The Netherlands Press

Journal of Green Energy and Transition to Sustainability

Article

WIND ENERGY FORECSTING USING RECURRENT NEURAL NETWORKS

K L Sai Shankar[1]*, Jayanth Kuchipudi[2]

[1,2]Department of Computer Science,
University of Dayton,
300 College Park, Dayton, OH 45469, United States.

Orchid: 0000-0002-0006-6400[1]

E-mail: kurmal1@udayton.edu[1]

Abstract: Forecast of wind power is an estimation of the output production required of one or more wind turbines. Because of the variations and the probabilistic characteristics of the wind energy, forecasting it accurately becomes essential for designing reliable, economic operation and power control strategies. Changes in the nature and characteristics of the wind are probabilistic and a variety of machine learning model based on statistical parameters are used to characterize the randomness in the wind power production. The drawbacks of different approaches include their computational complexity and their inability to adapt to time-series processes. This paper describes Recurrent Neural Network (RNN) Long-Short Term Memory (LSTM) for time series prediction of wind power. LSTM units based RNN models have the ability to learn from the important past observations and decide whether this learned information is useful for future prediction. The experimental study showed better performance of the LSTM model compared with other traditional models.

Keywords: Long short-term memory, Recurrent neural networks, statistical parameters, time series prediction.

1. Introduction

Three rotating blades are usual for wind turbines, which can be connected to a generator directly or via a gearbox. The generator is housed inside the rotation shaft, and the rotor blades revolve around a horizontal hub attached to it. Other electrical parts and a rotating mechanism are also a part of the nacelle system, which turns the turbine to point towards the wind. The tower head is rotated to align with the wind direction while the sensor is utilised to track the wind direction [1]. When the wind speed varies, the generator's output is automatically adjusted. The tower-mounted wind generators (WEG) range in height from 25 to 80 metres, while the diameter of the propeller blades ranges from 30 metres (m) to roughly 90 m. The wind turbines' energy output is adequately conditioned to power the nearby grid. WEG units can now operate at wind speeds of 2.5 m/s (metres per second) to 25 m/s and have capacities ranging from 225 kW to 2 MW.

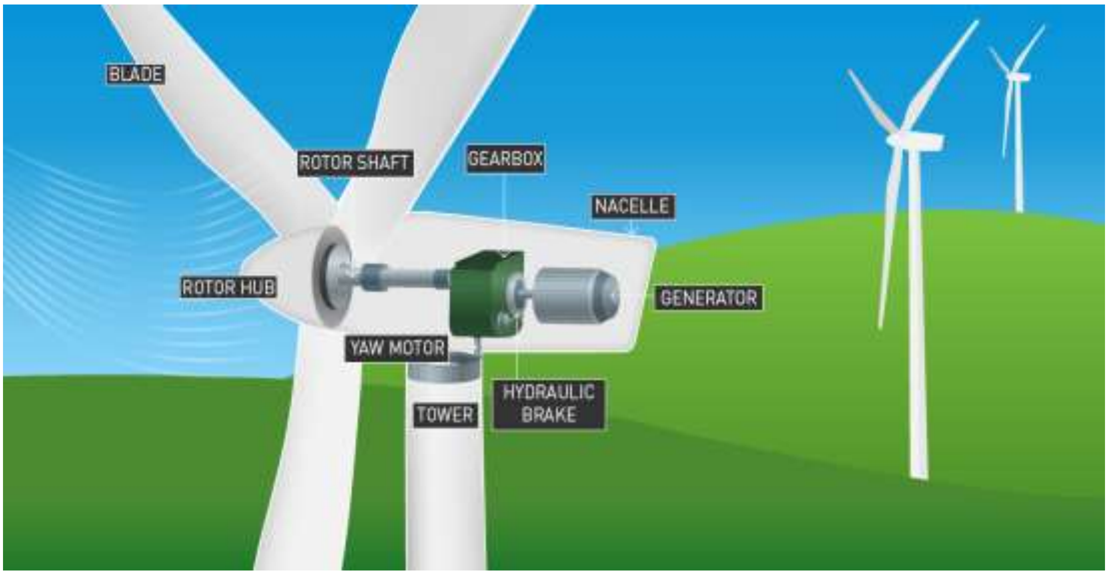

Figure 1:Wind energy [2].

The layers of a cyclic neural network are constructed using the units (or blocks) of long-term memory (LSTM) (RNN)[3]. An LSTM network is a common name for an RNN made out of LSTM units. A cell, an input port, an output port, and a forget gate make up a standard LSTM unit. Since cells are expected to "remember" information at any time, the word "memory" is used in the LSTM. In a multilayer (or forward) neural network, each of the three gates functions as a "regular" artificial neuron; that is, they compute the activations (using functional activations) of a weighted sum [4]. The "door" designation refers to the visual representation of them as value-current regulators moving through the connections of the LSTM. These ports are linked to the cell through connections. The phrase "short-term, long-term" alludes to the short-term memory model LSTM's ability to endure over extended periods of time. Given time delays of undetermined magnitude and length between important occurrences, LSTM is ideally suited for categorising, analysing, and forecasting time series [5]. To address the issue of bursting and disappearing gradients when training conventional RNNs, LSTMs were created [10]. In many cases, LSTM outperforms alternative RNNs, hidden Markov models, and other sequence learning techniques due to their relative insensitivity to gap length [7].

Open loop prediction predicts the following time step in a series using just input data, whereas closed loop prediction predicts the following time step in a sequence using output data. You get actual values from

Journal of Green Energy and Transition to Sustainability Volume 1 Issue 1.
Received Date 20 May 2022
Accepted Date 12 June 2022
Published Date 23 July 2022
Online at https://www.jgets.org/

your data source and utilise them as input for making predictions for the upcoming time steps [6]. Using the information gathered from time steps 1 through t 1, for instance, one may forecast the value of the sequence's time step t. Let's wait till you have recorded the actual value for the time step t and use that as input to generate the forecast for the time step t + 1. Before generating the next prediction, feed the network with real values using open-loop prediction [8].

With the use of prior predictions as input, closed-loop prediction forecasts the following time steps in a series. In this instance, the model can generate predictions without real-world data. Using only the data gathered at time steps 1 to t-1, for instance, one may forecast the value of the time steps from t to t + k in a series [9]. Use the expected value for time step i-1 as input when making a prediction for time step i. Use closed-loop prediction when you need to make several future predictions at once or when you don't have any actual data to feed the network before generating the subsequent forecast [11].

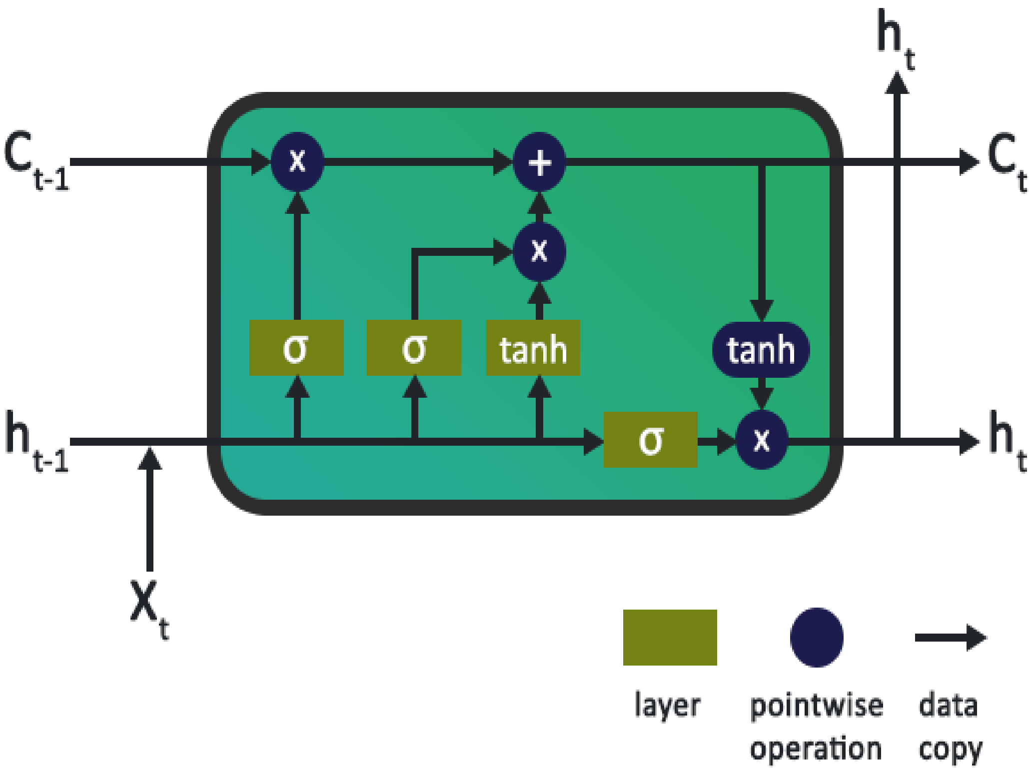

Figure 2: LSTM Block[14].

2. Implementing the wind energy data using neural networks

The deep learning model uses LSTM (Long Short-Term Memory), or a modified cyclic neural network, to estimate the power produced by wind turbines in wind power facilities.

2.1. Dataset preparation

NREL provided the historical wind energy data needed for this investigation. In this experiment, six years' worth of wind power production data were employed. Following pre-processing, the data includes timestamp, air temperature (C), pressure (atm), wind direction (degrees), wind speed (m/s), and power produced by the system (kW) in addition to other parameters [13].

Table 1: Date/Time

	Date/Time	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
0	01 01 2018 00:00	380.047791	5.311336	416.328908	259.994904
1	01 01 2018 00:10	453.769196	5.672167	519.917511	268.641113
2	01 01 2018 00:20	306.376587	5.216037	390.900016	272.564789
3	01 01 2018 00:30	419.645905	5.659674	516.127569	271.258087
4	01 01 2018 00:40	380.650696	5.577941	491.702972	265.674286

A mean and median approach is required to characterise the data collected to forecast power production. 1492.17 kWh are generated on average theoretically, and the average wind direction is 123.68 degrees. The data's maximum power output is 3600 kWh. This information can be used to execute an operation to pass into the model. The forecast needs to go through pre-processing. Data were first scaled using a common scale. The scale receives all data values and converts them to 0s and 1s. The model can function correctly if the data are converted to 0 and 1.

Table 2: LV ActivePower

	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
count	50530.000000	50530.000000	50530.000000	50530.000000
mean	1307.684332	7.557952	1492.175463	123.687559
std	1312.459242	4.227166	1368.018238	93.443736
min	-2.471405	0.000000	0.000000	0.000000
25%	50.677890	4.201395	161.328167	49.315437
50%	825.838074	7.104594	1063.776283	73.712978
75%	2482.507568	10.300020	2964.972462	201.696720
max	3618.732910	25.206011	3600.000000	359.997589

Take time series analysis in its purest form. Predict the system's power output in the Prediction section without taking future time into account. This is significant because predicting the weather in the future is a prediction issue on its own, requiring machine learning to solve problems with unknown future wind speed, temperature, or pressure. LSTM is used to forecast the strength simply by examining the pattern in historical data. The LSTM specifies that the date, time, and system-generated power must all be provided as monitored data for this model. The LSTM will examine historical data in an effort to learn relevant information about historical data trends. And it will make a prediction based on this information. Future values are forecasted, and outcomes are evaluated via direct validation.

2.2. Building the Model

Using a sequential model, a cyclic neural network is constructed. The LSTM layer is added with counted neurons for the sequential model, followed by the dense layer, which is added with a single neuron passing after the LSTM layer, and the data is divided into test and training sets. training and experience. The mean squared error is referred to as passing ship data in the model with epochs and loss functions. In essence, a loss function measures how well your predictive model is able to forecast the anticipated result (or value), turn a learning problem into an optimization problem, specify a loss function, and then optimise the algorithm. Adam is the optimization function employed. The Adam optimizer combines the following two methods of gradient descent: Momentum: By examining the "exponentially weighted average" of the slopes, this approach is utilised to accelerate the gradient descent procedure. The algorithm reaches the minimum more quickly when the mean is used.

```python
# fit an LSTM network to training data
def fit_lstm(train, batch_size, nb_epoch, neurons):
    X, y = train[:, 0:-1], train[:, -1]
    X = X.reshape(X.shape[0], 1, X.shape[1])
    model = Sequential()
    model.add(LSTM(neurons, batch_input_shape=(batch_size, X.shape[1], X.shape[2]), stateful=True))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam',metrics='accuracy')
    for i in range(nb_epoch):
        model.fit(X, y, epochs=1, batch_size=batch_size, verbose=1, shuffle=False)
        model.reset_states()
    return model
```

3. Results and discussions

By feeding the test data into the model validation accuracy and the observed loss, the metrics generated in these findings are the precision and loss per epoch. During the validation of the test data, the data must be forward for predictions. and execute one-step prediction by supplying the class to the one-step prediction function that was previously created. Predicted values should be used in place of experimental scale values. The output is the projected value and the result is presented along with the time and forecast after scaling and discriminating the values.

```python
# walk-forward validation on the test data
predictions = list()
expectations = list()
predictions_plot = list()
expectations_plot = list()
test_pred = list()
for i in range(len(test_scaled)):
    # make one-step forecast
    X, y = test_scaled[i, 0:-1], test_scaled[i, -1]
    yhat = forecast_lstm(lstm_model, 1, X)#batch_size_exp to 1
    '''# Start Debug prints
    print("X: %", X)
    print("yhat: %", yhat)
    # End Debug prints'''
    # Replacing value in test scaled with the predicted value.
    test_pred = [yhat] + test_pred
    if len(test_pred) > lag_exp+1:
        test_pred = test_pred[:-1]
    if i+1<len(test_scaled):
        if i+1 > lag_exp+1:
            test_scaled[i+1] = test_pred
        else:
            test_scaled[i+1] = np.concatenate((test_pred, test_scaled[i+1, i+1:]),axis=0)

    # invert scaling
    yhat = invert_scale(scaler, X, yhat)
    # invert differencing
    yhat = inverse_difference(raw_values, yhat, len(test_scaled)+1-i)
    # store forecast
    expected = raw_values[len(train) + i + 1]
    predictions_plot.append(yhat)
    expectations_plot.append(expected)
    if expected != 0:
        predictions.append(yhat)
        expectations.append(expected)
    print('Hour=%d, Predicted=%f, Expected=%f' % (i+1, yhat, expected))
```

It is possible to visualise windmills by utilising the wind flower library. Meteorologists use the graphical tool Windrose to give a quick overview of the broad distribution of wind speed and direction at a certain area. It may also be used to describe the things that pollute the air. The backend of the rose compass tool is Matplotlib. Pandas data frames or Numpy arrays can be used to transmit data into the packet.

A Python module called Windrose is used to manage wind data, draw wind roses (also known as polar rose graphs), and fit Weibull probability density functions. This library's first application was for a technical report on wind exposure and dispersion study.

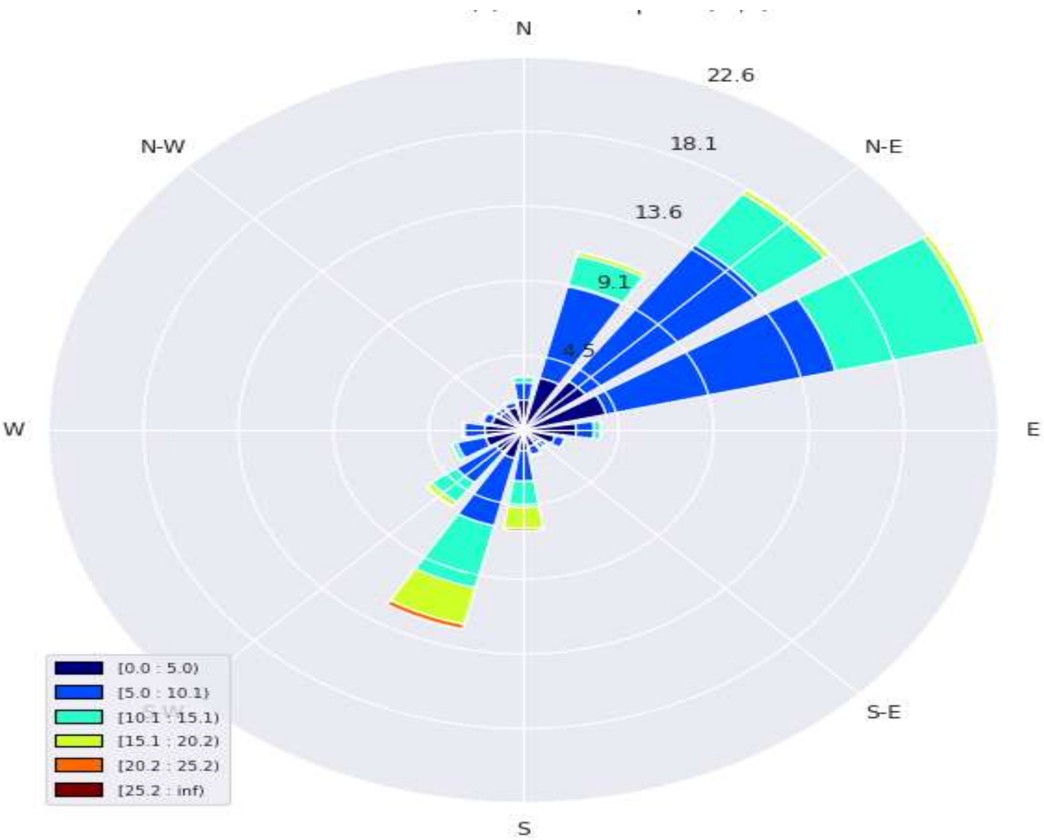

Figure 3: Wind speed vs wind angle.

The data must be plotted in order to be visualised, and in this case, the data is plotted against theoretical power. From this graph, various points may be noticed where power grows first, then declines and increases, varying unevenly. These values rely on additional variables when a neural network prediction- system is constructed.

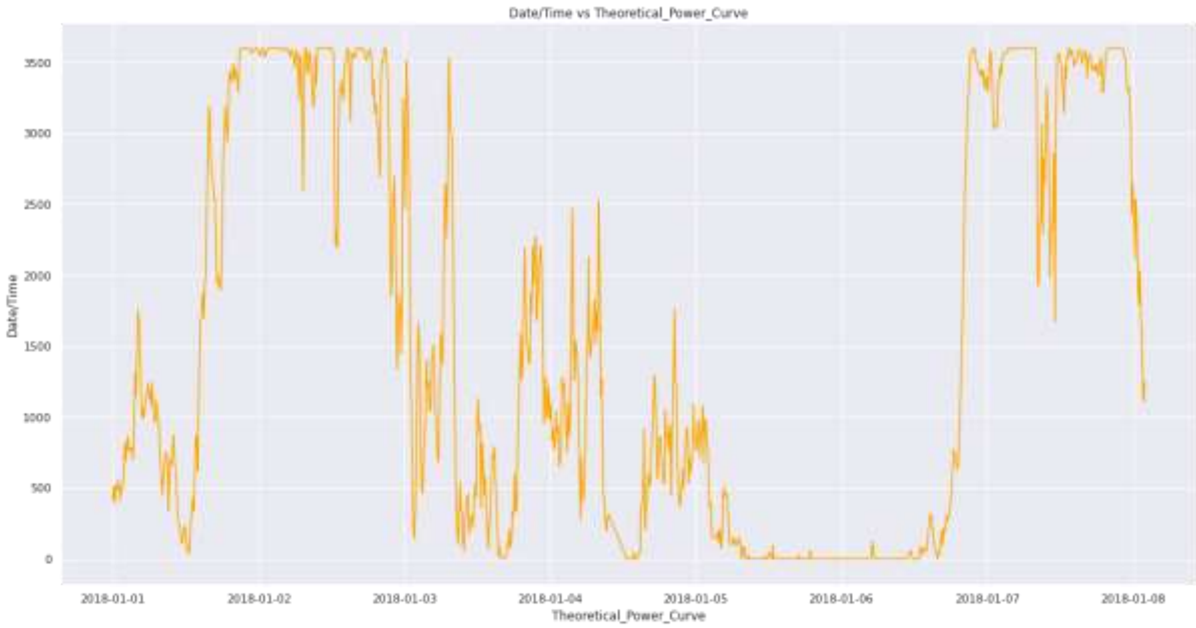

Figure 4: Date vs theoretical power graph.

Due to the created model's accuracy of 92.5 percent, testing may be done on it. After training, the model is put to the test by processing test pictures, and the obtained results are compared to those anticipated. Loss and accuracy graphs for the test and training parts of the model are obtained during model training. The model may be used to anticipate wind energy time series if the losses are reducing.

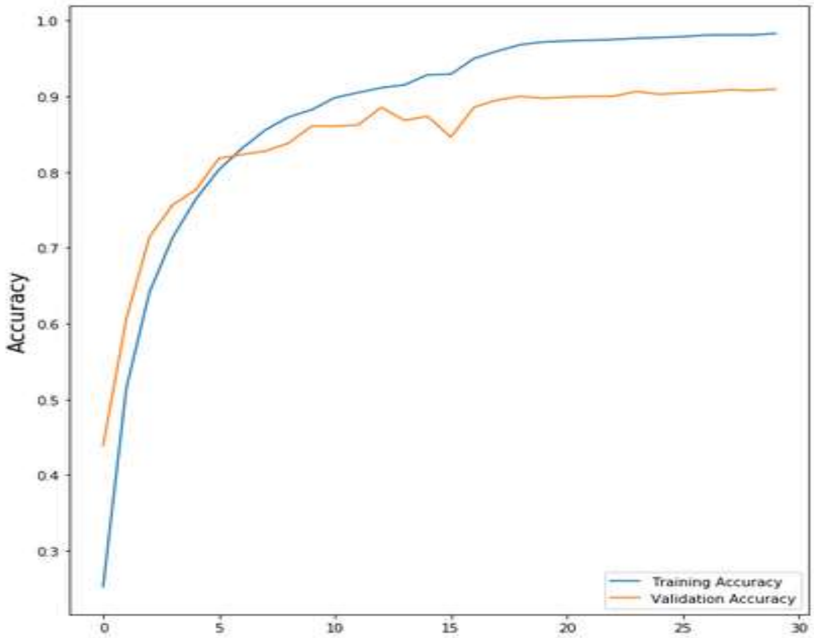

Figure 5: Accuracy

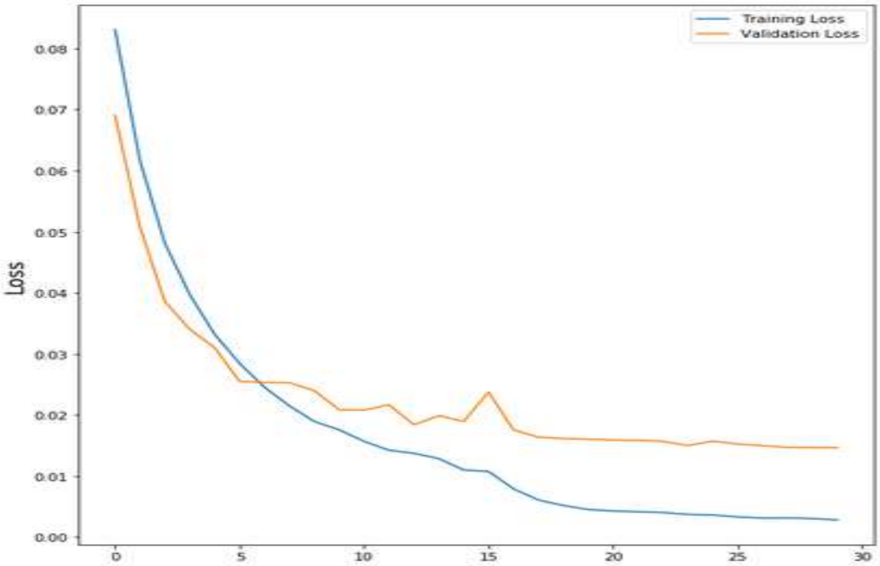

Figure 6:Loss

For the provided data, the output power and anticipated power graphs are presented. The average absolute error achieved is 0.4614, which is quite low, indicating that this model can predict well but may not be 100% correct in some forecast cases.

```
expectations = np.array(expectations)
predictions = np.array(predictions)
print("Mean Absolute Percent Error: ", (np.mean(np.abs((expectations - predictions) / expectations))))
```

```
Mean Absolute Percent Error:  0.072714952886022825
```

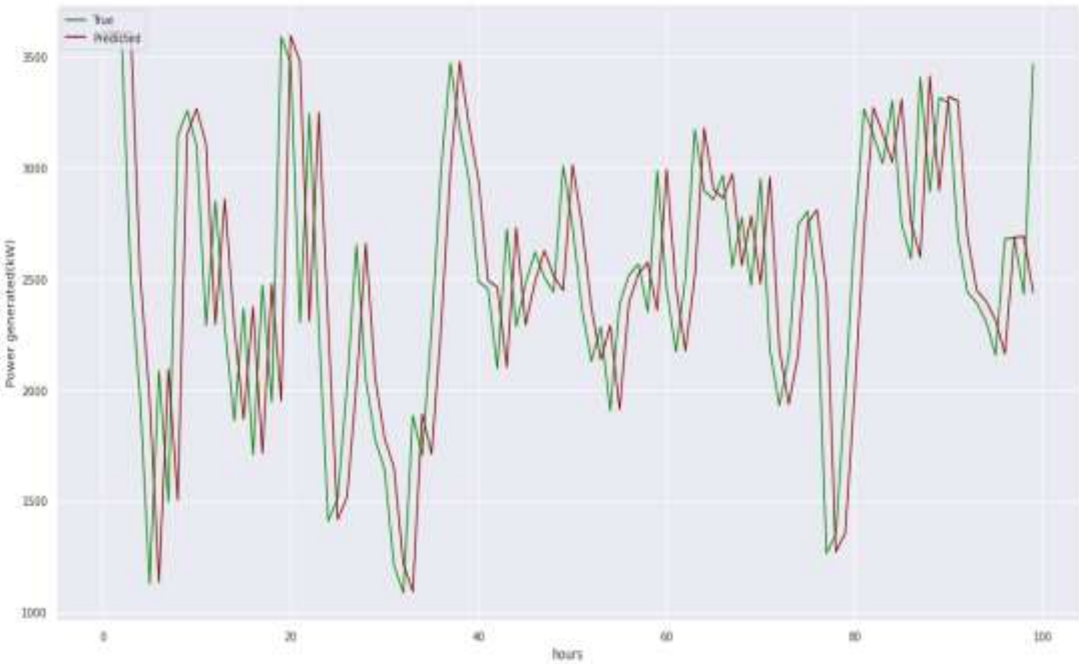

Figure 7: True vs prediction graph

4. Conclusion

Predictions are simplified by employing machine learning techniques, and models continue to advance. By utilising the capabilities of the biological genetic algorithm framework, this controls the optimization of the window size and neuron count of the LSTM layers. The prediction error from the LSTM findings is 0.07271, which is very near to zero. However, LSTM provides comparable measurement accuracy to artificial neural network findings developed via genetic programming. Because LSTM uses fewer algorithms and has less computational complexity, the error is different and occasionally even somewhat lower, but this is acceptable. In order to enhance the optimization of LSTM parameters, which can be used to several different time series issues for prediction, this novel concept combines the strengths of LSTM for improved prediction. With simple access to high-speed computer capabilities, further parameter optimization may be carried out for a wide range of potential combinations for greater output and optimization.

References

[1]. Shokri Gazafroudi Amin. Assessing the Impact of load and renewable Energies uncertainty on a hybrid system. International Journal of Energy and Power Engineering; 2015.

[2]. "wind power | Capacity & Facts," *Encyclopedia Britannica*. https://www.britannica.com/science/wind-power (accessed Jul. 07, 2022).

[3]. Lange M, Focken U. Physical approach to short-term wind power prediction, vol. 208. Springer; 2006.

[4]. Foley AM, et al. Current methods and advances in forecasting of wind power generation. Renew Energy 2012;37(1):1e8.

[5]. Taylor JW, Buizza R. Using weather ensemble predictions in electricity demand forecasting. Int J Forecast 2003;19(1):57e70.

[6]. Marciukaitis M, Katinas V, Kavaliauskas A. Wind power usage and prediction prospects in Lithuania. Renew Sustain Energy Rev 2008;12(1):265e77.

[7]. Louka P, et al. Improvements in wind speed forecasts for wind power prediction purposes using Kalman filtering. J Wind Eng Ind Aerod 2008;96(12): 2348e62.

[8]. Nielsen Torben Skov, et al. Zephyr - The Prediction Models. 2006.

[9]. Nielsen HA, et al. Improvement and automation of tools for short term wind power forecasting. In: Scientific Proceedings of the European Wind Energy Conference & Exhibition. Milan, Italy, 2007b; 2007. http://www. ewec2007proceedings.info/allfiles2/475Ewec2007fullpaper.pdf.

[10]. Focken Ulrich, et al. Regional wind power prediction with risk control. Proc Global Windpower Conf 2002.

[11]. Negnevitsky Michael, Johnson PL, Santoso Surya. Short Term Wind Power Forecasting Using Hybrid Intelligent Systems1; 2007.

[12]. Aggarwal SK, Saini LM. Solar energy prediction using linear and non-linear regularization models: a study on AMS (American Meteorological Society) 2013e14 Solar Energy Prediction Contest. Energy 2014;78:247e56.

[13]. Pinson P. Very-short-term probabilistic forecasting of wind power with generalized logitenormal distributions. J Roy Stat Soc: Series C (Applied Statistics) 2012;61(4):555e76.

[14]. Fan G-f, et al. Wind power prediction based on artificial neural network. In: Proceedings of the CSEE; 2008.