Continuous RBM Based Deep Neural Network for Wind Speed Forecasting in Hong Kong

Yanxing Hu
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
Email: csyhu@comp.polyu.edu.hk

Jane You
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
Email: csyjia@comp.polyu.edu.hk

James N.K. Liu
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
Email: jame.liu@polyu.edu.hk

Pak Wai Chan
Hong Kong Observatory
134A Nathan Road, Kowloon, Hong
Email: pwchan@hko.gov.hk

***Abstract*—The wind speed forecasting in Hong Kong is more difficult than in other places in the same latitude for two reasons: the great affect from the urbanization of Hong Kong in the long term, and the very high wind speeds brought by the tropical cyclones. Therefore, prediction model with higher learning ability is in need for the wind speed forecast in Hong Kong. In this paper, we try to employ the Deep Neural Network (DNN) to solve the time series problem of wind speed forecasting in Hong Kong since it is believed that Neural Network (NN) with deep architectures can provide higher learning ability than shallow NN model. Especially, in our paper, we use the continuous Restricted Boltzmann Machine (CRBM) to build the network architecture of the DNN. The CRBM is the continuous valued version of the classical binary valued Restricted Boltzmann Machine (RBM). Compared with the Stacked Auto-Encoder (SAE) model applied in our previous study, this CRBM model is more generative, and therefore more suitable for simulating the data in wind speed domain.**

In our research, we employ the DNN to process the massive wind speed data involving millions of hourly records provided by The Hong Kong Observatory (HKO)[1]. The results show that the applied approach is able to provide a better features space for computational models in wind speed data domain, and this approach is also a new potential tool for the feature fusion of continuous valued time series problems.

***Keywords*—Deep Neural Network, Continuous Restricted Boltzmann Machine, Wind Speed Forecasting, Feature Representation**

I. Introduction

Wind speed forecasting has great significance not only in atmospheric related area but also in every aspect of people's life [1]. e.g., in the wind energy industry the forecasting of wind speed can guide the selection of the site position [2]; Engineers frequently utilize information based on wind speed forecasts in the design and construction of large wind-resistantstructures such as bridges, high-rise buildings, and off-shore oil platforms [3]; even in financial markets, wind speed forecasts also play a critical role as weather derivatives and the need to manage weather-related risks, including wind risk and grows[4]. Therefore, the academical and practical value of efficient wind speed forecast approach is obvious.

Currently, there are mainly two families of approaches employed on wind speed forecasting problem: using the numerical models and using the Computational Intelligent(CI) models. Different from numerical models that are too dependent on the psychical restrictive conditions[5], the advantage of using the CI models is that the CI models can "learn" the disciplines from the historical information itself in a statistical manner. One of the mainstream ideas of using the CI models on forecasting is to apply the Neural Networks (NNs) to deal with the given time series data. NNs can recognize the hidden patterns or relationships from the historical observations, meanwhile, additional advantages of the NN approach over the numerical models include data error tolerance, ease of adaptability to online measurements,etc. [6].

On the other hand, in the very recent years, theories about NNs and learning systems have experienced a fast development. More specifically, the applications of DNN or Deep Learning (DL) make breakthroughs in many difference areas [7]. DNN represents a series of multi-layer architecture NNs that training with the greedy layer-wise unsupervised pre-training algorithms[8], [9]. Albeit controversial, this family of NNs have won great success in some fields including Computer Vision, Speech Recognition, Natural Linguistic Programming and Bioinformation Processing. By applying the greedy layer-wise unsupervised pre-training mechanism, DNN can reconstruct the raw data set, in other words, DNN can "learn" features from the original data with a learning system mechanism instead of selecting features manually that we did traditionally[10]. And the intelligent models, like classifiers or regressors usually can obtain higher accuracy and better generalization with the learned features.

As its name suggested, DNN is a kind of NNs that structured by multiple layers. The word "Deep" indicates that such NN contains more layers than the "shallow" ones,

[1]http://www.hko.gov.hk/contente.htm

which mainly includes the most widely used three-layer (single hidden layer) Feed Forward NNs in the past 30 years. Actually, multi-layer NN is not a new conception, some earlier studies have been conducted since 1990s [11], [12], but the successful implementation of multi-layer NNs was not realized until the provision of the novel training mechanism by Hinton in 2006 that a so-called Layer-wise unsupervised Pre-training mechanism is employed to solve the training difficulties efficiently [8]. Via the Layer-wise unsupervised Pre-training mechanism, a DNN represents the raw data set projected from the original feature space into a learned feature space layer by layer in the training process. In each layer, the unsupervised training may provide a kind of regularization to the data set and minimize the variance .

Although theoretically, a shallow NN with three layers trained with Back-Propagation(BP) training algorithm has been proved that can approximate any nonlinear functions with arbitrary precision [13], once the number of hidden neurons is limited, the learning ability of a shallow NN may not be enough and poor generalization may be expected when using an insufficiently deep architecture for representing some functions. The significance of “deep” is that compared with a simple and shallow model, NN with deep architecture can provide a higher learning ability: functions that can be compactly represented by a deep architecture might be required to handle an exponential number of computational elements (parameters) to be represented by a deep architecture. More precisely, functions that can be compactly represented by a depth k architecture might require an exponential number of computational elements to be represented in a depth $k-1$ architecture [9]. Therefore, by adding the number of layers in the network architecture, DNN can provide higher learning ability with less hidden neurons in each layer, this advantage may be more useful for the big data cases. In general, compared with shallow NNs, the DNN model can learn from the massive raw data and map the raw data into a new feature space, classifiers or regressors thus may have chances to obtain higher accuracy and better generalization.

The main work of this paper is an extension of our previous work published in WORLDCOMP’14 last year [14]. In this work, we are continually exploring the potential of DNN in time series problems, especially in weather forecasting domain. In previous research, we noted that for time series problem, a good representation of original feature space may be helpful for the applied model to get better performance [15]. Meanwhile, in time series problems, the correlations among features are obvious but not easy to be identified. If we can analyze the correlations and have the features represented, the prediction accuracy is expected to be improved, and the DNN is a reasonable and suitable tool to analyze the time series features. Moreover, in [14], we have shown that the Stacked Auto Encoder(SAE) [16] can provide positive results on our weather data sets. However, the SAE is considered as a discriminative approach, and for time serise problem, we hope to use a more generative method to build the DNN in order to get the prior knowledge from the data sequence in the model training process. Therefore, in this paper, we applied the Continuous Restricted Boltzmann Machine (CRBM) to build the DNN to deal with the wind speed forecasting problem in Hong Kong. In detail, in the experiment, the CRBM model based DNN is employed to predict the wind speed in the next few hours. The massive data involving millions of weather records employed in this study is provided by The Hong Kong Observatory (HKO).

The contribution and significance of our investigation demonstrate that: we give a further investigation to show that NNs with deep architectures can improve the prediction accuracy in weather forecasting domain; moreover, the modified version of the Restricted Boltzmann Machine(RBM), CRBM, is empolyed in our paper to show that the RBM with continuous stochastic units can partly solve the limitation of classical RBM; more importantly, in our work, we focus on the wind speed forecasting of Hong Kong. For wind speed forecasting in Hong Kong case, we have some special challenges: (i) for wind data at Hong Kong, there is urbanization effect over the long term; (ii) there are tropical cyclones in Hong Kong bringing high speed winds, which are difficult to predict [17], the results of our experiment demonstrate that our model can learn the wind speed change trends better than the previous models.

II. The Wind Speed Prediction Problem in Hong Kong

Unlike data sets in other domain, weather data has some particularities. Specifically, there is season-to-season, and year-to-year variability in the trend of weather data. The cycle could be multi-month, multi-season or multi-year, and the main difficulty of investigations on weather data is to capture all the possible cycles. Hong Kong is characterized by a long coastline and numerous islands for such a relatively small territory.The mesoscale weather system of Hong Kong is quite different from other places since it is heavily affected by rainstorms and tropical cyclones[18], moreover, the high building density may also affect the weather condition of Hong Kong. Therefore, finding the disciplines and capturing the possible cycles of wind speed change in Hong Kong is more difficult than other places in sub-tropical regions.

The changes of wind speed may greatly impact Hong Kong people’s daily life, for example, the government’s plan of wind power generation system is greatly depending on the long-term wind speed prediction [17], or, the short term forecasting may affect the operation of airport and harbor in Kong Kong. Therefore researchers on Hong Kong put great efforts on wind speed forecasting. Many significant investigations including artificial intelligence technologies have been accepted as appropriate means for wind speed forecasting and reported encouraging results since 1980s [19], [20].

Among many different intelligent models, univariate time series regression is the most fundamental and most widely applied one in wind speed forecasting, especially short-term predictions. In this paper, we also concentrate on employing DNN to represent the feature space for univariate time series model. Generally speaking, for a certain variable, the objective of univariate time series regression is to find the relationship between its status in a certain future time point and its status in a series of past time points, and estimate its future status via:

$$v_t = f(v_{t-1}, v_{t-2}, \ldots, v_{t-n}) \tag{1}$$

The function f, can be obtained by employing different intelligence models such as Linear Regression, Generalized

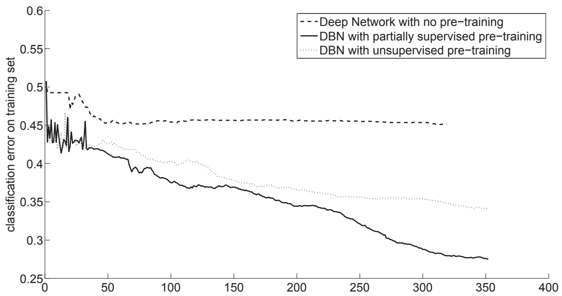

Fig. 1. Training classification error vs training iteration on DNNs, which shows the optimization difficulty for DNNs and the advantage of pre-training methods.

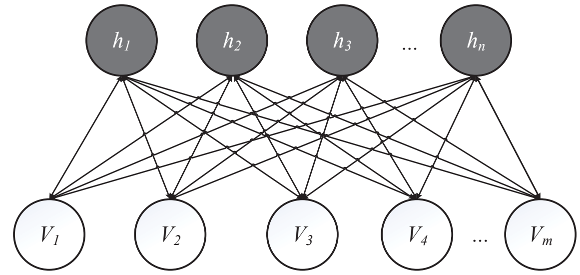

Fig. 2. The typical architecture of a classical RBM model with two layers, m neurons in the visible layer and n neurons in the hidden layer, all neurons a binary-valued and no connection between any two neurons in the same layer.

Linear Model, Auto Regressive Integrated Moving Average Mode (ARIMA), etc.

In our investigation, we target on the wind speed data in the next few hours. We will input the raw data sets into our DNN model, the input n-dimension vector is composed of the status in $(t-1)th$, $(t-2)th$, ..., $(t-n)th$ time points, we try to use the DNN to represent these statuses, and employ a regressor to estimate the status in the tth time point. We hope the seasonal cycles can be captured via massive volume of data by the superior learning ability of DNN.

III. Greedy Layer-wise Unsupervised Pre-training and Layer Model Selection in Deep Learning

Although the idea of Multi-layer(Deep) NN has been proposed for more than twenty years, it wasn't widely used until 2006 since Hinton solved the training difficulties efficiently in [8].

The essential challenge in training deep architectures is to deal with the strong dependencies that exist during training between the parameters across layers [21]. Multi-layer NN has more parameters than NN with shallow architectures. Moreover, in a multi-layer NN, due to the non-convexity of the complex model, the optimization with traditional BP training approach may fall in a local minimum rather than global minimum. This may bring poor generalization to the model.

This problem wasn't well solved until Hinton et al. introduced Deep Belief Network (DBN) that greedily trained up one layer with a Restricted Boltzmann Machine (RBM) at a time in 2006 [8]. Shortly after, strategies for building deep architectures from related variants were proposed by Bengio [22] and Ranzato[23]. They solved the training problem of deep NN in two phases: in the first phase, unsupervised pre-training, all layers are initialized using this layer-wise unsupervised learning signal; in the second phase, fine-tuning, a global training criterion (a prediction error, using labels in the case of a supervised task) is minimized. Such training approach is called the Greedy Layer-wise Unsupervised Pre-training. Fig.1 [21] shows the comparison among different training methods for NNs with deep architectures.

The advantage of learning features from data via a unsupervised approach is that the plentiful unlabeled data can be utilized and that potentially better features than hand-crafted features can be learned. This advantage reduce the need for expertise of the data and often the learned feature space may provide a better regularization effect on the raw data so that can improve the accuracy of the applied model[1], [24]. There are a number of NN architectures categorized into the family of Greedy Layer-wise Unsupervised Pre-training approaches, for example, the Auto Encoder and the Sparse Auto Encoder that obtaining the connection weights of the hidden layer by learning an approximation of the input variables; the RBM, which models the static data via an energy function and the joint distribution for a given visible and the hidden vector; the Convolutional Neural Networks(CNN), which learns the features via a convolutional kernel in each layer, etc. We cannot say which unsupervised pre-training model is definitely better than others since each of the models have its own properties. The choices of model and how the data should be presented to the model are highly dependent on the properties of the data sets [24]. In our previous paper [14], we empolyed the SAE model to do the weather forecasting in the short term. However, compared with SAE, which is a discriminative model, the RBM model is a generative model. A generative model means that it can generate observable data given a hidden representation and this ability is mostly used for generating synthetic data of future time steps [25], [24]. Thus for time series problems, a generative model based DNN is reasonably expected to be able to provide a better performance than the stacked Auto Encoder.

IV. The Restricted Boltzmann Machine and the continuous Restricted Boltzmann Machine

The RBM is a two-layer networking with one visible layer and one hidden layer. Fig.2 gives an illustration of RBM architecture. As shown in Fig.2, the standard type of RBM has binary-valued (Boolean/Bernoulli) m hidden and n visible neurons, and consists of a matrix of weights $W = (w_{i,j})$ (size $m \times n$) associated with the connection between hidden neurons h_j and visible neuron v_i, as well as bias weights (offsets) a_i for the visible units and b_j for the hidden units. The word "restricted" means that there is no connection between any two neurons in the same layer.

Given these, the energy function of a configuration (pair of boolean vectors) (v, h) is defined as:

$$E(v,h) = -\sum_i a_i v_i - \sum_j a_j v_j - \sum_i \sum_j v_i w_{ij} h_j \qquad (2)$$

Since the neurons is binary-valued, the probabilities of the states of the visible and hidden neurons can be obtained via the sigmoid function:

$$p_{vi} = p(vi = 1) = \frac{1}{1 + exp(-\sum_i w_{ij} h_j)} \tag{3}$$

and

$$p_{hj} = p(hj = 1) = \frac{1}{1 + exp(-\sum_i w_{ij} v_i)} \tag{4}$$

respectively.

In RBM, the probability distributions over hidden and/or visible vectors are defined in terms of the energy function in Eq.(1):

$$P(v) = \frac{1}{Z} \sum_h e^{E(v,h)} \tag{5}$$

The RBMs are trained to maximize the product of probabilities assigned to some training set V:

$$\arg\max_W \prod_{v \in V} P(v) \tag{6}$$

In the traning process of the RBM model, the Minimising Contrastive Divergence (MCD) training rule for an RBM replaces the computationally expensive relaxation search of the Boltzmann Machine [26] with a single step of Gibbs sampling [27]. In each iteration, we update the w_{ij} according MCD rule by:

$$\Delta w_{ij} = \varepsilon(v \cdot h^T - \hat{v} \cdot \hat{h}^T) \tag{7}$$

where $\hat{v}$,$\hat{h}$ is the reconstructed states of the node in the last iteration.

As discussed above, we choose the layer component of DNN according to the type of the data sets and the property of the model. From the brief description of the RBM, we can see that the neurons in this model are binary value, this is why we chose Auto-Encoder rather than RBM approach in our previous work to forecast the wind speed data that was continuous-valued. However, according to [24], the RBM is a more generative model than the Auto Encoder, that means, from the aspetacts of model properties, the RBM model maybe a better choice in wind speed forecasting application. Therefore, in this paper, for using the RBM to process the continuous valued wind speed data, a CRBM is employed to build the DNN architectures.

The CRBM is introduced by H. Chen and A.F. Murray in [28]. The continuous stochastic neurons are employed to take the places of the binary-value neurons by adding a zero-mean Gaussian noise to the input of a sampled sigmoid neuron. The binary-value neurons in CRBM have the form:

$$s_j = \varphi_j \cdot (\sum_i w_{ij} s_i + \sigma \cdot N_j(0,1)) \tag{8}$$

with

$$\varphi_j(x_j) = \theta_L + (\theta_H - \theta_L) \cdot \frac{1}{1 + exp(a_j x_j)} \tag{9}$$

where $N_j(O,1)$ represents a Gaussian random variable with zero mean and unit variance. The constant σ and $N_j(O,1)$

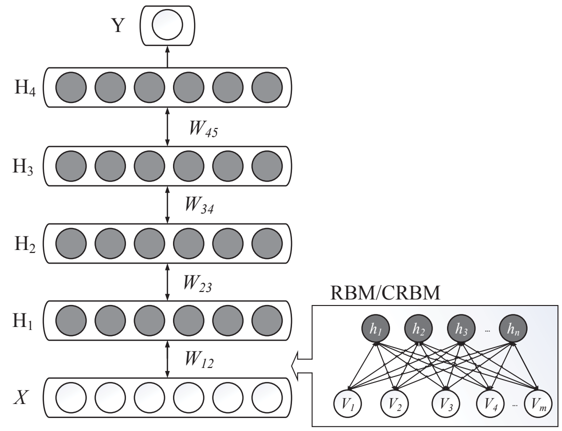

Fig. 3. A 4-hidden-layer DNN with RBM model, by which each layer is greedily pre-trained with an unsupervised RBM model to learn a nonlinear transformation of its input (the output of the previous layer) that captures the main variations in its input by a MCD training methods.

thus constitute a noise input component $n_j = \sigma \cdot N_j(O,1)$ according to a probability distribution

$$p(n_j) = \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-n_j^2}{2\sigma^2}) \tag{10}$$

The parameters θ_L, θ_H and a_j control the asymptotes and slop of the sigmoid function in the neurons. By this way, the nature and extent of the neurons stochastic behavior is simulated [29]. Such behaviour is similar to the noisy units in [30], where the variance of the added noise is tuned [28].

From [28],the energy function of CRBM is analogous to that of the continuous Hopfield model:

$$E_{CRBM} = -\frac{1}{2} \sum_{i \neq j} w_{ij} s_i s_j + \sum_j \frac{1}{a_j} \int_0^{s_j} \varphi^{-1}(s) ds \tag{11}$$

By using the MCD rule, in each iteration, parameters in CRBM model can be updated via:

$$\Delta w_{ij} = \varepsilon_w (s_i \cdot s_j^T - \hat{s}_i \cdot \hat{s_j}^T) \tag{12}$$

and

$$\Delta a_j = \frac{\varepsilon_a}{a_j^2} (s_j \cdot s_j^T - \hat{s_j} \cdot \hat{s_j}^T) \tag{13}$$

Consequently, we need to combine the CRBMs layer by layer with a stacked structure to build the DNN. We follow the method introduced in [8], In each layer, we use a CRBM to train the connection wight in this layer, and then have these layers combined together. Specifically, in the training process of each layer, as shown in Fig.1, the input vectors need to pass through the two layers, meanwhile, the vectors in hidden layers are representations of the input vectors and can be used to reconstruct the input vectors. Thus, in each layer of the DNN, the input of the current layer is the output of the previous layer, then we train the input data via a CRBM, and use the transformed vectors as the output of the current layer. Fig.3 shows the detailed mechanism of CRBM based DNN. We can see that through a DNN, the raw data can be represented into new feature spaces layer by layer, in other words, DNN can learn features from the original data sets. And consequently, we need to employ a proper regression approach to compute the output with the learned features.

V. Experimental Results and Analysis

In this section, we will describe the experiment and give the results and discussions

A. Wind Speed Data Collection and Pre-processing

The HKO has provided great support to our investigation. Based on our collaboration with HKO, a massive volume of high quality real weather data could be applied in our experiment. The time range of the historical wind speed data sets is almost 30-year long, which covers the period from January, 1, 1983 to December, 31, 2012. The total number of records is more than 230,000. Please note that our data set contains massive records that cover data all the year round of the 30 years in Hong Kong, by this way, we hope the model can catch the urbanization effect change over the long term and learn the rules of the daily, monthly and yearly cycles as well as the seasonal rules of the tropical cyclones in Hong Kong [31].

The wind speed data provided by the HKO has two dimensions: the polar coordinate for the wind direction (measured with degree angle) and the speed (measured with meters per second), moreover, for a certain time points, the direction of the air motion is not stable, i.e. the wind direction at that time point is not fixed. such condition is denoted as "variable" in the raw data. Therefore, according to the requirement of our algorithm, we have to do some pre-processing on the data sets: since the wind speed data (in a fixed horizontal plane) is a vector quantity that has two dimensions in the polar coordinate (as Fig.4), i.e. the angle to show its direction and the speed to measure the magnitude in this direction: the polar coordinate and the speed [32]. However, since our model is focused on single variable time series problems, we have to transform the data set to satisfy the model's requirement. According to the physical significance of the two dimensions, we denote the angle as θ and the speed as v to obtain:

$$v^0 = cos\theta \cdot v \tag{14}$$

where v^0 is the vector components of the wind speed in 0 degree angle direction (as Fig.5). Thus, what we actually simulate is the time series of the speed components of the air motion in 0 degree angle direction. Moreover, there are about 3% of wind speed data with the direction valued as "variable", for such condition, we consider it as a missing value in the data set and use the average value of the wind direction in its previous time point and its next time point to replace the value "variable".

B. Experiment Configuration

In our experiment, the whole data set is divided into two parts, the training set contains the samples of the first 27 years, and the testing set contains the samples of the last 3 years. Thus the ratio of the sizes between the training set and the testing set is 9:1.

To learn the complex effect of the seasonal and yearly cycle of the wind speed change in Hong Kong, we don't input training data to the model randomly as we did in [14] last year. In this investigation, we use shift windows to organize the input model, and there are 7-day data contained in each window. The windows are input into the model according to the time sequence.

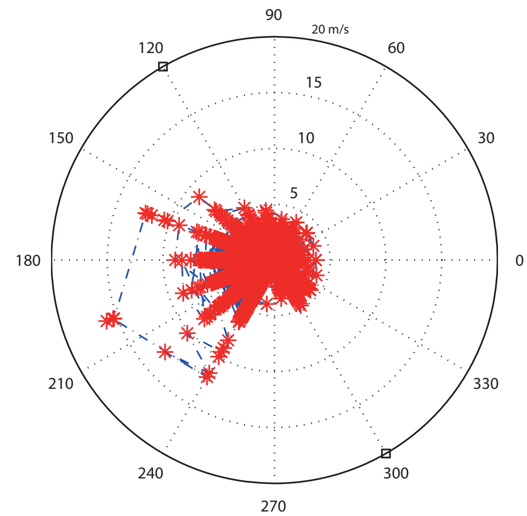

Fig. 4. The distribution of wind speed data in polar coordinate

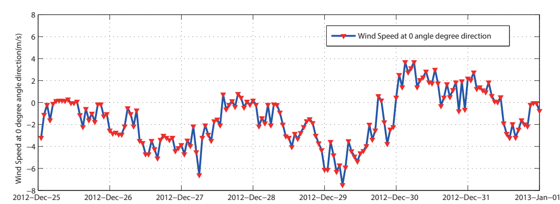

Fig. 5. The distribution of wind speed at a fixed diection

In our experiment, we build a four-layer network and employed it to predict the wind speed in Hong Kong. The large size of the data set can avoid the overfitting problem of the complex model. Actually, there is a feature reconstruction of the data sets in each layer, and we hope to obtain a better feature space after 3 feature reconstructions so that the output layer can provide a higher accuracy in the finally obtained feature space. In the top layer, in our model, we choose the Support Vector Regression (SVR) with the Gaussian kernel to give the forecasting output [33]. The parameter configuration of the whole model is given in Table I.

TABLE I. The parameter configuration of the networking

Parameter	Value
Number of neurons in hidden layer 1	168
Number of neurons in hidden layer 2	96
Number of neurons in hidden layer 3	84
Learning rate	0.001
Max Iteration	1000
Parameters in SVR	Default as LibSVM [33]

TABLE II. THE COMPARISON OF WIND SPEED PREDICTION BY THE FOUR MODELS

Model	NMSE	DS	R^2
Single Layer ANN	0.4547	0.694	0.791
CRBM DNN with SVR	0.2213	0.727	0.921
SAE DNN with SVR	0.2395	0.830	0.901
Classical SVR	0.2947	0.741	0.871

C. Experiment Results

In this paper, to evaluate the performance of the CRBM based DNN, other three models are also applied to predict the wind speed in Hong Kong, and the results are compared. Specifically, the four models are the single layer Artificial Neural Networking(ANN), the Classical SVR, the SAE DNN followed with an SVR and the proposed model. From the results comparison, We hope to study the advantages and disadvantages of the SVR and NN models; also, the results will show that whether a feature representation is helpful for improving the accuracy of wind speed prediction; and more importantly; ,the performances of the SVR in feature spaces obtained via the SAEs and via CRBMs are also compared. Table II gives the comparison of the results on three major criteria, and the performance of the four models is respectively shown in Fig 6, Fig 7, Fig 8 and Fig 9.

From the results, we can observe that, all of the four models can catch the main trends of the wind speed change in Hong Kong, but the performances of the four models are not in the same level. The single layer ANN provides the worst results: the single-layer ANN model only has R^2 value less than 0.8; and also provides relatively poorer performance in other two criteria. However, we believe that if we can add more hidden neurons in ANN, the performance will be better, but the computational cost will also be higher. From the results of other three models, we can see that the performance of SVR can be improved (0.03 as the least improvement on R^2 value) by using the DNN feature representation, moreover, compared with the SAE model, DNN with CRBM can provide a 3% improvement of accuracy on weather data prediction. These results demonstrate that as a generative model, CRBM is more suitable than SAE for the time series problem, e.g., wind speed forecasting.

VI. CONCLUSION, LIMITATION AND FUTURE WORK

The wind speed forecasting in Hong Kong is more difficult than that of other places in the same latitude for two reasons: the great affect from the urbanization of Hong Kong in the long term, and the very high speeds of winds brought by the tropical cyclones. In our investigation, we modified the model that applied in our previous paper [14], using the continuous valued RBM model to build the architecture of the DNN instead of the SAE that we used before. The RBM model is more generative than the SAE models and more suitable for time series problem, and we applied the continuous version of the RBM so that the model can be employed to process the wind speed data.

We use massive volume of wind speed data in Hong Kong to test our model. The comparison results are positive: the CRBM based DNN model can learn a better feature space from the raw wind speed data so that the SVR can obtain higher accuracy in this learned feature space. The network can provide lower NMSE by using the CRBM than using the SAE.

The main future work of our investigation is that, we will try to employ the CRBM model on more difficult weather data, such as rain fall data set; and moreover, we will continue exploring the theoretical principle of computational intelligence, especially, we will try to give the mathematical explanation of the DNN.

ACKNOWLEDGMENT

The authors wish to thank the financial support from the Hong Kong Polytechnic University by its Central Research Grants G-YK53 and G-YN39.

REFERENCES

[1] Q. Cao, B. T. Ewing, and M. A. Thompson, "Forecasting wind speed with recurrent neural networks," *European Journal of Operational Research*, vol. 221, no. 1, pp. 148–154, 2012.

[2] T. J. Considine, C. Jablonowski, B. Posner, and C. H. Bishop, "The value of hurricane forecasts to oil and gas producers in the gulf of mexico," *Journal of Applied Meteorology*, vol. 43, no. 9, pp. 1270–1281, 2004.

[3] R. Horonjeff, F. McKelvey, W. Sproule, and S. Young, *Planning and design of airports*. Granite Hill Publishers, 2010.

[4] A. Khanduri and G. Morrow, "Vulnerability of buildings to windstorms and insurance loss estimation," *Journal of wind engineering and industrial aerodynamics*, vol. 91, no. 4, pp. 455–467, 2003.

[5] G. Li and J. Shi, "On comparing three artificial neural networks for wind speed forecasting," *Applied Energy*, vol. 87, no. 7, pp. 2313–2320, 2010.

[6] A. More and M. Deo, "Forecasting wind with neural networks," *Marine structures*, vol. 16, no. 1, pp. 35–49, 2003.

[7] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning." *Journal of Machine Learning Research-Proceedings Track*, vol. 27, pp. 17–36, 2012.

[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[9] Y. Bengio, *Learning deep architectures for AI*. Now Publishers Inc., 2009, vol. 2, no. 1.

[10] X. Wang and Q. He, "Enhancing generalization capability of svm classifiers with feature weight adjustment," in *Knowledge-Based Intelligent Information and Engineering Systems*. Springer, 2004, pp. 1037–1043.

[11] J. Schmidhuber, "Curious model-building control systems," in *Neural Networks, 1991. 1991 IEEE International Joint Conference on*. IEEE, 1991, pp. 1458–1463.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[14] J. N. Liu, Y. Hu, J. J. You, and P. W. Chan, "Deep neural network based feature representation for weather forecasting."

[15] J. N. Liu and Y. Hu, "Application of feature-weighted support vector regression using grey correlation degree to stock price forecasting," *Neural Computing and Applications*, pp. 1–10, 2013.

[16] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 9999, pp. 3371–3408, 2010.

[17] H. Yang, L. Lu, and J. Burnett, "Weather data and probability analysis of hybrid photovoltaic–wind power generation systems in hong kong," *Renewable Energy*, vol. 28, no. 11, pp. 1813–1824, 2003.

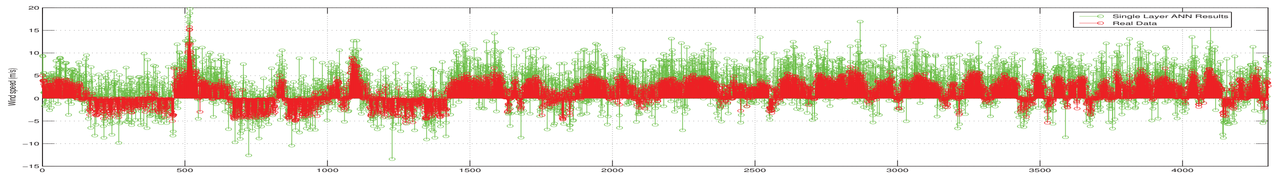

Fig. 6. The prediction results of a singly layer ANN

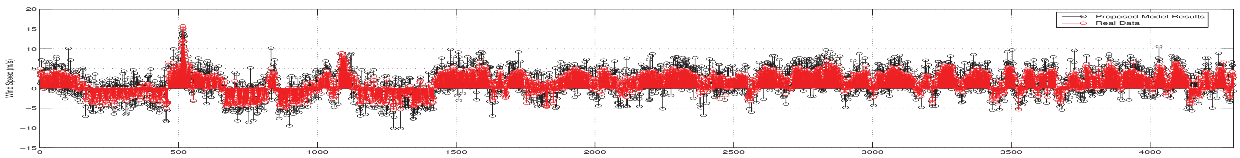

Fig. 7. The prediction results of the proposed model

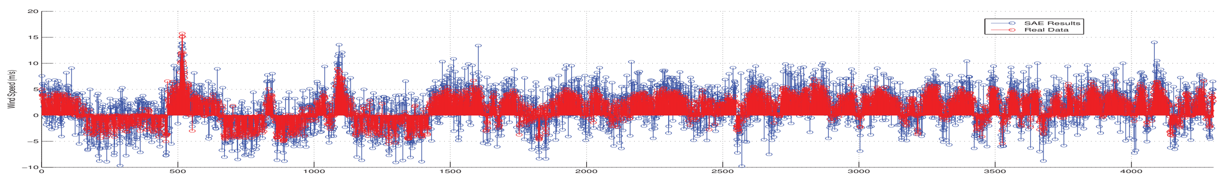

Fig. 8. The predictioin results of the SAE model

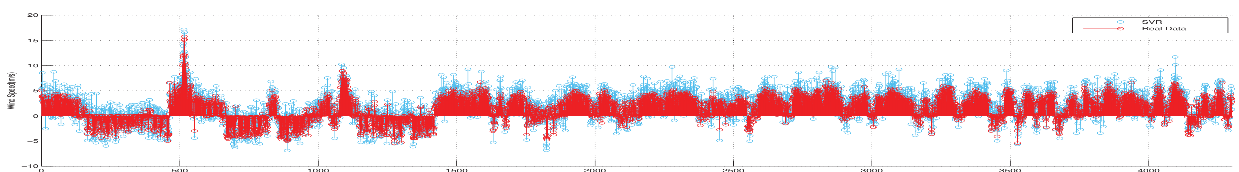

Fig. 9. The prediction results of the SVR model

[18] P. Li and E. S. Lai, "Short-range quantitative precipitation forecasting in hong kong," *Journal of Hydrology*, vol. 288, no. 1, pp. 189–209, 2004.

[19] S.-M. Chen and J.-R. Hwang, "Temperature prediction using fuzzy time series," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 30, no. 2, pp. 263–275, 2000.

[20] K. Kwong, M. H. Wong, J. N. Liu, and P. Chan, "An artificial neural network with chaotic oscillator for wind shear alerting," *Journal of Atmospheric and Oceanic Technology*, vol. 29, no. 10, pp. 1518–1531, 2012.

[21] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *The Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[22] S. M. Miller, Y. Geng, R. Z. Zheng, and A. Dewald, "Presentation of complex medical information: Interaction between concept maps and spatial ability on deep learning," *International Journal of Cyber Behavior, Psychology and Learning (IJCBPL)*, vol. 2, no. 1, pp. 42–53, 2012.

[23] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. Lecun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[24] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014.

[25] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Advances in Neural Information Processing Systems*, 2013, pp. 899–907.

[26] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in boltzmann machines," *MIT Press, Cambridge, Mass*, vol. 1, pp. 282–317, 1986.

[27] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[28] H. Chen and A. F. Murray, "Continuous restricted boltzmann machine with an implementable training algorithm," in *Vision, Image and Signal Processing, IEE Proceedings-*, vol. 150, no. 3. IET, 2003, pp. 153–158.

[29] H. Chen and A. Murray, "A continuous restricted boltzmann machine with a hardware-amenable learning algorithm," in *Artificial Neural NetworksICANN 2002*. Springer, 2002, pp. 358–363.

[30] B. J. Frey, "Continuous sigmoidal belief networks trained using slice sampling," *Advances in Neural Information Processing Systems*, pp. 452–458, 1997.

[31] J. C. Chan, J.-e. Shi, and C.-m. Lam, "Seasonal forecasting of tropical cyclone activity over thewestern north pacific and the south china sea," *Weather and forecasting*, vol. 13, no. 4, pp. 997–1004, 1998.

[32] R. A. Pielke, *Mesoscale meteorological modeling*. Academic press, 2002.

[33] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.