MINIMIZING WIND TURBINE DOWNTIMES USING RADIAL BASIS FUNCTION NETWORKS

Mikkel Schwarz, Hugo Johnson, Peter Hildorf, Lucia Ciprian, Julia Lackinger

Project 3, Group 2

ABSTRACT

Failures in the hydraulic pitch system of wind turbines significantly contribute to downtime, affecting economic yield. Accurate and real-time prediction of the momentum M_z, indicative of pitch system disturbances, is essential to mitigate these failures. This study evaluates neural network-based approaches for real-time M_z estimation, focusing on Radial Basis Function (RBF) networks and their hybridization with Long Short-Term Memory (LSTM) models.
The results indicate that the RBF network trained with gradient descent achieves the best performance, with a validation mean squared error (MSE) of 0.0029, outperforming both the standalone LSTM model (MSE: 0.0036) and the combined RBF+LSTM architecture ((MSE: 0.0038)). While the hybrid model demonstrated worse predictive accuracy than both standalone RBF network, despite increasing computational cost, it highlights the need for further refinement of this architecture. While the hybrid model demonstrated worse predictive accuracy than the standalone RBF network, despite increasing computational cost, it highlights the need for further refinement of this architecture.
The RBF model demonstrates a strong ability to approximate M_z dynamics, though minor deviations from actual values persist. These results suggest that RBF networks are a promising solution for minimizing wind turbine downtime. However, further work is required to refine these models and establish the precision needed for practical implementation.

***Index Terms*—** Radial Basis Function, RBF Network, LSTM
xx

1. INTRODUCTION

Minimizing the downtime of wind turbines is crucial to maximizing their economic yield. A significant source of downtime is failures in the hydraulic pitch system, which plays a critical role in regulating the blades' angle for optimal performance. Friction monitoring has emerged as a promising approach to assess the wear status of this system. Central to this effort is the real-time calculation of the momentum M_z, a parameter indicative of disturbances in the pitch system. Although M_z can be determined through complex simulations, these methods are computationally intensive and do not support real-time applications.
Neural networks offer a potential solution for estimating M_z in real-time, given their ability to model complex, nonlinear relationships. Classic approaches for time-series data often rely on Recurrent Neural Networks, such as Long Short-Term Memory (LSTM) networks, due to their feedback loops, which incorporate information from previous time steps [1]. However, Radial Basis Function (RBF) networks present an alternative approach that may be effective for time-series tasks, given their localized response to input features [2]. In this report, we explore the potential of RBF networks for estimating M_z in real-time. Specifically, we implement and evaluate an RBF network trained using gradient descent, and an RBF network extended by an LSTM layer. These models are then compared to a benchmark LSTM model to assess their performance.

2. METHODOLOGY

2.1. Network architectures

The given regression problem is based on complex nonlinear functions that need to be solved in order to calculate the exact M_z in real-time. The proposed solution to capture the time-delayed relation is to integrate an RBF into the neural network. This approach allows the network to find localized patterns in the data by emphasizing inputs that are closer to a neuron's center x_c. When an input x is fed into the network, a Radial Basis Function (RBF), also known as Gaussian kernel with standard deviation σ, is used to calculate each neuron's weight with regard to the euclidian distance of x and x_c [3].

$$K(\mathbf{x}, \mathbf{x_c}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x_c}\|^2}{2\sigma^2}\right) \tag{1}$$

The fundamental idea of an RBF is that if two items have similar inputs, their outputs will likely be similar as well. Therefore, the RBF kernel gives more weight to data points that are close together [4] [5]. The network consists out of one hidden layer.

The following architectures were implemented to investigate the potential of RBFs:

- RBF network trained using gradient descent
- RBF in combination with LSTM layer
- LSTM as a benchmark for the RBF+LSTM model

In the following, the different methods will be introduced further from a theoretical perspective.

RBF network trained using gradient descent

In this approach, the centers, the standard deviations, and the output weights w_i are learned through iterative training using gradient-based optimization. Training involves these two main steps:

- **Forward propagation**: Inputs are passed through the RBF layer, where the activations $\phi_i(x)$ are computed for each kernel. These activations are then used to calculate the final output.
- **Backward propagation**: The error between the predicted and target outputs is computed, and gradients are calculated with respect to the centers, standard deviations, and weights. These parameters are updated using gradient descent.

While this approach allows for flexibility and precise tuning of the model parameters, it has weaknesses. The training process can be unstable, especially for larger networks, due to sensitivity to initializations and gradient vanishing issues. Additionally, it is computationally more expensive than the closed-form solution due to the iterative nature of training [6].

LSTM

An LSTM model is known for its good performance in sequential and time-series data like in the given problem. In contrast to recurrent networks, it addresses the vanishing and exploding gradient problem by controlling the information flow [7]. Three gates (Forget, Input, and Output) are used to decide on the share of long-term memory that is supposed to be remembered and how the long- and short-term memory are updated. Sigmoid and Tanh are used as activation functions for the different gates [1, 8].

RBF in combination with LSTM layer

The LSTM model is extended by an RBF layer. Specifically, the output of the RBF layer, $\phi_i(x)$, is used as the input to the LSTM layer. This hybrid architecture aims to combine the strengths of RBF networks in feature extraction with the temporal modeling capabilities of LSTMs.

2.2. Optimizations

Moody et al. [6] details the steps involved in maximizing the performance of a single-layered RBF model. These are (1) to use a clustering method (e.g. k-means) to distribute the clusters in the input space, in order to ensure the function is closely set to the expected inputs. (2) Setting the sigma values to a local heuristic (e.g. the distance to the nearest other unit.) (3) Normalize the data if some features have a vastly different scale. These steps were implemented in each model, with the exception of (2): we found that changing the standard deviation had little effect on the mean squared error (MSE) and so we kept σ

2.3. Data preparation

Data from each of the wind speeds was first loaded into a dictionary and then concatenated into a Pandas data frame. The data was then normalized to fit within a range between zero and one. Next, the selected input features were filtered and finally, data was split into train, tests, and validation sets. For models that use time-delayed features, an algorithm was created to programmatically generate the time-delayed features just before the data is split into sets.
Where appropriate, the processed data was loaded from a pickle file to save computation time.

3. RESULTS AND DISCUSSION

In this section the MSE and prediction performances of the different architectures are compared and discussed.

3.1. RBF network trained using gradient-descent

Figure 1 shows the evolution of the training and validation MSE over increasing epochs. The validation MSE at the final epoch is 0.0029. While the training MSE decreases steadily, the validation MSE shows some fluctuations. This behavior may be attributed to the instability of RBF networks, as discussed in Section 2.1.

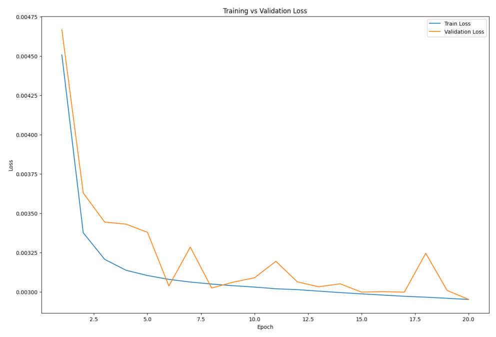

Fig. 1. RBF: Test and Validation MSE over 20 epochs.

Figure 2 presents a comparison of the actual (blue) and predicted (orange) values of M_z over one second.

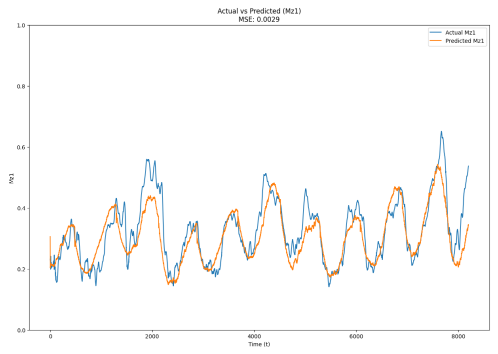

Fig. 2. RBF: Actual values vs. predictions of M_z made by RBF network between 0 to 1 seconds.

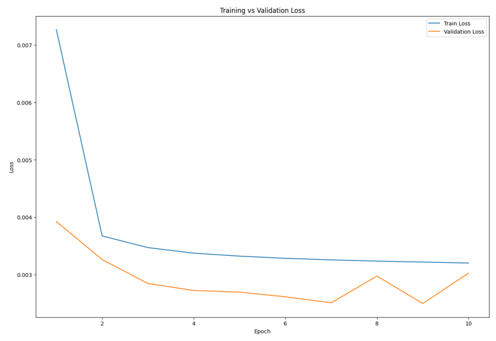

Fig. 3. LSTM: Test and Validation MSE over 10 epochs.

3.2. Model Performance and Suitability Analysis

Among the tested architectures, the standalone RBF network achieved the best results with a validation MSE of 0.00290, demonstrating higher accuracy in predicting M_z. This highlights its ability to effectively capture localized patterns within the dataset.

The LSTM standalone model, with a validation MSE of 0.0039, exhibited higher prediction errors, likely because of its focus on modeling long-term dependencies, which was less critical for this specific dataset. While the RBF+LSTM hybrid model achieved a slightly lower validation MSE of 0.0038 compared to the LSTM, it did not surpass the standalone RBF network. Furthermore, the hybrid architecture introduced additional computational complexity without delivering performance gains.

These results underline the standalone RBF network's suitability for minimizing wind turbine downtime in this study. However, the relatively higher errors in both, the LSTM and the hybrid model as well as the sign of underfitting in the LSTM network suggest that further optimization

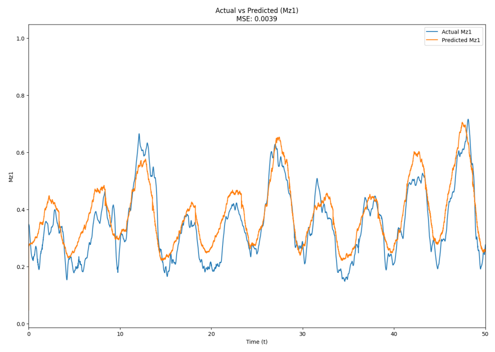

Fig. 4. LSTM: Actual values vs. predictions of M_z over time from 0 to 50 seconds

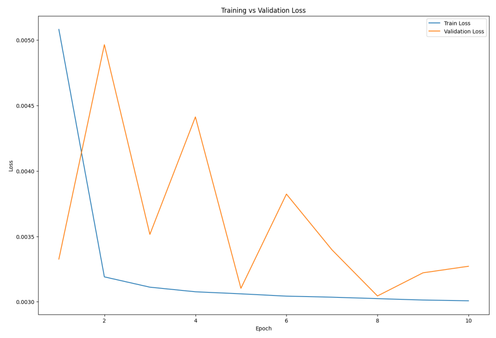

Fig. 5. LSTM+RBF: Test and Validation MSE over 10 epochs.

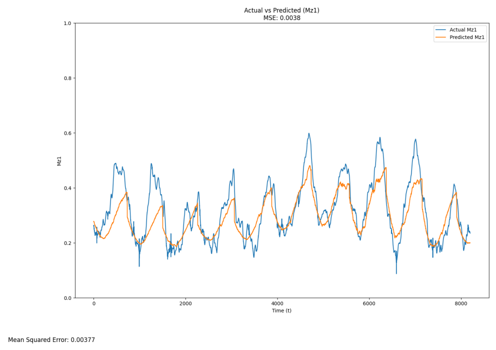

Fig. 6. LSTM+RBF: Actual values vs predictions of M_z over time 0 to 1.5 second

for the LSTM-based architectures may enhance their performance in future implementations.

The LSTM standalone model, with a validation MSE of 0.0039, exhibited higher prediction errors than the RBF and the hybrid LSTM + RBF model. Although the LSTM is well-suited for sequential and time-series data due to its ability to incorporate long-term dependencies, it was less effective in capturing localized patterns within the dataset. The hybrid RBF+LSTM model did not outperform the standalone RBF network, achieving a validation MSE slightly higher of 0.0038. The added computational complexity of the hybrid model did not translate into performance improvements. Drawn from the results, the RBF had the best performance making it the better architecture for minimizing wind turbine downtime in this study.

4. CONCLUSION

In this study, several neural network architectures were evaluated for their suitability in predicting M_z values in real time to minimize the downtime of wind turbines. Among the models, the RBF network demonstrated the best performance, achieving a validation MSE of 0.0029. The LSTM model as well as the RBF+LSTM network performed worse, with a validation MSE of 0.0039 and 0.0038 RBF, respectively. The combined version introduced higher computational cost due to the additional LSTM layer, making it less suitable for this application. When comparing the predicted M_z values with the actual values, the RBF model showed good approximations of the dynamics, though slight deviations were observed. While the results are promising, they are not perfectly precise. To determine whether this level of accuracy is sufficient to effectively minimize wind turbine downtime and maximize economic yield, further analysis is required to establish how accurate the M_z predictions need to be for practical application.

5. FUTURE WORK

Future work will focus on refining the combined RBF+LSTM network to better balance computational efficiency and predictive accuracy. Despite the increased complexity of the combined model, the validation error did not show significant improvement over the standalone RBF network. To address this, the model's complexity could be increased step-by-step by adding additional layers and experimenting with hyperparameters such as learning rate and dropout rates. Further, incorporating advanced regularization techniques will be essential to avoid overfitting and improve generalization. These steps may help unlock the full potential of the combined architecture and achieve a more favorable trade-off between complexity and performance.

6. REFERENCES

[1] Maria Kaselimi, Nikolaos Doulamis, Anastasios Doulamis, Athanasios Voulodimos, and Eftychios Protopapadakis, "Bayesian-optimized bidirectional lstm regression model for non-intrusive load monitoring," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 2747–2751.

[2] Martin Buhmann, "Radial basis function," *Scholarpedia*, vol. 5, no. 5, pp. 9837, 2010, Curator: Martin Buhmann. Prof. Martin Buhmann, Mathematisches Institut, Justus-Liebig-Universität Giessen, Germany. Accepted on: 2010-05-26 07:52:16 GMT.

[3] Xiaojian Ding, Jian Liu, Fan Yang, and Jie Cao, "Random radial basis function kernel-based support vector machine," *Journal of the Franklin Institute*, vol. 358, 2021.

[4] Yin-Wen Chang, Cho-Jui Hsieh, and Kai-Wei Chang, "Training and testing low-degree polynomial data mappings via linear svm," in *Journal of Machine Learning Research*, 2010.

[5] Fabian Wurzberger and Friedhelm Schwenker, "Learning in deep radial basis function networks," in *Institute of Neural Information Processing, Ulm University*, 2024.

[6] John Moody and Christian J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989, From Yale Comp Sci.

[7] Alex Graves, "Chapter 4: Long short-term memory," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer Berlin, Heidelberg, 2012.

[8] Sajjad Ali Haider, Syed Rameez Naqvi, Tallha Akram, Gulfam Ahmad Umar, Aamir Shahzad, Muhammad Rafiq Sial, Shoaib Khaliq, and Muhammad Kamran, "Lstm neural network based forecasting model for wheat production in pakistan," *Agronomy*, vol. 9, no. 2, 2019.

7. APPENDIX

Github repository link: https://github.com/hugojjohnson/02456-deep-learning-project