Azerbaijan Journal of High Performance Computing, Vol 5, Issue 2, 2022, pp. 254-272
https://doi.org/10.32010/26166127.2022.5.2.254.272

Short-term Wind Speed Forecasting Using Deep Variational LSTM

Navid Atashfaraz[1] and Mohammad Manthouri[2]

[1]North Tehran Branch Azad University, Tehran, Iran, n.atashfaraz@iau-tnb.ac.ir
[2]Shahed University, Tehran, Iran, mmanthouri@shahed.ac.ir

*Correspondence: Mohammad Manthouri, Shahed University, Tehran, Iran, mmanthouri@shahed.ac.ir

Abstract

Wind speed and power at wind power stations affect the efficiency of a wind farm, so accurate wind forecasting, a nonlinear signal with high fluctuations, increases security and better efficiency than wind power. We are looking for wind speed for a wind farm in Iran. In this research, a combined neural network created from variational autoencoder (VAE), long-term, short-term memory (LSTM), and multilayer perceptron (MLP) for dimension Reduction and encoding is proposed for predicting short-term wind speeds. The data used in this research is related to the statistics of 10 minutes of wind speed in 10- meter, 30-meter, and 40-meter wind turbines, the standard deviation of wind speed, air temperature, and humidity. To compare the proposed model (V- LSTM-MLP), we implemented three deep neural network models, including Stacked Auto-Encoder (SAE), recurrent neural networks (Regular LSTM), and hybrid model Encoder-Decoder recurrent network (LSTM-Encoder-MLP) presented on this dataset. According to the RMSE statistical index, the proposed model is worth 0.1127 for a short time and performs better than other types on this dataset.

Keyword: LSTM, VAE, MLP, Wind Speed Prediction, Dimension Reduction, Encoder-Decoder

1. Introduction

Wind energy is a type of renewable energy that has the potential to provide a sustainable energy source. It has many advantages, including reducing pollution and protecting the environment. Additionally, with the lack of fossil energy resources, governments support using renewable energy. However, due to random fluctuations, intermittent, nonlinear, and uncertainty in wind data, consuming energy and its integration with electricity networks in economic development takes much work (Meng et al., 2016). This uncertainty about wind speed and power can jeopardize the reliability and quality of electrical systems; therefore, the main issues of electricity network integration, such as balanced management and storage capacities, can be questioned (Georgilakis, 2008; Smith et al., 2007; Sder et al., 2007).

Study of the previous studies in this field, it can be concluded that wind speed prediction methods are divided into five general categories. These categories include stability, physical, statistical, and artificial intelligence methods such as neural networks (Philippopoulos & Deligiorgi, 2012), Fuzzy methods (Eseye et al., 2017), and combined methods which have been used and improved to ameliorate predictive accuracy. The sustainability method is the cheapest and most straightforward method of forecasting applied in wind farms. It operates to consider the future wind speed equal to the current wind speed. As the forecasting horizon increases, the performance of the sustainability method decreases, so this model is only reliable for short-term goals. Physical methods require information such as temperature, pressure, obstacles, and roughness. Predict wind speed; this method's problem is the calculation's complexity and time spent on the process.

Statistical methods such as AR and ARMA examine mathematical pattern correlation Between time series data and show better performance in short-term forecasting than in sustainability and physical methods. Artificial intelligence-based approaches, including artificial neural networks (ANN) (Welch et al., 2009; Bhaskar & Singh, 2012), support vector machine (SVM) (Zhou et al., 2011), and fuzzy logic (Damousis et al., 2004) have led to new ways to predict wind speeds in the short-term. Artificial neural networks can be divided into two categories, shallow architectures, and deep learning models.

A) Shallow models: including feedforward (Welch et al., 2009; Bhaskar and Singh; 2012), recurrent neural networks (Barb2ounis & Theocharis, 2007), and wavelet neural networks (Ricalde et al., 2011) are designed to use a hidden layer to capture time series features.

B) Deep Learning Architecture: These models can teach several layers of high-performance hidden computing units. This recent study (Khodayar et al., 2019; Liu et al., 2018) has effectively predicted short-term wind speeds. Hybrid methods combine several algorithms and methods that use the unique power of each algorithm to provide a better output. According to work done in wind speed forecasting, each forecasting model has its strengths and weaknesses compared to the others. Therefore, many hybrid models have been proposed that take advantage of the strengths of different methods. A recent study (Memarzadeh & Keynia, 2020; Tascikaraoglu & Uzunoglu, 2014) has shown that merging several methods could lead to a highly improved output.

For example, (Z. Liu et al. 2020) enhance the performance and stability of wind speed forecasting. This state runs a pre-processing data strategy to control noises and a multi-objective optimization algorithm to achieve predictive accuracy and stability. The results of their prediction model perform incredibly better than other types. A recent study used recurrent neural networks with optimization algorithms; for example (Vinothkumar & Deeba, 2019) proposed two models, including recurrent neural networks and support vector machines for prediction. To optimize the parameters of the models, they use a combination of particle swarm optimization algorithms and ant lion optimization algorithms. Their simulation results show that their research could have more efficient outcomes.

Autoencoder neural networks have been used for prediction in recent years. (Khodayar et al. 2017) The rough set theory incorporates the autoencoder's deep models to improve the prediction's accuracy. This study introduces the four architectures: SAE, SDAE, RSAE, and RSDAE. Among the Autoencoder networks provided by them, the RSDAE network has a more accurate answer due to the use of the rough set theory and the elimination of noise. Because of randomness, high fluctuations, and unpredictability of wind speeds, the issue of noise elimination is essential in wind speed data, and it produces the preferred output.

(Peng et al., 2020) provides a deep learning model by removing noise. Their noise cancellation model wavelet soft threshold denoising (WSTD) filters out additional data from time series data. The GRU neural network is also used for prediction.

They have shown that this method increased the predictive speed. Recurrent neural networks have shown their strength in many time series prediction tasks in recent years. In most cases, these networks are commonly used; for example, (Hu & Chen, 2018) offers a new nonlinear hybrid model (LSTM-DE-HELM) to improve wind speed forecasting performance. To enhance the performance of the Extreme Learning Machine, they incorporate a biological, neurological property into an ELM activating neurons and also use a differential evolutionary algorithm to optimize and determine the number of LSTM layers. Finally, they compared the created model with several single and combined models and explained that the performance of their model is functionally better in predicting wind speed.

(Mirzapour et al. 2017) Provides a Potential of k-Means Clustering-Based Fuzzy Logic for the Prediction of Temperature in an Ambient Atmosphere. The study uses the database of maximum temperature, corresponding pressure, relative humidity, wind speed, and historical temperature to develop a fuzzy rule base domain prediction methodology to estimate the next-day maximum temperature for Mumbai, India. They have proved that this method increases the predictive speed. (Mir et al., 2019) provides a new hybrid prediction engine consisting of three main stages: empirical state analysis, an intelligent algorithm, and back propagation neural network. The structure of their proposed model is based on the non-stationary nature of the wind speed signal. The effectiveness of the proposed model is tested with real-world hourly data from wind farms in Spain and Texas.

To improve the accuracy of short-term wind speed forecasting, a deep hybrid neural network model was developed, which includes three modules: the LSTM Recurrent Neural Network, the probabilistic VAE neural network, and the MLP neural network for predicting time series. We first teach an Encoder-Decoder model that includes the LSTM neural network and the VAE probabilistic network to reduce the dimensions of the input set. Combining LSTM, a network suitable for time series data, with VAE layers, a production model that applies a regular geometry to the data according to probability theory, could help us reduce the size of the input set. Continuing this process, we are in the next step with a single feature built by VAE-LSTM that (in addition to improving predictive accuracy) significantly reduces model execution time.

Ultimately, the output of the previous step injects into an MLP as input, by which the wind speed is predicted. To validate the performance of the proposed model to predict short-term wind speeds, the dataset from the first 660 kW wind turbine in the Lutak region of Zabol city, which was commissioned by the New Energy Organization of Iran in 2006, was used. The results show that the proposed combined neural network performs well in predicting wind speed. The paper is organized as follows: In the second part, the framework of the proposed model is presented; in the third part, the presented model is to predict wind speed as described; in the fourth part, data sets and simulation results are presented.

Thanks to increased calculation power and storage of devices, this has become even easier nowadays. The calculation and storage of processes in local devices and not on servers is called edge computing and several works were already done in this field.

2. The Framework of the Proposed Model

This paper proposes a deep hybrid neural network model developed from LSTM, VAE, and MLP (V-LSTM- MLP), to predict wind speed. Figure 1 shows the framework of the proposed model, and the steps are as follows:

1. In the first step, 13 climate characteristics are used as input to the VAE-LSTM hybrid network to per- form the operation of reducing the initial input signal dimensions. Details of LSTM and VAE are pro- vided in Sections 3.1 and 3.2, respectively.
2. At this stage, the first 13 features convert into six features by VAE-LSTM, which we extract.
3. Therefore, in the new data set, we have two features: 1 – 6 Encoded features and 2- The maximum wind speed in 40 meters of wind.
4. New dataset as input to MLP network for prediction. Details of MLP are provided in Section 3.3.
5. The number of encoding features is systemically available to us, and after a few experiments, we have reached six features for the optimal model.

3. Methodology

This section explains LSTM, VAE, and MLP and their architecture. We also look at formulas and their relationships and see their architecture in forms.

3.1. Long Short-Term Memory (LSTM)

LSTM neural networks are a particular type of recurrent neural network that can learn long-term dependencies. These networks were first introduced by (Hochreiter & Schmidhuber, 1997). The goal of designing LSTM networks was to solve the obstacle of long-term affiliation (Vanishing gradient). It is significant to note that retaining information for long periods is the default and regular conduct of LSTM networks. Their structure is designed so that they learn distant information properly, which is a unique attribute of their structure. This network is appropriate for taxonomy, processing, and predicting time series. It trains the pattern by using backpropagation. Due to the three gates, i.e., input gates, output gates, and forget gates, the LSTM network can add or remove data to the cell situation. Updating the situation of the cell and calculating the output of the LSTM network can be calculated as follows:

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \tag{1}$$
$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \tag{2}$$
$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \tag{3}$$
$$\widetilde{C} = tanh(x_t U^g + h_{t-1} W^g) \tag{4}$$
$$C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t) \tag{5}$$

Here i, f, and o call the input, forget, and output gates. W is the recurrent connection between the previously hidden layer and the current hidden layer. U is the weight matrix connecting the inputs to the running hidden layer. The $\tilde{C}$ element is a "candidate" hidden state based on the current input and the previously hidden state, and C is the unit's internal memory. LSTM architecture is shown in Fig 2.

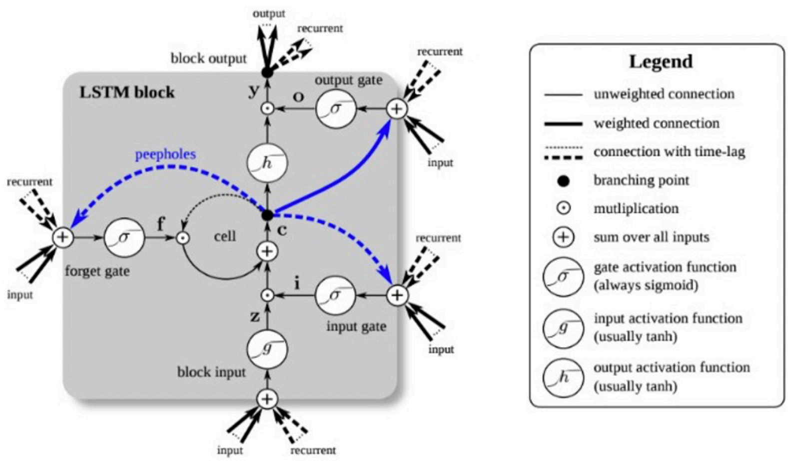


Fig. 2 Framework of LSTM model. (Long Short-Term Memory (LSTM), 2020)

3.2. Variational Autoencoder (VAE)

The variational autoencoder (Kingma & Welling, 2014) is a generative model based on a regularized prescription of the standard autoencoder. Variational autoencoders compress the input information into a constrained multivariate latent distribution (encoding) to reconstruct it as accurately as possible (decoding).

Let us consider the samples of some continuous or discrete variable x from the dataset, and we hypothesize that an unobserved continuous random variable z generates the variable x. An auto-encoder (Goodfellow et al., 2016; Patidar et al., 2017) usually consists of two parts, an encoder representation function (latent representation) and a decoder that produces a reconstruction function. Variational auto-encoder is an unsupervised learning approach for modeling complicated significant data distributions.

Now we describe some notions as follows:

X= data that we demand to model

Z= hidden variable

P (X) = eventuality distribution of the data.

P (z) = eventuality distribution of the hidden variable.

P (X | z) = distribution of generating data given hidden variable.

At this point, the purpose is to model the data; accordingly, we want to find P(X). Using the law of probability, we could discover it concerning z as follows:

$$P(X) = \int P(X|z)P(z)dz \tag{6}$$

The idea of VAE is to derive P (z) using P (z| X). In VAE, as its name suggests, we derive P (z | X) using Variational Inference (VI). Variational Inference (VI) is one of the desired method choices in Bayesian inference, and the other is the Markov Chain Monte Carlo (MCMC) procedure. The primary idea of VI is to set the inference by approaching it as an optimization problem. In statistics, variational inference (VI) is a technique to approximate complex distributions. The idea is to set a parametrized family of distributions (for example, the family of Gaussians, whose parameters are the mean and the covariance) and to look for the best approximation of our target distribution among this family.

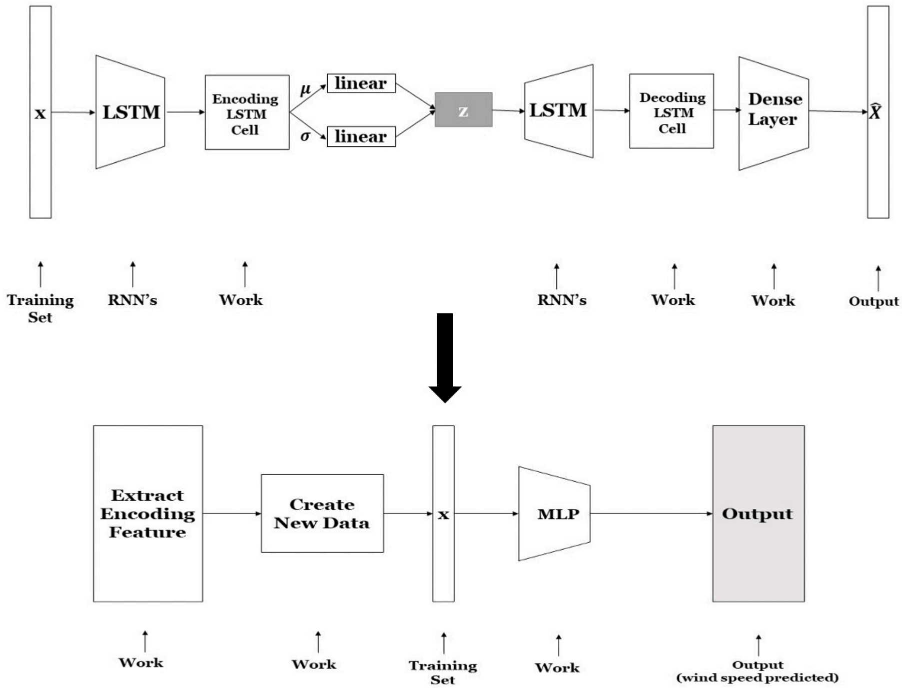


Fig. 1: Framework of the proposed method.

The best element in the family is the one that minimizes a given approximation error measurement (most of the time, the Kullback-Leibler divergence between approximation and target) and is found by gradient descent over the parameters that describe the family. We intend to derive P (z | X) using Q (z | X). The KL divergence is then formulated as follows:

$$D_{KL}[Q(z|X)||P(z|X)] = \sum_z Q(z|X) log\frac{Q(z|X)}{P(z|X)} = E\left[log\frac{Q(z|X)}{P(z|X)}\right] = E[\log Q(z|X) - P(z|X)] \quad (7)$$

There are two objects that we have yet to use, namely P(X), P (X | z), and P(z). However, with Bayes' rule, we could emerge it in the following equation:

$$D_{KL}[Q(z|X)||P(z|X)] = E\left[\log Q(z|X) - log\frac{P(X|z)P(z)}{P(X)}\right] = E[\log Q(z|X) - (\log P(X|z) + \log P(z) - \log P(X))] = E[\log Q(z|X) - \log P(X|z) - \log P(z) + \log P(X)] \quad (8)$$

Notice that the expectancy is over z, and P(X) does not pertain to z, so that we could move it out of the expectation.

$$D_{KL}[Q(z|X)||P(z|X)] = E[\log Q(z|X) - logP(X|z) - \log P(z) + \log P(X)$$

$$D_{KL}[Q(z|X)||P(z|X)] - \log P(X) = E[\log Q(z|X) - logP(X|z) - \log P(z) \quad (9)$$

Looking carefully at the right side of the equation, we notice that it could be rewritten as another KL divergence. So, let us do that first by rearranging the mark.

$$D_{KL}[Q(z|X)||P(z|X)] - \log P(X) = E[\log Q(z|X) - logP(X|z) - \log P(z)$$

$$\log P(X) - D_{KL}[Q(z|X)||P(z|X)] = E[\log P(X|z) - (logQ(z|X) - \log P(z))] = E[\log P(X|z)] - E[logQ(z|X) - \log P(z)] = E[\log P(X|z)] - D_{KL}[Q(z|X)||P(z)] \quad (10)$$

And this is it, the VAE objective function:

$$\log P(X) - D_{KL}[Q(z|X)||P(z|X)] = E[\log P(X|z)] - D_{KL}[Q(z|X)||P(z)] \quad (11)$$

At this stage, we have these three steps:

Q (z | X) = that projects our data X into hidden variable space.

z= latent variable

P (X | z) = that generates data given latent variable.

Q (z| X) is the encoder net, z is the encoded delegacy, and P (X | z) is the decoder net. Variational Autoencoder architecture is shown in Fig. 3.

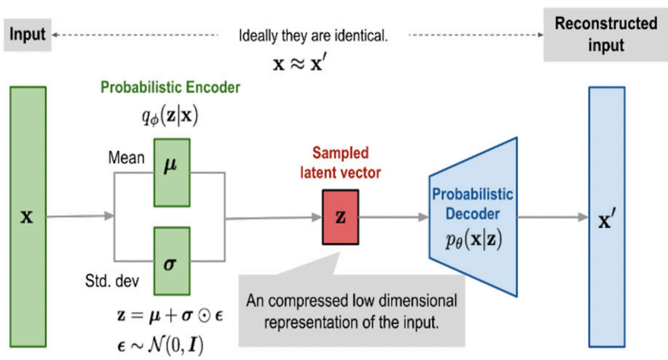


Fig. 3: The architecture of the Variational autoencoder model. (Weng, 2018)

3.3. Multilayer Perceptron Neural Networks (MLP)

In this article, we perform a multilayer perceptron (MLP) algorithm on wind speed prediction. The MLP model is a particular case of the well-established ANN model (Kawamoto et al., 1989). An MLP consists of at least three layers of knots: an input layer, a hidden layer, and an output layer. Exclude the input nodes; each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning method called backpropagation for training. (Rosenblatt 1961; Rumelhart et al. 1986) Its numerous layers and nonlinear activation individualize MLP from a linear perceptron. It can distinguish data that is not linearly separable (Cybenko, 1989). this mode applies in many previous studies in renewable energy, e.g. (Deo & Samui, 2017). In its pivotal form, where no optimizer algorithm merges, the multilayer feedforward perceptron backpropagation learning algorithm consists of the input layer, hidden layer, and output layer, and it is considered one of the beloved neural network architectures. In an error backpropagation algorithm during training, each neuron is connected to the neurons in the opposite layer by weights, called a fully-connected neural network.

The sigmoid and the linear activation functions apply in the hidden and output layer, respectively. More detailed narrations about the MLP method can be spotted in (Ghorbani et al., 2013) MLP architecture, shown in Fig. 4.

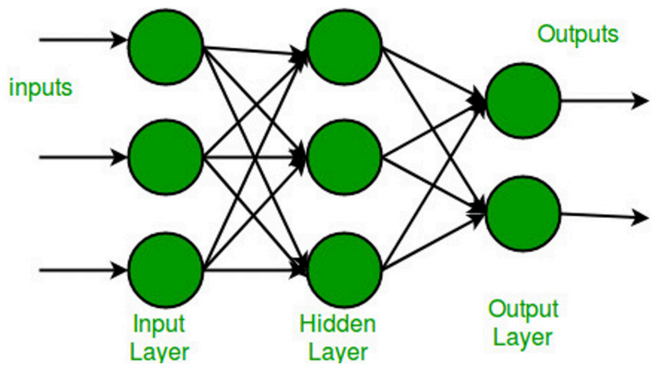


Fig. 4: The architecture of MLP. (Multilayer Perceptron Learning in TensorFlow, 2021)

4. Simulation and Results

This section describes the dataset, the study area, and the elements used in the model simulation. It describes the performance evaluation criteria of the models (Absolute Deviation (MAD), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and finally, the output of four deep models). These models include the proposed three models, REGULAR LSTM, SAE, and ENCODER-DECODER-LSTM, as the standard networks discuss and review.

4.1. Dataset, Study Area and Simulation Parameters

This study deals with the "Lutak region" of "Zabol," a city in "Sistan" and "Baluchestan" province in Iran. This region has been known as a windy area in Iran. Fig. 5 shows the location of this region inside Iran. The data set Includes an annual wind speed of 10 minutes be- tween 2006 and 2010.

At this station, 660 kWh and 660 volts with transverse power are injected into the 20 kV network. Fig. 6 shows the wind speeds in the different meters of wind turbines. In this study, 80% of the total data belonged to the training, and the remaining 20% were introduced to the model as test data—137,000 wind speeds measured at 10-minute intervals. Therefore, sufficient data are available to teach and test the proposed approach. We use different weather features in the implementation of the model. 13 climate characteristics are entered as inputs to predict the maximum wind speed within forty meters of the wind tower. Table 1 shows the input and output variables and their statistical information. Fig. 7 shows the actual wind speed data in 40 m wind turbines for one year in 4 seasons.

Fig. 5: Map of Iran, including the case study area.

4.2. Index of Performance

Mean Absolute deviation (MAD): Mean Absolute Deviation measures the precision of the prediction by averaging the alleged error (the absolute value of each error). MAD is useful when measuring prediction errors in the same unit as the original series. The amount of MAD can be calculated using the following formula.

$$MAD = \left(\frac{\sum|Y_t - \hat{Y}_t|}{N}\right) \quad (12)$$

Table 1: Input and Output Variable.

variable	high	mean	low	position
TEMPERATURE	45.60	22.94	-9.30	Input
REL HUMIDITY	100	27.58	4	Input
WIND SPEED MAX10m	29	7.82	0	Input
WIND SPEED MIN10m	12.30	14.3	0	Input
WIND SPEED AVG10m	18	5.25	0	Input

WIND SPEED SDev10m	6.12	0.78	0	Input
WIND SPEED MAX 30m	31.20	28.9	0	Input
WIND SPEED MIN 30m	17	4.97	0	Input
WIND SPEED AVG 30m	23.40	8.7	0	Input
WIND SPEED SDev30m	7.87	0.80	0	Input
WIND SPEED MIN 40m	18.70	5.51	0	Input
WIND SPEED AVG 40m	24.10	7.61	0	Input
WIND SPEED SDev40m	7.90	0.79	0	Input
WIND SPEED MAX 40m	32.10	9.75	0	Output

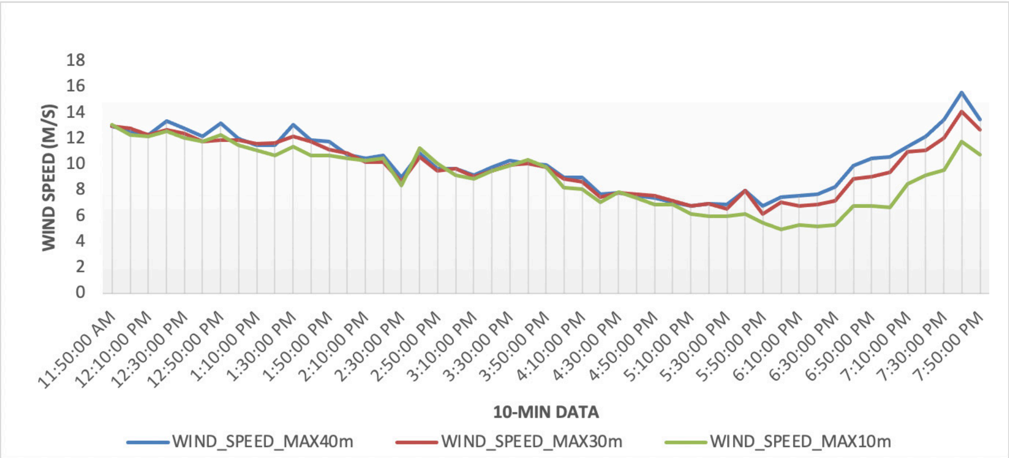


Fig. 6: Original Wind Speed.

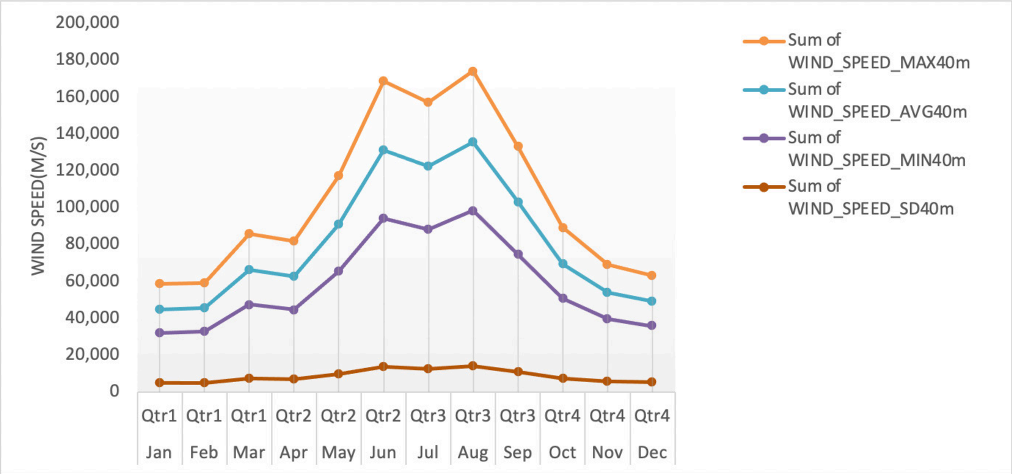


Fig. 7: Original Wind Speed in Monthly quarter.

Mean Squared Error (MSE): mean squared error (MSE) of an estimator (of a method for estimating an unobserved value) measures the average of the squares of the errors, that is, the average squared variation between the estimated values and the actual value. The formula for mean squared error is given below:

$$MSE = \frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{N} \tag{13}$$

Root Mean Squared Error (RMSE): Root Mean Squared Error is an absolute error measure that squares the deviances to keep the positive and negative deviations from canceling one another out. The composition for calculating RMSE:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{N}} \tag{14}$$

Mean Absolute Percentage Error (MAPE): Mean Absolute Percentage Error (MAPE) is calculated using the absolute error in each cycle divided by the observed values for that cycle. Then, averaging those fixed percentages. This approach is functional when the size or size of a prediction variable is significant in evaluating the accuracy of a prediction. MAPE indicates how much error in predicting compared with the exact amount.

$$\mathrm{MAPE} = \frac{\sum \frac{|\mathrm{Y_t} - \hat{\mathrm{Y}}_\mathrm{t}|}{\mathrm{Y_t}} * 100}{\mathrm{N}} \tag{15}$$

Y_t Is the factual value of a point for a given time cycle t, N is the entire number of fixed points, and $\hat{Y}_t$ is the fixed forecast value for the time cycle t.

4.3. Numerical Result and Discussion

As it turns out, the time series of wind speed is a nonlinear and random signal. This paper presents a combined neural network model, including VAE, LSTM, and MLP, to solve problematic time series, reduce training time, and improve output in predicting wind speed. Initially, wind speed data is sent as input to an ENCODER-DECODER model consisting of LSTM and VAE. After learning the ENCODER-DECODER model, we extract the encoded features. We have many encrypted features and have done a few tests to improve the output. Then a matrix with 6 encrypted features is created and sent as input to the MLP neural network, and finally, the output of the wind speed forecast within forty meters of the wind turbine. One of the features of the work provided is the reliance on the use of LSTM and VAE networks as a suitable method for data encryption because 1- LSTM is a suitable network for time series data, 2- VAE network is a probabilistic production model which implements a regular geometry on the data and allows for proper sampling of the data set.

After the encryption operation, the training time is significantly reduced in addition to improving the error rate compared to other hybrid and single models.

So, to predict wind speeds using the method described in this article, you should follow the steps below:

In the first step, 13 climate characteristics are input to the VAE-LSTM hybrid network to perform dimensional reduction and encryption of the initial input signal.

In this step, the 13 primary features are converted to 6 encrypted features by VAE-LSTM, which we extract.

Then, in the new data set, we have 7 features: 1- Six encrypted features 2- The maximum wind speed within forty meters of the wind tower, which is our target variable for prediction.

We target the new data set as input to the MLP network to predict the variable.

In this study, to evaluate the proposed V-LSTM-MLP model, we compared it with 3 deep LSTM, SAE (Khodayar et al., 2017), and LSTM-ENCODER-MLP models. Table 2 shows the error criteria for all models. Fig. 8 shows the output of accurate data and the output predicted by all models. We implemented the SAE model, introduced by (Khodayar et al., 2017), on this data set, and the output that is shown in Fig. 11 performs better than the LSTM-ENCODER-MLP, one of the reasons that SAE performs better than the LSTM-ENCODER-MLP neural network is that the LSTM-ENCODER-MLP has an encrypted feature as output. Still, we set up the SAE neural network in a way that, like the proposed model, includes six encrypted features. The output of accurate data and data predicted by ENCODER-DECODER-LSTM is shown in Fig. 12. However, the single Regular-LSTM model has better output than the two models (SAE, LSTM-ENCODER-DECODER) encrypting the data. The main reasons for the better output of Regular LSTM memory architecture are to solve the problem of vanishing gradient by this network and its regularity; the output for Regular-LSTM can be seen in Fig. 10.

As shown in the table, the proposed RMSE and MAPE models with 0.0982 and 57.3%, respectively, perform better than other models on this data set. The output of the proposed model is shown in Fig. 9, and Fig. 13 shows the PRECENTAGE ERROR of the proposed model relative to the actual wind speed data. To better understand and view the data in Fig. 14, a SCAATER output of the accurate data set and data predicted by V-LSTM-MLP is provided. The output of the Regular LSTM model and our proposal, which had a better output than the other two models, are shown in Fig. 15 for comparison. The topic that distinguishes our model from other models is the proper encryption with the VAE-LSTM network, which can be determined by the amount of RMSE and MAPE indicators VAE is a productive neural network to apply a regular geometry to the data and allow for appropriate sampling of the data set.

Table 2: Performances Index of short-term forecasting methods

Models	RMSE	MSE	MAD	MAPE
Encoder–Decoder-LSTM	0.1715	0.0294	0.1319	142.35%
Stacked Auto-Encoder (SAE) (Khodayar et al., 2017)	0.1470	0.0216	0.1102	123.24%
Regular LSTM	0.1309	0.0171	0.0826	63.45%
Proposed Model (V-LSTM-MLP)	0.1127	0.0127	0.0827	40.70%

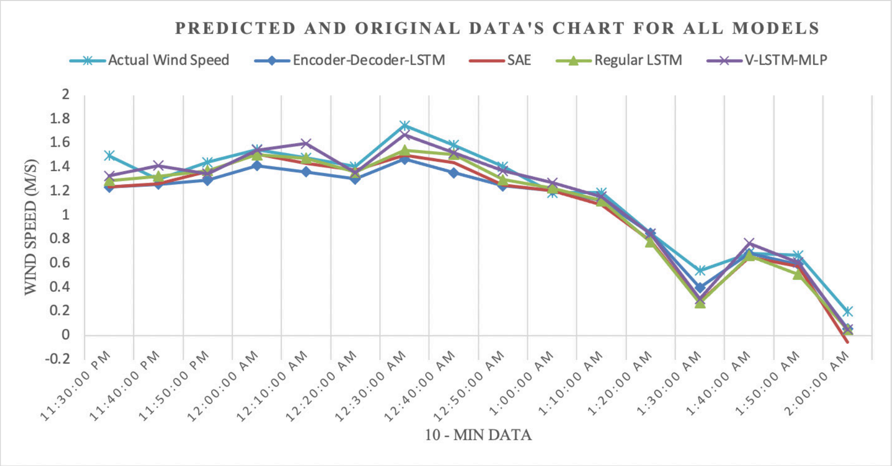


Fig. 8: ALL MODELS Output: 10 min Prediction.

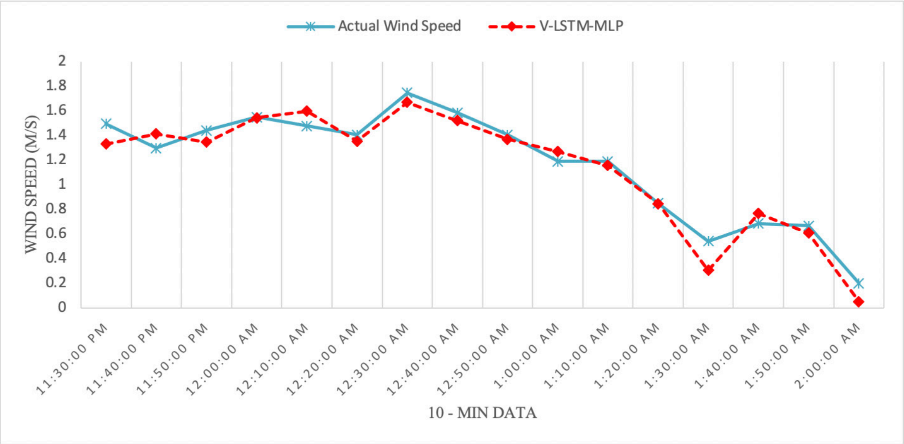


Fig. 9: V-LSTM-MLP MODELS Output: 10 min Prediction.

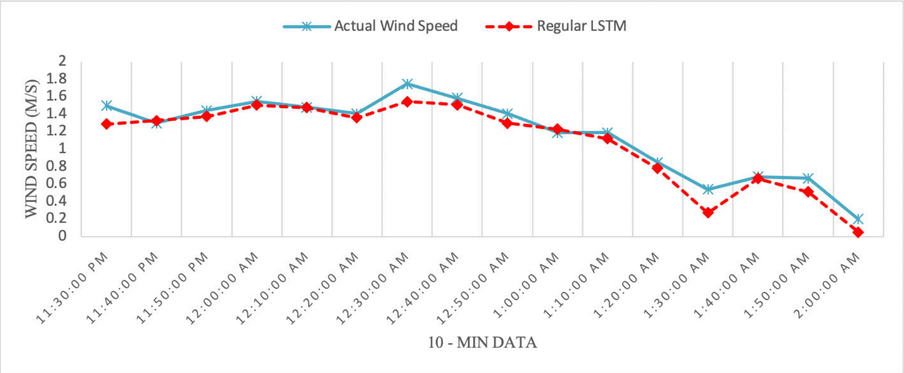


Fig. 10: Regular LSTM MODELS Output: 10 min Prediction.

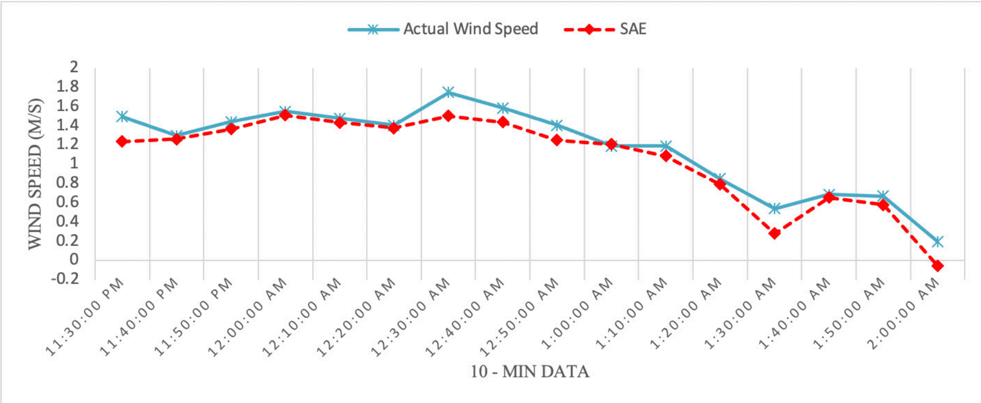


Fig. 11: SAE MODELS Output: 10 min Prediction.

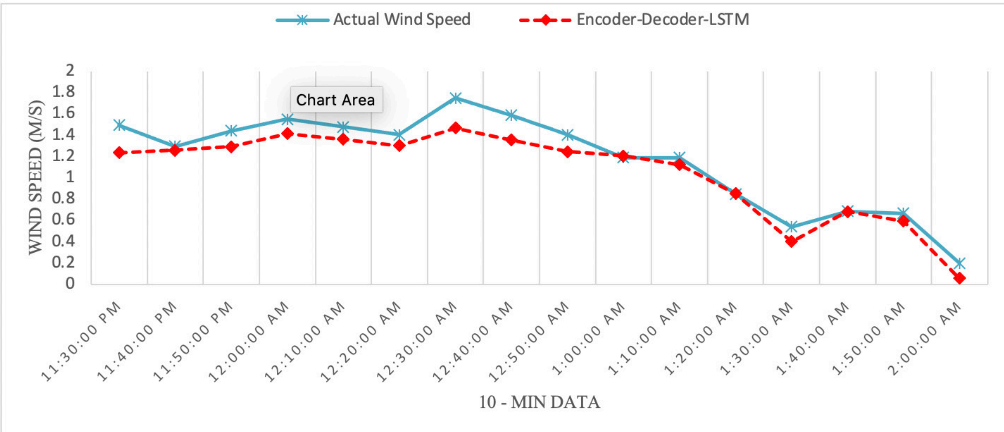
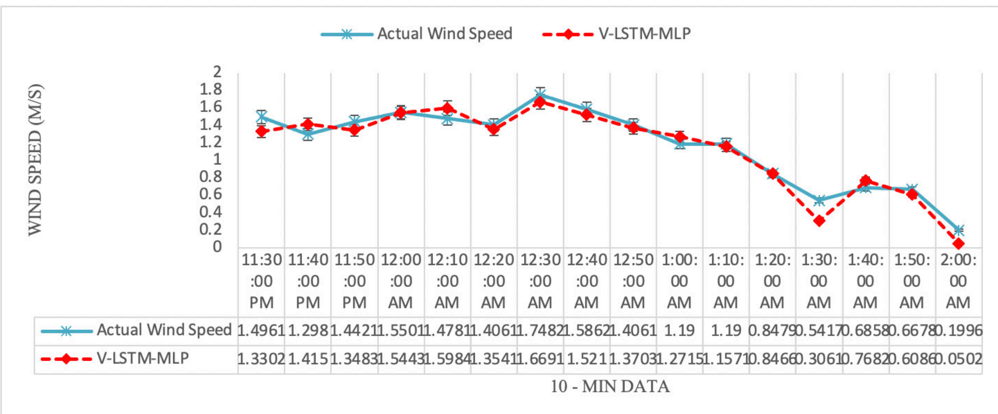


Fig. 12: ENCODER-DECODER-LSTM Error Output: 10 min Prediction.



	11:30:00 PM	11:40:00 PM	11:50:00 PM	12:00:00 AM	12:10:00 AM	12:20:00 AM	12:30:00 AM	12:40:00 AM	12:50:00 AM	1:00:00 AM	1:10:00 AM	1:20:00 AM	1:30:00 AM	1:40:00 AM	1:50:00 AM	2:00:00 AM
Actual Wind Speed	1.4961	1.298	1.4421	1.5501	1.4781	1.4061	1.7482	1.5862	1.4061	1.19	1.19	0.8479	0.5417	0.6858	0.6678	0.1996
V-LSTM-MLP	1.3302	1.415	1.3483	1.5443	1.5984	1.3541	1.6691	1.521	1.3703	1.2715	1.1571	0.8466	0.3061	0.7682	0.6086	0.0502

Fig. 13: V-LSTM-MLP and Percentage Error Output: 10 min Prediction.

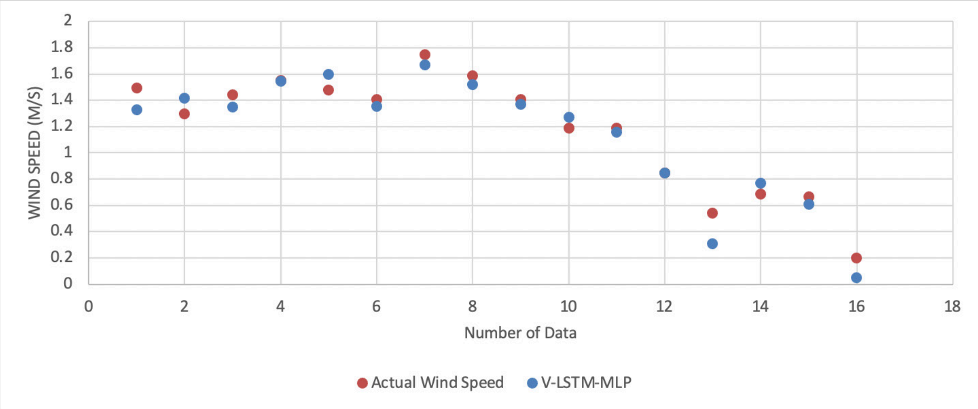


Fig. 14: Scatter Plot Actual Wind Speed and V-LSTM-MLP: 10 min Prediction.

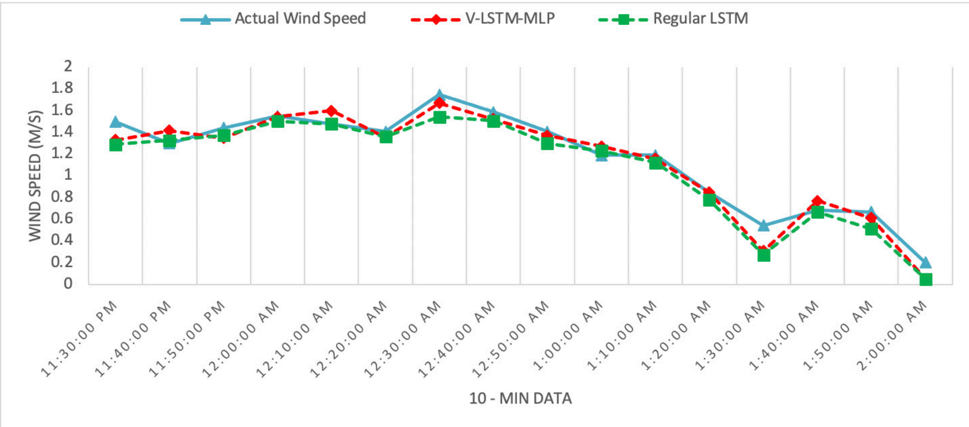


Fig. 15: V-LSTM -MLP and Regular LSTM Output: For comparison.

Conclusion

Wind energy is considered a flexible and natural source of natural energy and is in the position of high potential capability worldwide. However, the wind speed could also be irregular and has a wide range of fluctuations, affecting wind energy development and its integration with power grids. The possibility of accurately predicting the reactions of wind turbines in a wind farm makes the power transmission operator control the wind turbine and power transmission efficiency. Therefore, accurate wind speed forecasting is essential to increase efficiency and improve the performance of the electricity market. In this work, a combined neural network consisting of VAE-LSTM-MLP was designed and implemented to predict short-term wind speeds. In particular, the advantages of our hybrid model lie in the following aspects. We implemented the ENCODER-DECODER model, which consists of LSTM and VAE, to encrypt and reduce data dimensions; the design and training of this network have two advantages before performing the prediction operation: 1- It reduces the dimensions of our data and significantly increase the speed of the prediction operation in the next step. 2- Encryption using VAE, a production model, is probabilistic, implements a probability distribution and regular geometry on the input signal, and acts with power in selecting the appropriate sample.

To optimize the output of the encoder model, we performed several experiments and encrypted the result of the input signal graphing to 6 different features to achieve an improved model. Therefore, we used MLP neural network to predict wind speed in the short term. We are a collection of wind turbine wind speed data located in the Lutak area of Zabol city, which includes (maximum, minimum, average, and standard deviation of wind speed in the range of 10-30-40 meter wind turbine, air temperature, and humidity). We used 10 minutes of recorded time between 2006 and 2010. To compare the proposed model, we implemented three deep neural networks, LSTM-ENCODER-MLP and SAE (Khodayar et al., 2017), and Regular LSTM on this data set with the same conditions.

Finally, the proposed network with values of MAPE = 40.70% and RMSE = 0.1127 compared to ENCODER-DECODER LSTM networks with values of MAPE = 142.35% and RMSE = 0.1715 and SAE with MAPE = 123.24% and RMSE = 0.1470 and Regular LSTM with MAPE values = 63.45% and RMSE = 0.1309 perform better on our data set. Therefore, our combined strategy successfully increases wind speed forecasting accuracy and is an efficient model for predicting short-term speed at the Lutak wind power station.

Our future work is based on aspects such as 1 - To improve the accuracy of prediction models, consider new strategies to allocate the best configuration of modules in the combination of methods and combined networks such as capsule neural network with probabilistic models. 2- Using methods that select the best weight coefficients for neural networks. Optimizing weight ratios can significantly improve the predictability of models. 3. Using powerful pre-processing to control noise data effectively, in fact, using methods such as rough set theory and wavelet to increase predictive accuracy. 4- It can be extended by integrating a hybrid network with fuzzy logic.

References

Barbounis, T., & Theocharis, J. (2007). Locally recurrent neural networks for wind speed prediction using spatial correlation. *Information Sciences, 177*(24), 5775–5797. https://doi.org/10.1016/j.ins.2007.05.024

Bhaskar, K., & Singh, S. N. (2012). AWNN-Assisted Wind Power Forecasting Using Feedforward Neural Network. *IEEE Transactions on Sustainable Energy, 3*(2),306–315. https://doi.org/10.1109/tste.2011.2182215

Cybenko, G. (1989a). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems,2*(4),303–314. https://doi.org10.1007/

bf02551274.

Damousis, I., Alexiadis, M., Theocharis, J., & Dokopoulos, P. (2004). A Fuzzy Model for Wind Speed Prediction and Power Generation in Wind Parks Using Spatial Correlation. *IEEE Transactions on Energy Conversion, 19*(2),352361.https://doi.org/10.1109/tec.2003.821865

Deo, R. C., & Samui, P. (2017). Forecasting Evaporative Loss by Least-Square Support-Vector Regression and Evaluation with Genetic Programming, Gaussian Process, and Minimax Probability Machine Regression: Case Study of Brisbane City. *Journal of Hydrologic Engineering, 22*(6), 05017003. https://doi.org/10.1061/(asce)he.1943-5584.0001506

Eseye, A. T., Zhang, J., Zheng, D., Ma, H., & Jingfu, G. (2017, March). A double-stage hierarchical ANFIS model for short-term wind power prediction. *In 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)* (pp. 546–551). IEEE.

Georgilakis, P. S. (2008). Technical challenges associated with the integration of wind power into power systems. *Renewable and Sustainable Energy Reviews, 12*(3),852–863. https://doi.org/10.1016/j.rser.2006.10.007

Ghorbani, M. A., Khatibi, R., Hosseini, B., & Bilgili, M. (2013). Relative importance of parameters affecting wind speed prediction using artificial neural networks. *Theoretical andApplied Climatology, 114*(1-2),107–114. https://doi.org/10.1007/s00704-012-0821-9

Goodfellow, I., Yoshua Bengio, & Courville, A. (2016). Deep Learning. MIT Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hu, Y. L., & Chen, L. (2018). A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and Differential Evolution algorithm. *Energy Conversion and Management, 173,* 123–142. https://doi.org/10.1016/j.enconman.2018.07.070

Kawamoto, A. H., McClelland, J. L., & Rumelhart, D. E. (1989). Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises. *The American Journal of Psychology, 102*(3), 435. https://doi.org/10.2307/1423065

Khodayar, M., Kaynak, O., & Khodayar, M. E. (2017). Rough Deep Neural Architecture for Short-Term Wind Speed Forecasting. *IEEE Transactions on Industrial Informatics, 13*(6),2770–2779. https://doi.org/10.1109/tii.2017.2730846

Khodayar, M., Wang, J., & Manthouri, M. (2019). Interval Deep Generative Neural Network for Wind Speed Forecasting. *IEEE Transactions on Smart Grid, 10*(4),3974–3989. https://doi.org/10.1109/tsg.2018.2847223

Kingma, D.P., & Welling, M. (2014, April). Stochastic gradient VB and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR (Vol. 19, p. 121).* https://doi.org/10.48550/arXiv.1312.6114

Zhou, J., Shi, J., & Li, G. (2011). Fine tuning support vector machines for short-term wind speed forecasting. *Energy Conversion and Management, 52*(4), 1990–1998. https://doi.org/10.1016/j.enconman.2010.11.007

Liu, H., Mi, X., & Li, Y. (2018). Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. *Energy Conversion and Management, 159,* 54–64. https://doi.org/10.1016/j.enconman.2018.01.010

Liu, Z., Jiang, P., Zhang, L., & Niu, X. (2020). A combined forecasting model for time series: Application to short-term wind speed forecasting. *Applied Energy, 259, 114137.*https://doi.org/10.1016/j.apenergy.2019.114137

Long Short-Term Memory (LSTM). (2020, February 21). NVIDIADeveloper.https://developer.nvidia.com/discover/lstm

Memarzadeh, G., & Keynia, F. (2020). A new short-term wind speed forecasting method based on fine-tuned LSTM neural network and optimal input sets. *Energy Conversion and Management, 213, 112824.* https://doi.org/10.1016/j.enconman.2020.112824

Meng, A., Ge, J., Yin, H., & Chen, S. (2016). Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm. *Energy Conversion and Management, 114,75–88.* https://doi.org/10.1016/j.enconman.2016.02.013

Mir, M., Shafieezadeh, M., Heidari, M. A., & Ghadimi, N. (2019). Application of hybrid forecast engine based intelligent algorithm and feature selection for wind signal prediction. *Evolving Systems, 11*(4), 559–573. https://doi.org/10.1007/s12530-019-09271-y

Mirzapour, F., Lakzaei, M., Varamini, G., Teimourian, M., & Ghadimi, N. (2017). A new prediction model of battery and wind-solar output in hybrid power system. *Journal of Ambient Intelligence and Humanized Computing, 10*(1), 77–87. https://doi.org/10.1007/s12652-017-0600-7

Multilayer Perceptron Learning in TensorFlow. (2021, November3).GeeksforGeeks. https://www.geeksforgeeks.org/multi-layer-perceptron-learning-in-tensorflow/

Patidar, M., Agarwal, P., Vig, L., & Shroff, G. (2017). Correcting Linguistic Training Bias in an FAQ-bot using LSTM-VAE. In DMNLP Workshop of ECML-PKDD.

Peng, Z., Peng, S., Fu, L., Lu, B., Tang, J., Wang, K., & Li, W. (2020). A novel deep learning ensemble model with data denoising for short-term wind speed forecasting. Energy Conversion and Management, 207, 112524. https://doi.org/10.1016/j.enconman.2020.112524

Philippopoulos, K., & Deligiorgi, D. (2012). Application of artificial neural networks for the spatial estimation of wind speed in a coastal region with complex topography. *Renewable Energy, 38*(1), 75–82. https://doi.org/10.1016/j.renene.2011.07.007

Ricalde, L. J., Catzin, G. A., Alanis, A. Y., & Sanchez, E. N. (2011). Higher Order Wavelet Neural Networks with Kalman learning for wind speed forecasting. 2011 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG). https://doi.org/10.1109/ciasg.2011.5953332

Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Cornell Aeronautical Lab Inc Buffalo NY.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536. https://doi.org/10.1038/323533a0

Sder, L., Hofmann, L., Orths, A., Holttinen, H., Wan, Y. H., & Tuohy, A. (2007). Experience From Wind Integration in Some High Penetration Areas. *IEEE Transactions on Energy Conversion, 22*(1), 4–12. https://doi.org/10.1109/tec.2006.889604

Smith, J. C., Milligan, M. R., DeMeo, E. A., & Parsons, B. (2007). Utility Wind Integration and Operating Impact State of the Art. *IEEE Transactions on Power Systems, 22*(3), 900–908. https://doi.org/10.1109/tpwrs.2007.901598

Tascikaraoglu, A., & Uzunoglu, M. (2014). A review of combined approaches for prediction of short-term wind speed and power. *Renewable and Sustainable Energy Reviews, 34,* 243–254. https://doi.org/10.1016/j.rser.2014.03.033

Vinothkumar, T., & Deeba, K. (2019). Hybrid wind speed prediction model based on recurrent long short-term memory neural network and support vector machine models. *Soft Computing, 24*(7), 5345–5355. https://doi.org/10.1007/s00500-019-04292-w

Welch, R. L., Ruffing, S. M., & Venayagamoorthy, G. K. (2009, June). Comparison of feedforward and feedback neural network architectures for short term wind speed prediction. In *2009 International Joint Conference on Neural Networks (pp.23335-3340).* IEEE. https://doi.org/10.1109/ijcnn.2009.5179034

Weng, L. (2018, August 12). From Autoencoder to Beta-VAE. Lil'Log. https://lilianweng.github.io/posts/2018-08-12-vae/

Zhou, J., Shi, J., & Li, G. (2011). Fine tuning support vector machines for short-term wind speed forecasting. *Energy Conversion and Management, 52*(4), 1990–1998. https://doi.org/10.1016/j.enconman.2010.11.007

Submitted: 30.09.2022
Accepted: 17.11.2022