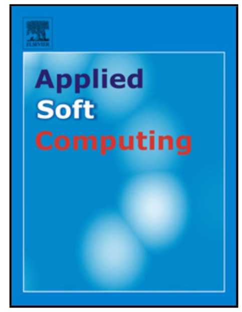
Accepted Manuscript



Title: Wind Power Prediction using Deep Neural Network based Meta Regression and Transfer Learning

Authors: Aqsa Saeed Qureshi, Asifullah Khan, Aneela Zameer, Anila Usman

PII: S1568-4946(17)30294-6
DOI: http://dx.doi.org/doi:10.1016/j.asoc.2017.05.031
Reference: ASOC 4235

To appear in: *Applied Soft Computing*

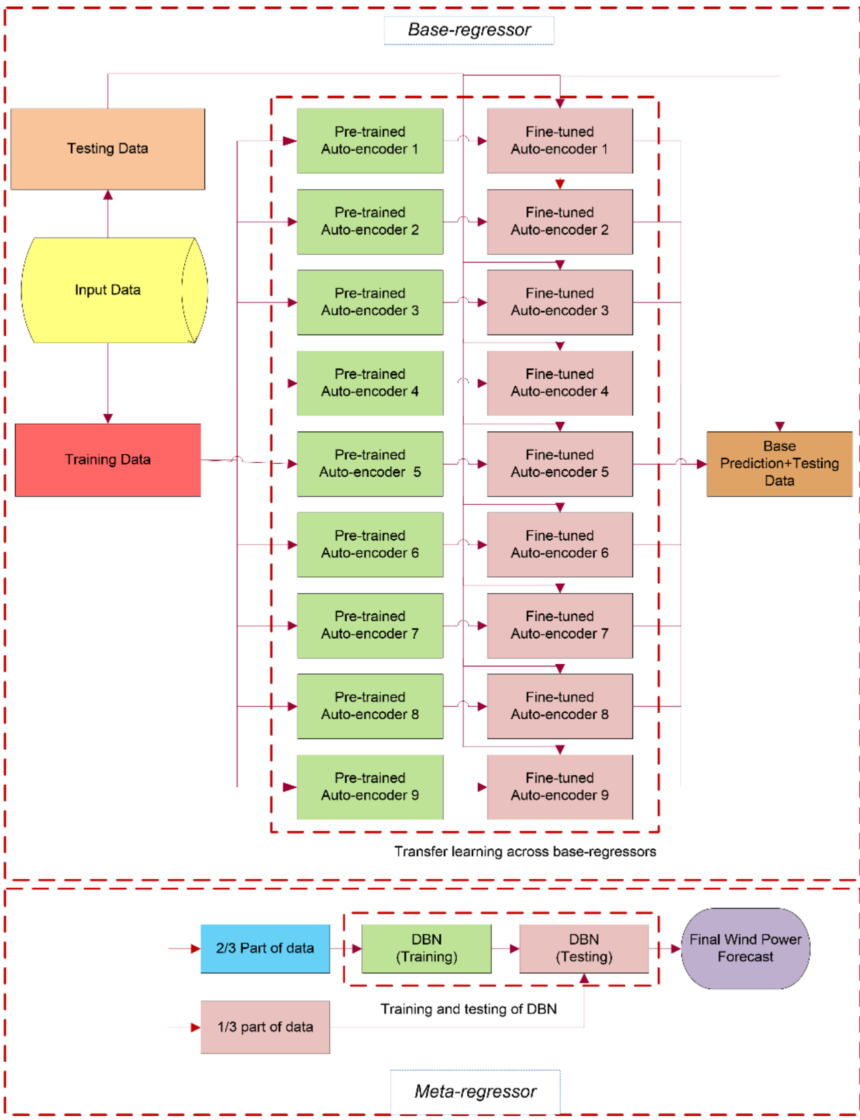
Received date: 21-2-2017
Revised date: 12-5-2017
Accepted date: 14-5-2017

Wind Power Prediction using Deep Neural Network based Meta Regression and Transfer Learning

Aqsa Saeed Qureshi[1], Asifullah Khan[*1], Aneela Zameer[1], Anila Usman[1]

[1]Department of Computer Science, Pakistan Institute of Engineering and Applied Sciences, Nilore-45650, Islamabad; asif@pieas.edu.pk

Graphical abstract



Highlights:

- Wind power prediction system using Deep Neural Network based Ensemble and Transfer Learning.
- Ensemble learning is shown to enable collective and robust decision on unseen data.
- Use of Transfer Learning is shown to enable quick learning on a new wind farm.
- Deep Auto-encoders act as base, whereas Deep Belief Network acts as Meta regressor. • The proposed wind predictor has good generalization and thus robust to small variations in metrological properties.

Abstract

An innovative short term wind power prediction system is proposed which exploits the learning ability of deep neural network based ensemble technique and the concept of transfer learning. In the proposed DNN-MRT scheme, deep auto-encoders act as base-regressors, whereas Deep Belief Network is used as a meta-regressor. Employing the concept of ensemble learning facilitates robust and collective decision on test data, whereas deep base and meta-regressors ultimately enhance the performance of the proposed DNN-MRT approach. The concept of transfer learning not only saves time required during training of a base-regressor on each individual wind farm dataset from scratch but also stipulates good weight initialization points for each of the wind farm for training. The effectiveness of the proposed, DNN-MRT technique is expressed by comparing statistical performance measures in terms of root mean squared error (RMSE), mean absolute error (MAE), and standard deviation error (SDE) with other existing techniques.

Keywords ---- Wind power prediction; sparse denoising auto-encoders; meta-regressor; transfer learning; meteorological properties

1. Introduction

For smooth generation of power from turbine and also for the integration of wind power towards grid, exact prediction of wind power is indispensible; however, due to the fluctuating behavior of wind and uncertainty in the geographical properties and climatic conditions, accurate wind power forecast is a crucial task. Wind power prediction can be classified into two major categories i.e. short term and long term based wind power prediction. In case of short term wind power forecast, prediction is carried out ranging from a few minutes to a day, whereas, in case of long term forecast, power is predicted on the basis of days, months or even years in advance. In order to ensure the reliable forecast system, short term wind power prediction system is considered preferable. Countries having sufficient wind energy exploit various wind power prediction approaches. These approaches are based either on physical or statistical models; hybrid models based on a combination of both of these are also popular. Physical models are affected with the physical attributes of location such as obstacle, coarseness of surface, turbulence etc. Mostly physical models are used for long term prediction of wind power [1]. Statistical model employ previous data for the prediction of power in the near future [2,3], and can exploit different artificial intelligence techniques such as Support Vector Regressor (SVR) [4] , Multi-layer perceptron [5], deep neural network [6], and combination of different approaches [7], [8].

Recently, research in wind power prediction is gaining importance, owing to its useful applications on commercial scale without polluting the environment. Kartz and Murphy in 1984 proposed time-series wind power forecasting method [1]. For efficient wind power prediction system, different types of methods based on regression have been presented by different researcher. Liu et al. [9] reported short term wind power prediction system that is based upon wavelet transform, attributes of turbine and support vector machine. Xuan et al. [10] proposed data mining and K-means clustering based clustering technique that analyze wind power data for extracting the redundant data and finally wind power is forecasted using Artificial Neural Network (ANN). A hybrid approach that is based on least square support vector machine (LSSVM) and gravitational search algorithms (GSA) in which parameters of LSSVM are optimized using GSA is proposed by Yuan et al. [4]. Another short term based wind power prediction in which 52 ANNs and five gaussian sub model are trained on historical data in parallel is proposed by Lee et al. [11]. In Lee's technique each model generates power forecast for the same hour; and at the end final forecast is predicted on the basis of forecast predicted by different ANNs and gaussian models. Forecasting of wind power based on ensemble predictions method is proposed by Penson et al. [12]. Grazia et al. [13] trained different model on wind farm real data from three different wind turbines. Auto Regression Moving Average based model is used for the linear mapping of input data to output data, other models used in Grazia's technique are ANNs and Adaptive Neuro Fuzzy Interface systems.

An efficient wind power prediction system is proposed by Bhaskar et al. [14], in which two phases are involved during training. In the first phase wavelet is used for the decomposition of wind series and in second phase NN is used for transforming wind speed into its corresponding power. Amjady et al. [15] reported a wind power prediction system that is trained using ANN in which optimal choice of hidden neuron is chosen by heuristic method, because in ANN performance is dependent on the size of hidden layer. Carolin et al. [16] also used ANN for the final prediction of wind power on seven wind farm data sets that contains three years of data. Wind power prediction system that uses clonal search algorithm and wavelet neural network during training is proposed by chitsaz et al. [17]. Grassi et al. [18] reported an interesting approach that consists of two layered ANN, having three neuron in each layer and a single neuron in output layer that predicts the final wind farm energy. For the first layer hyperbolic tangent, while for second hidden layer sigmoid function is used as an activation function. A Global Energy Forecasting Competition is carried out in 2012, in this competition different methods for reliable wind power prediction system are suggested by different researchers Hong et al.[19] revealed an overview of competition along with the description of data. Another review that is based on different wind power prediction related technique is also reported by Lei et al. [20].

Various hybrid approaches are also reported by many researchers [21], [22], [23], a comprehensive review on different hybrid approaches use for wind power prediction is reported by Tascikaraoglu et al.[24].Wu et al. [25] proposed an Action Dependent Heuristic Dynamic Programming based method that uses Radial basis neural network to determine maximum power of wind. Another hybrid technique is stated by Peng et al. [26] that is based on statistical and physical methods. Bao et al. [21] proposed wind power prediction in which empirical model decomposition (EMD) decomposes the power associated with wind into several parts that contain high frequency information and also into parts that contain information related to low frequencies. For forecasting of wind power relevance vector machine (RVM) is used which sum up the prediction associated with high as well as low frequencies. Pinson et al. [27] used statistical method for efficient wind power prediction by using prediction error generated in series after conversion of multivariate Gaussian random variable. Another Quantile regression for probalistic wind power forecast is proposed by Nielsen et al. [28], purpose of technique is to extend the recent wind power prediction systems. Different ensemble based approaches related to wind speed are also reported in literature.

Technique proposed by Sanz et al. [29] utilizes bank of ANNs, which explore the diversity among input data sample and show that the prediction related to wind speed outperforms the predictions of single neural network (NN). Troncoso et al. [30] used various regression trees for predicting wind speed related to different wind farms and also showed comparison among different NNs and SVR. Garcia et al.

[31] proposed ensemble based technique in which bank of SVRs are used to forecast the final speed of wind. Moreover Garcia's approach shows better performance in comparison to bank of multilayer perceptron. Another forecasting technique based on hybrid model for predicting speed of wind is proposed by Zhang et al. [32] in which, dataset is taken from northwest of China. To overcome the shortcoming of ANN for forecasting wind speed, Ma et al. [33] developed model that dependents on dynamic fuzzy based ANN. There are many statistical and hybrid approaches for efficient wind power prediction; however with every approach different disadvantages are associated [34].

Amjady's et al. [35] proposed an efficient short term wind power prediction system in which features are selected using irrelevancy and redundancy filter, that exploits the concept of mutual information (MI). After selecting features, wind power is predicted through forecast engine that consists of Modified Hybrid Neural Network (MHNN), whereas Particle Swarm Optimization (PSO) is used for the optimization of weights. In Amjady's et al. [35] technique prediction from Radial Basis Function (RBF) Networks are provided to Second Levenberg Marquardr (LM) Neural Network, and weights are optimized through enhanced PSO. After the adjustment of weights output along with adjusted weight are then passed to second Neural Network. In the same way similarly procedure is performed with Broyden Fletcher-Goldfarb-Shanno (BFGS) and Baysian Regularization (BR) based Neural Network, but drawback is that error produced by first predictor is not only amplified but also gets propagate in the remaining predictors. Wang et al. [36] used hybrid approach based on Elman neural network and empirical mode decomposition to efficiently predict speed of wind.

Another ensemble based technique (GPeANN) for the short term prediction of wind power using GP (Genetic programing) was recently proposed by Zameer et al. [37]. In GPeANN, five ANN based base-regressors are trained using features that are selected using MI based feature selection technique, whereas GP is employed as a meta-regressor. After the training phase of meta-regressor, the best evolved GP tree is used to check performance of meta-regressor on unseen data (test data). Hu et al. [38] use concept of transfer learning for efficient wind speed prediction and used hidden layer of pre-trained denoising auto-encoder on old wind farm data and modification in architecture is performed according to newly developed wind farm data to predict the final wind speed. Another Deep Belief Network (DBN) based technique that uses time series data for forecasting load demand is proposed by Qiu et al. [39]. Wang et al. [40] used ensemble of deep learning based approach for efficient forecasting of wind power. In all of the previously proposed short term based wind power prediction strategies for different wind farms data, training is needed to be performed from scratch which consumes time. Transfer learning can help to reduce the time required during the training across different wind farms. Moreover, another drawback of previously proposed sequentially connected regressors based techniques for efficient wind power prediction is that error generated by one predictor propagates and also gets amplified by the next predictor. Use of ensemble reduces error propagation due to robustness, as compared to sequentially connected regressors. This explains the main motivation and contribution of the proposed DNN-MRT technique.

In this work, we investigate the idea of utilizing Deep Neural Network both as base-regressors and as a meta-regressor. The concept of meta-regressor is used to smooth-out the rapid transients in predictor behavior due to abrupt changes in atmospheric properties. Proposed deep neural network based meta-regressor technique, DNN-MRT, is comprised of two phases. In the first phase, nine deep auto-encoders are trained as base-regressors, while in the second phase after the learning of base-regressors; predictions on test data from base-regressors along with the test data features are provided as an input to the meta-regressor. The meta-regressor used in the proposed technique is Deep Belief Network, which is formed by stacking Restricted Boltzmann Machine (RBMs). As the proposed DNN-MRT technique is exploiting the learning capabilities of an ensemble of regressors therefore, it is more robust as compared to previously proposed techniques. Moreover, the involvement of deep architecture in base-regressors as well as in meta-regressor improves the representation power and enables better learning of the underlying mathematical function, which ultimately provides enhanced performance. To overcome the short comings of the previously proposed methods, proposed DNN-MRT techniques is based on the ensemble method that uses the concept of transfer learning. Table 1 shows the list of abbreviation used in this paper. Section 2 is based on methodology of the proposed DNN-MRT technique, whereas, Section 3 presents the implementation details, while related results are discussed in Section 4. In the end, conclusion is drawn in Section 5.

2. Proposed DNN-MRT Technique for Wind Power Forecasting

The Proposed DNN-MRT approach consists of two phases. In the first phase, base-regressors are trained, where as in second phase, meta-regressor forecasts the final wind power based on the predictions generated by base-regressors. In the proposed DNN-MRT technique, deep learning is exploited during the training of base and meta-regressors. Also transfer learning is used during the training of base-regressors for different wind farm datasets. Block diagram of the proposed DNN-MRT technique in terms of transfer learning, base and meta-regressor is shown in Fnig. 1

2.1. Wind Power Dataset:

Dataset used in the proposed DNN-MRT technique contains collection of three years of data from five different wind farms situated in Europe. The datasets contain power measurement and also the meteorological forecast related to components of wind. Zonal (ZS), meridional component of surface wind (MS), corresponding speed (S) and direction (D) of wind are included as features against the actual power measurements. Forecast is released by Europe center for medium range weather Forecast (ECMWF). All the forecasts are released twice a day.

Wind Speed, wind direction, zonal, and meridional component of surface wind are weather forecast for each of wind power at time t. These forecasts are issued on t-12, t-24, t-36 and t-48 hours. In order to select more informative feature set among the four feature sets, MI is used. Feature set having highest value of MI between the weather forecast and power are selected, while remaining feature sets are discarded. Current weather forecasts are thus dependent on previous weather forecasts and power measurements. So, in order to explore the dependency of power measurement on the previous predicted powers and associated feature set, previous forecast of last 24 hours along with associated features (selected through MI) are provided as input features to the model. Mathematically feature set for wind power forecast is expressed as:

$$S(\mathrm{t})=\begin{bmatrix} \mathrm{S(t-1),S(t-2),......,S(t-24),} \\ \mathrm{D(t),D(t-1),D(t-2),.....,D(t-24),} \\ \mathrm{ZS(t),ZS(t-1),ZS(t-2),......,ZS(t-24),} \\ \mathrm{MS(t),MS(t-1),MS(t-2),....MS(t-24),} \\ \mathrm{P(t),P(t-1),P(t-2),.....P(t-24)} \end{bmatrix} \qquad \textbf{(1)}$$

P(t), S(t), ZS(t) and MS(t) are the prediction of wind power, wind speed, and zonal, meridional component of surface wind at time t respectively, whereas t-1,t-2,………,t-24, indicate the prediction forecast at the earlier time.

2.2. Division of Dataset among Base and Meta Regressors

Fig. 2 demonstrates the division of dataset among Meta- and base-regressors. To evaluate the performance of proposed technique, five different wind farm datasets are examined. For each wind farm dataset, percentage of data among base and meta-regressor for training and testing is divided in the same manner. Effectively, 70% of data (D1) is reserved for training of base learners, where by 90% of D1 data is used to train the base learners, and 10% of D1 is used for validation purpose. After finding the optimal parameters from validation dataset then, the whole data (D1) is used to train the model according to parameters that are selected during validation step. While remaining 30% of data (D2) is divided into two parts, 2/3rd part of D2 (named as D3) is reserved for training of DBN, which is used as a meta-regressor, while 1/3rd part of D2 is used for testing of DBN. Similarly, 90% of D3 is used for training of DBN and remaining 10% is used for validation purpose. Finally after finding the suitable parameters during validation step then the whole data (D3) is used to train the model on selected parameters.

2.3. Deep Sparse Auto-encoder as a Base-regressors:

Auto-encoder is a type of neural network, which tries to copy its input to its output in an unsupervised manner. Encoding and decoding phase is involved during the training of auto-encoders. During the encoding phase, auto-encoder encodes its input $x_{en}^{'}$ in the form of encoding function as in equation 2.

$$x_{en} = f(x_{en}^{'}) \tag{2}$$

After the encoding phase next phase during training is to decode the encoded input (x_{en}) for sake of reconstructing the original input $x_{en}^{'}$ .Decoding phase of auto-encoder is represented in equation 3.

$$d_{in} = f_r(x_{en}) \tag{3}$$

However, during training if the auto-encoder only focuses on copying input to output, then it is not effective, as it over fits the training data and thus performance on the test data may not be satisfactory. By introducing sparsity as regularization term during training, the generalization capability of an auto-encoder can be enhanced. The underlying reason is that it learns some useful properties from data, which ultimately increases the generalization performance.

Basically an auto-encoder consists of input layer, output layer and a single hidden layer (which performs encoding function during training). Auto-encoder is called under complete if hidden layer neurons are less in comparison to number of neurons in input layer. During training of under complete auto-encoder, use of least Mean-Squared-Error (MSE) as loss function does not help to increase the generalization performance and only learn salient features during training phase. Loss function of sparse auto-encoder during training phase is given below

$$L = \frac{1}{N}\sum_{i'=1}^{m'}\sum_{j'=1}^{n'}(X_{i'j'} - \overline{X_{i'j'}})^2 + \lambda' * \Omega_w + \beta' * \Omega_s \quad \textbf{(4)}$$

where $\frac{1}{N'}\sum_{i'=1}^{m'}\sum_{j'=1}^{n'}(X_{i'j'} - \overline{X_{i'j'}})^2$ is the MSE, Ω_w is L2 weight regularization term and can be represented in terms of weights as.

$$\Omega_w = \frac{1}{2}\sum_{l'}^{L'}\sum_{i'}^{m'}\sum_{j'}^{n'}(W_{i'j'}{}^{l'})^2 \quad \textbf{(5)}$$

L' represents number of hidden layers, m shows the number of examples, however number of used variables in the data is denoted by n. In representation of loss function, L2 regularization helps to increase the generalization performance of auto-encoder. Another sparsity regularization term (Ω_s) used in loss function is represented mathematically as.

$$\Omega_s = \sum_{i'=1}^{s'} KL_{D'}\left(\frac{p'}{p'_{i'}}\right) = \sum_{i'=1}^{s'} p'\log\left(\frac{p'}{p'_{i'}}\right) + (1-p')\log\left(\frac{1-p'}{1-p'_{i'}}\right) \quad \textbf{(6)}$$

Sparsity regularization term tends to enforce the sparsity constraint on the output from the hidden layer. If desire value of activation for particular neuron i' is p' whereas $p'_{i'}$ is the neuron’s average activation value, whenever desire and actual value of a neuron i' have exactly same value, then sparsity regularization term will have zero value. On the other hand, there will be increase in Ω_s value, when difference between p' and $p'_{i'}$ increases. Sparsity regularization as well as Regularization term helps in acquiring optimal weights during the training phase, and ultimately generalization performance will increase. Whereas β and λ are the coefficient of sparsity regularization and weight regularization term respectively.

There are different variants of auto-encoders which have been reported widely in literature for machine learning related tasks. Sun et al. [41] used sparse auto-encoder based approach to classify faults in induction motor. Denoising auto-encoder is another variant of auto-encoder, which is also used to extract useful features from data, Lu et al. [42] proposed deep sparse denoising auto-encoder for the identification of rotatory component of machinery in order to diagnose fault.

Auto-encoders consist of single layer, but we can make it deep by individual training of auto-encoder in such a way that output of one auto-encoder acts as an input for the training of next auto-encoder (stacking auto-encoders). Hence, ones the individual auto-encoders, are trained next step is stacking of pre-trained auto-encoders to form a deep architecture, which provides good weight initialization points for stacked feed forward network. Finally supervised fine tuning of stacked feed forward network is performed using back propagation learning algorithm. There are many advantages of using deep auto-encoder over simple auto-encoder; one advantage is that a thin (but deep) architecture can reduce the total number of neurons required in comparison to a shallow (but fat) architecture for learning the same mathematical function [43]. Similarly, as number of total neurons is less thus, number of parameters reduces and consequently, less training data is needed to achieve good generalization. Owing to the importance of sparsity in deep auto-encoder, in the proposed DNN-MRT technique, nine deep sparse auto-encoders are used as base-repressors. All of these base regressors are trained by tuning some learning parameters such as number of stacked auto-encoders, maximum number of epochs, sparsity regularization term, sparsity proportion term, etc. After the training of base regressors, predictions from base regressors (along with the original data features) are provided as an input to meta-regressor.

2.3.1. Use of Transfer Learning During the Training Phase of Base-regressors

Transfer learning is a type of machine learning technique in which information gained during the training of one machine learning task is used to solve another task. Transfer learning shows remarkable results if two tasks (source and target tasks) are related to one another, otherwise performance on target task may not be encouraging. In other words, transfer learning is sharing of knowledge learnt during the training of one task with another related task. Another advantage of transfer learning is that it avoids “starting from scratch” for the training on individual tasks. Thus related task uses the information from previous task and hence takes less time for its training. Transfer learning is especially useful during the training of deep architecture, because during training of deep architecture more computational time is required as compared to shallow architecture. In case of ensemble of classifier, for which deep learning is involved in training of base and Meta learner, there too transfer learning helps to reduce the training time.

In the proposed DNN-MRT technique transfer learning is used during the training phase of base-regressors for five different wind farm datasets. Without the use of transfer learning, all nine deep auto-encoder based base regressors are needed to be trained from scratch (using pre-training and fine-tuning steps) for all wind farms. By using transfer learning approach, each deep sparse auto-encoder just pre-trained on wind farm 1. Final predictions on all five wind farm datasets are taken by fine tuning of pre-trained network, there is no need to train the deep auto-encoders from scratch for all of the wind farm datasets. In this way trained deep auto-encoders learn some useful properties from wind farm1 and later use the gained knowledge for the training of remaining wind farm datasets, this concept of transfer learning not only saves time but also improves the performance of proposed, DNN-MRT, technique. Fig. 3 shows the concept of transfer learning used during the training phase of deep auto-encoders for different wind farm datasets. Fig. 3 shows the concept of transfer learning used during the training phase of deep auto-encoders for different wind farm datasets.

2.4. Deep Belief Network as a Meta-Regressor:

After the training of base-regressors, predictions from base-regressors along with original data (on which predictions were taken) are provided as input to the meta-regressor (DBN). Basic building block of DBN is RBM. During the pre-training phase, required numbers of RBMs are trained. After the training of RBMs, next step is to stack them to form feed forward network, in which weights after individual training of RBMs act an initial point for the fine tuning (using back propagation as learning algorithm) of stacked network. The aim of meta-regressor is to improve the performance of wind power prediction as well as the generalization performance of the proposed DNN-MRT technique (due to the diversity in the prediction of individual base-regressors).

2.5. Advantages of Using Ensemble Learning Methodology

Ensemble learning is used to solve machine learning related problem in which different base-learners have their own hypothetical space. Instead of using single hypothesis space for the training of dataset, set of hypothesis spaces are combined in order to make final robust decision. There are two basic steps involve in constructing an ensemble. In the first step, base-learners are trained either in parallel or sequential manner. After that, most of the time base learners are combined using majority voting in case of classification and weighing average in case of regression. In ensemble methods, base-learners need to be accurate and diverse. In order to estimate the accuracy of base-learners different cross validation strategies are used. For attaining the diversity among the base-learners different techniques like sub sampling, manipulating the attributes, manipulating the feature space or also introducing the randomness in the learning algorithm is used [44].

Ensemble methods are not only used for classification and regression but they are also useful in clustering techniques. In general there is no such ensemble method that always out performs. Following are the advantages of using ensemble methodology in machine learning related problems.

i. Generalization capability of ensemble method is usually better than the individual base-learners.
ii. For some problems, there exists single best and unique global optimal space; however, reaching that global optimal space is difficult because running different algorithms result in sub optimal spaces. Ensemble method helps in compensation of such imperfect optimal search space; in short, ensemble method mostly gives good approximation of the search space.
iii. Ensemble methods boost the diversity in opinion.

3. Implementation Details

All the experiments of the proposed DNN-MRT techniques were carried out on a personal desktop machine, having RAM: 16 GB, 64 bit operating system, Intel(R) Core(TM) i7-33770 , processor CPU@3.4 GHz. 3.46 GHz. The operating system was windows 7 professional and Matlab 2015(b).

3.1. Parameter Setting of Deep Auto-encoder based Regressor

Tables 2-10 show parameters during the training of nine different deep auto-encoders on wind farm 1 dataset. All parameters are selected on the basis of performance of deep auto-encoders on validation part of data used during training phase. As proposed technique uses the concept of transfer learning therefore, on remaining wind farm datasets trained base regressors from wind farm1 are used in order to obtain the results from base-regressors.

3.2. Parameter Settings of Deep Belief Network based Meta-regressor

After training of the deep auto-encoders, the predictions for the test samples along with the original features are provided as an input to the DBN. For the different wind farm datasets, a separate

DBN is trained accordingly. Number of layers along with number of neurons required in each layer for each of the wind farm dataset is shown in Table 11. Parameters like batch size, number of epochs, momentum, and learning rate for all five wind farm datasets are kept the same as, shown in Table 12. All parameters are selected on the basis of performance of DBN on validation data (10% of total training data kept for training of DBN) during training phase.

3.3. Performance Evaluation Measures

For evaluating the performance of the proposed DNN-MRT technique, root mean squared error (RMSE), mean absolute error (MAE), and standard deviation of error (SDE) are used as a performance evaluation metrics. These performance matrices have following formula.

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(Wp_{actual} - Wp_{predicted}\right)^{2}} \quad \textbf{(7)}$$

$$MAE = \frac{1}{M}\sum_{i=1}^{M}\left| Wp_{actual} - Wp_{predicted} \right| \quad \textbf{(8)}$$

$$SDE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(Error - Mean\ Error\right)^{2}} \quad \textbf{(9)}$$

Wp_{actual} is the desired value of power while $Wp_{predicted}$ is the predicted power value provided by the proposed technique. Error=$Wp_{actual} - Wp_{predicted}$, while average value of error is called Mean Error.

4. Experimental Results

In the proposed DNN-MRT technique, after training of base and meta-regressors, performance measures in terms of RMSE, MAE, and SDE are evaluated on test data. To check the baseline performance on wind farm datasets, commonly used baseline regression models such as Autoregressive Integrated Moving Average (ARIMA) and SVR are used. In order to ensure fair comparison of the base line regressors performance with that of the proposed DNN-MRT technique, same percentage of test data (10% of total data) is used to evaluate the performance of baseline regressors (whereby remaining 90% of total data is used during training). During the training of baseline regressors, 10% of total training data is used as a validation data (for selecting the best parameters). The selected parameters during validation phase are used to train model using whole training data (90% of total data). Table 13 shows performances of ARIMA and SVR on test data, which depicts that the performance of SVR using linear kernel is the best among all of the baseline regressors. ARIMA is basically linear time series model, based on autoregressive degree (P), differencing degree (D), and third important parameter is moving average degree (q). Table 14 shows the parameters selected during the training phase of baseline regressors.

4.1. Performance of Base-regressors on Test Data

Table 15 illustrates the performance of base-regressors on test data. For this purpose, evaluation is carried out by training base-regressor ten times independently, and average performance along with standard deviation of error is reported against the test data. As deep auto-encoders act as base-regressors, so there is a variation in the performance in terms of error measures across each base-regressor for different wind farms. The diversity in performance by the base-regressors helps the meta-regressor to predict robust predictions.

4.2. Performance of Meta-regressor on Test Data

Performance evaluation along with standard deviation of error using ten independent runs in case of meta-regressor is shown in Table 16. Comparing Table 15 results with that of Table 16, it is shown that performance on test data by the meta-regressor is better than any of the individual base regressors. Also comparison with Amjady's et al. [35], Grassi's et al. [18] and Zameer et al. [37] techniques show that our proposed technique is better in terms of almost all evaluation metrics on training as well as on testing data. Moreover, performance of proposed DNN-MRT technique has shown better results than the performance on baseline models, presented in Table 13. Lower value of standard deviation of error against performance metrics show that the proposed technique is more robust and generalized. This is mainly because ensemble learning increases the robustness of a model, while deep learning helps to increase the representational power and is able to extract information rich features from base regressor. Comparison in terms of RMSE, MAE, and SDE of the proposed technique with that of ARIMA, SVR(linear kernel), SVM (rbf kernel),Amjady's et al. [12], Grassi's et al. [21], and Zameer et al. [37] techniques, in the form of bar charts, is presented in Fig. 4, Fig. 5, and Fig. 6, respectively. All of the three bar charts clearly depict that proposed technique is better in comparison to baseline (SVR with linear and rdb kernel) and existing techniques.

4.3. Graphical Representation of Actual and Predicted Power

Graphical representation in terms of predicted and actual power for each of the five wind farm datasets is shown in Fig. 7-11. All of the five figures depict that the predicted power values are very close to actual power, especially in case of Fig. 7 (wind farm1) and Fig. 8 (wind farm2). Overall close matching between the predicted and actual power measurements for all of the wind farms, demonstrate that our proposed method is robust. This is because predicted power is calculated by averaging the predicted powers on test data that are obtained from ten independent training of the proposed DNN-MRT technique. Also, all the wind farm datasets are closely related to each other in terms of features and task that is why "transfer of learning" based on transfer of "trained model" from wind farm1 is successful in training the base-regressors of remaining wind farms. Graphical representation shows that not only predicted power for wind farm 1 is close to the actual power, but also remaining wind power predictions are closed to the actual power, which indicates that transfer learning is effective in the proposed, DNN-MRT technique and there is no need to independently train base-regressor for each of the wind Farm.

4.4. Generalization Analysis: Performance evaluation on different Splits of training and test samples

Table 17 and 18 show the performance of proposed DNN-MRT technique on different Splits of training and test samples. Pre-trained auto-encoders from wind farm 1 are used to facilitate transfer learning in target domain (wind farm 2), but now performance of target domain is evaluated on different distribution of training and test data. Table 17 shows the performance of wind farm2 on test data in terms of MAE, SDE, and SDE, whereby 50% of data is used for the training of base-regressors, 20% of data is used for the training of meta-regressor, and 30% is kept for the testing purpose. Similarly, in another split (to check generalization of the proposed technique), MAE, SDE, and MAE against test data is shown in Table 18, but now 40% of total wind farm2 data is used for the training of base-regressor , 40% is used for the training of meta-regressor, and 20% is reserved as testing data.

Performance on different splits of training and testing data show that the proposed DNN-MRT technique offers good generalization (Table 17 and 18). It is evident that in case of the proposed technique, even stringent splits of training and test distribution (increase of unseen samples) do not show large variations in performance using the different evaluation metrics.

4.5. Statistical Test

In order to show the strength of association between the actual and predicted wind power, Pearson's correlation is evaluated against each wind farm dataset. Karl Pearson presented the idea of Pearson correlation coefficient, which shows strength of linear relationship between two variables. Zero value of Pearson's correlation indicates no relationship, whereas value of 1 indicates the perfect linear relationship between the predicted and actual wind power and negative relationship is indicated by -1 value. Let's assume actual values power measurement against specific wind farm is $\{x_{A1}, x_{A2}, x_{A3}............x_{An}\}$ and predicted power of wind according to DNN-MRT technique is $\{x_{P1}', x_{P2}', x_{P3}'..........x_{Pn}'\}$ then Pearson correlation coefficient between predicted and actual power of wind is expressed as.

$$p_{corr} = \frac{\sum_{k=1}^{n}(x_{Ak} - x_{Am})(x_{Pk}' - x_{Pm}')}{\sqrt{\sum_{i=1}^{n}(x_{Ai} - x_{Am})^2 \sum_{j=1}^{n}(xP_{Pj}' - x_{Pm}')^2}} \tag{10}$$

x_{Am} and x_{Pm}' represent mean of actual and predicted values against power of wind farms.

Table 19 shows that for all the wind farm datasets, value of correlation coefficient is close to 1, which indicates that our proposed technique has best fitted the data points. In order to evaluate significance of the Proposed DNN-MRT paired T-test is carried out. Results of the proposed technique for ten independent runs are compared with Amjady's et al. [12] technique. For all wind farm datasets, p-value is close to zero which indicates the rejection of null hypothesis, which further strengthens the improved performance of proposed techniques.

5. Conclusion

A novel wind power prediction system using deep neural network based ensemble method and the concept of transfer learning is proposed in this work. The proposed DNN-MRT model exploits an ensemble technique, whereby the concept of transfer learning is used. It is practically shown that base-regressors trained on one wind farm can be fine-tuned to work for the remaining wind farms. This has added advantage of saving extra time needed to train from scratch a new model on each of the wind farms. It is observed that prediction of base-regressors varies because of variation in meteorological properties, however due to deep learning based ensemble approach; the final results of the DNN-MRT are robust to small variations in metrological properties and are thus more generalized. The proposed technique is evaluated on datasets from five Wind Farms located at different regions. Although pattern of generated power by wind farm located at different regions may be different but the concept of transfer learning helps in efficiently and quickly predicting wind power for different wind farms. Transfer learning helps to reduce the time required for the training of wind farm dataset, so in future by using parallel processing technique, we can further reduce time for the training of base-regressors. Performance evaluation in terms of MAE, RMSE, and SDE show that our proposed technique is better than previously reported techniques.

ACKNOWLEDGMENT

This work is supported by the Higher Education Commission of Pakistan under Indigenous Ph.D. Fellowship Program. We also acknowledge Pakistan Institute of Engineering and Applied Sciences (PIEAS) for healthy research environment which lead to the research work presented in this article

References

[1] Brown BG, Katz RW, Murphy AH. Time Series Models to Simulate and Forecast Wind Speed and Wind Power. J Clim Appl Meteorol 1984;23:1184–95. doi:10.1175/1520-0450(1984)023<1184:TSMTSA>2.0.CO;2.

[2] Gomes P, Castro R. Wind Speed and Wind Power Forecasting using Statistical Models : AutoRegressive Moving Average (ARMA) and Artificial Neural Networks (ANN). Int J Sustain Energy Dev 2012;1:36–45.

[3] Sideratos G, Hatziargyriou ND. An Advanced Statistical Method for Wind Power Forecasting. IEEE Trans Power Syst 2007;22:258–65. doi:10.1109/TPWRS.2006.889078.

[4] Yuan X, Chen C, Yuan Y, Huang Y, Tan Q. Short-term wind power prediction based on LSSVM–GSA model. Energy Convers Manag 2015;101:393–401. doi:10.1016/j.enconman.2015.05.065.

[5] Li S, Wang P, Goel L. Wind Power Forecasting Using Neural Network Ensembles With Feature Selection. IEEE Trans Sustain Energy 2015;6:1447–56. doi:10.1109/TSTE.2015.2441747.

[6] Dalto M, Matusko J, Vasak M. Deep neural networks for ultra-short-term wind forecasting. 2015 IEEE Int. Conf. Ind. Technol., vol. 2015–June, IEEE; 2015, p. 1657–63. doi:10.1109/ICIT.2015.7125335.

[7] Jiang Y, Song Z, Kusiak A. Very short-term wind speed forecasting with Bayesian structural break model. Renew Energy 2013;50:637–47. doi:10.1016/j.renene.2012.07.041.

[8] Zeng J, Qiao W. Short-Term Wind Power Prediction Using a Wavelet Support Vector Machine. IEEE Trans Sustain Energy 2012;3:255–64. doi:10.1109/TSTE.2011.2180029.

[9] Liu Y, Shi J, Yang Y, Lee W-J. Short-Term Wind-Power Prediction Based on Wavelet Transform–Support Vector Machine and Statistic-Characteristics Analysis. IEEE Trans Ind Appl 2012;48:1136–41. doi:10.1109/TIA.2012.2199449.

[10] Xu Q, He D, Zhang N, Kang C, Xia Q, Bai J, et al. A Short-Term Wind Power Forecasting Approach With Adjustment of Numerical Weather Prediction Input by Data Mining. IEEE Trans Sustain Energy 2015;6:1283–91. doi:10.1109/TSTE.2015.2429586.

[11] Lee D, Baldick R. Short-Term Wind Power Ensemble Prediction Based on Gaussian Processes

and Neural Networks. IEEE Trans Smart Grid 2014;5:501–10. doi:10.1109/TSG.2013.2280649.

[12] Pinson P, Nielsen HA, Madsen H, Kariniotakis G. Skill forecasting from ensemble predictions of wind power. Appl Energy 2009;86:1326–34. doi:10.1016/j.apenergy.2008.10.009.

[13] De Giorgi MG, Ficarella A, Tarantino M. Error analysis of short term wind power prediction models. Appl Energy 2011;88:1298–311. doi:10.1016/j.apenergy.2010.10.035.

[14] Bhaskar K, Singh SN. AWNN-Assisted Wind Power Forecasting Using Feed-Forward Neural Network. IEEE Trans Sustain Energy 2012;3:306–15. doi:10.1109/TSTE.2011.2182215.

[15] Abedinia O, Amjady N. Short-term wind power prediction based on Hybrid Neural Network and chaotic shark smell optimization. Int J Precis Eng Manuf Technol 2015;2:245–54. doi:10.1007/s40684-015-0029-4.

[16] Carolin Mabel M, Fernandez E. Analysis of wind power generation and prediction using ANN: A case study. Renew Energy 2008;33:986–92. doi:10.1016/j.renene.2007.06.013.

[17] Chitsaz H, Amjady N, Zareipour H. Wind power forecast using wavelet neural network trained by improved Clonal selection algorithm. Energy Convers Manag 2015;89:588–98. doi:10.1016/j.enconman.2014.10.001.

[18] Grassi G, Vecchio P. Wind energy prediction using a two-hidden layer neural network. Commun Nonlinear Sci Numer Simul 2010;15:2262–6. doi:10.1016/j.cnsns.2009.10.005.

[19] Hong T, Pinson P, Fan S. Global Energy Forecasting Competition 2012. Int J Forecast 2014;30:357–63. doi:10.1016/j.ijforecast.2013.07.001.

[20] Lei M, Shiyan L, Chuanwen J, Hongling L, Yan Z. A review on the forecasting of wind speed and generated power. Renew Sustain Energy Rev 2009;13:915–20. doi:10.1016/j.rser.2008.02.002.

[21] Bao Y, Wang H, Wang B. Short-term wind power prediction using differential EMD and relevance vector machine. Neural Comput Appl 2014;25:283–9. doi:10.1007/s00521-013-1482-z.

[22] Costa A, Crespo A, Navarro J, Lizcano G, Madsen H, Feitosa E. A review on the young history of the wind power short-term prediction. Renew Sustain Energy Rev 2008;12:1725–44. doi:10.1016/j.rser.2007.01.015.

[23] Foley AM, Leahy PG, Marvuglia A, McKeogh EJ. Current methods and advances in forecasting

of wind power generation. Renew Energy 2012;37:1–8. doi:10.1016/j.renene.2011.05.033.

[24] Tascikaraoglu A, Uzunoglu M. A review of combined approaches for prediction of short-term wind speed and power. Renew Sustain Energy Rev 2014;34:243–54. doi:10.1016/j.rser.2014.03.033.

[25] Zhong-Qiang W, Wen-Jing J, Li-Ru Z, Chang-Han W. Maximum wind power tracking for {PMSG} chaos systems – {ADHDP} method. Appl Soft Comput 2015;36:204–9. doi:http://dx.doi.org/10.1016/j.asoc.2015.07.024.

[26] Peng H, Liu F, Yang X. A hybrid strategy of short term wind power prediction. Renew Energy 2013;50:590–5. doi:10.1016/j.renene.2012.07.022.

[27] Pinson P, Madsen H, Papaefthymiou G, Klöckl B. From Probabilistic Forecasts to Wind Power Production. Wind Energy 2009;12:51–62. doi:10.1002/we.284.

[28] Nielsen HA, Madsen H, Nielsen TS. Using quanti le regression to extend an existing wind power forecasting system With probabilistic forecasts. Wind Energy 2006;9:95–108. doi:10.1002/we.180.

[29] Ortiz-garcı EG, Portilla-figueras A, Salcedo-sanz S, Prieto L, Correoso F. Neurocomputing Accurate short-term wind speed prediction by exploiting diversity in input data using banks of artificial neural networks 2009;72:1336–41. doi:10.1016/j.neucom.2008.09.010.

[30] Troncoso A, Salcedo-sanz S, Casanova-mateo C, Riquelme JC, Prieto L. Local models-based regression trees for very short-term wind speed prediction. Renew Energy 2015;81:589–98. doi:10.1016/j.renene.2015.03.071.

[31] Ortiz-garcía EG, Salcedo-sanz S, Pérez-bellido ÁM, Gascón-moreno J, Portilla-figueras JA, Prieto L. Short-term wind speed prediction in wind farms based on banks of support vector machines 2011:193–207. doi:10.1002/we.

[32] Zhang W, Wang J, Wang J, Zhao Z, Tian M. Short-term wind speed forecasting based on a hybrid model. Appl Soft Comput J 2013;13:3225–33. doi:10.1016/j.asoc.2013.02.016.

[33] Ma X, Jin Y, Dong Q. A generalized dynamic fuzzy neural network based on singular spectrum analysis optimized by brain storm optimization for short-term wind speed forecasting. Appl Soft Comput J 2017;54:296–312. doi:10.1016/j.asoc.2017.01.033.

[34] Jung J, Broadwater RP. Current status and future advances for wind speed and power forecasting. Renew Sustain Energy Rev 2014;31:762–77. doi:10.1016/j.rser.2013.12.054.

[35] Amjady N, Keynia F, Zareipour H. Wind Power Prediction by a New Forecast Engine Composed of Modified Hybrid Neural Network and Enhanced Particle Swarm Optimization. IEEE Trans Sustain Energy 2011;2:265–76. doi:10.1109/TSTE.2011.2114680.

[36] Wang JJ, Zhang W, Li Y, Wang JJ, Dang Z. Forecasting wind speed using empirical mode decomposition and Elman neural network. Appl Soft Comput 2014;23:452–9. doi:10.1016/j.asoc.2014.06.027.

[37] Zameer A, Arshad J, Khan A, Raja MAZ. Intelligent and robust prediction of short term wind power using genetic programming based ensemble of neural networks. Energy Convers Manag 2017;134:361–72.

[38] Hu Q, Zhang R, Zhou Y. Transfer learning for short-term wind speed prediction with deep neural networks. Renew Energy 2016;85:83–95. doi:10.1016/j.renene.2015.06.034.

[39] Qiu X, Ren Y, Suganthan PN, Amaratunga GAJ. Empirical Mode Decomposition based Ensemble Deep Learning for Load Demand Time Series Forecasting. Appl Soft Comput 2017;54:246–55. doi:10.1016/j.asoc.2017.01.015.

[40] Wang H, Li G, Wang G, Peng J, Jiang H, Liu Y. Deep learning based ensemble approach for probabilistic wind power forecasting. Appl Energy 2017;188:56–70. doi:10.1016/j.apenergy.2016.11.111.

[41] Sun W, Shao S, Zhao R, Yan R, Zhang X, Chen X. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. Measurement 2016;89:171–8. doi:10.1016/j.measurement.2016.04.007.

[42] Lu C, Wang Z, Qin W, Ma J. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identi fi cation. Signal Processing 2017;130:377–88. doi:10.1016/j.sigpro.2016.07.028.

[43] Schulz H, Behnke S. Deep Learning. KI - Künstliche Intelligenz 2012;26:357–63. doi:10.1007/s13218-012-0198-z.

[44] Choraś M, Bhanu B, Chen H, Champod C, Komatsu N, Nakano M, et al. Ensemble Learning. Encycl. Biometrics, Boston, MA: Springer US; 2009, p. 270–3. doi:10.1007/978-0-387-73003-

5_293.

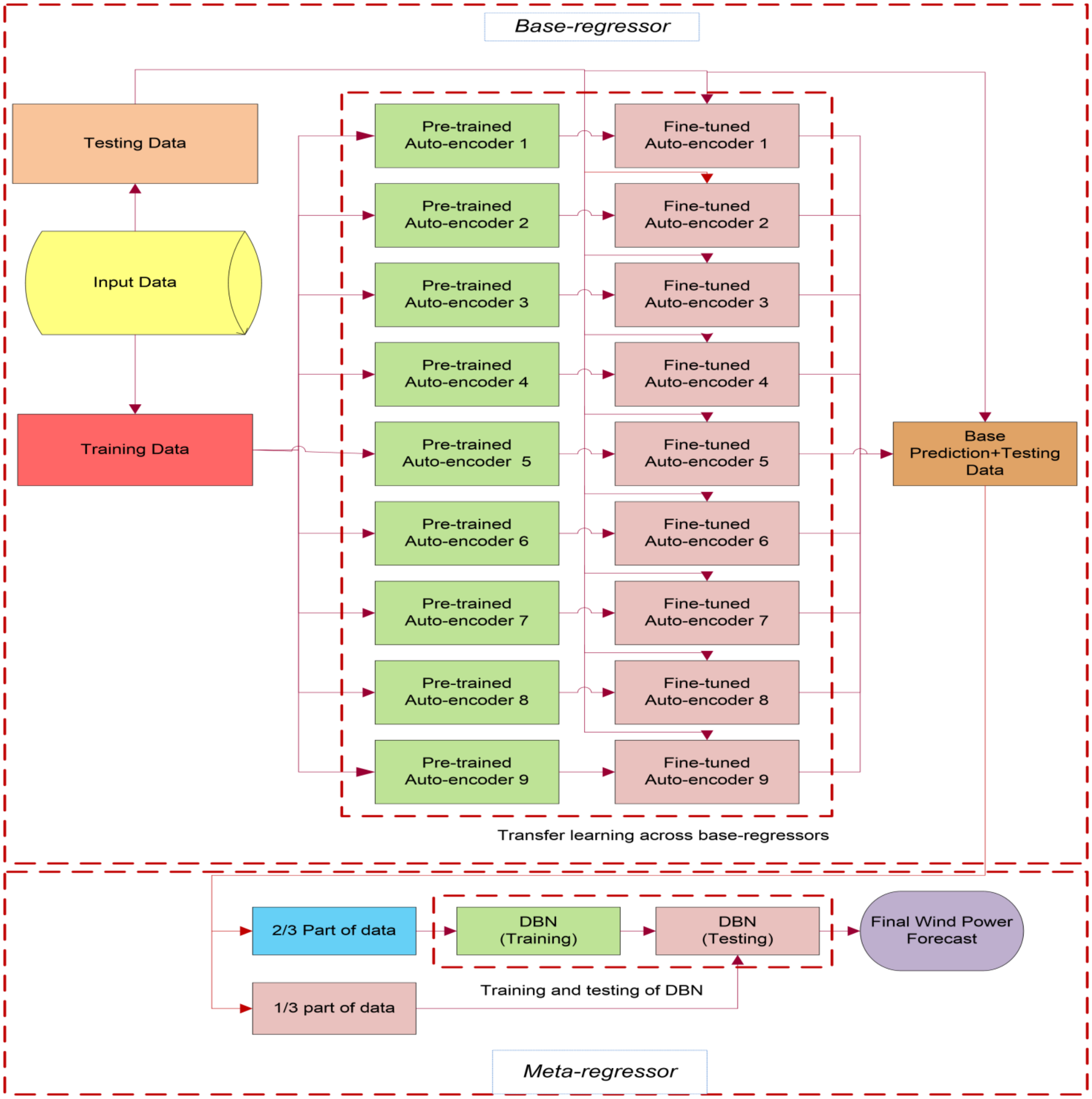


Figure 1: Overall work flow diagram depicting all stages used in proposed DNN-MRT technique

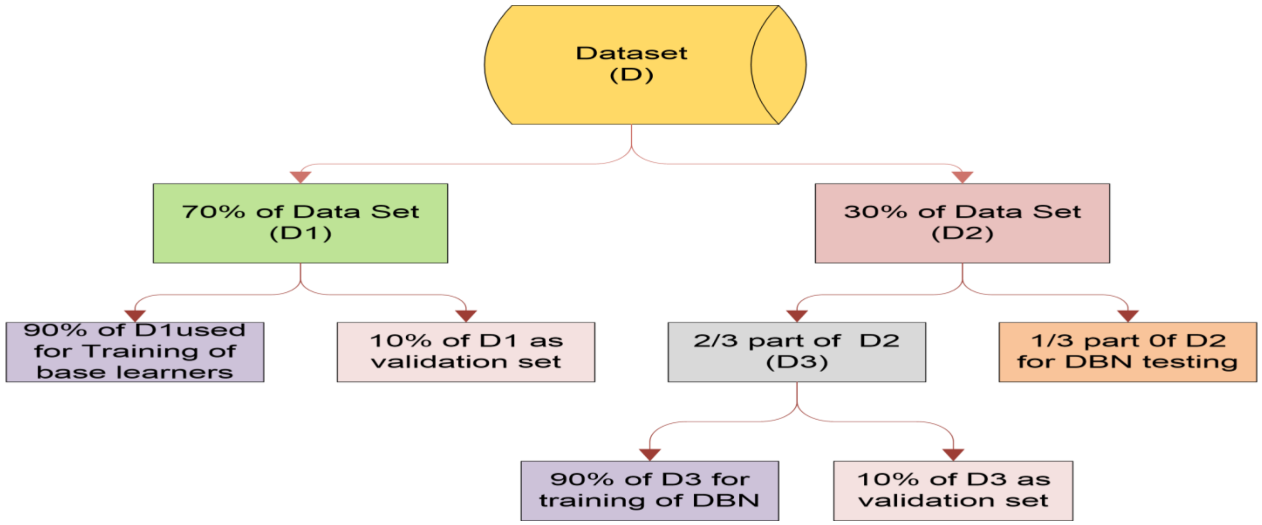


Figure 2: Dataset Distribution

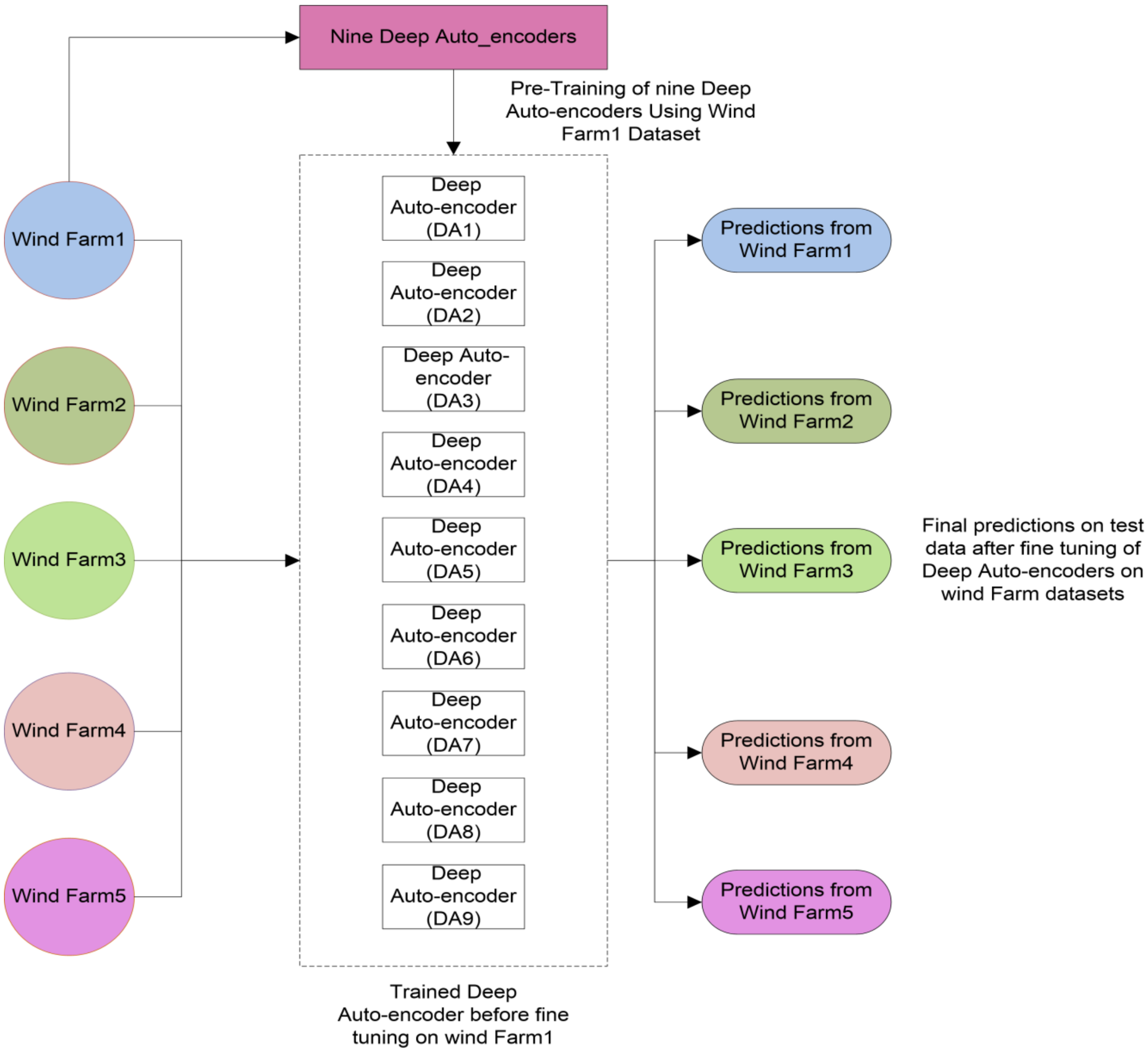


Figure 3: Transfer learning architecture for training of base-regressors of different wind farm Dataset

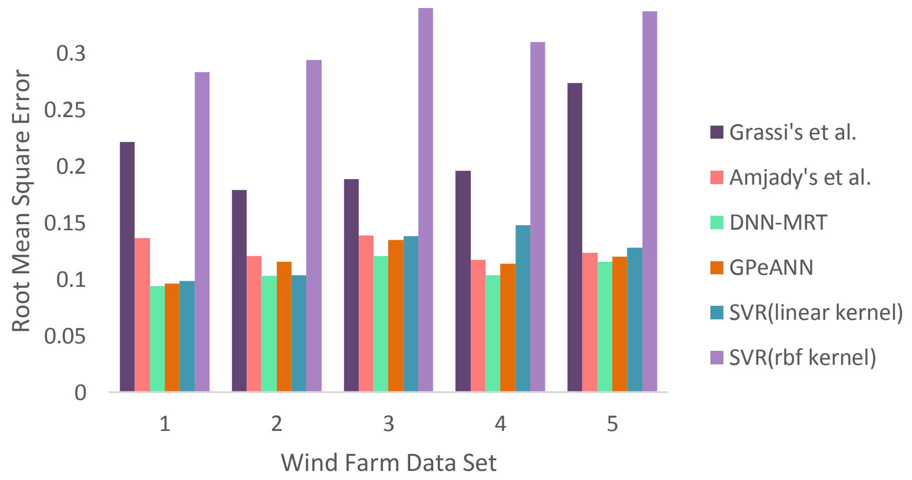


Figure4: Comparison of proposed DNN-MRT in term of RMSE with existing techniques

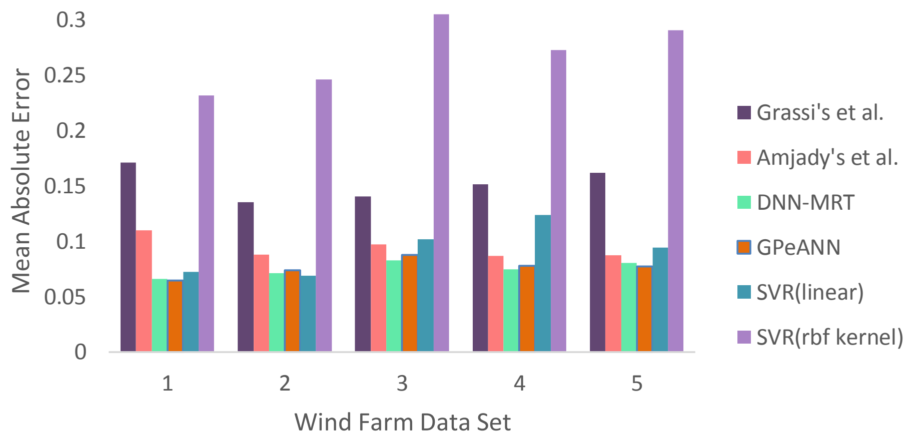


Figure5: Comparison of proposed DNN-MRT in term of MAE with existing techniques

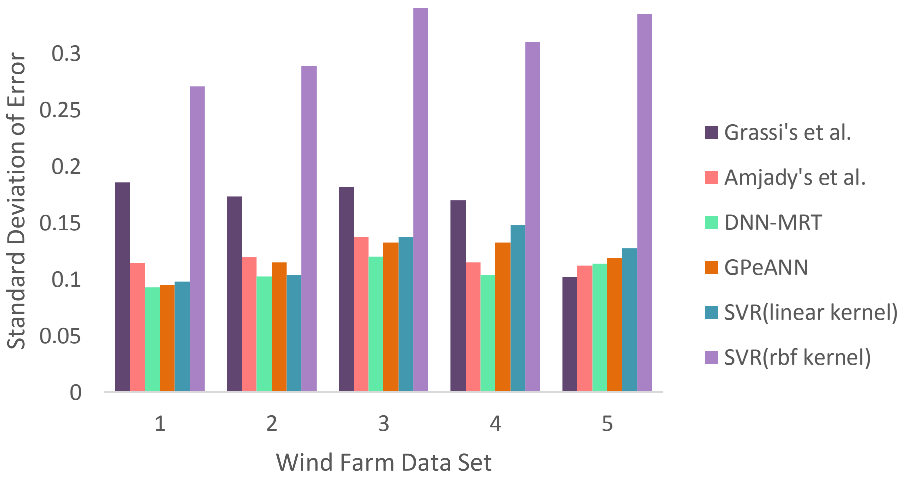


Figure 6: Comparison of proposed DNN-MRT in term of SDE with existing techniques

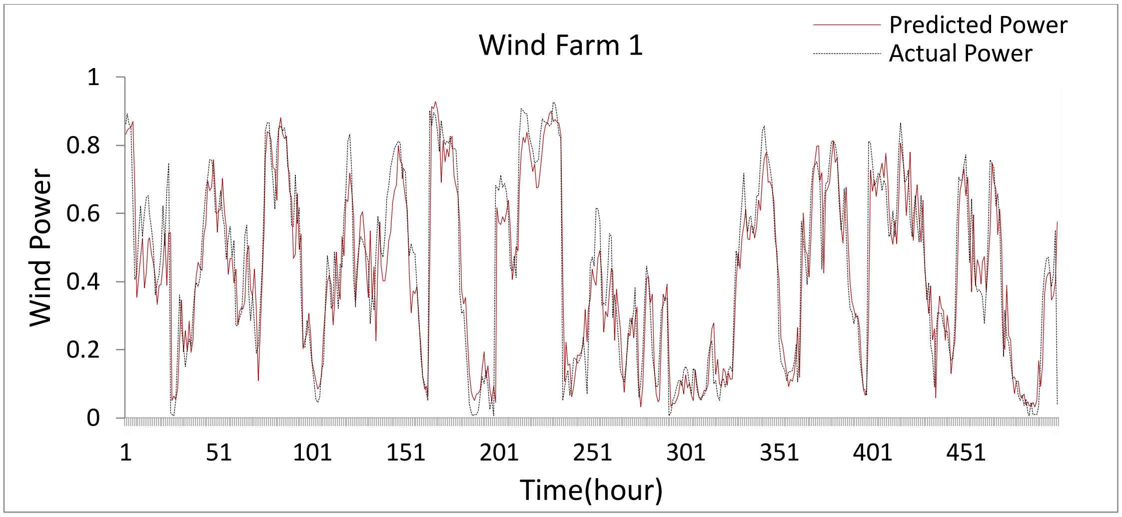


Figure7: Comparison btween actual and predicted power for Wind Farm 1

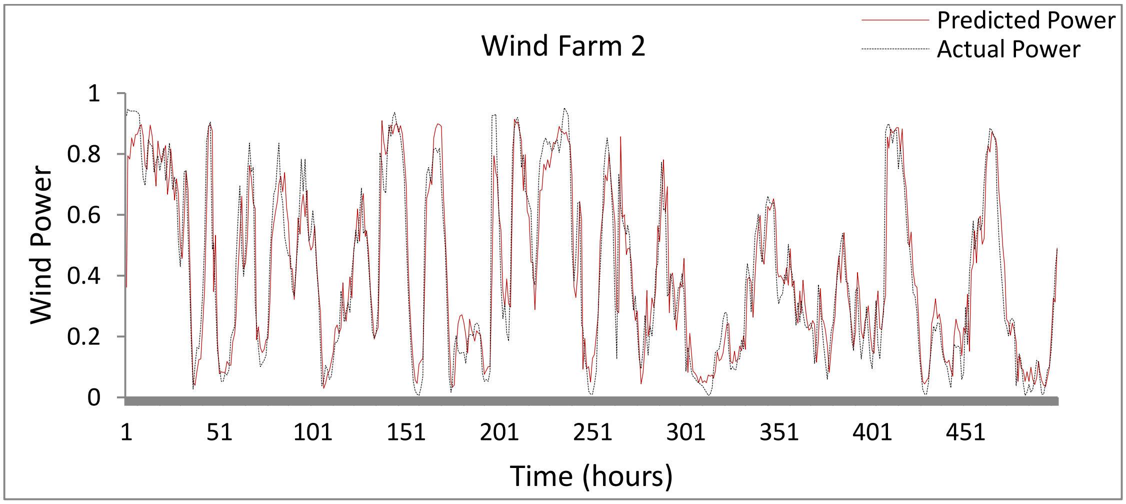


Figure8: Comparison of Actual and Predicted power for Wind Farm 2

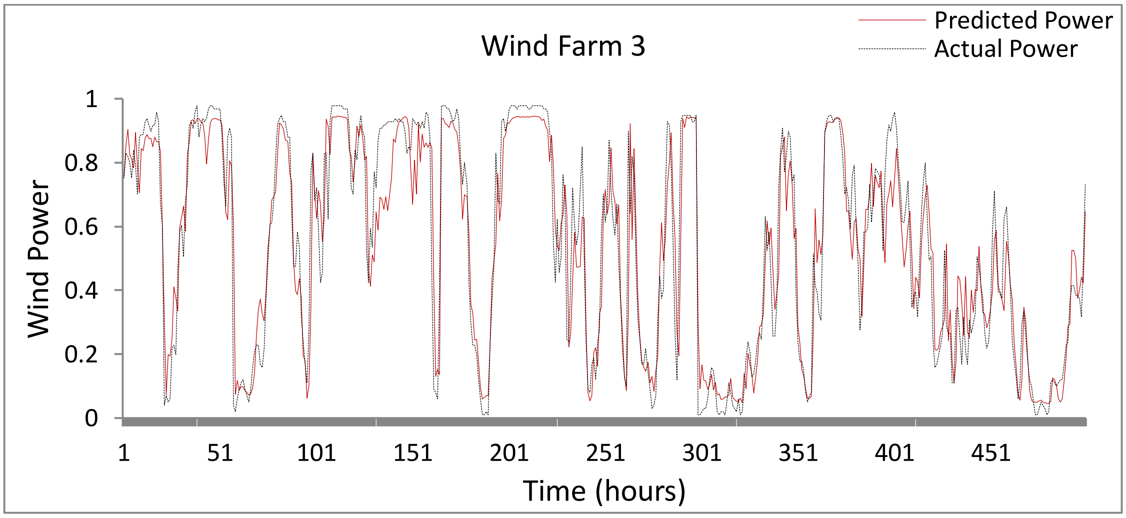


Figure9: Comparison between Actual and Predicted Power for Wind Farm 3

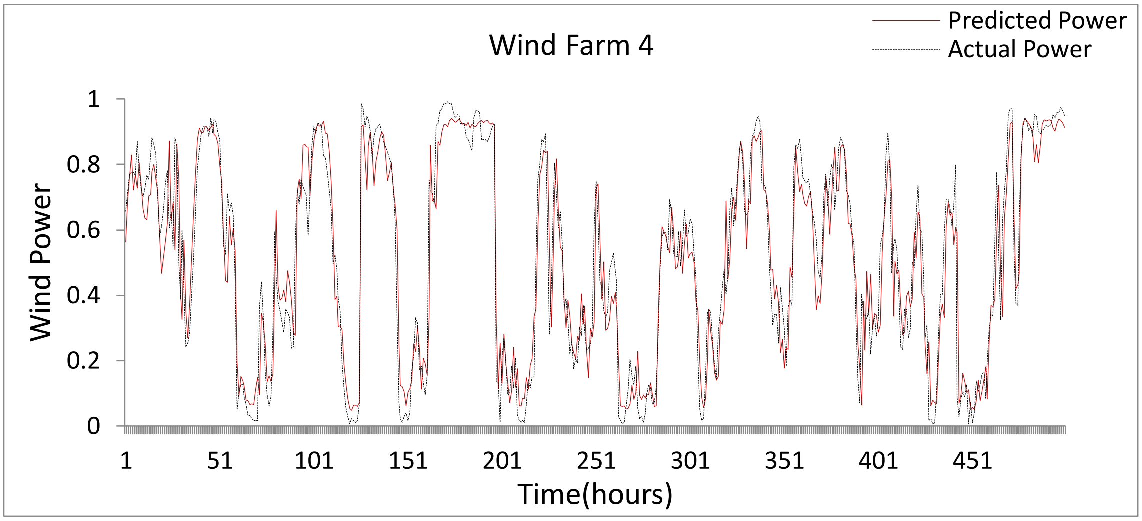


Figure 10: Comparison of Actual and Predicted Power for Wind Farm 4

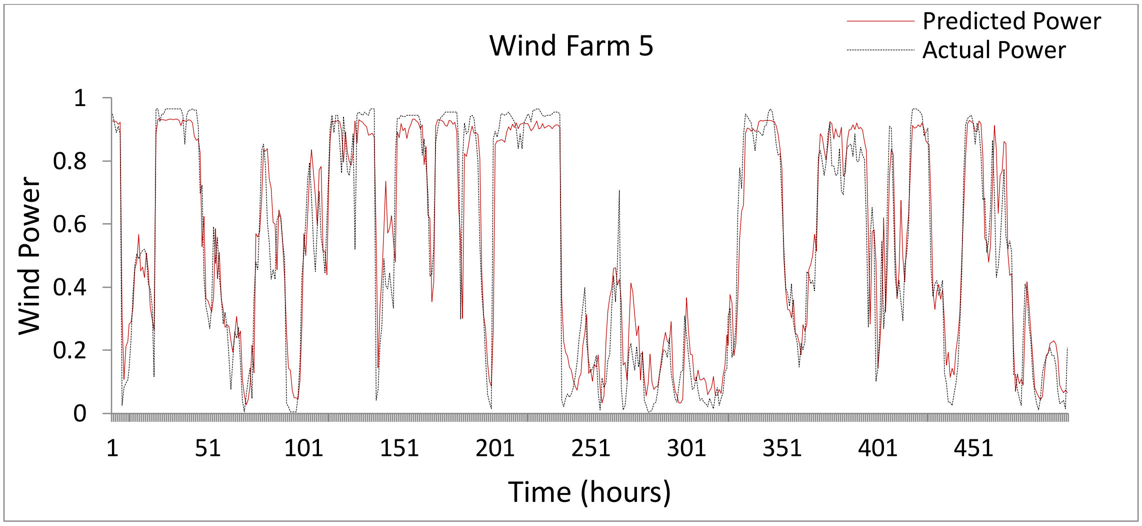


Figure 11 Comparison of Actual and Predicted Power for Wind Farm 5

Table 1: List of Abbreviation

DNN-MRT	Deep Neural Network based Meta Regression and Transfer Learning
ANN	Artificial Neural Network
GSA	Gravitational Search Algorithms
LSSVM	Least Square Support Vector Machine
EMD	Empirical Model Decomposition
RVM	Relevance Vector Machine
MHNN	Modified Hybrid Neural Network
LM	Levenberg Marquardr
PSO	Particle Swarm Optimization
BFGS	Broyden Fletcher-Goldfarb-Shanno
BR	Baysian Regularization
ZS	Zonal Component of Surface Wind
MS	Meridional Component of Surface Wind
S	Speed of Wind
D	Direction of Wind
ECMWF	Europe Center for Medium Range Weather Forecast
MI	Mutual Information
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
SDE	Standard Deviation of Error
MAE	Mean Absolute Error
RBM	Restricted Boltzmann Machine
DBN	Deep Belief Network
CD	Contractive Divergence Theorem
ARIMA	Autoregressive Integrated Moving Average
SVR	Support Vector Regressor

Table 2:Parameter Setting of Auto-encoder 1

Layer	No of Neurons	Maximum Epoch	L2 Weight Regularization	Sparsity Regularization	Sparsity Proportion
1	250	500	.00003	4	.15
2	220	250	.00001	4	.1
3	200	200	.00001	4	.1
4	175	175	.00001	4	.1
5	150	150	.00001	4	.1
6	100	150	.00001	4	.1
7	75	200	.00002	4	.1
8	50	200	.00002	4	.1

Table 3:Parameter Setting of Auto-encoder 2

Layer	No of Neurons	Maximum Epoch	L2 Weight Regularization	Sparsity Regularization	Sparsity Proportion
1	250	500	.00003	4	.15
2	220	250	.00001	4	.1
3	200	200	.00001	4	.1
4	175	175	.00001	4	.1
5	150	150	.00001	4	.1
6	100	150	.00001	4	.1
7	75	200	.00002	4	.1
8	50	200	.00002	4	.1
9	20	200	.00002	4	.1

Table 4:Parameter Setting of Auto-encoder 3

Layer	No of Neurons	Maximum Epoch	L2 Weight Regularization	Sparsity Regularization	Sparsity Proportion
1	30	500	.00003	4	.15
2	5	250	.00001	4	.1

Table 5:Parameter Setting of Auto-encoder 4

Layer	No of Neurons	Maximum Epoch	L2 Weight Regularization	Sparsity Regularization	Sparsity Proportion
1	300	500	.00003	4	.15
2	250	250	.00001	4	.1
3	220	200	.00001	4	.1
4	205	175	.00001	4	.1
5	140	150	.00001	4	.1
6	90	150	.00001	4	.1
7	60	200	.00002	4	.1
8	55	200	.00002	4	.1
9	30	200	.00002	4	.1
10	15	200	.00002	4	.1

Table 6:Parameter Setting of Auto-encoder 5

Layer	No of Neurons	Maximum Epoch	L2 Weight Regularization	Sparsity Regularization	Sparsity Proportion
1	300	500	.00003	4	.15
2	220	250	.00001	4	.1
3	200	200	.00001	4	.1
4	175	175	.00001	4	.1
5	150	150	.00001	4	.1
6	100	150	.00001	4	.1
7	75	200	.00002	4	.1
8	50	200	.00002	4	.1
9	20	200	.00002	4	.1

Table 7:Parameter Setting of Auto-encoder 6

Layer	No of Neurons	Maximum Epoch	L2 Weight Regularization	Sparsity Regularization	Sparsity Proportion
1	160	500	.00004	4	.15
2	220	250	.00001	4	.1
3	200	200	.00001	4	.1
4	175	175	.00001	4	.1
5	150	150	.00001	4	.1
6	100	150	.00001	4	.1
7	75	200	.00002	4	.1
8	50	200	.00002	4	.1
9	20	200	.00002	4	.1

Table 8:Parameter Setting of Auto-encoder 7

Layer	No of Neurons	Maximum Epoch	L2 Weight Regularization	Sparsity Regularization	Sparsity Proportion
1	200	500	.00004	4	.15
2	150	250	.00001	4	.1
3	120	200	.00001	4	.1
4	100	175	.00001	4	.1
5	75	150	.00001	4	.1
6	50	150	.00001	4	.1
7	25	250	.00001	4	.1

Table 9:Parameter Setting of Auto-encoder 8

Layer	No of Neurons	Maximum Epoch	L2 Weight Regularization	Sparsity Regularization	Sparsity Proportion
1	150	500	.00004	4	.15
2	100	275	.00001	4	.1
3	70	275	.00001	4	.1
4	50	275	.00001	4	.1
5	5	275	.00001	4	.1

Table 10:Parameter Setting of Auto-encoder 9

Layer	No of Neurons	Maximum Epoch	L2 Weight Regularization	Sparsity Regularization	Sparsity Proportion
1	200	500	.00004	4	.15
2	150	250	.00001	4	.1
3	120	200	.00001	4	.1
4	100	175	.00001	4	.1
5	75	150	.00001	4	.1
6	50	150	.00001	4	.1

Table 11: Parameter setting of DBN for each layer of wind farm dataset

Wind Farm1	Number of neuron in layer 1	80
	Number of neuron in layer 2	50
	Number of neuron in layer 3	20
	Number of neuron in layer 4	5
Wind Farm2	Number of neuron in layer 1	280
	Number of neuron in layer 2	80
	Number of neuron in layer 3	50
	Number of neuron in layer 4	20
	Number of neuron in layer 5	5
Wind Farm3	Number of neuron in layer 1	370
	Number of neuron in layer 2	80
	Number of neuron in layer 3	50
	Number of neuron in layer 4	20
	Number of neuron in layer 5	5
Wind Farm4	Number of neuron in layer 1	355
	Number of neuron in layer 2	80
	Number of neuron in layer 3	50
	Number of neuron in layer 4	20
	Number of neuron in layer 5	5
Wind Farm5	Number of neuron in layer 1	250
	Number of neuron in layer 2	80
	Number of neuron in layer 3	50
	Number of neuron in layer 4	20
	Number of neuron in layer 5	5

Table 12: General Parameter Setting of DBN for all the wind farm dataset

Number of Epochs	**250**
Batch size	10
Momentum	.05
Learning Rate	.001

Table 13: Baseline performance on five wind farm datasets using ARIMA and SVR

	ARIMA			**SVR (linear kernel)**			**SVR (rbf kernel)**		
	MAE	SDE	RMSE	MAE	SDE	RMSE	MAE	SDE	RMSE
Wind farm1	0.4470	0.3776	0.5410	0.0722	0.0982	0.0988	0.2319	0.2710	0.2838
Wind farm2	0.4432	0.3956	0.5543	0.0688	0.1040	0.1040	0.2463	0.2894	0.2941
Wind farm3	0.5711	0.4929	0.6832	0.1019	0.1380	0.1384	0.3047	0.3415	0.3520
Wind farm4	0.4897	0.4433	0.6117	0.1234	0.1471	0.1479	0.2724	0.3100	0.3101
Wind farm5	0.5071	0.4589	0.6178	0.0943	0.1275	0.1282	0.2906	0.3354	0.3377

Table 14: Parameters setting of baseline regressors

	SVM					**ARIMA**		
	Epsilon		**Value of C**		**Value of sigma**			
	linear kernel	rbf kernel	linear kernel	rbf kernel	rbf kernel	P (Degree of auto-regression polynomial)	D (Degree of non-seasonal integration)	Q (Degree of non-seasonal moving average performance)
Wind farm1	0.1	0.1	1	0.2824	1	1	1	7
Wind farm2	0.1	0.1	0.1	0.2824	0.5	2	1	1
Wind farm3	0.1	0.1	1	0.2824	0.5	3	1	1
Wind farm4	0.1	0.2	0.3	0.2824	0.5	1	1	15
Wind farm5	0.1	0.2	1	0.2824	1	2	1	3

Table 15: Performance in terms of MAE, SDE, and RMSE for base-regressors (Auto-encoder) on test data

		Auto-1	**Auto-2**	**Auto-3**	**Auto-4**	**Auto-5**	**Auto-6**	**Auto-7**	**Auto-8**	**Auto-9**
Wind Farm1	MAE	0.0630 ±.0012	0.0625 ±.0004	0.0632 ±.0005	0.0622 ±.0004	0.0630 ±.0012	0.0643 ±.0022	0.0627 ±.0005	0.0654 ±.0009	0.0639 ±.0025
	SDE	0.0944 ±.0011	0.0937 ±.0004	0.0942 ±.0005	0.0939 ±.0005	0.0944 ±.0011	0.0949 ±.0016	0.0939 ±.0004	0.0957 ±.0009	0.0955 ±.0025
	RMSE	0.0947 ±.0011	0.0940 ±.0004	0.0946 ±.0005	0.0942 ±.0005	0.0947 ±.0011	0.0955 ±.0016	0.0942 ±.0004	0.0960 ±.0009	0.0953 ±.0023
Wind Farm2	MAE	0.6840 ±.0008	0.0681 ±.0008	0.0695 ±.0011	0.0680 ±.0004	0.0678 ±.0003	0.0701 ±.0041	0.0684 ±.0009	0.0713 ±.0010	0.0695 ±.0031
	SDE	0.1103 ±.0007	0.1099 ±.0006	0.1113 ±.0007	0.1101 ±.0005	0.1100 ±.0003	0.1117 ±.0038	0.1101 ±.0008	0.1123 ±.0012	0.1112 ±.0031
	RMSE	0.1103 ±.0007	0.1099 ±.0006	0.1113 ±.0007	0.1101 ±.0005	0.1100 ±.0003	0.1117 ±.0039	0.1101 ±.0007	0.1124 ±.0011	0.1112 ±.0031
Wind Farm3	MAE	0.0799 ±.0023	0.0787 ±.0005	0.0800 ±.0009	0.0786 ±.0005	0.0785 ±.0004	0.0802 ±.0015	0.0800 ±.0015	0.0845 ±.0013	0.0804 ±.0025
	SDE	0.1234 ±.0018	0.1221 ±.0004	0.1223 ±.0008	0.1222 ±.0004	0.1220 ±.0003	0.1232 ±.0011	0.1226 ±.0008	0.1274 ±.0016	0.1225 ±.0008
	RMSE	0.1238 ±.0019	0.1225 ±.0004	0.1227 ±.0008	0.1226 ±.0005	0.1223 ±.0004	0.1236 ±.0011	0.1228 ±.0009	0.1275 ±.0015	0.1235 ±.0020
Wind Farm4	MAE	0.0718 ±.0012	0.0718 ±.0013	0.0737 ±.0014	0.0713 ±.0006	0.0715 ±.0007	0.0664 ±.0209	0.0716 ±.0008	0.0754 ±.0011	0.0729 ±.0024
	SDE	0.1057 ±0008	0.1054 ±.0007	0.1057 ±.0016	0.1055 ±.0004	0.1052 ±.0004	0.1063 ±.0014	0.1053 ±.0006	0.1087 ±.0014	0.1056 ±.0009
	RMSE	0.1061 ±.0008	0.1059 ±.0008	0.1072 ±.0013	0.1059 ±.0005	0.1057 ±.0005	0.1068 ±.0015	0.1057 ±.0007	0.1092 ±.0014	0.1071 ±.0027
Wind Farm5	MAE	0.0750 ±.0019	0.0741 ±.0005	0.0758 ±.0008	0.0737 ±.0005	0.0736 ±.0005	0.0762 ±.0030	0.0740 ±.0009	0.0783 ±.0017	0.0755 ±.0026
	SDE	0.1190 ±.0015	0.1179 ±.0004	0.1182 ±.0012	0.1188 ±.0006	0.1181 ±.0004	0.1197 ±.0020	0.1178 ±.0005	0.1220 ±.0021	0.1194 ±.0020
	RMSE	0.1194 ±.0015	0.1182 ±.0005	0.1186 ±.0012	0.1191 ±.0006	0.1185 ±.0004	0.1203 ±.0025	0.1182 ±.0006	0.1223 ±.0020	0.1198 ±.0022

Table 16: Comparison of proposed technique in terms of MAE, SDE, and RMSE with Amjady,s et al. [35],Grassi's et al. [18], and Zameer et al. [37]

		Grassi Model	Amjady Model	GPeANN	DNN-MRT	Grassi Model	Amjady Model	GPeANN	DNN-MRT
		Training Data				Testing Data			
Wind Farm1	MAE	0.1256	0.1143	0.0575	**0.0508 ±.0002**	0.1708	0.1100	0.0643	**0.0658 ±.0003**
	SDE	0.1621	0.0988	0.0868	**0.0719 ±.0004**	0.1860	0.1144	0.095	**0.0929 ±.0005**
	RMSE	0.1621	0.1335	0.0869	**0.0721 ±.0004**	0.2215	0.1365	0.0966	**0.0939 ±.0005**
Wind Farm2	MAE	0.1309	0.0797	0.0623	**0.0564 ±.0002**	0.1352	0.0879	0.0739	**0.0713 ±.0007**
	SDE	0.1743	0.1074	0.0957	**0.0782 ±.0012**	0.1738	0.1195	0.1152	**0.1025 ±.0013**
	RMSE	0.1743	0.1048	0.0975	**0.0809 ±.0022**	0.1792	0.1206	0.1157	**0.1032 ±.0012**
Wind Farm3	MAE	0.1242	0.0835	0.0694	**0.0607 ±.0007**	0.1405	0.0972	0.0874	**0.0825 ±.0005**
	SDE	0.1672	0.1070	0.1071	**0.0897 ±.0010**	0.1821	0.1378	0.1329	**0.1203 ±.0003**
	RMSE	0.1672	0.1100	0.1071	**0.0903 ±.0003**	0.1888	0.1388	0.135	**0.1207 ±.0002**
Wind Farm4	MAE	0.1251	0.0835	0.0681	**0.0603 ±.0003**	0.1516	0.0867	0.078	**0.0748 ±.0002**
	SDE	0.1618	0.1074	0.1061	**0.0827 ±.0002**	0.1703	0.1153	0.1118	**0.1036 ±.0002**
	RMSE	0.1618	0.1074	0.1061	**0.0838 ±.0003**	0.1963	0.1174	0.1118	**0.1036 ±.00019**
Wind Farm5	MAE	0.1154	0.0997	0.0715	**0.0705 ±.0003**	0.1616	0.0872	0.077	**0.0804 ±.0008**
	SDE	0.1603	0.1189	0.1135	**0.0952 ±.0004**	0.1020	0.1125	0.119	**0.1137 ±.0007**
	RMSE	0.1603	0.1355	0.1135	**0.1008 ±.0006**	0.2124	0.1236	0.1203	**0.1156 ±.0009**

Table 17:Performance of Proposed technique on 30% of total wind farm2 data as test data

		Auto-1	Auto-2	Auto-3	Auto-4	Auto-5	Auto-6	Auto-7	Auto-8	Auto-9	DBN
Wind Farm2	MAE	0.0679 ±.0135	0.0694 ±.0180	0.0669 ±.0045	0.0680 ±.0153	0.0680 ±.0150	0.0687 ±.0123	0.0681 ±.0146	0.0709 ±.0097	0.0641 ±.0020	0.0626 ±.0003
	SDE	0.1068 ±.0155	0.1090 ±.0225	0.1047 ±.0036	0.1069 ±.0179	0.1075 ±.0187	0.1069 ±.0139	0.1070 ±.0174	0.1092 ±.0122	0.1022 ±.0020	0.0968 ±.0002
	RMSE	0.1069 ±.0157	0.1091 ±.0227	0.1048 ±.0037	0.1069 ±.0180	0.1075 ±.0189	0.1070 ±.0140	0.1071 ±.0176	0.1093 ±.0124	0.1022 ±.0020	0.0988 ±.0003

Table 18: Performance of Proposed technique on 20% of total wind farm2 data as test data

		Auto-1	Auto-2	Auto-3	Auto-4	Auto-5	Auto-6	Auto-7	Auto-8	Auto-9	DBN
Wind Farm2	MAE	0.0680 ±.0172	0.0675 ±.0171	0.0655 ±.0052	0.0674 ±.0179	0.0679 ±.0190	0.0692 ±.0196	0.0683 ±.0195	0.0739 ±.0175	0.0628 ±.0017	0.0647 ±.0002
	SDE	0.1038 ±.0215	0.1038 ±.0222	0.1006 ±.0049	0.1034 ±.0222	0.1042 ±.0242	0.1054 ±.0245	0.1045 ±.0248	0.1104 ±.0226	0.0972 ±.0014	0.0966 ±.0003
	RMSE	0.1039 ±.0215	0.1039 ±0.0222	0.1007 ±.0049	0.1035 ±.0222	0.1043 ±.0242	0.1054 ±.0245	0.1045 ±.0248	0.1108 ±.0226	0.0973 ±.0014	0.0968 ±.0002

Table 19: Statistical Results of Proposed (DNN-MRT) Technique

Dataset	Hypothesis test	P-value	Pearson correlation
Wind Farm1	1	1.1577e-276	0.9402
Wind Farm2	1	2.6732e-06	0.9358
Wind Farm3	1	8.8086e-38	0.9306
Wind Farm4	1	8.8650e-40	0.9428
Wind Farm5	1	1.3059e-10	0.9447