2010). Developing a Time Series Model Based on Particle Swarm Optimization for Gold Price Forecasting. *Proceedings of the 2010 Third International Conference on Business Intelligence and Financial Engineering*, 337–340. doi:10.1109/BIFE.2010.85

Handoyo, S., Efendi, A., Jie, F., & Widodo, A. (2017). Implementation of particle swarm optimization (PSO) algorithm for estimating parameter of arma model via maximum likelihood method. *ECU Publications Post*, *2013*(7), 1337–1363. Advance online publication. doi:10.17654/MS102071337

Seo, J.-H., Im, C.-H., Heo, C.-G., Kim, J.-K., Jung, H.-K., & Lee, C.-G. (2006). Multimodal function optimization based on particle swarm optimization. *IEEE Transactions on Magnetics*, *42*(4), 1095–1098. doi:10.1109/TMAG.2006.871568

Junior, P. R., Salomon, F. L. R., & Pamplona, E. de O. (2014). ARIMA: An Applied Time Series Forecasting Model for the Bovespa Stock Index. *Journal of Applied Mathematics*, *05*(21), 3383–3391. doi:10.4236/am.2014.521315

Lei, M., Shiyan, L., Chuanwen, J., Hongling, L., & Yan, Z. (2009). A review on the forecasting of wind speed and generated power. *International Journal of Renewable and Sustainable Energy Reviews*, *13*(4), 915–920. doi:10.1016/j.rser.2008.02.002

Li, H., He, H., & Wen, Y. (2015). Dynamic particle swarm optimization and K-means clustering algorithm for image segmentation. *Journal of Optics*, *126*(24), 4817–4822. doi:10.1016/j.ijleo.2015.09.127

Li, Z., Ye, L., Zhao, Y., Song, X., Teng, J., & Jin, J. (2016). Short-term wind power prediction based on extreme learning machine with error correction. *International Journal of Protection and Control of Modern Power Systems*, *1*(1), 1–8. doi:10.1186/s41601-016-0016-y

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, *10*(3), 281–295. doi:10.1109/TEVC.2005.857610

Madhiarasan, M., & Deepa, S. N. (2016). Performance Investigation of Six Artificial Neural Networks for Different Time Scale Wind Speed Forecasting in Three Wind Farms of Coimbatore Region. *International Journal of Innovation and Scientific Research*, *23*(2), 380–411.

Mao, Y., & Shaoshuai, W. (2016). A review of wind power forecasting amp; prediction. *2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 1–7. doi:10.1109/PMAPS.2016.7764085

Ong, C.-S., Huang, J.-J., & Tzeng, G.-H. (2005). Model identification of ARIMA family using genetic algorithms. *Journal of Applied Mathematics and Computation*, *164*(3), 885–912. doi:10.1016/j.amc.2004.06.044

Parsopoulos, K. E., & Vrahatis, M. N. (2010). *Particle Swarm Optimization and Intelligence: Advances and Applications*. Retrieved from https://www.igi-global.com/book/particle-swarm-optimization-intelligence/37246

Patwal, R. S., Narang, N., & Garg, H. (2018). A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units. *Journal of Energy*, *142*, 822–837. doi:10.1016/j.energy.2017.10.052

Samson, J., & Kainkwa, R. (2019). Prediction of Wind Power Density for Electricity Generation at Makambako, Tanzania Using Auto-Regression Integrated Moving Average (ARIMA) Model. *Tanzania Journal of Science, 45*(2), 209-215–215.

Sankardoss, V., & Geethanjali, P. (2017). PMDC Motor Parameter Estimation Using Bio-Inspired Optimization Algorithms. *IEEE Access : Practical Innovations, Open Solutions*, *5*, 11244–11254. doi:10.1109/ACCESS.2017.2679743

Saxena, N., Tripathi, A., Mishra, K. K., & Misra, A. K. (2015). Dynamic-PSO: An improved particle swarm optimizer. *Proc. of 2015 IEEE Congress on Evolutionary Computation (CEC),* 212–219. doi:10.1109/CEC.2015.7256894

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *Proc. of 1998 IEEE Int. Conf. on. Evolutionary Computation,* 69–73. doi:10.1109/ICEC.1998.699146

Soman, S. S., Zareipour, H., Malik, O., & Mandal, P. (2010). A review of wind power and wind speed forecasting methods with different time horizons. *North American Power Symposium 2010*, 1–8. doi:10.1109/NAPS.2010.5619586

Wang, H., & Zhao, W. (2009). ARIMA Model Estimated by Particle Swarm Optimization Algorithm for Consumer Price Index Forecasting. *Int. Conf. on Artificial Intelligence and Computational Intelligence,* 48–58. doi:10.1007/978-3-642-05253-8_6

Wang, J., Liang, J., Che, J., & Sun, D. (2008). ARMA Model identification using Particle Swarm Optimization Algorithm. *2008 International Conference on Computer Science and Information Technology*, 223–227. doi:10.1109/ICCSIT.2008.60

Zhao, H., & Feng, L. (2014). An Improved Adaptive Dynamic Particle Swarm Optimization Algorithm. *Journal of Networks*, *9*(2), 488–494. doi:10.4304/jnw.9.2.488-494

Zhao, X., Wang, S., & Li, T. (2011). Review of Evaluation Criteria and Main Methods of Wind Power Forecasting. *Journal of Energy Procedia*, *12*, 761–769. doi:10.1016/j.egypro.2011.10.102

Pavan Kumar Singh had the Diploma in Electrical Engineering from Institute of Engineering and Rural technology, Allahabad in 1984. He was awarded AMIE degree in Electrical Engineering from The Institution of Engineers (India), Calcutta in 1988, Master of Technology degree from Motilal Nehru National Institute of Technology, Allahabad in 2005 and is a research scholar in the same institute. He served for 20 years in engineering department of Indian Air Force and 11 years as an Assistant Professor in different engineering colleges in India. His main research area of interests are forecasting, applications of artificial intelligence and optimization techniques.

Nitin Singh (PhD) is currently working as an Associate Professor in Department of Electrical Engineering, MNNIT Allahabad, Prayagraj, India. He is an alumnus of Motilal Nehru National Institute of Technology Allahabad, India. His primary areas of research are Power System Planning, Electricity markets, Artificial Intelligence, implementation of optimization algorithms and machine learning techniques for power system optimization. He has published more than 45 publications in International Journals and in Proceedings of International Conferences of repute. He is serving as reviewer of several Scopus, SCI and E-SCI-indexed journals. He has organized several international conferences under the banner of IEEE and Springer. He is Member of IEEE, member of IEEE Computational Intelligence Society, Power Energy Society, and Industrial Application society. He is currently serving as Secretary, IEEE Joint Chapter of Industrial Electronics/ Power Electronics /Control System at MNNIT Allahabad.

Richa Negi received the B.Tech. degree in electrical engineering and the M.Tech. degree in power systems from REC, Kurukshetra, India, in 1992 and 1994 respectively, and the Ph.D. degree in control systems from MNNIT, Allahabad, in 2013. Presently, she is working as professor in the Department of Electrical Engineering, MNNIT Allahabad. Her research interests include multidimensional control system, nonlinear control and systems, WMAS, and smart grid. She has many research publications in journal and conferences of international repute.