Prediction of Wind Farm Power Output Based on an Enhanced Recurrent Neural Network.

E. C. Eze*[1], E.S. Ibrahim[2], C.K. Ihim[1], C.R Chatwin[1], T. C. Yang[1]

[1]Department of Engineering and Informatics, University of Sussex, Brighton, UK.
[2]Department of Geography, Humboldt University, Berlin, Germany.
*Corresponding Author e.eze@sussex.ac.uk

Abstract

This paper studies wind-farm power output prediction based on recurrent neural network. First, a hybrid recurrent neural network (RNN) regularization method involving dropout and long short-term memory (LSTM) is presented. In this model, a regularization scheme is applied to modify and adapt the stochastic nature of the wind. Secondly, a new data structure is presented to the model. Thirdly, the method is developed for wind farm power output (WFPO) prediction. This algorithm is based on the dropout method, which has made WFPO capable of better prediction irrespective of the non-deterministic wind speed. The LSTM solves the RNN limitation of overfitting. The proposed method is demonstrated by investigating the WFPO on a fourteen wind-turbines, provides up to 80% accurate result over ARIMA model.

Keywords: Wind-power Output Prediction, Recurrent Neural Network, LSTM, eLSTM, ARIMA, MSE.

I Introduction.

The intermittency of wind speed introduces challenges to wind power operation during energy integration. Wind-power reliability planning relies on fast and strong wind speed fluctuation and response to system dynamics for better wind power production. The global energy report [1] shows that power generation from the wind rose to 54.6 gigawatts (GW) of installed capacity across the globe. From the report, China and the USA are leading with installed capacity of about 23.4 and 8.2 GW, respectively. Hence, countries like Germany and India are showing a strong appetite for wind energy generation [2]. However, wind is non-deterministic; thus, efficient and reliable wind prediction is required to learn wind variations for sustainable wind power generation. Wind data is stochastic and complex to predict using linear approaches. Forecasting horizons and methods correlates with accuracy. These horizons are short-term, ultra-short-term, medium and long term as discussed extensively by [3]. Real power balance from the wind are classified into four major categories for wind speed prediction. These categories supplement the forecasting horizons accuracy. They are the persistent models [4], Physical methods using numerical weather prediction (NWP) [5], statistical methods, using ARIMA and Bayesian approaches [6, 7] and the artificial intelligence (AI) method. The AI method includes artificial neural networks (ANN) [8-12], support vector regressions (SVR) [10, 13] and recurrent neural networks (RNN) [14-17]. The RNN properties has proven to be among the most efficient procedures [17, 18]. One of the challenges is the vanishing gradient problem on sequence growths. This is because RNNs' properties involve mapping between the input and output sequences for learning purposes. To address this, researchers came up with many variants [19, 20], one of which introduces the concept of the cell state to control these growths; this is called long short-term memory (LSTM). Although LSTM controls these growths by mitigating vanishing gradients, LSTM suffers from overfitting or perfect learning due to the nature of time series data; hence, it requires further regularisation.

Regularisation is a method of controlling model complexities and numerical stability in neural network (NN) systems [21]. To obtain regularisation in a neural network, an additive penalty term is introduced into the cost function; in the form of dropout, and L1L2 to favour simpler models over complex ones [3]. From the literature, the L1 sums the weight coefficients while L2 sums the squared weight coefficients [22]. The dropout method on the other hand prevents co-adaptation on training data and is an efficient method of model averaging in NN, hence, it reduces overfitting of RNNs [23]. The structural implementation in LSTM threatens memory ability in time series prediction, which in turn, results in poor performance. Reference [24] described dropout applied during the prediction of protein structure confirming that [24, 25] experienced smooth training and better prediction due to model averaging implementations. This research is therefore inspired by [12, 23, 26, 27] where the idea of sequential modelling is introduced for time series sequence applying dropout on LSTM to forecast sequence generation for speech, handwriting recognition, and machine translation. In addition, the paper is leveraging on the concept of applied LSTM sequence representation [22] seen in image label annotation. Also, the investigation of the prediction of energy consumption and wind power for households using LSTM [21] and effective learning of measured energy consumption profile. In view of the above, the main contributions of the research are as follows:

- Improvement on RNN by synchronising long short-term memory (LSTM) and the dropout regularization methods to obtain significantly better results when compared to other models.
- Inspection of the dropout prediction from a sequence modelling perspective and implementation of temporal models to solve wind speed time series prediction problems.
- Modelling of a typical wind farm for power output prediction.

To the best of our knowledge, the research work makes the first attempt on both wind-farm power output prediction using a combination of LSTM and dropout methods for prediction.

The rest of the paper is organised as follows; in section II, wind power distributions and the data used in the research is discussed. The derived wind power data and analysis as related to the research is as presented and described in Section III. Section IV presents data preparation and model configuration. The research experimentation is presented in section V. The results evaluation and presentation is as discussed in section VI while the paper conclusion is presented in section VII.

II. Field Data Description

The data is extracted from the PHM society web portal used for the data challenge in 2011 by [28]. Feature extraction, de-noising and filtering were as described by [29]. Figure 1 explores randomly selected data within the wind farm.

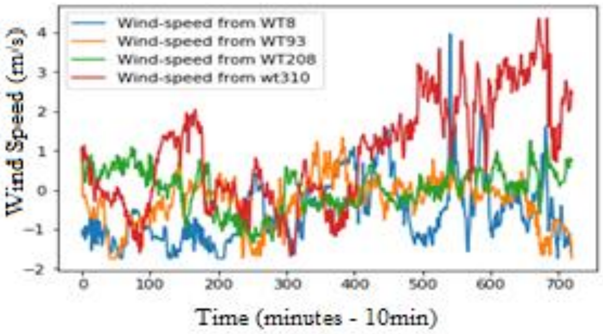

Figure 1. Wind speed of Different Turbines.

Visualisations at each of these wind turbines, shows wind speed plots share similar patterns, although data requires several verifications for predictability.

A. Wind Power Modelling

Wind power modelling over the years has relied on the traditional power curves. This curve models wind power as a function of wind speed at turbine hub height, hence, adjusted for air density. In the literature, the algorithm used for calculating the induction factors are the force dF_N and torque dQ of Eq. (1) and (2) respectively [30], which defines wind power as the conversion of atmospheric forecasts into power output from many turbines or a single turbine.

$$dF_N = B * \frac{1}{2} * \rho U_{rel}^2 cdr(C_l cos\theta + C_d sin\theta) \quad (1)$$

$$dQ = B * \frac{1}{2} * \rho U_{rel}^2 crdr(C_l cos\theta - C_d sin\theta) \quad (2)$$

From the equation, U is the relative wind speed and its angle θ as it approaches the blade. Once torque is computed, the power generation from the wind is the torque multiplied by angular velocity. The rotor disk generates power by the summation of each wind on the blades.

Furthermore, to obtain the characteristics of wind in a wind farm, wind speed data collected at the farm is used. Secondly using statistical estimation – predefined probability distributions. In the previous, anemometer can be used to collect wind speed as in Figure 1, which depicts the variation of wind speed as a function of time. The wind speed is subdivided into bins such that the data point that falls within each bin is counted to form Figure 2, called the wind speed histogram (WSH). In the histogram, each data point is associated with an average wind speed within certain time interval to obtain the total amount of time during which wind is blowing at a speed associated with the bin. Using this approach, the annual energy production, AEP is obtained as shown in Eq. (4).

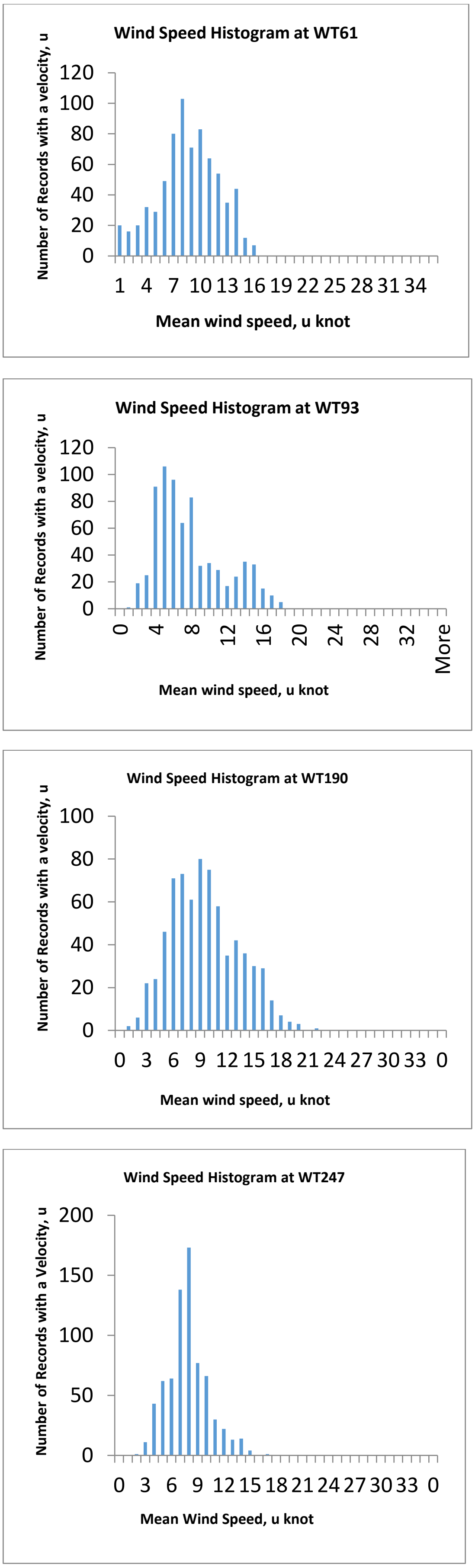

Figure 2: WSH of four Turbines in the Wind farm.

The second approach, which requires statistical methods, shows that the Weibull distribution can provide a good fit to the wind speed histogram. The probability density function (PDF) has two parameters that allow users to adjust the shape of wind power within the wind farm, see Eq. (3). From the equation, c is the scale factor while k is the shape factor. The shape factor controls the location and peak of the distribution while the width of the function is controlled by the scale factor c, often selected at average wind speed [31].

$$f(t) = \frac{k}{c}\left(\frac{t}{c}\right)^{k-1} e^{-\left(\frac{t}{c}\right)^{k}} \qquad (3)$$

$$AEP = \sum_{i=1}^{NB}\left\{\exp[-(\frac{U_{i-1}}{c})^{k}] - \exp[-\left(\frac{U_i}{c}\right)^{k}]\right\}P_w(\frac{U_{i-1}+U_i}{2}) \qquad (4)$$

Wind power density (WPD) is the indicator that shows wind resource capacity in a specific wind farm. It is calculated based on available power in the wind farm and the Weibull parameters method of Eq. (5) [47].

$$\frac{P}{A} = \int_0^{\infty}\frac{1}{2}\rho U^3 f(V)dV = \frac{1}{2}\rho c^3 \Gamma(\frac{k+3}{k}) \qquad (5)$$

The average WPD in terms of wind speed is calculated using Eq. (6). This is because wind power is proportional to the cube of wind speed, hence, the root mean cube (rmc) of wind speed results in Eq. (7)

$$WPD = \frac{\sum_{i=1}^{N} 0.5\rho U_i^3}{N} \qquad (6)$$

$$U_{rmc} = \sqrt[3]{\frac{1}{N}\sum_{i=1}^{N} U_i^3} \qquad (7)$$

N is 14, equivalent to the number of Turbines in the Wind Farm as shown in Figure 3.

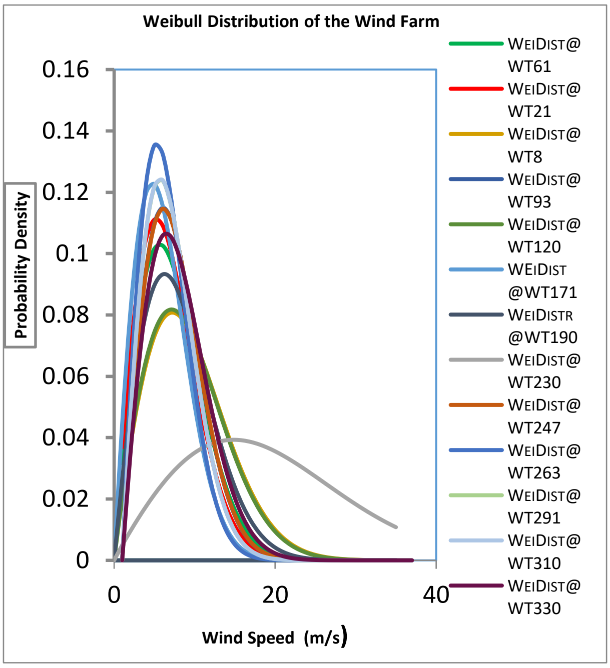

Figure 3: Weibull Distribution of the Wind farm.

B. Wind Farm Power Output Modelling

To demonstrate wind farm power output prediction from a wind farm, a 14-turbine wind farm is considered in this paper, adapted from the design of [32] although not considering the wake effect, however, other configurations are as described in [33]. From the windfarm layout, Eq. (8) models the wind-farm power output.

$$eP_{\alpha,y}(\phi, \delta, \boldsymbol{\mathcal{L}})[\text{Kwh}] = \frac{1}{2}\rho V_{\alpha,y}(\phi, \delta, \mathcal{L})^3 \beta C_p N_m \qquad (8)$$

Where β is the swept rotor area (m^2), C_p is the rotor coefficient of efficiency or capacity factor, which we assume to be 90%, N_m is the efficiency for converting the rotor mechanical power into electricity; assumed to be 92%. ρ is air density and $V_{\alpha,y}$ is the wind velocity of the wind farm.

III. Machine Learning Modelling.

The wind data undergoes transformations using machine learning (ML) to fit to the model. The research employed these three basic steps to achieve this:

- Transform the generated wind speed (WS) data to be stationary using the Dickey-Fuller test of figure 4. Compute first level (d = 1) differencing using the difference between current series (γ_t) and previous series (γ_{t-1}) as in $\Delta\gamma_t = \gamma_t - \gamma_{t-1}$.
- Window model is used to transform the data into a supervised learning problem to have input/output patterns such that at prior steps, observations are used as input to predict observation at the current time step.
- Normalise the data to have a specific scale between -1 and 1
- These transformations are converted after the prediction to return them into their original scale before errors are calculated and scored.

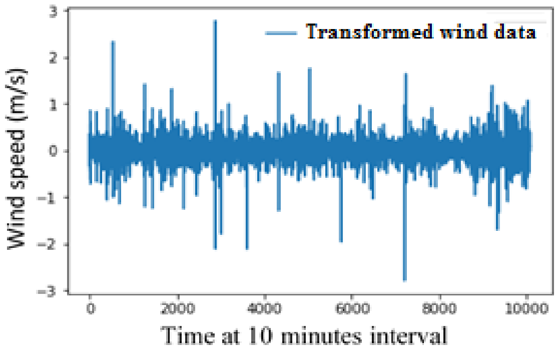

Figure 4: Stationary Test at D=1

The insight gained in Figure 4 led to the data being split; 80% for training whilst 20%, was used for testing the regularised models.

a. Recurrent Neural Network (RNN) Model

This is a type of neural network, which perform the same task for every element of a sequence, with the output dependent on previous computations in a memory-like manner [12] as shown in Figure 5. RNN models are unfolded into full network for training tasks to be applied in the network.

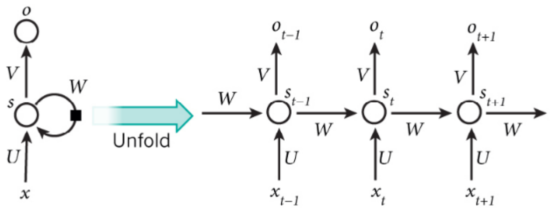

Figure 5: Simple RNN Architecture.

During training, the sequence is back propagated such that the input x_t at time step t is a vector corresponding to s_t which is calculated based on previous hidden state by considering current state input steps of $f(Ux_t + Ws_{t-1})$. Here, s_{-1} is required to calculate the first hidden state that is set to zero. Therefore, the output o_t at step t is set to $o_t = softmax(Vs_t)$. This process fails at long sequence of data and requires LSTM.

b. Long Short Term Memory Model (LSTM).

LSTM improves RNN by the concept of the cell state C_t The cell state keeps updating by having the input i_t and output, o_t perform element-wise multiplication on the input and output of the cell as shown in Figure 6. The previous state of the cell is multiplied by the forget gate f_t, which results in the control of exponential bursts, hence correcting vanishing gradients. Because time series requires single value prediction, the gate activation function i_t, o_t, f_t *uses* sigmoid activation for output blocks. However, the rest of the mathematical configuration is as shown in Eq. (9).

$$\begin{aligned}
f_t &= \sigma_g(\Theta_{xf}x_t + \Theta_{hf}h_{t-1} + b_f)\\
i_t &= \sigma_g(\Theta_{xi}x_t + \Theta_{hi} + b_i)\\
o_t &= \sigma_g(\Theta_{xo}x_t + \Theta_{ho}h_{t-1} + b_o)\\
g_t &= Tanh\,(\Theta_{xg}x_t + \Theta_{hg}h_{t-1} + b_g)\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t\\
h_t &= o_t \odot \mathrm{Tanh}\,(c_t) \qquad (9)
\end{aligned}$$

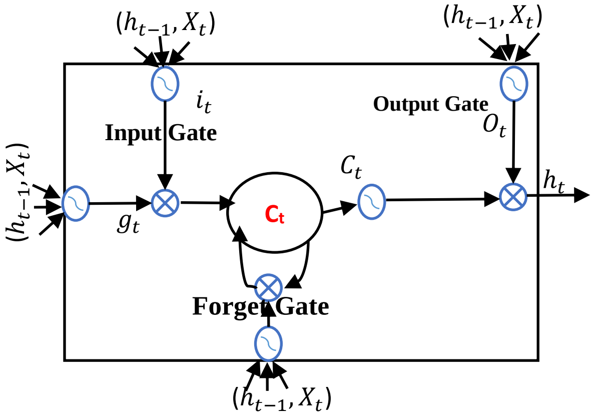

Figure 6: LSTM Architectural design

Hence, LSTM has seen implementation in [19, 34]. Figure 6 depicts the LSTM architecture with a single node cell implementation. LSTM struggles with time series kind of problems since the generalisation often results to overfitting. Therefore, adoption of further regularisation using dropout method is required.

c. Dropout Model.

This is a method proposed by [35] for correcting weight values due to over-adaptation, which in turn causes diminishing accuracy on new samples while training RNN, described [36-38]. This is achieved [39] using the Bernoulli random variable generator δ_i to remove neuron at random from a neural network as described by Eq. (10)

$$\begin{aligned}
E[y^{(i)}] &= \textstyle\sum_{k=1}^{n} w_k \odot x_k^{(i)} E[\,\delta_k] < bE[\delta_k]\\
&= \textstyle\sum_{k=1}^{n} w_k \odot x_k^{(i)} p_k < bp_b \qquad (10)
\end{aligned}$$

Where w_k is the weight-vector and $x_k^{(i)}$ is the neural shape parameter. At independent identical distribution of the δ_i, the q becomes the random number generator that ensures the shape of the network is kept at every iteration while p is the probability of keeping a neuron at random. Therefore, during training Eq. 11 is applied to train individual nodes of a RNN

$$E[y^{(i)}] = \frac{1}{q} \cdot [\textstyle\sum_{k=1}^{n} w_k \odot x_k^{(i)} q + b] \qquad (11)$$

Eq. (11) is the **inverted dropout** representation. During backpropagation, p is element-wise multiplied ʘ by the weight parameters w_k of the reduced nodes to present a zero-out neurons in the hidden layer by reducing co-adaptation among the neurons, this scenario results in an LSTM network that is insensitive to specific neuron weights, thereby influencing better generalisation with relatively less likelihood for overfitting training data.

d. ARIMA Model Configuration.

Machine learning (ML) models like RNN, can be applied directly to the raw data [36, 41, 42]. ARIMA $_{(P, D, Q)}$ models are state space models, which require model improvement due to outliers inherited from the data. The p and q parameters were as modelled in [43, 44]. The p, d, q parameters are obtained using a grid search machine learning method. The grid search technique is tuneable to RMSE statistical quantity for best estimation. In this paper, about 0.82% of evaluated RMSE errors were reported meaning that the search has the best (p, d, q) components at (0.4, 1, 2) respectively.

e. The Proposed Machine Learning Model for wind speed prediction.

The combination of LSTM and Dropout in this paper is eLSTM. Here, the set of inputs are multiplied by a set of a probable-weights $(w_{\Theta i})$ of Eq. (11), which are further processed by individual deep units of 30-hidden layers with output ϴ dimension as in Eq.(12). However, the corresponding Eq. (13) shows the ARIMA model counterpart used to compare the performance of the proposed model.

$$eLSTM_{\Theta}(t) = \frac{1}{q}[g(\textstyle\sum_{i=1}^{\Theta} w_{\Theta i} X_i(t) q + b) \qquad (12)$$

$$ARIMA_{p,d,q} = \textstyle\sum_{i=1}^{\Theta} X_i(t) \qquad (13)$$

The proposed model for comparison is shown using Eq. (14) for demonstration purposes.

$$y_\theta(t) = P\left(f\left(eLSTM_\theta(t), ARIMA_{p,d,q}\right) \mid \{ws_{t-1},, \dots\}\right) \qquad (14)$$

Where *ws* = *wind speed*, t represents a 10min interval of wind data record, while the sigmoid function implements the non-linear output of Eq. (12), hence, derived from Eq. (15)

$$f(x) = \frac{1}{1+ e^{-x}} \qquad (15)$$

Modelling 6-hours ahead as proposed in Eq. (15), where N is the number of hours considered in the dataset, results to 300 minutes ahead as formulated in Eq. (16)

$$W_d^{nod} = \{\dot{W}_s^{nod}[t_h + min], \dot{W}_{t,w,t,h}^{nod}[t_h + min]) \mid min = 1, 2, \dots., 300\text{minutes}\} \qquad (16)$$

Here $h = 1,2,3$ and $nod \in \{position\ of\ turbine\ of\ wind\ Turbine\}$ – not disclosed in the research dataset. Also, $\dot{W}_{t,w,t,h}^{nod}$ *and* $\dot{W}_s^{nod}$ denotes the turbine's node prediction of wind power respectively at time $t = t_h + min$ given wind speed.

V. Experiments

A. Training and Testing Results.

Framing feature series is implemented using a window method that requires samples featured in current time (t) to predict the next time sequence (t+1) knowing prior times t-1, t-2, t-3,…, t-n as input variables. This is because from the literature, LSTM's gating parameters decides whether to update the current state **ɱ** to a new candidate state **ɯ** to learn from the input sequence of the previous state resulting in 6-times, 12-times and 18-times steps ahead respectively. Training and testing accuracy is shown in Figure 7 below for LSTM and the proposed eLSTM counterpart.

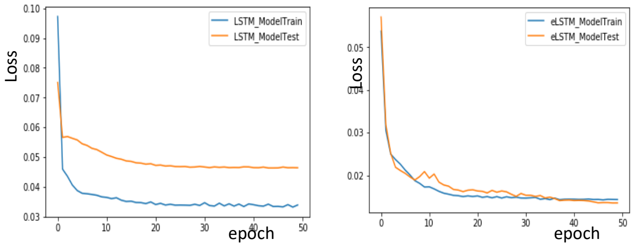

Figure 7a: 18-times steps-ahead Training and Testing.

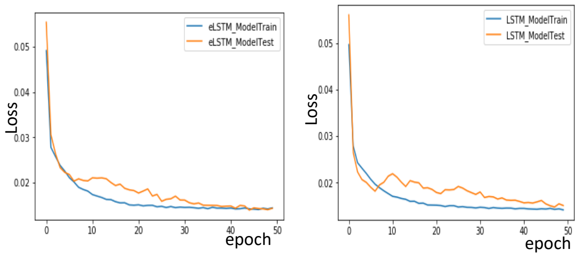

Figure 7b: 12-times steps-ahead Training and Testing.

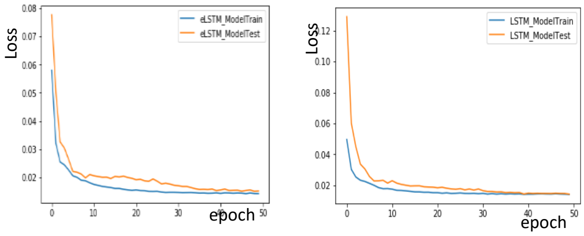

Figure 7c:6-times steps-ahead Training and Testing.

While training RNN, error is imminent. The algorithm experiences errors that need to be minimised. Error function **E(x)** depends on internal learnable parameters of the RNN model [40]. During training, we implemented RMSprop as in [22, 26, 27] for optimisation and error minimisation.

VI. Evaluation and Result Presentation

The paper is evaluated by sequence generation on mean squared error (MSE) as in [23], which is averaged over the features in the training and test set. MSE score is a metric measuring the mean difference between the predicted and the actual features. Comparing regularizers on RNN, the MSE on training and testing is as shown in Table 5. However, in Table 6 ARIMA and the eLSTM performance are compared using the RMSE metric.

$$e_{MSE} = \frac{1}{N}\sum_{t=1}^{N}(X_t - \ddot{X}_t)^2 \qquad (17)$$

Sequence generation on root mean squared error (RMSE) and MSE criterion is as evaluated in Eq. (18) since RMSE punishes large error accumulation over the sequence.

$$ze_{RMSE} = sqrt(e_{MSE}) \qquad (18)$$

Table 5: Regularisation MSE Results.

	eLSTM MSE (%)	LSTM MSE (%)
Exp. 2	0.6241	0.7101
Exp. 4	0.6730	0.7800
Exp. 6	0.6221	0.7410
Exp. 8	0.6100	0.7512
Exp. 10	0.6013	0.7304

Table 6: Comparison of ARIMA and eLSTM.

Dropout proportion	RMSE-ARIMA (%)	RMSE eLSTM (%)
20%	0.7612	0.6228
30%	0.8423	0.5095
50%	0.8531	0.6520

Furthermore, to illustrate the research point, Figure 8 provides a comparison between ARIMA and the proposed model.

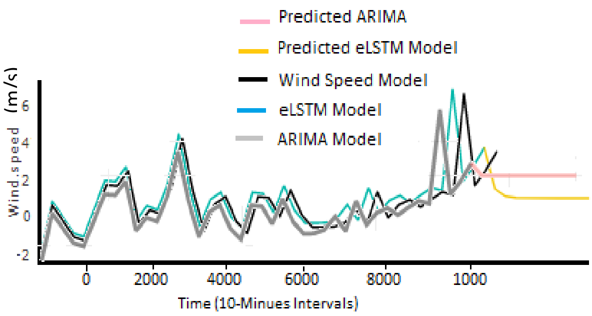

Figure 8: Predicted Wind Speed Model

From the figure, eLSTM tends to learn the data pattern and have better mapping for better prediction than the traditional LSTM.

VII. Conclusion

This paper has presented a comparative study of a hybrid LSTM and dropout regularisation methods for medium-term, 6 hours ahead WFPO prediction based on wind speed and considering the stochastic wind variation and processes. Wind speed data models from PHM are used mainly for training and testing the regression models. The MSE and RMSE, tables (5) and (6), illustrate how the modelling error has improved at various time steps seen in Figure 7 due to the implementation of eLSTM. In addition, the modelling results implements the integration of multiple wind turbine data, which enhances the windfarm performance prediction.

References

[1] CNBC, "China and US lead way with wind power installations, says global energy report," *Sustainable Energy,* Online Report 13 February, 2017 2017.

[2] CNBC, "China and US lead way with wind power installations, says global energy report.," 05 May, 2017 13 Feb 2017

[3] D. Q. Wang, Y. L. Gao, J. X. Liu, C. H. Zheng, and X. Z. Kong, "Identifying drug-pathway association pairs based on L1L2,1-integrative penalized matrix decomposition," (in English), *Oncotarget,* Article vol. 8, no. 29, pp. 48075-48085, Jul 2017.

[4] S. Balluff, J. Bendfeld, and S. Krauter, "Short term wind and energy prediction for offshore wind farms using neural networks," in *2015 International Conference on Renewable Energy Research and Applications (ICRERA)*, 2015, pp. 379-382.

[5] T. Wardah, A. A. Kamil, A. B. S. Hamid, and W. W. I. Maisarah, "Statistical verification of numerical weather prediction models for quantitative precipitation forecast," in *2011 IEEE Colloquium on Humanities, Science and Engineering*, 2011, pp. 88-92.

[6] M. Yesilbudak, S. Sagiroglu, and I. Colak, "A new approach to very short term wind speed prediction using k-nearest neighbor classification," *Energy Conversion and Management,* vol. 69, pp. 77-86, 2013/05/01/ 2013.

[7] H. Liu, H.-q. Tian, and Y.-f. Li, "Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction," *Applied Energy,* vol. 98, pp. 415-424, 2012/10/01/ 2012.

[8] H. Liu, C. Chen, H.-q. Tian, and Y.-f. Li, "A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks," *Renewable Energy,* vol. 48, pp. 545-556, 2012.

[9] Q. Hu, R. Zhang, and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy,* vol. 85, pp. 83-95, 2016.

[10] J. Liu and E. Zio, "SVM hyperparameters tuning for recursive multi-step-ahead prediction," (in English), *Neural Computing & Applications,* Article vol. 28, no. 12, pp. 3749-3763, Dec 2017.

[11] J. M. Hu, J. Z. Wang, and L. Q. Xiao, "A hybrid approach based on the Gaussian process with t-observation model for short-term wind speed forecasts," *Renewable Energy,* vol. 114, pp. 670-685, Dec 2017.

[12] R. Maalej and M. Kherallah, "Improving MDLSTM for Offline Arabic Handwriting Recognition Using Dropout at Different Positions," in *Artificial Neural Networks and Machine Learning – ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II*, A. E. P. Villa, P. Masulli, and A. J. Pons Rivero, Eds. Cham: Springer International Publishing, 2016, pp. 431-438.

[13] S. Salcedo-Sanz, E. G. Ortiz-Garcı´a, Á. M. Pérez-Bellido, A. Portilla-Figueras, and L. Prieto, "Short term wind speed prediction based on evolutionary support vector regression algorithms," *Expert Systems with Applications,* vol. 38, no. 4, pp. 4052-4057, 2011/04/01/ 2011.

[14] C. F. Heuberger, I. Staffell, N. Shah, and N. Mac Dowell, "A systems approach to quantifying the value of power generation and energy storage technologies in future electricity networks," *Computers & Chemical Engineering,* vol. 107, pp. 247-256, Dec 2017.

[15] M. Padhee and R. Karki, "Reliability/environmental impacts of wind energy curtailment due to ramping constraints," (in English), *International Journal of System Assurance Engineering and Management,* Article vol. 8, no. 4, pp. 663-672, Dec 2017.

[16] K. Moustafa, "A clean environmental week: Let the nature breathe," (in English), *Science of the Total Environment,* Article vol. 598, pp. 639-646, Nov 2017.

[17] I. Tanaka and H. Ohmori, "Method Evaluation for Short-Term Wind Speed Prediction Considering Multi Regions in Japan," (in English), *Journal of Robotics and Mechatronics,* Article vol. 28, no. 5, pp. 681-686, Oct 2016.

[18] P. Chatziagorakis *et al.*, "Enhancement of hybrid renewable energy systems control with

neural networks applied to weather forecasting: the case of Olvio," (in English), *Neural Computing & Applications,* Article; Proceedings Paper vol. 27, no. 5, pp. 1093-1118, Jul 2016.

[19] M. Coto-Jimenez and J. Goddard-Close, "LSTM Deep Neural Networks Postfiltering for Enhancing Synthetic Voices," (in English), *International Journal of Pattern Recognition and Artificial Intelligence,* Article; Proceedings Paper vol. 32, no. 1, p. 24, Jan 2018, Art. no. 1860008.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," (in eng), *Neural Comput,* vol. 9, no. 8, pp. 1735-80, Nov 15 1997.

[21] D. T. Mirikitani and N. Nikolaev, "Recursive Bayesian Recurrent Neural Networks for Time-Series Modeling," *IEEE Transactions on Neural Networks,* vol. 21, no. 2, pp. 262-274, 2010.

[22] V. Pham, C. Kermorvant, and J. Louradour, *Dropout Improves Recurrent Neural Networks for Handwriting Recognition*. 2013.

[23] T. Bluche, C. Kermorvant, and J. Louradour, "Where to apply dropout in recurrent neural networks for handwriting recognition?," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR),* 2015, pp. 681-685.

[24] M. Fei and D. Y. Yeung, "Temporal Models for Predicting Student Dropout in Massive Open Online Courses," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW),* 2015, pp. 256-263.

[25] S. Hochreiter, M. Heusel, and K. Obermayer, "Fast model-based protein homology detection without alignment," *Bioinformatics,* vol. 23, no. 14, pp. 1728-1736, 2007.

[26] R. Maalej, N. Tagougui, and M. Kherallah, "Recognition of Handwritten Arabic Words with Dropout Applied in MDLSTM," in *Image Analysis and Recognition: 13th International Conference, ICIAR 2016, in Memory of Mohamed Kamel, Póvoa de Varzim, Portugal, July 13-15, 2016, Proceedings,* A. Campilho and F. Karray, Eds. Cham: Springer International Publishing, 2016, pp. 746-752.

[27] T. Moon, H. Choi, H. Lee, and I. Song, *RnnDrop: a novel dropout for RNNs in ASR*. 2015, pp. 65-70.

[28] P. Society, "phm11 data challenge - condition monitoring of anemometers," https://www.phmsociety.org/competition/pmh/11/problems. , Case study 10/11/2014 2011.

[29] L. Sun, C. Chen, and Q. Cheng, "Feature extraction and pattern identification for anemometer condition diagnosis," *Int. J. Progn. Heal. Manag,* vol. 3, pp. 8-18, 2012.

[30] S. W. Vera Bulaevskaya, Andy Clifton, Wayne Miller, "statistical analysis and modeling for wind power forecasting," *Article,* Conference Preceedings 10/06/2014 10/06/2014.

[31] M. Mazhar, S. Kara, and H. Kaebernick, "Remaining life estimation of used components in consumer products: Life cycle data analysis by Weibull and artificial neural networks," *Journal of operations management,* vol. 25, no. 6, pp. 1184-1193, 2007.

[32] K. Attias and S. P. Ladany, "Optimal layout for wind turbine farms," in *World Renewable Energy Congress-Sweden; 8-13 May; 2011; Linköping; Sweden*, 2011, no. 57, pp. 4153-4160: Linköping University Electronic Press.

[33] X. Gao, H. Yang, and L. Lu, "Investigation into the optimal wind turbine layout patterns for a Hong Kong offshore wind farm," *Energy,* vol. 73, pp. 430-442, 2014/08/14/ 2014.

[34] Y. Fei, J. Hu, K. Gao, J. Tu, W.-q. Li, and W. Wang, "Predicting risk for portal vein thrombosis in acute pancreatitis patients: A comparison of radical basis function artificial neural network and logistic regression models," *Journal of critical care,* vol. 39, pp. 115-123, 2017.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[36] G. Vaca-Castano, S. Das, J. P. Sousa, N. D. Lobo, and M. Shah, "Improved scene identification and object detection on egocentric vision of daily activities," *Computer Vision and Image Understanding,* vol. 156, pp. 92-103, 3// 2017.

[37] S. V. Ravuri and A. Stolcke, "Recurrent neural network and LSTM models for lexical utterance classification," in *INTERSPEECH,* 2015, pp. 135-139.

[38] T. N. Sainath *et al.*, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks,* vol. 64, pp. 39-48, 2015.

[39] E. Phaisangittisagul, "An Analysis of the Regularization between L2 and Dropout in Single Hidden Layer Neural Network," *IEEE Computer society,* Conference Paper vol. DOI 10.1109/ISMS.2016.14, no. 2166-0670/16, p. 6, 2016 2016.

[40] A. S. Walia, "Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent," Blog 2017.

[41] S. Ryu, S. Kim, J. Choi, H. Yu, and G. G. Lee, "Neural sentence embedding using only in-domain sentences for out-of-domain sentence detection in dialog systems," *Pattern Recognition Letters,* vol. 88, pp. 26-32, 2017.

[42] U. Güçlü and M. A. J. van Gerven, "Modeling the Dynamics of Human Brain Activity with Recurrent Neural Networks," (in English), *Frontiers in Computational Neuroscience,* Original Research vol. 11, no. 7, 2017-February-09 2017.

[43] L. Hui, T. Hong-qi, and L. Yan-fei, "Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction.," *Applied Energy,* vol. 98, pp. 415-424, 2012.

[44] R. G. Kavasseri and K. Seetharaman, "Day-ahead wind speed forecasting using f-ARIMA models," *Renewable Energy,* vol. 34, no. 5, pp. 1388-1393, 5/5/2009 2009.