Wind Power Prediction based on Recurrent Neural Network with Long Short-Term Memory Units

Danting Dong*, Zhihao Sheng*
Computer Engineering, University of Toronto
Toronto, Canada
e-mail: [danting.dong, zhihao.sheng]@mail.utoronto.ca

Tiancheng Yang*
Computer Science, University of Waterloo
Waterloo, Canada
e-mail: t77yang@edu.uwaterloo.ca

Abstract— **Wind power is one of the most promising renewable energy sources, it is clean, safe and inexhaustible. However, predicting wind signal has always been challenging because the time series data is nonlinear, non-stationary and chaotic. In this paper, we provide a novel predicting framework including a recurrent neural network (RNN) structure model with long-short term memory (LSTM) units and an effective forecasting map adapted to different prediction horizons. We compare our new approach with concurrent methods and show that our new method is more effective in predicting wind power.**

Keywords-component; wind power; renewable energy; prediction model; deep learning; recurrent neural network; long short-term memory

I. INTRODUCTION

The energy cost and consumption grow rapidly recently, which makes the prediction of the future energy usage gradually important. Moreover, there are increasingly requirements in different industries to make important decisions relied on the accuracy of the future forecasting. Among many modern energy sources, wind power plays a significant effect in various industries and business like electric energy industry, construction engineering and transportation [1]. However, the prediction of wind power has always been a challenge due to its uncertainty and non-stationary characteristics.

Three traditional methodologies in wind power prediction are physical modeling, statistical methods and soft-computing techniques [2]. The physical modeling use information from geography and meteorology to capture the wind signal trend. It can predict certain wind turbines with satisfying accuracy but is not very practical due to the high computational cost. [3, 4, 5]. The statistical modelling resolves an optimal process with high accuracy based on collected data and is usually applied in short term prediction. Additionally, soft-computing technique which including neural network is an interconnected group and is based on formal logical system. The soft-computing technique can deal with uncertainty and achieve solution with a low cost [3]. With the rapid development of machine learning and deep learning techniques, new approaches have attracted attention for wind power prediction. Linear regression is a commonly used method because it is easy to implement [1]. But in practice it may over-simplify the question and is sensitive to extreme observations. Another non-linear model to predict wind power is the random forest model [2, 6], which may also easily over fit the data [7]. Based on the temporal structure of wind speed data, we propose a recurrent neural network (RNN) with Long-Short Term Memory (LSTM) unit for modelling [8]. An important feature of RNN is that it could carry over the previous information over time. The LSTM unit can further persist the memory to a long period time. In this project, we focus on the wind data from Global Energy Forecasting Competition in to conduct the analysis. Our goal is to develop a model which can forecast wind speed in a variety of time period based on the past historical data.

The remaining of this paper is organized as follows: section II gives a detailed explanation of our model and the new prediction framework. Section III shows the results in both quantitative and qualitative ways. The comparison between our model and three tradition models mentioned above will also be discussed. The conclusion will be drawn at the end of the paper.

II. METHODOLOGY

A. RNN and LSTM

The wind power records are chronological, hence the information is expected to be persisted so that the power trend would help to predict more accurate results. Therefore, RNN discussed above could be our natural wind power prediction models [9].

One of the most important features of RNN is that it can memorize information from previous data. The feedback loops inside RNN will allow the model to keep information over time. The left-hand side of the equal sign in Figure 1 illustrates the basic structure of an RNN unit. It has an arrow pointing to itself, indicating that the data inside block "A" will be recursively used. Once expanded, its structure is equivalent to the chain shown in the right-hand side of Figure 1.

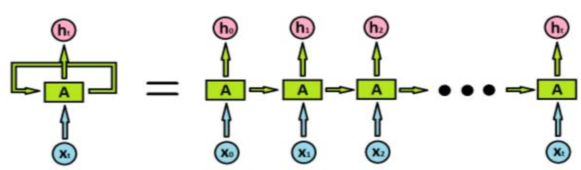

Figure 1. An unrolled recurrent neural network

However, it is needed to be cautious about the long-term dependencies of the RNN model. This problem could be more obvious once the dataset becomes very large in practice. Fortunately, the LSTM can address this issue [10]. The LSTM is an extension of RNN, and it contains a "memory cell", which is designed to remember information passed through long period of time. It is the main reason LSTM is better than general RNN models in our research.

The basic structures of the standard RNN and LSTM are very similar. However, in each cell of the RNN, there is only one single tanh layer [Figure 2]. While the LSTM model has a set of layers used to control data into and out of each analysis and determine whether to add or remove that information into the cell state [Figure 3] [11]. As a result, the "memory cell" of LSTM combines a sigmoid layer and a multiplier together as a gate: the sigmoid layer outputs a number between zero and one, which then multiplies the original information. If output is 1 all information will pass the gate while zero does not. The "memory cell" in LSTMs contains three different gates: a "forget gate" to decide whether the information should be kept or not; an "input gate" to activate the updated information from last cell; and an "output gate" to choose what information will be outputted to the next cell [11].

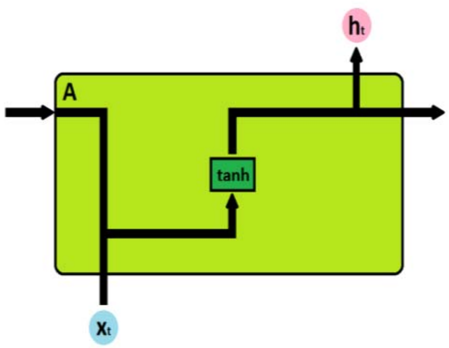

Figure 2. A standard RNN's cell

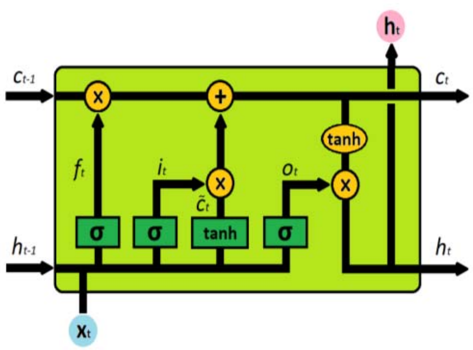

Figure 3. A LSTM's "memory cell"

The more important key of LSTMs is the cell state, which travels along the entire chain with some minor linear interactions [12]. It allows the information to flow through the chain while remained unchanged to keep the integrity from long period of time. As shown in the Figure 3, C is the cell state at top of the diagram running horizontally.

LSTM calculates the output sequence h_t based on input sequence x_t as follows:

$$f_t = \sigma\left(W_t \cdot [h_{t-1}, x_t] + b_f\right) \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

In these equations, f_t is the output of the forget gate, i_t and o_t are the outputs of the input gate and the output gate respectively. W represents weight metrics, b is the bias, and C is the cell state [13]. The sigmoid layer (σ) is used to output a number between zero and one which will control the data' pass rate. The *tanh* layer creates a vector, $\tilde{C}_t$, which will be added to the state. Then the cell state will go through another *tanh* layer and multiply by the sigmoid layer to output the result, h_t.

B. Prediction System

Our proposed prediction framework consists of three steps, a preliminary processing to the raw data, a self-update prediction map with a moving window, and a trained model with optimized LSTM structure.

To pre-process the original data, firstly cut them into sequential pairs, then divide these pairs into training and validation sets. The steps are as following.

(1) Determine a number n, as the size of data for training, and divide the complete dataset into a training set and a validation set
(2) Choose a sequence length p, which is the length of historical data that is needed to look back. Then construct a dataset S with sequence sets Ji. The set Ji = {i, i+1, i+2, …, i+p}, i is chosen from 0 to the length of entire data. Hence, each Ji has a size of p+1 data.
(3) Parse each set Ji into two parts, Xi and Yi, Xi = {i, i+1, …, i+l-1} Yi = {i+l}. Then, combine all the Xi together into the dataset train_X and eval_X and combine all the Yi together into the dataset train_Y and eval_Y.

After preparing the data, train_X and train_Y are used to train a RNN model with LSTM units. A series of according experiments are carried out for structure optimization and parameter tuning.

After the deep training of the model, a prediction map is constructed as follows:

(1) Take the first pair of data from eval_X with a size of p as the initial state of a prediction window, use the pre-trained model to predict the next following one data point.

(2) Move the window forward with one step, the new prediction window consists of p-1 data same as the last window, and a new-predicted data from step 1 as the the last one data point.
(3) Repeat (1) and (2) for m times, where m is the length of the prediction horizon.
(4) After m times, take the next pair of data from eval_X, and repeat (1), (2) and (3) until the end of entire eval_X.

The pseudo code of the prediction system is shown in Figure 4, and an illustration of the prediction window is given in Figure 5.

Algorithm 1 The prediction process

Require: Sequence length L, steps of prediction s, trained model h, testing data set T, raw data set R

Ensure: Prediction data set P

```
P ← {}
t ← current testing location
n ← length of T
while t < n do
    for i ← 0 to l do
        x ← T[t+i, t+i+1, ..., t+i+L−1]
        p ← h(x)
        P ← P + p
        T[t+i+L] ← p
    end for
    t ← t + s
    T ← R
end while
```

Figure 4: The prediction process

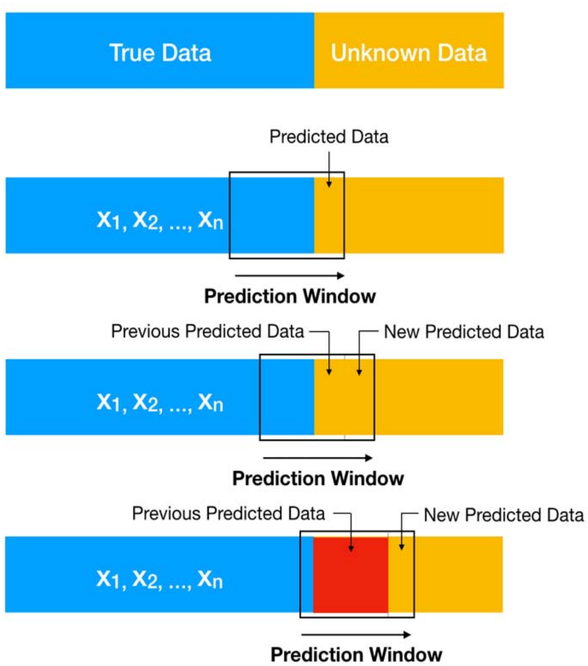

Figure 5: Graph for prediction process.

III. RESULTS

In this section, the results of the wind power prediction is presented with the real data in comparative experiments.

A. *Data and Prediction Horizon*

The data set used for analysis is from Global Energy Forecasting Competition 2012. The wind speed is recorded at a wind farm hourly and then normalized to a range of 0 and 1. The data is recorded from July, 2009 to June, 2012 with 26245 entries. Figure 6 shows the first 1000 raw data.

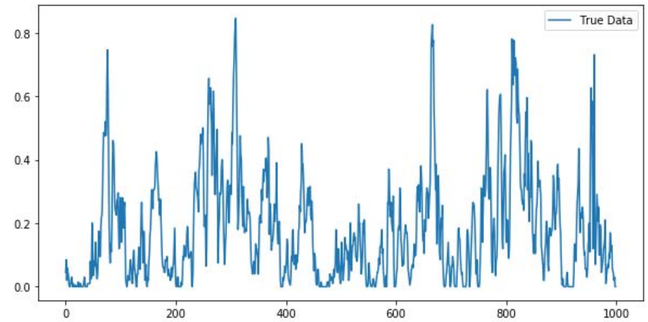

Figure 6: First 1000 raw data

In experiment, the dataset is separated into two parts: training set and evaluation set. 23000 (87.6%) data are selected as training data to train the model and use the remaining part to evaluate the model performance. Before training, a data processing step is carried out to cut the data into sequential pairs with a specific length. This sequence length will be tuned as a parameter in the experiment below. Experiments are also carried on wind power prediction for a various coverage from short-term to long-term. The definition of the prediction horizon is shown in the following table. [14]

TABLE I: WIND SPEED PREDICTION TIME SCALE

Time Horizon	***Range***
Long term	Larger than 1 day
Medium term	6 hours to 1 day
Short term	30 minutes to 6 hours
Very short term	Few seconds to 30 minutes

B. *Point-to-point Prediction Results*

First of all, point-to-point prediction is implemented, which predicts a single step using the actual data set with size of 120 each time. The window size is determined by cross validation. A line graph of comparing a part of actual data and predicted data is given by Figure 7. It is noted that point-to-point prediction produces the predicted result close to the actual one. However, it is not practical in the real production since industry usually wants to have longer horizon, and point-to-point prediction can only make one step forward.

C. *Comparison with different Prediction Horizon*

The prediction method for different time periods is also tested(predict_len, one unit means one hour), from one hour to two days (48 hours). This covers short term period to long term period defined in Table 1 and also meets the needs of practical forecasting. From Table 2 and Figure 8, it can be seen that the result with shorter prediction length produced smaller error than longer prediction length when sequence length is fixed. The error grows when prediction period covers more. In addition, for each prediction period, it is also needed to compare the result with different sequence length. Within this process, the best value of this parameter (seq_len, one unit means one hour) for our model could be found and the value can be used to do comparison cross different models. Inside most rows of Table 2, when sequence length equals 120 the forecasting model has smallest error. The prediction results with sequence length equal to 120 are shown in Figure 9, with a comparison to the true data. The way to calculate error is mean square error (MSE). It calculates average square difference between predicted data and true data [15]. Smaller MSE identifies better prediction result.

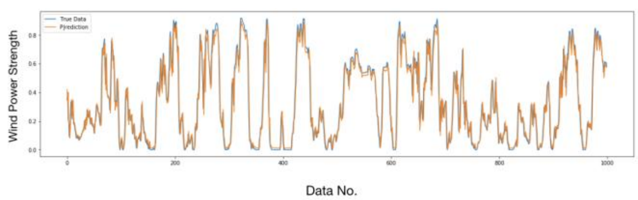

Figure 7. Point-by-point prediction compared with true data

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (P_i - A_i)^2 \qquad (1)$$

where P_i is model output value and A_i is real value at prediction step i.

TABLE II: LSTM RESULT FOR DIFFERENT PARAMETER VALUE

Seq_len (hour) / **Predict_len (unit)**	60	120	240
1	0.005998223	0.005961354	0.005945017
6	0.024276134	0.023023771	0.023165724
12	0.038745614	0.037682586	0.040955302
24	0.051733093	0.046380745	0.050997839
48	0.063649691	0.059187898	0.068807822

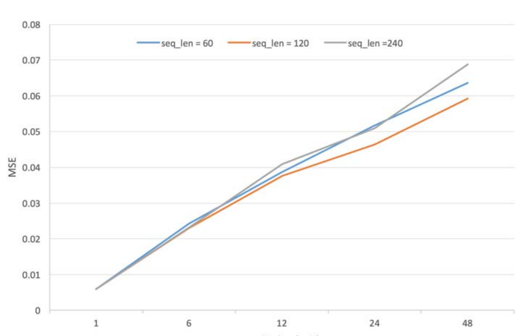

Figure 8: LSTM model prediction result with different sequence length and prediction length

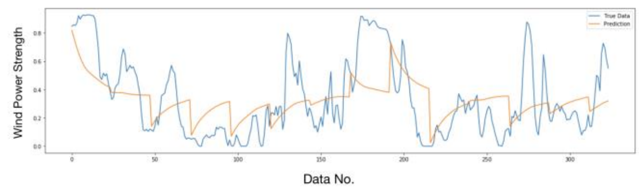

Figure 9: LSTM prediction result compared with true data

D. Comparision with Other Models

For comparison, mean square error (MSE) is used as an accuracy evaluation criterion. The higher the MSE is, the worse the result is produced. 120 is used as the sequence length since 120 gives the output with smallest MSE from the previous experiment. From the table and line chart of the experiment data, the accuracy of prediction from the LSTM model is higher than that of the three classic machine learning algorithms, namely linear regression [16], random forest [17], and gradient boosting [18]. As can be seen from Figure 10 that with the growing of prediction length, MSE for all methods increased rapidly. Furthermore, when prediction length is short, such as one or six, the forecast results from LSTM, linear regression, and gradient boosting are very similar. With the growth of the prediction length, LSTM produces a better performance than these two methods. The statistics also show that the results given by random forest model have the largest error comparing to the actual ones among the four models compared.

TABLE III: LSTM RESULT OF MULTIPLE MODELS WITH DIFFERENT PREDICTION LENGTH

Predict_len	1	6	12	24	48
LSTM	0.0059613	0.0230237	0.0376825	0.0463807	0.0591878
Linear Regressio	0.0059221	0.0230864	0.0379108	0.0471406	0.0598618
Random Forest	0.0067417	0.0262763	0.0436307	0.0596441	0.0751653
Gradient Boosting	0.0059673	0.0227639	0.0374811	0.0481438	0.0639614

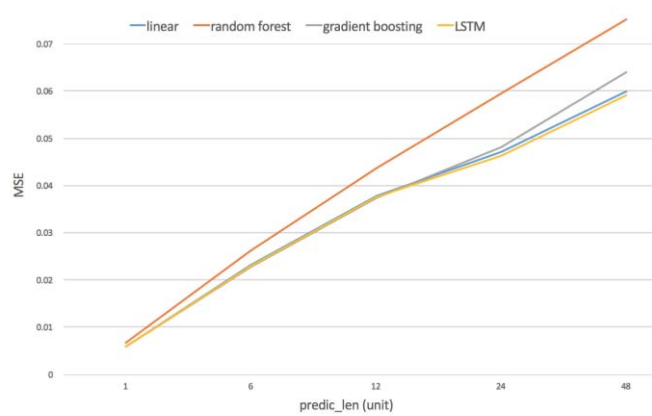

Figure 10. Prediction result of multiple models

IV. CONCLUSION

In this paper, we propose a novel prediction framework based on recurrent neural network and long short-term memory which performs well for predicting various time periods. This approach takes the advantage of LSTM model that can carry long historical data to prediction, and a self-updated prediction map that allows to make long time period prediction. The performance of our model is compared with three other wind power prediction models and the results show that our approach outperforms the others in all types of prediction horizons.

Furthermore, the proposed prediction system has some aspects to be further discussed and developed. During experiment, we realize the model performance is heavily influenced by the sequence length of input. Hence, further improvement can contribute to the pre-processing step of the raw time sequence data including cleaning, clustering and other techniques to further optimize the prediction. Another aspect that can be improved is the prediction algorithm. It is promising that an ensemble of multiple existing models could give a better result than a sole model. Moreover, in the future work, the number of layers and the structure of neurons in each layer could be further explored, and more models such as Gated Recurrent Units (GRU) could be used as a control model as well.

REFERENCES

[1] T. Hong, P. Pinson and S. Fan, "Global Energy Forecasting Competition 2012", International Journal of Forecasting, vol. 30, no. 2, pp. 357-363, 2014.

[2] J. Tan, H. Liu, M. Li and J. Wang, "A prediction scheme of tropical cyclone frequency based on lasso and random forest", Theoretical and Applied Climatology, vol. 133, no. 3-4, pp. 973-983, 2017.

[3] H. Wang, G. Li, G. Wang, J. Peng, H. Jiang and Y. Liu, "Deep learning based ensemble approach for probabilistic wind power forecasting", Applied Energy, vol. 188, pp. 56-70, 2017.

[4] J. Zhao, Z. Guo, Z. Su, Z. Zhao, X. Xiao and F. Liu, "An improved multi-step forecasting model based on WRF ensembles and creative fuzzy systems for wind speed", Applied Energy, vol. 162, pp. 808-826, 2016.

[5] A. Haque, M. Nehrir and P. Mandal, "A Hybrid Intelligent Model for Deterministic and Quantile Regression Approach for Probabilistic Wind Power Forecasting", IEEE Transactions on Power Systems, vol. 29, no. 4, pp. 1663-1672, 2014.

[6] M. Peng, L. Xie and L. Pietrafesa, "Tropical cyclone induced asymmetry of sea level surge and fall and its presentation in a storm surge model with parametric wind fields", Ocean Modelling, vol. 14, no. 1-2, pp. 81-101, 2006.

[7] Amateurdatascientist.blogspot.com. (2018). Random Forest algorithm. [online] Available at: http://amateurdatascientist.blogspot.com/2012/01/random-forest-algorithm.html [Accessed 20 Sep. 2018].

[8] D. Lee, M. Lim, H. Park, Y. Kang, J. Park, G. Jang and J. Kim, "Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus", China Communications, vol. 14, no. 9, pp. 23-31, 2017.

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

[10] "The fall of RNN / LSTM – Towards Data Science", Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0. [Accessed: 22- Sep- 2018].

[11] D. Learning and E. Memory, "Essentials of Deep Learning : Introduction to Long Short Term Memory", Analytics Vidhya, 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/12/fundamental s-of-deep-learning-introduction-to-lstm/. [Accessed: 22- Sep- 2018].

[12] "Understanding LSTM Networks -- colah's blog", Colah.github.io, 2018. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 29- Sep- 2018].

[13] Y. Jia, Z. Wu, Y. Xu, D. Ke and K. Su, "Long Short-Term Memory Projection Recurrent Neural Network Architectures for Piano's Continuous Note Recognition", Journal of Robotics, vol. 2017, pp. 1-7, 2017.

[14] S.M. Lawan, W. A. W. Z. Abidin, W. Y. Chai, A. Baharun and T. Masri, "Different Models of Wind Speed Prediction; A Comprehensive Review", International Journal of Scientific & Engineering Research, vol. 5, no. 1, 2014.

[15] L. Bisaglia and S. Bordignon, "Mean square prediction error for long-memory processes", Statistical Papers, vol. 43, no. 2, pp. 161-175, 2002.

[16] Neter, John, Michael H. Kutner, Christopher J. Nachtsheim, and William Wasserman. Applied linear statistical models. Vol. 4. Chicago: Irwin, 1996

[17] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2, no. 3, pp. 18-22, 2002

[18] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics, pp. 1189-1232, 2001