Time Series Forecasting of Generated Power From Texas Wind Turbine

Sara Antonijevic
Department of Statistics
Texas A&M University
College Station, Texas
saraant@tamu.edu

Nicholas A. Hegedus
Department of Mechanical Engineering
University of Nevada
Las Vegas, Nevada
hegedn1@unlv.nevada.edu

Nuri J. Omolara
Department of Industrial and Systems Engineering
North Carolina A&T State University
Greensboro, North Carolina
njomolara@aggies.ncat.edu

Kishore Bingi
Department of Electrical and Electronics Engineering
Universiti Teknologi PETRONAS
Seri Iskandar, Malaysia
bingi.kishore@ieee.org

Om Prakash Yadav
Department of Industrial and Systems Engineering
North Carolina A&T State University
Greensboro, North Carolina
oyadav@ncat.edu

Rosdiazli Ibrahim
Department of Electrical and Electronics Engineering
Universiti Teknologi PETRONAS
Seri Iskandar, Malaysia
rosdiazli@utp.edu.my

***Abstract*—As the demand for renewable energy rises, optimizing wind power as an energy source is crucial. Wind power is one of the cleanest forms of energy available, and understanding the energy that wind turbines generate over time is necessary for building a better foundation for wind energy reliance. Previous research has explored using Long-Short-Term Memory (LSTM) and LSTM-based algorithms for practical wind turbine data analysis and predictions. In those cases, the LSTM-based forecasts showed the most robust wind turbine prediction rates based on various variables, mainly wind speed and direction, and generated active power. This research implements LSTM, Nonlinear Autoregressive (NAR), and Nonlinear Autoregressive Exogenous (NARX) networks on simulated Texas wind turbine data to compare the techniques that produce better data predictions. The data is normalized using correlation analysis techniques on the following data features: system power generated, wind speed, wind direction, pressure, and air temperature. The data is separated for training and testing and run through the LSTM, NAR, and NARX algorithms. After obtaining the mean squared error (MSE) of the testing data, the algorithms are compared to determine the best predicting algorithm. The results show which algorithm on time-series data holds the most robust prediction of generated energy from wind turbines.**

***Keywords*—forecasting, LSTM, NAR, NARX, prediction, wind energy, wind turbine**

I. INTRODUCTION

Wind power is a significant and reliable source of energy that is collected through the use of wind turbines. These turbines absorb the energy of the wind when it hits the propeller blades, and the motion of the blades allows a generator to gather kinetic energy. In order to better understand the impact of the power generated, the turbines collect various variables such as wind speed, air temperature, pressure, and wind direction [1]. One can understand how these variables interact under different circumstances and create patterns by tracking them in a time series. For this research, each variable is recorded hourly for an entire year, allowing an AI algorithm to be incorporated to predict future variable trends. Previous research has explored the effectiveness of machine learning algorithms in predicting wind power generation. For instance, in [2], the LSTM model was measured for its effectiveness through the mean absolute error (MAE) of a normalized data set before being split into training and testing data. Multiple hyperparameter tuning techniques have been utilized on the training data to record error rates between the moving-average approach and the multi-layer perceptron model. The study concluded that the LSTM system had the smallest error estimates. In study [3], a basic LSTM algorithm was used to predict data pre-processed into three forms of experiments (single-sensor, multi-sensor, and small-number data). The multi-sensor data underwent hyperparameter tuning once the data was split into training and testing data. They were compared to other machine learning algorithms in which the LSTM's results showed higher accuracy.

Study in [4] suggested using an LSTM-based algorithm on pre-processed and normalized data ran through the Aquila optimizer (AO). The algorithm's performance is compared to a basic LSTM model, revealing that the AO-LSTM has the lowest recorded errors. Similarly, in [5], LSTM and Convolutional Neural Network (CNN) utilize pre-processed data through outlier removal and normalization before running it through the Adam algorithm optimizer. The performance is measured through various errors in the CNN and LSTM values, with the LSTM system having the smallest error. In [6], research shows the effectiveness of LSTM mixed with the Gaussian algorithm in forecasting short-term wind power data. The data was pre-processed by removing nonoperational data points and normalized using the min-max method. The Adam algorithm then optimizes the data before recording error metrics. These metrics, along with their respective forecast confidence intervals, are compared with various techniques to reinforce the accuracy of the LSTM method. Furthermore, in [7], the LSTM's strength in data is pre-processed by outlier removal, and the testing data undergoes a time-sequence probability-based optimization. Once the accuracy of the variables is found, it is compared to different techniques, showing that employing the LSTM-based algorithm over other machine learning variations holds a stronger accuracy percentage.

In their research, the authors of [8] explored combining the LSTM algorithm and optimizer by filtering data based on the sequential correlation features of target turbines. The Spectral Clustering algorithm was then used to optimize the results, and the resulting error value was compared to other algorithms such as Support Vector Regression (SVR), K-Nearest Neighbors (KNN), and a conventional LSTM algorithm to determine the accuracy. In [9], the potency of the LSTM method on wind pattern time series was further emphasized through the K-Means-LSTM technique. This technique involved cleaning the data with the culling method and normalizing it through Pearson correlation coefficient analysis. The data was split into testing and training sets and underwent the Adam algorithm to optimize the loss function. Dropout technology was also used to prevent model overfitting. The method's effectiveness was measured through error metrics and compared to other models such as SVR, Elman, and a conventional LSTM model. According to performance measurement rates, the K-Means clustering and LSTM combination showed the lowest residual rates compared to the other methods.

In the study reported in [10], CNN-LSTM measurement was performed using linear regression and 10-fold validation processing. The input variables included wind speed and direction, elevation, temperature, air pressure, and electrical generation, which have been used to make different comparisons. The total power output of the generator was the output variable derived using the 10-fold validation as a hyperparameter. The paper concluded that linear regression is an effective method for testing CNN-LSTM. In the referenced study [11], the LSTM-based system was applied to Supervisory Control and Data (SCADA). It was first pre-processed by outlier removal and normalized through correlation analysis to prepare for training and testing. Hyperparameter tuning used an optimizer to record the error metrics and correlation coefficient R^2 for LSTM and CNN-LSTM algorithms. The study found that the CNN-LSTM model had negligible errors between actual and predicted.

Based on the literature presented, this paper suggests conducting a comparative analysis of three techniques for predicting the output of a wind turbine in Texas. These techniques include LSTM, NAR, and NARX networks.

The paper is organized into several sections. Section II outlines the methodology proposed for the study. Section III explores the results and provides a discussion. Finally, section IV concludes the paper.

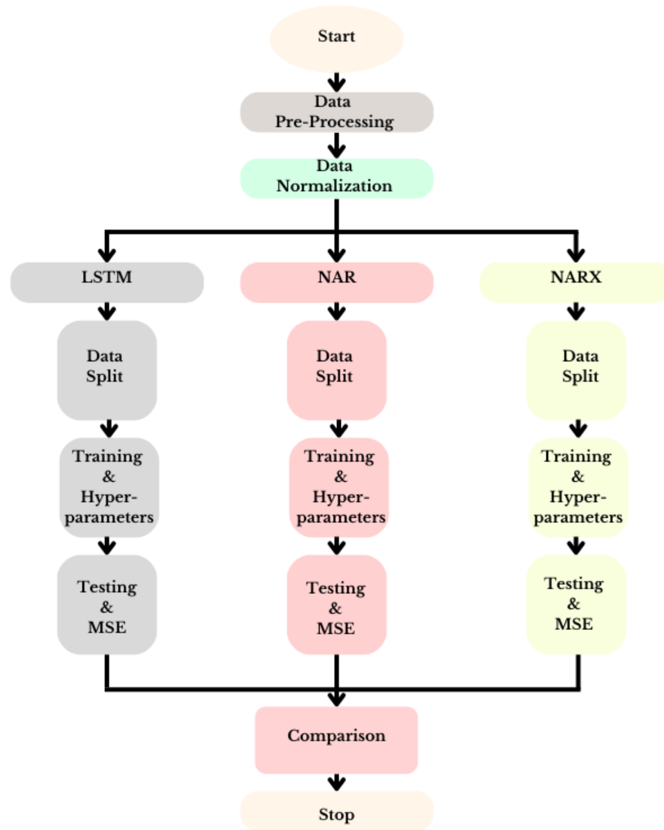

Fig. 1. Flowchart for proposed time series forecasting of Texas wind turbine's generated power

II. Methodology

The methodology this research follows is illustrated in Fig. 1 and involves utilizing and contrasting three machine learning algorithms, namely LSTM, NAR, and NARX. The details and architectures of these algorithms are explained underneath.

A. Long-Short-Term Memory (LSTM)

When it comes to artificial intelligence (AI) learning algorithms for long-term dependencies, one common option is the LSTM, a Recurrent Neural Network (RNN). LSTM programs use past and present data to make predictions, making them particularly useful for time-series-based projects. There are several advantages to using an LSTM. For one thing, it considers the data's previous behavior when making predictions about its current behavior. This can lead to solid predictions on time-based data sets. LSTM also has excellent performance across a variety of time scales. Finally, these programs can be trained using back-propagation, which means that the weights of different parts of the data can be adjusted depending on how much the algorithm expects that information to be needed for future predictions [12], [13].

The design of an LSTM is illustrated in Fig. 2. This model has three gates: input, forget, and output. The input gate decides which data components to use for adjusting the algorithm's memory. Mathematically, the algorithm utilizes a

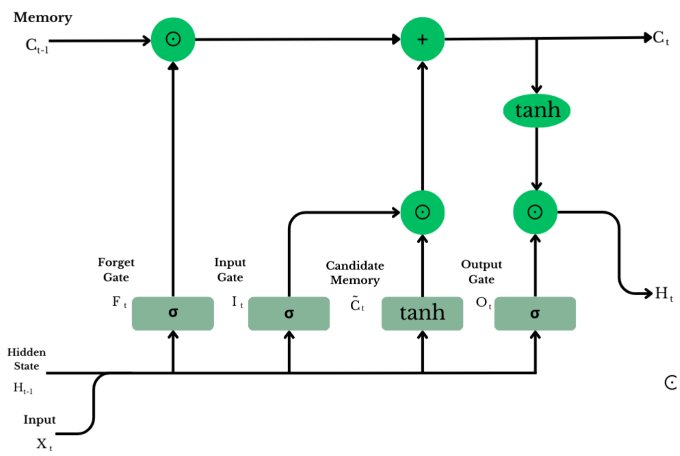

Fig. 2. Architecture of LSTM

sigmoid function to assign a value of either 0 or 1 to data points. If the value is 0, the data is not considered in the prediction process. If the value is 1, the data is passed through and given a weight between -1 and 1 based on the tanh function. The weight the tanh function assigned the influence the point will have on the prediction-making process.

The forget gate is a crucial architectural component that evaluates which data pieces are unnecessary to create the next set of predictions. Similar to the input gates, the forget gate accepts values of 0 or 1 to determine which pieces of data should be kept or omitted from its memory. The output gates use the input and weighted data memory to determine the algorithm's output. It identifies which input value should be used to adjust the memory. The sigmoid function decides which values to allow through while the tanh function assigns weight to the passed values.

To ensure optimal performance of the LSTM, the data is divided into training and testing data. About 70-80% of the data is used for training and 20-30% for testing. Two variables are created during training to accept input sequences and target values. These variables work together to teach the algorithm to predict the next step's value. The normalization used for training and testing data is to prevent divergence during the training process, and the same normalization is used for testing variables. Next, the training variables are refined by adjusting the data through hyperparameter tuning. The effectiveness of the training is evaluated using statistical measurements like MSE and root mean square error (RMSE). This methodology is also applied to the testing data before the LSTM can be used for predicting future steps.

This study utilizes the LSTM method to forecast the power output of wind turbines. It is crucial to train the LSTM model since wind turbines' energy production is unpredictable, making it challenging to determine the required power at specific times. By integrating the LSTM into the system, one can better anticipate the instability of power output and take appropriate measures.

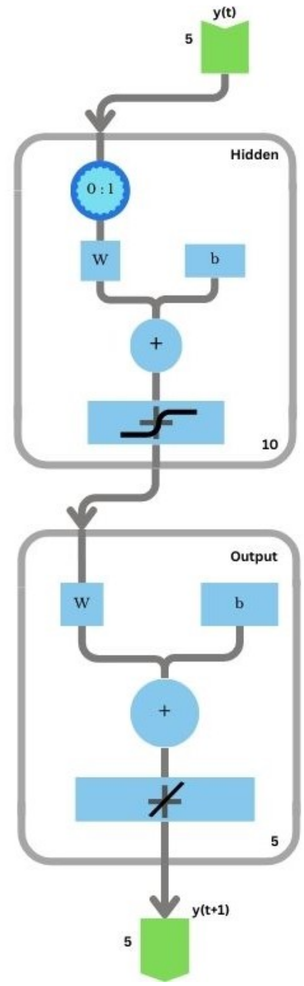

Fig. 3. Architecture of NAR

B. Nonlinear Autoregressive (NAR)

NAR is an artificial neural network that predicts time series based on past values [14]. It consists of an input layer, a hidden layer, and an output layer. The output at a specific time depends heavily on previous outputs. Like an LSTM, a NAR prediction is based on the weight given to data. To evaluate the algorithm's accuracy, autocorrelation, and prediction error are assessed. See Fig. 3 for a visual representation.

The NAR algorithm confidently predicts wind turbines' power output (in kW), much like the LSTM. Its training utilizes historical time series data as input to the network, allowing the model to fully grasp the complex interactions between the input and target variables. As a result, the algorithm produces accurate predictions for future periods.

C. Nonlinear Autoregressive with Exogenous (NARX)

NARX, a nonlinear autoregressive with exogenous inputs, is a neural network for series forecasting [15]. Unlike traditional time series models, NARX models consider past values of the target variable and past exogenous values to make predictions (See Fig. 4). The autoregressive component of NARX looks at the relationship between the current target variable and its

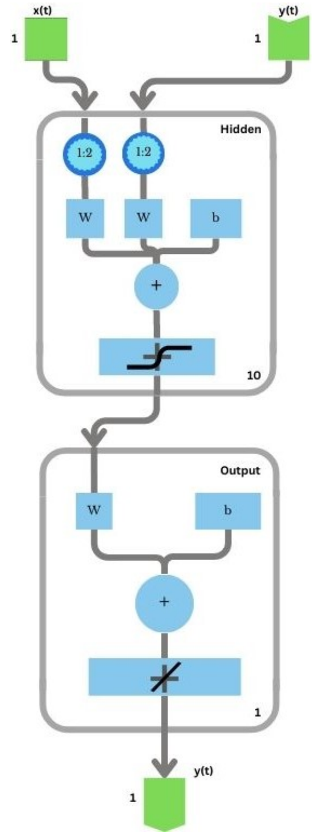

Fig. 4. Architecture of NARX

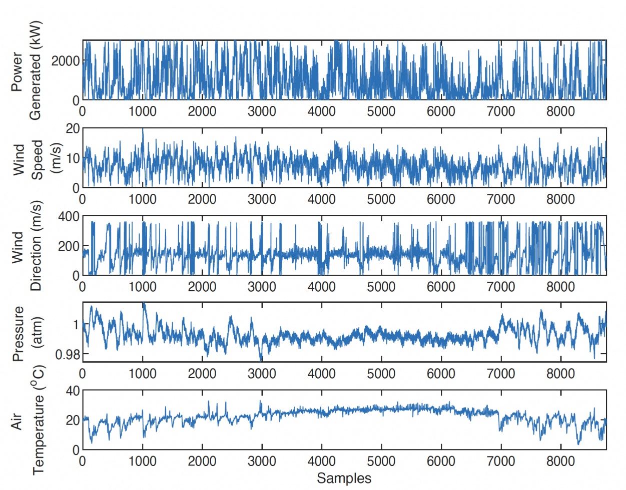

Fig. 5. Dataset of simulated Texas wind turbine

previous outputs. Previous predictions are reintroduced into the model using a feedback loop to improve future forecasts. This part of NARX can identify both linear and nonlinear patterns.

In addition, an external factor impacts the target variables, called the exogenous element. Additional inputs of exogenous variables may affect the performance of the time series, but they are not part of the target variable's past. To forecast a

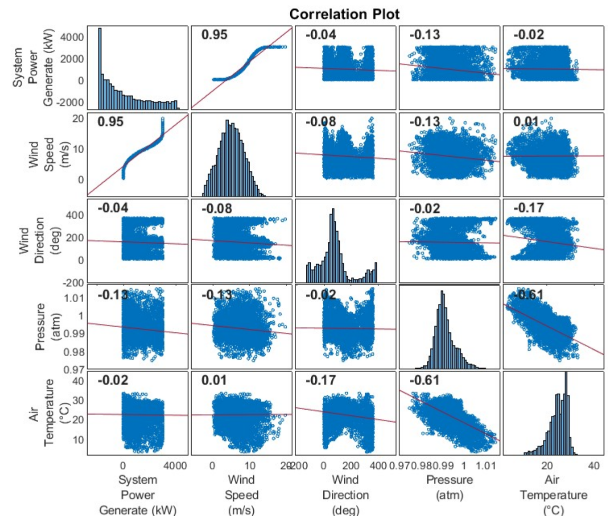

Fig. 6. Correlation plot of Texas wind turbine variables

series, the NARX model is utilized through a neural network with the help of recurrent neural network frameworks like LSTM. During the training process of the NARX model, the network is fed with historical time series data. This enables the model to grasp the complex interactions between the target and exogenous variables, leading to accurate predictions.

III. Results and Discussion

A. Data Description

The study's data is accessible at [16]. The simulated wind turbine used in the data set is located in onshore Texas and has a rotor diameter of 111 meters, producing a rated output of 3600 kW. With an overall height of 80 meters, the turbine collects data points every hour for a year, recording five essential variables: system power generated (kW), wind speed (m/s), wind direction (deg), pressure, and air temperature(C°). Each variable consists of 8,760 samples generated hourly for a full year (see Fig. 5). The data set provided has been simulated using National Renewable Energy Laboratory software to have perfect completeness and no noisy data, allowing for quick data split between training and testing data points.

B. Data Correlation

Figure 6 used a correlation plot to examine the relationship between variables. The results show a strong positive correlation of approximately 0.95 between wind speed and wind power generated. The Pearson correlation coefficient confirms that the stronger the wind speed, the more power the turbine generates.

There is a relationship worth monitoring between pressure and air temperature, which is portrayed as a moderate negative correlation (-0.61). This means these variables act opposite each other for a certain period. For example, if there is higher pressure, a lower value from the air temperature should be expected. A full multicollinearity test is unnecessary, as the relationship between the variables is expected.

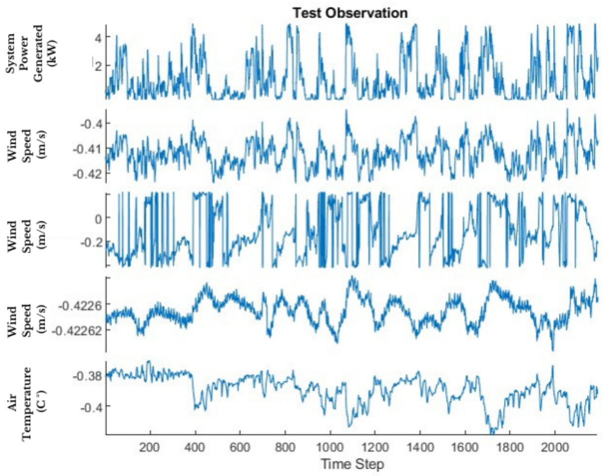

Fig. 7. Response of LSTM during training and testing.

The other variables have weak correlations, also known as negligible correlations, on Pearson's chart. A negligible correlation suggests insufficient statistical evidence of a correlation between variables occurring. In other words, the correlation between the two variables is likely a chance occurrence. These correlations include wind direction and air temperature (-0.17), wind speed and pressure (-0.13), wind speed and wind direction (-0.8), wind direction and pressure (-0.02), system power generation and air temperature (-0.02), and wind speed and air temperature (0.01).

C. LSTM

When training the LSTM algorithm, the RMSE decreases significantly during training and stabilizes at around 0.6 once more data samples are introduced. The LSTM algorithm is trained for 200 epochs, during which parameters such as the weights on data points are adjusted to minimize the difference between predicted and target outputs. In the loss function of the LSTM algorithm, initially, there is a high loss error between actual and predicted models due to random weights. Still, as training continues, the algorithm captures the model's parameters more accurately, minimizing loss. Once the weights stabilize, the LSTM algorithm shows an average loss of around 0.2 between actual and predicted data points. The MSE model is approximately 0.2823 after training. Later, the same normalization technique is applied to the testing variables, and the algorithm is run with the trained network. The average MSE of the testing variables is approximately 1.5757. After calculating the MSE of the overall LSTM, predictions can be generated. The test is done for 100 steps for all five variables and is pictured in Fig. 7. The graphs depict the test sequence plot applying input time series to forecast future time steps.

D. NAR

Figure 8 illustrates the difference between the intended targets and actual outcomes over time. Though most error values are small, a few stand out. The NAR response of the output plot above is trained using a double array of 8760-time steps, with five variable features acting as predictors. Figure 8 displays the fluctuations in error rates across all time samples. This type of machine learning algorithm records relatively high error rates, with the training algorithm's MSE being 1.8410×10^4, which increases to 2.1093×10^4 after training.

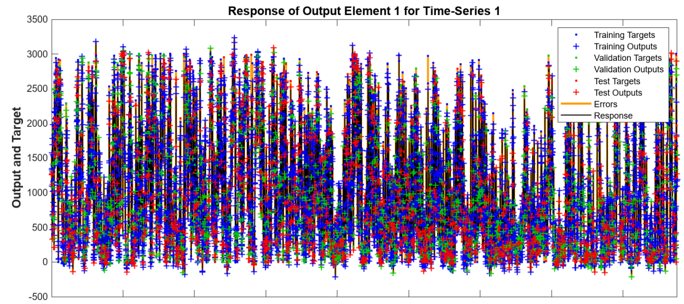

Fig. 8. Response of NAR during training, validation and testing.

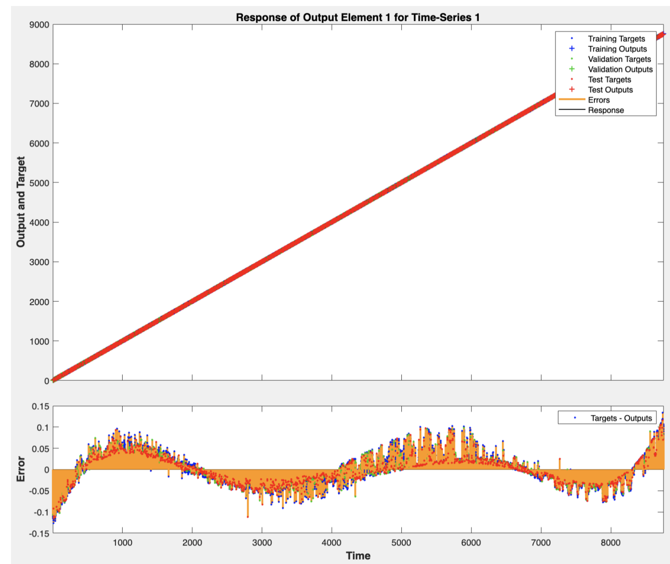

Fig. 9. Response of NARX during training, validation and testing.

E. NARX

The first case study involves the NARX Response, trained using two double arrays with 8760 time steps each. One array has one feature for the predictors, the system power generated, while the other has one feature for the responses: the hours. Based on the training, it can be observed from Fig. 9 that the estimated forecast is closer to being accurate than not, with the evaluated MSE for training, validation, and testing being 0.0013. Furthermore, the current training session shows a positive correlation between the system power generated variable and the hours.

Moving on to the second case study, the first array has 8760 time steps and one feature for the predictors, the Wind Speed, while the other has 8760 time steps and one feature for the

TABLE I
MSE COMPARISON OF VARIOUS TECHNIQUES

Technique	Stage	MSE
LSTM	Training	0.2823
	Testing	1.5757
NAR	Training	1.8410×10^{4}
	Validation	1.9018×10^{4}
	Testing	2.1093×10^{4}
NARX	Training	2.8684×10^{-4}
	Validation	3.0183×10^{-4}
	Testing	3.1311×10^{-4}

responses, the Hours. After the test, the output MSE values for the training, validation, and testing are 2.8684×10^{-4}, 3.0183×10^{-4}, and 3.1311×10^{-4}, respectively. This indicates the estimated forecast is accurate, as the MSE values are notably low. It's worth noting that wind speed and system power generated have a higher correlation regarding more precise forecasting.

F. Comparison

This research employs comparative analysis to evaluate the performance differences between the three proposed models, as presented in Table I. The comparison showcases the difference in MSE values of each algorithm after training and testing. There is also a validation area in the cases of NAR and NARX as these algorithms implement that into their overall evaluation. The lower the MSE value, the better the algorithm grasps the wind's dynamic behavior. The best MSE is captured by the NARX algorithm, which mainly focuses on the system-generated power variable. While this algorithm has the lowest MSE score, it does not account for the multivariate nature of the data. Thus, while LSTM holds the second-lowest MSE values, it also accounts for the five variables to create predictions. The MSE value of 1.5757 also showcases very little error between predicted and actual variables. The MSE is also skewed towards the right, meaning that most errors are closer to 0, with outliers being the main reason the MSE is the second-best performer.

IV. CONCLUSION

Wind energy represents a highly sustainable and eco-friendly source of power. It promises to reduce global warming by identifying the most reliable and strong wind energy points. However, predicting the surplus or shortage of energy proves challenging, as wind speed and power generated are recorded over time. Machine learning algorithms train using the simulated Texas Wind turbine data to overcome this. This data analysis uses correlation-based techniques to ensure accurate predictions and visualize variable patterns. The NARX algorithm performs best by capturing the dynamic behavior of wind energy with the smallest MSE value. Comparing the MSE results, the NARX significantly outperforms single variable analysis, whilst LSTM is best on multivariate data.

ACKNOWLEDGMENTS

The authors would like to thank Universiti Teknologi PETRONAS (UTP) in Malaysia, University of Nevada Las Vegas (UNLV), Texas A&M University (TAMU), North Carolina Agricultural Technical State University (NCAT) in the USA. This work was supported by the National Science Foundation's International Experience for Students (IRES) Site grant (Grant Numbers: OISE# 1952490-TAMU, 2208801-NCAT, and 1952497-UNLV). Any opinions, findings, conclusions, or recommendations presented are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] M. Sunder, R. Abishek, M. Maiti, K. Bingi, P. A. M. Devan, and M. Assaad, "Forecasting of wind turbines generated power with missing input variables," in *2022 International Conference on Future Trends in Smart Communities (ICFTSC)*. IEEE, 2022, pp. 98–103.

[2] L. Xiang, P. Wang, X. Yang, A. Hu, and H. Su, "Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism," *Measurement*, vol. 175, p. 109094, 2021.

[3] J. Lei, C. Liu, and D. Jiang, "Fault diagnosis of wind turbine based on Long Short-term memory networks," *Renewable energy*, vol. 133, pp. 422–432, 2019.

[4] J. Zhang, J. Yan, D. Infield, Y. Liu, and F.-s. Lien, "Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and Gaussian mixture model," *Applied Energy*, vol. 241, pp. 229–244, 2019.

[5] X. Chen, X. Zhang, M. Dong, L. Huang, Y. Guo, and S. He, "Deep learning-based prediction of wind power for multi-turbines in a wind farm," *Frontiers in Energy Research*, vol. 9, p. 723775, 2021.

[6] Z. Li, X. Luo, M. Liu, X. Cao, S. Du, and H. Sun, "Short-term prediction of the power of a new wind turbine based on IAO-LSTM," *Energy Reports*, vol. 8, pp. 9025–9037, 2022.

[7] Y. Wu and X. Ma, "A hybrid LSTM-KLD approach to condition monitoring of operational wind turbines," *Renewable Energy*, vol. 181, pp. 554–566, 2022.

[8] R. Yu, J. Gao, M. Yu, W. Lu, T. Xu, M. Zhao, J. Zhang, R. Zhang, and Z. Zhang, "LSTM-EFG for wind power forecasting based on sequential correlation features," *Future Generation Computer Systems*, vol. 93, pp. 33–42, 2019.

[9] B. Zhou, X. Ma, Y. Luo, and D. Yang, "Wind power prediction based on LSTM networks and nonparametric kernel density estimation," *IEEE Access*, vol. 7, pp. 165 279–165 292, 2019.

[10] S. M. Malakouti, A. R. Ghiasi, A. A. Ghavifekr, and P. Emami, "Predicting wind power generation using machine learning and CNN-LSTM approaches," *Wind Engineering*, vol. 46, no. 6, pp. 1853–1869, 2022.

[11] I. Delgado and M. Fahim, "Wind turbine data analysis and LSTM-based prediction in SCADA system," *Energies*, vol. 14, no. 1, p. 125, 2020.

[12] B. Ramadevi and K. Bingi, "Time series forecasting model for sunspot number," in *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*. IEEE, 2022, pp. 1–6.

[13] M. B. Omar, R. Ibrahim, R. Mantri, J. Chaudhary, K. Ram Selvaraj, and K. Bingi, "Smart grid stability prediction model using neural networks to handle missing inputs," *Sensors*, vol. 22, no. 12, p. 4342, 2022.

[14] A. Ruhan and G. Cheng, "The application of NAR dynamical neural network on quantitative investment," in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. IEEE, 2018, pp. 2671–2674.

[15] T. Serikov, A. Zhetpisbayeva, S. Mirzakulova, K. Zhetpisbayev, Z. Ibrayeva, A. Tolegenova, L. Soboleva, and B. Zhumazhanov, "Application of the NARX neural network for predicting a one-dimensional time series," *Eastern-European Journal of Enterprise Technologies*, vol. 5, no. 4, p. 113, 2021.

[16] "Texas Wind Turbine Dataset." [Online]. Available: https://www.kaggle.com/datasets/pravdomirdobrev/texas-wind-turbine-dataset-simulated