Proceedings of the 8th North American International Conference on Industrial Engineering and Operations Management, Houston, Texas, USA, June 13-16, 2023

Applications of Artificial Intelligence Algorithms in Wind Speed

Raghad Alkhamis and Nuha Aloud
Industrial Engineering Department, College of Engineering,
Princess Nourah Bint Abdulrahman University, Riyadh, KSA

Ali AlArjani
Industrial Engineering Department, College of Engineering,
Prince Sattam Bin Abdulaziz University, AlKharj, KSA
a.alarjani@psau.edu.sa

Abstract

Wind energy plays a crucial component in the contest to fulfill environmental control objectives. Wind energy, on the other hand, will only be able to fulfill its essential importance if the wind turbines work efficiently. The paper aims to analyze the application of artificial intelligence (AI) algorithms in wind speed. In this paper, three network parameter optimization algorithms, AdaGrad, RMSprop, and Adam, are implemented and compared in the context of wind speed forecasting. This paper employs wind speed data obtained from the Jeddah Al Jazeera database in Saudi Arabia. Mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), and R-squared are the four metrics used to assess performance. The experiment results show that the Adam algorithm outperforms the other optimization algorithms regarding forecasting accuracy and training time. Therefore, researchers can use this study to help them choose optimization algorithms for wind energy forecasting.

Keywords
Wind energy, Artificial intelligence, Training, optimization algorithms

1. Introduction

Machine learning is a subset of artificial intelligence (AI) that allows computers to learn and evolve without being explicitly programmed (Han & Pei, 2011). Neural Network (NN) is an AI technique capable of instructing computers to analyse information similarly to the human brain by imitates the workings of the human brain using interconnected neurons in a layered structure called deep learning. ANNs are comprised of three layers, containing an input layer, one or more hidden layers, and an output layer. Each ANN consists of several neurons that are interconnected to one another in various layers of the networks, these neurons are known as nodes and has an associated weight and threshold. The output Z for layer i can be calculated by using Equation 1.

$$Z_i = W_{i-1}x + b_i$$

Equation 1: ANN output formula

Where W and b are the weight and bias.

Recurrent Neural Network (RNN) is one of the several algorithms of deep learning distinguished by its ability to internal memory, it can be used for applications like speech recognition, time series forecasting, and language translation since they are built to store information from previous inputs, enabling the use of context and dependencies between time steps. Gated Recurrent Unit Network is a type of RNN that use gating mechanisms to selectively update the hidden state at each time step using gating methods, effectively modelling sequential data. They have shown successful at a number of natural languages processing tasks, including language modelling, machine translation, and speech recognition. Long Short-Term Memory (LSTM) is one of the applications used in Recurrent Neural Networks (RNN). In LSTM, sequential data are used to overcome long-term dependency to achieve an excellent learning process (LSTM. Accessed. 1 November 2022). It has exceptional performance on a wide range of problems, such as avoiding the long-term reliance problem. A cell, an input gate, an output gate, and a forget gate make up a typical LSTM unit. The three gates control the flow of information into and out of the cell.

Optimization theory describes and analyses algorithms for solving well-structured optimization problems. For example, optimization algorithms in machine learning (particularly neural networks) seek to minimize an objective function (the loss or cost function), which is intuitively the difference between the predicted data and the expected values. Many fields of science and engineering rely heavily on stochastic gradient-based optimization. Many problems in these fields can be framed as the optimization of some scalar parameterized objective function that requires parameter maximization or minimization. Adagrad, RMSprop, and Adam are three first-order optimization algorithms applied in this article.

An adaptive gradient descent algorithm (AdaGrad) is one of a family of algorithms for stochastic optimization using multi-gradient descents. A member of that family of algorithms approximates the Hessian of the optimized function in a similar manner to second-order stochastic gradient descent. Its name comes from Adaptative Gradient. Generally, it assigns a higher learning rate to infrequent features, which ensures that parameter updates are less dependent on frequency and more dependent on relevance. Accordingly, the learning rate is adapted for each feature based on the estimated geometry of the problem; mainly, it tends to assign a higher learning rate to infrequent features. In 2011, AdaGrad was published in a highly cited paper in the Journal of machine learning research (Duchi et al., 2011). There is no doubt that it is one of the most popular algorithms for machine learning (particularly for training deep neural networks) and has influenced the Adam algorithm's development (Kingma & Ba, 2014).
The Adagrad, RMSprop, and Adam machine learning algorithms are employed in this study since they are among the most extensively used nowadays. The results of Adagrad, RMSprop, and Adam were compared to determine the best model for this study.

1.2 Data

Data was collected from the Jeddah Al Jazeera wind power station from 1/1/2016 to 30/11/2016. The focus was on an hourly average speed of 98m (in m/s) winds containing 8040 values of average hourly speed

2. Literature Review

In United States, a method for predicting wind speed using artificial neural networks was provided in the study. Hourly wind data in miles per hour and dry bulb temperature data in degrees Fahrenheit from four separate places in the United States were used in this manner for a whole year. In order to compare results across multiple data sets and prevent local extrema from skewing results, the data were standardized in the range between 0 and 1. NAR and NARX, two ANN prediction models, were applied. Both time series prediction models produced respectable results, but it was discovered that NARX's performance was superior to NAR's when external data on dry bulb temperature were included. Another benefit of NARX was that fewer delays were employed, which reduced the amount of previous data that went into forecasting. The NARX network's inaccuracy was one order of magnitude lower than the NAR network's when forecasting one hour in advance. The NARX network exhibited inaccuracy two orders of magnitude less than the NAR network when forecasting 5 hours in advance. This strengthens the claim that NARX networks use less information to make predictions that are more accurate than those made by NAR networks. By using other ANN algorithms and more weather data, the wind speed forecast may be enhanced. The amount of time these models can reliably anticipate may be extended by including numerical weather prediction methods (Blanchard, T., & Samanta, B. 2020).

The intermittent nature of wind makes accurate wind speed estimates difficult when using wind energy for the generation of electrical power. In Tehran, Iran, four precise models have been successfully constructed to predict wind speed. A thorough wind database is used to assess the short-term wind speed predicting capabilities of the ANN-RBF, ANFIS, ANN-GA, and ANN-PSO models. The ANN-GA model has lower mean square and root mean square error amounts than other models, and its determination coefficient R^2 is higher than that of the other models in comparison. Consequently, we can draw the conclusion that the ANN-GA model is the best technique for short-term wind speed forecasting in Tehran. It must be remembered, nonetheless, that all of the models that have been shown have produced accurate results and can be applied to the short-term forecasting of wind speed in Tehran (Fazelpour et al. 2016).
Despite the fact that convolutional neural networks were the most effective in Netherland study, a disadvantage of this approach is that it is challenging for a meteorologist to interpret. Although the coastline is clearly visible, these do not clearly indicate which input features are crucial. To visualize which elements of an input image are most important for a convolutional neural network's prediction, explain ability techniques like layer-wise relevance propagation would be a useful contribution to future research. This can be particularly helpful when fitting a truncated normal distribution since in such situation it might be feasible to discriminate between features that are important for predicting the spread and features that are important for correcting the bias. Convolutional neural networks also have the disadvantage of

requiring a lot of training data, which makes them less likely to be used for local postprocessing (Veldkamp et al. 2021).

3. Methodology

This article compares Adagrad, RMSprop, and Adam algorithms to determine the most accurate. In addition, a theoretical foundation for Adagrad, RMSprop, and Adam algorithms is presented in this section. We have collected and processed data and applied machine learning models for training and testing the model to conclude the study's results. The data collected was cleaned and filled with average values using the ¨fillna¨ function, and the cleaned data were normalized. To obtain the training and data from the cleaned and normalized data, we developed the computer code in python to divide the database into two data categories: (1) training and (2) testing. Also, the data need to be divided into inputs and outputs for training preparations while changing the percentage for each combination and with several the time step for all combinations fixed at 24. After that, the database was prepared to be used in the LSTM network. Furthermore, the number of neurons and optimizer was structured with the developed code in python while training the network and specifying the required number of epochs as structured using the developed code. Also, we developed the codes required to find the predicted values, then plotted the distribution of the actual and predicted values. Finally, we used the developed code to find the required target, Errors, and R-squared results.

3. Results and Discussion

In this section, the data of the average wind speed will be analyzed using three optimizers, three number of epochs and three percentages of training set, which means 27 combinations will be applied.

3.1 Adagrad Optimizer

Adagrad is an optimizer that measures the frequency of updated parameters. Using 100, 500 and 100 epochs, results are shown below (Table 1, 2 and 3; Figure 1, 2 and 3) (Adagard. Accessed 1 November 2023).

Table 1. Adagrad Optimizer with 100 Epochs

Training %	MAE	MSE	RMSE	R-Squared
70	0.102099456735904	0.015263722062200	0.123546436865656	-0.231409599604211
80	0.097291308433315	0.013447828114362	0.115964771005518	-0.269279013185804
90	0.101271369207758	0.014915734210527	0.122129988989300	-0.562218072698157

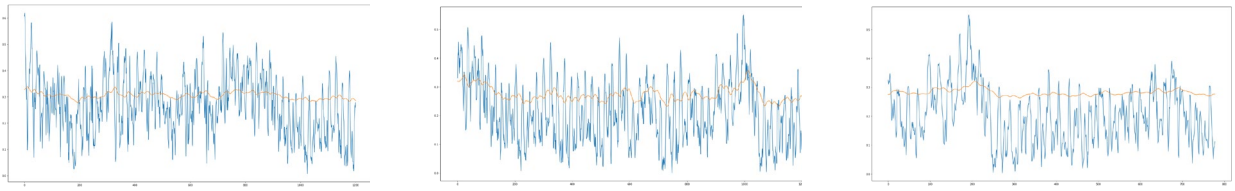

Figure 1. Adagrad Optimizer, 70%, 80 %, and 90 % Training with 100 Epochs

Table 2. Adagrad Optimizer with 500 Epochs

Training %	MAE	MSE	RMSE	R-Squared
70	0.08446909747322179	0.01028769153668553	0.10142825807774443	0.17003453912368194
80	0.08466749579129915	0.009997617049400115	0.09998808453710929	0.05637063511249485
90	0.07645400752627968	0.008385748194192497	0.09157373091772825	0.12170817760897923

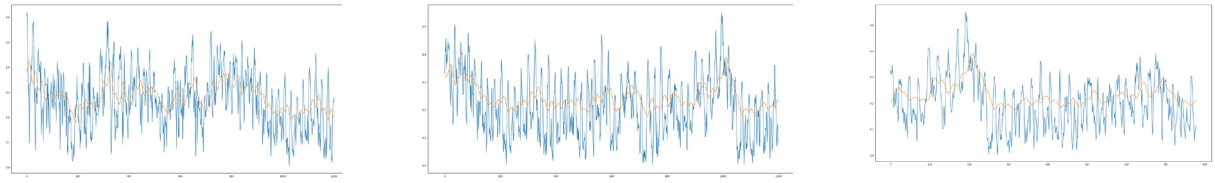

Figure 2. Adagrad Optimizer, 70%, 80 %, and 90 % Training with 500 Epochs

Table 3. Adagrad Optimizer with 1000 Epochs

Training %	MAE	MSE	RMSE	R-Squared
70	0.08351640905702855	0.010010875475428463	0.10005436260068055	0.1923668348617409
80	0.08455982816445731	0.009919344550410481	0.09959590629343397	0.0637584184357396
90	0.07597958388897234	0.008185787828402942	0.09047534375951793	0.14265127654401644

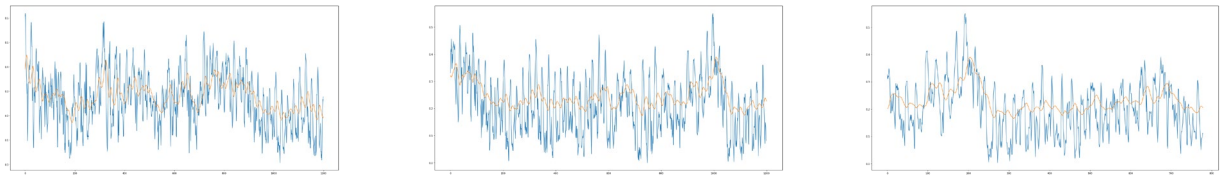

Figure 3. Adagrad Optimizer, 70%, 80 %, and 90 % Training with 1000 Epochs

Based on above MAE, MSE, RMSE, R2 and figures attached, Adagrad has proven to be an inadequate representation of the data. Due to the extremely low R-squared values and the fact that all plots indicates that the predicted values do not match the real values.

3.2 RMSprop Optimizer

Root Mean Squared Propagation is an extension version of Adagrad, which adapts the step size for each parameter to decay the average of partial gradients. Using 100, 500 and 100 epochs, results are shown below (Table 4, 5 and 6; Figure 4, 5 and 6) (A look at gradient descent and RMSprop optimizers. Accessed 1 November 2023).

Table 4. RMSprop Optimizer with 100 Epochs

Training %	MAE	MSE	RMSE	R-Squared
70	0.039447290315807744	0.002703104016556114	0.05199138406078563	0.7819255211047644
80	0.03606548843112073	0.002198012397863341	0.04688296490051947	0.7925396589245135
90	0.03349383553700266	0.0019043858998515513	0.043639270157182414	0.8005417615956156

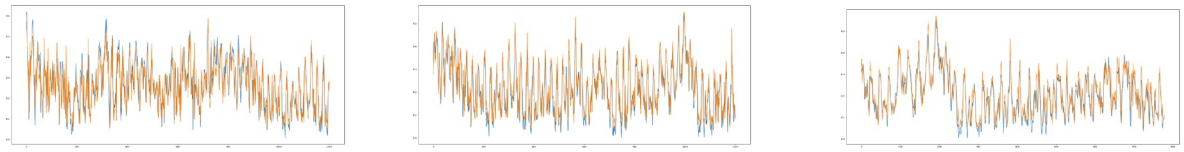

Figure 4. RMSprop Optimizer, 70%, 80 %, and 90 % Training with 100 Epochs

Table 5. RMSprop Optimizer with 500 Epochs

Training %	MAE	MSE	RMSE	R-Squared
70	0.045507254961848737	0.0035554697966667464	0.05962776028551422	0.713160419137813
80	0.041905216749210356	0.0029454458151438137	0.05427196159292396	0.7219928358806773
90	0.04108105863084746	0.0028685331208618396	0.05355868856555246	0.6995605968641474

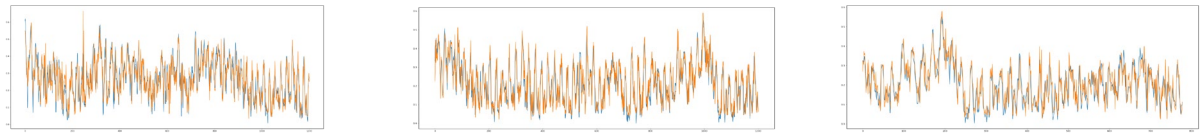

Figure 5. RMSprop Optimizer, 70%, 80 %, and 90 % Training with 500 Epochs

Table 6. RMSprop Optimizer with 1000 Epochs

Training %	MAE	MSE	RMSE	R-Squared
70	0.043514410152343364	0.003206427261929045	0.05662532350396812	0.7413196273699008
80	0.04041956623846255	0.0026554547200979793	0.05153110439431683	0.7493637695230648
90	0.03926618812989912	0.0025637793237426147	0.050633776510770105	0.7314793668598641

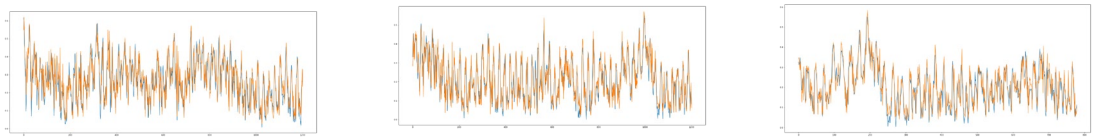

Figure 6. RMSprop Optimizer, 70%, 80 %, and 90 % Training with 1000 Epochs

RMSprop has shown to be an appropriate representation of the data based on the above MAE, MSE, RMSE, R2, and figures attached. Because of the high R-squared values and the fact that all plots show that the projected values match the actual values.

3.3 Adam Optimizer

Adam is a stochastic gradient descent optimizer for training deep learning models. Using 100, 500 and 100 epochs, results are shown below (Table 7, 8 and 9; Figure 7, 8 and 9) (Brownlee, 2021).

Table 7. Adam Optimizer with 100 Epochs

Training %	MAE	MSE	RMSE	R-Squared
70	0.03405026911273833	0.002047101753886266	0.0452449085962859	0.8348488828065774
80	0.034162093130618784	0.0019465738720801753	0.044119994017227326	0.8162717918138459
90	0.0334424818649305	0.0018574670105435077	0.04309834115767691	0.8054558701331773

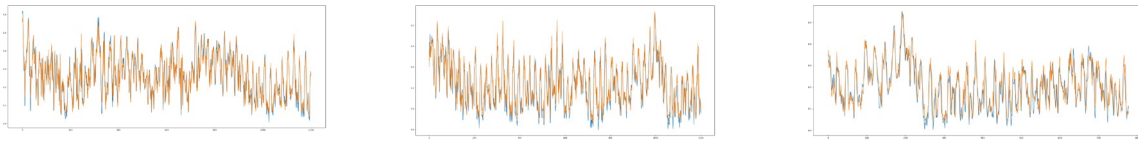

Figure 7. Adam Optimizer, Optimizer, 70%, 80 %, and 90 % Training with 100 Epochs

Table 8. Adam Optimizer with 500 Epochs

Training %	MAE	MSE	RMSE	R-Squared
70	0.04545510586539732	0.0033975986125436685	0.05828892358367641	0.7258967681644689
80	0.03964435055276435	0.002582258527788611	0.05081592789459434	0.7562724234672782
90	0.0419102632081567	0.002964750002642702	0.05444951792846932	0.6894832014443064

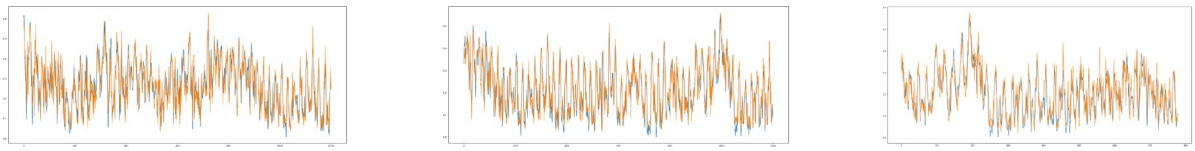

Figure 8. Adam Optimizer, Optimizer, 70%, 80 %, and 90 % Training with 500 Epochs

Table 9. Adam Optimizer with 1000 Epochs

Training %	MAE	MSE	RMSE	R-Squared
70	0.043210125093153584	0.0031795414220506047	0.05638742255193621	0.7434886580417658
80	0.038650884717791274	0.0024599962011752395	0.04959834877468442	0.7678122055015146
90	0.0396867550084522	0.0026302725380088184	0.051286182720191004	0.7245151169226975

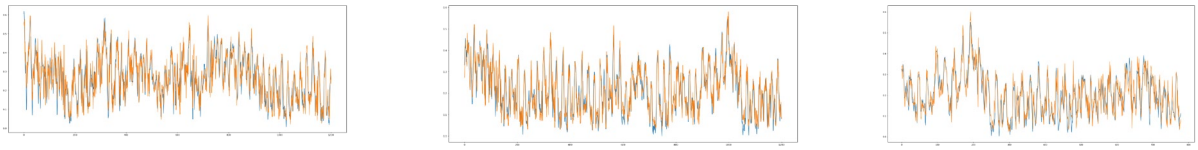

Figure 9. Adam Optimizer, Optimizer, 70%, 80 %, and 90 % Training with 1000 Epochs

Adam has proven to be a sufficient representation of the data based on the above MAE, MSE, RMSE, R2, and graphs attached. Because of the high R-squared values and the fact that all plots show that the projected values match the actual values.

According to the results of the above investigation, the Adagrad optimizer needs a better anticipated model of average wind speed. In contrast, the RMSprop and Adam optimizers have superior predicted outcomes in the Jeddah Al Jazeera database. Furthermore, as demonstrated in table 10, training the above models across 100 epochs for both RMSprop and Adam has proven to be the best training replications.

Table 10. Summary of all measures

Optimizer	Epochs	Training (%)	MAE	MSE	RMSE	R-Squared
Adagrad	100	70	0.1020995	0.0152637	0.1235464	-0.2314096
		80	0.0972913	0.0134478	0.1159648	-0.2692790
		90	0.1012714	0.0149157	0.1221300	-0.5622181
	500	70	0.0844691	0.0102877	0.1014283	0.1700345
		80	0.0846675	0.0099976	0.0999881	0.0563706
		90	0.0764540	0.0083857	0.0915737	0.1217082
	1000	70	0.0835164	0.0100109	0.1000544	0.1923668
		80	0.0845598	0.0099193	0.0995959	0.0637584
		90	0.0759796	0.0081858	0.0904753	0.1426513
RMSprop	100	70	0.0394473	0.0027031	0.0519914	0.7819255
		80	0.0360655	0.0021980	0.0468830	0.7925397
		90	0.0334938	0.0019044	0.0436393	0.8005418
	500	70	0.0455073	0.0035555	0.0596278	0.7131604
		80	0.0419052	0.0029454	0.0542720	0.7219928
		90	0.0410811	0.0028685	0.0535587	0.6995606
	1000	70	0.0435144	0.0032064	0.0566253	0.7413196
		80	0.0404196	0.0026555	0.0515311	0.7493638
		90	0.0392662	0.0025638	0.0506338	0.7314794
Adam	100	70	0.0340503	0.0020471	0.0452449	0.8348489
		80	0.0341621	0.0019466	0.0441200	0.8162718
		90	0.0334425	0.0018575	0.0430983	0.8054559
	500	70	0.0454551	0.0033976	0.0582889	0.7258968
		80	0.0396444	0.0025823	0.0508159	0.7562724
		90	0.0419103	0.0029648	0.0544495	0.6894832
	1000	70	0.0432101	0.0031795	0.0563874	0.7434887
		80	0.0386509	0.0024600	0.0495983	0.7678122
		90	0.0396868	0.0026303	0.0512862	0.7245151

As seen from the above figures and tables, the Adam optimizer with 100 epochs and 90% of training data produces the best results, followed by RMSprop.

Nonetheless, Adam has proven to be the greatest optimizer with 100 epochs across all training percentages. Logically, as the number of epochs increases, the MAE, MSE, and RMSE fall, while R-squared increases, and as the percentage of training increases, the MAE, MSE, and RMSE decrease, while R-squared increases. The results of Adam were studied to test this model, as shown in figure 10 below.

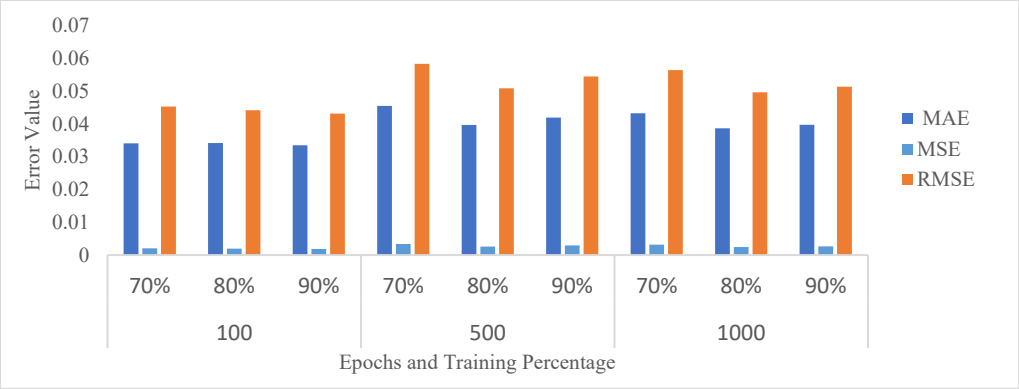

Figure 10. Measures of accuracy of Adam optimizer

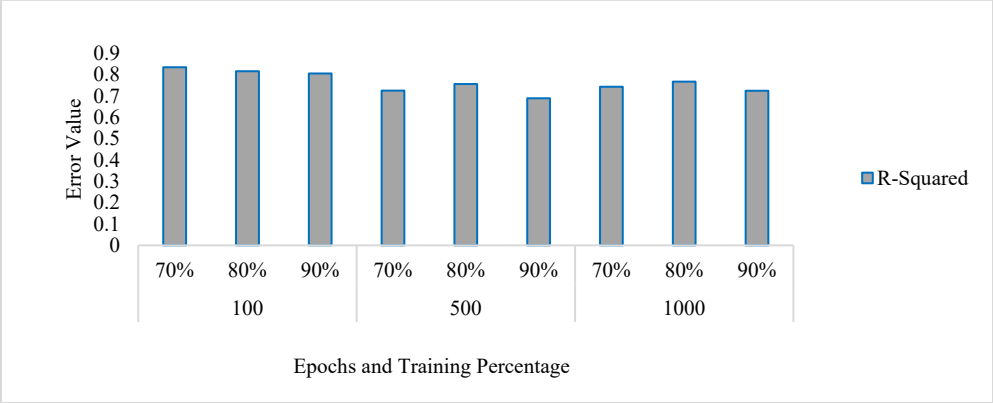

Figure 11. R-Squared of Adam optimizer

Table 10 and the Figure 11 above show that the 500 epochs with a 70% training percentage had a higher MAE, MSE, and RMSE than all 1000 epochs with a 10% training percentage. Furthermore, the MAE, MSE, and RMSE of the 500 epochs with a 90% training percentage are greater than those of the same epochs with an 80% training percentage. Likewise, all epochs with 80% training showed a higher R-Squared value than epochs with 90% training. As a result, the preceding theory is abandoned.

3.4 Sensitivity Analysis

The above results represent the current state of the Jeddah Al Jazeera average wind speed database. Changing the results above to a different database may result in changes that support the above hypothesis.

The change will be to use the aforementioned approach and data analysis from a different database of Hafar Al-Batin District wind average speed, which has similar results in the 100 and 500 epochs, as shown in Figure 12 below.

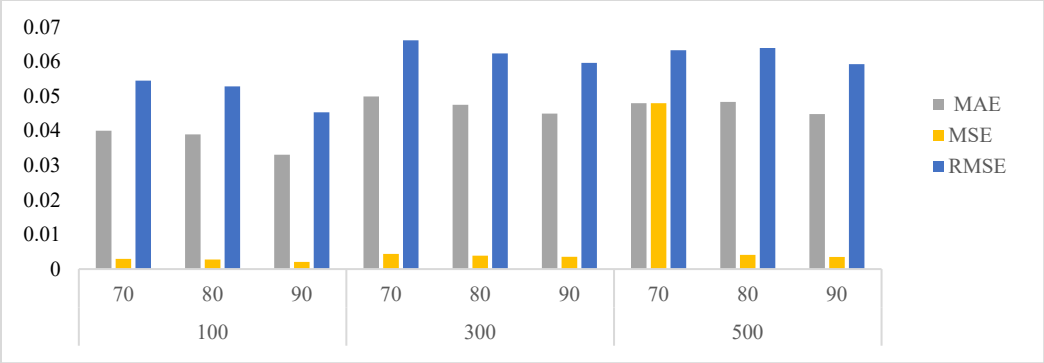

Figure 12. Sensitivity Analysis Adam optimizer

4. Conclusion

The machine learning concept, as well as artificial intelligence principles, are discussed in this report. The above concept was implemented using Python from Google Colab. The average wind speed database for Jeddah Al Jazeera has been characterized and analyzed using Adagrad, RMSprop, and Adam optimizers over 100, 500, and 1000 epochs with training percentages of 70%, 80%, and 90%.

Based on the results, hypotheses of a relationship between the number of epochs and training percentage and MSE, MAE, RMSE, and R-Squared have been generated and demonstrated to be declined by evaluating the present database and adjusting it to do sensitivity analysis. The combination of Adam optimizer through 100 epochs with 90% training has proved to be the best analyzer in this sample. However, it is recommended to test this database on a more significant percentage of training to improve the results.

References

LSTM, https://towardsdatascience.com/lstm-by-example-using-tensorflow-feb0c1968537, Accessed on November 1, 2022.

Han, J., Kamber, M., & Pei, J. , Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.

Team, K. Keras Documentation: Adagrad, Keras, 2014, https://keras.io/api/optimizers/adagrad/#:~:text=Adagrad%20is%20an%20optimizer%20with,point%20value%2C%20or%20a%20tf, Accessed on November 1, 2022.

Duchi, J., Hazan, E., & Singer, Y., Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7), 2011.

Kingma, D. P., & Ba, J., Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. Accessed on February 27, 2023.

RMSProp, https://optimization.cbe.cornell.edu/index.php?title=RMSProp, Accessed on February 27, 2023.

A. Agnes Lydia and, F. Sagayaraj Francis, Adagrad - An Optimizer for Stochastic Gradient Descent, Department of Computer Science and Engineering, Pondicherry Engineering College, 2019.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: neural networks for machine learning, 4(2):26–31, 2012.

Adam, https://optimization.cbe.cornell.edu/index.php?title=Adam, Accessed on February 27, 2023.

ICLR 2015, https://www.iclr.cc/archive/www/doku.php%3Fid=iclr2015:main.html, Accessed on February 27, 2023.

A look at gradient descent and RMSprop optimizers, https://towardsdatascience.com/a-look-at-gradient-descent-and-rmsprop-optimizers-f77d483ef08b, Accessed on November 1, 2022.

Brownlee, J. (2021) Gentle introduction to the adam optimization algorithm for deep learning, Machine Learning Mastery. Available at: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/, Accessed on November 1, 2022.

Blanchard, T., & Samanta, B., Wind speed forecasting using neural networks. *Wind Engineering*, *44*(1), 33-48, 2020.

Fazelpour, F., Tarashkar, N., & Rosen, M. A., Short-term wind speed forecasting using artificial neural networks for Tehran, Iran. *International Journal of Energy and Environmental Engineering*, *7*, 377-390, 2016.

Veldkamp, S., Whan, K., Dirksen, S., & Schmeits, M., Statistical postprocessing of wind speed forecasts using convolutional neural networks. *Monthly Weather Review*, *149*(4), 1141-1152, 2021.

Biographies

Raghad Alkhamis is a senior Industrial and Systems Engineering student at Princess Nourah University (PNU), Riyadh, KSA. She is interested in data analytics and applying operation research techniques for the purpose of optimization.

Nuha Aloud is a senior Industrial and Systems Engineering student at Princess Nourah University (PNU), Riyadh, KSA. She is an IEOM member in PNU Chapter as a Treasurer from 2019 till 2022, she won the ninth place in Gulf Student Simulation Competition that held by IEOM in King Fahad University of Petroleum and Minerals Chapter in her freshman year 2019.

Ali AlArjani is an associate professor and currently the head of the department of industrial engineering, at Prince Sattam bin Abdulaziz University (PSAU), AlKharj, Riyadh, KSA. His research interest includes operation research techniques, algorithms, operation management, forecasting methods, applied statistical methods. In addition, he has keen interest on mathematical modeling. He also has published number of articles in different reputed peer-reviewed scientific journals. Also he is leading the PSAU budget spending efficiency team and work as consultant of Prince Sattam bin Abdulaziz University rector.