East African Journal of Engineering
eaje.eanso.org
Volume 8, Issue 1, 2025
Print ISSN: 2707-5389 | Online ISSN: 2707-5397
Title DOI: https://doi.org/10.37284/2707-5397

EAST AFRICAN NATURE & SCIENCE ORGANIZATION

Original Article

Wind Speed Prediction Using BiLSTM Deep Learning Model and Comparable Batch Sizes of Training Data: A Case of Singida Wind Farm Site, Tanzania

Isakwisa Gaddy Tende[1*]

[1] Dar es Salaam Institute of Technology, P. O. Box 2958, Dar es Salaam, Tanzania.
[*] Correspondence ORCID ID; https://orcid.org/0009-0000-2264-7815; Email: isakwisa.tende@dit.ac.tz

Article DOI : https://doi.org/10.37284/eaje.8.1.3441

Publication Date:

06 August 2025

Keywords:

Wind Speed Prediction, BiLSTM, Deep Learning, Batch Size, Singida, Tanzania.

ABSTRACT

Singida region, located in central Tanzania, has long been identified as a potential location for installing wind farms to generate electric power due to the steady annual wind speed. Apart from the huge potential of contributing to the national grid, wind power also helps to address carbon emissions and environmental problems associated with generating electric power using fossil fuels. Failure to accurately predict wind speed can lead to poor harvest of wind power and low contribution to the national grid, and in the end, affect consumers. Bidirectional Long Short-Term Memory (BiLSTM) is one of the Deep Learning models which can be used to predict time series parameters such as wind speed. In a BiLSTM model, a batch size is an important hyperparameter as it is used to set the number of training data samples to be processed together before the weights of a Deep Learning model are updated. Despite its importance, there is still a research gap on the impact of batch size on the prediction performance of BiLSTM models, especially in the context of predicting wind speed at the Singida Wind Farm Site, located in the Singida region, Tanzania. The goal of the study was to fill this gap by developing a BiLSTM model and comparing the performance of three batch sizes (16, 32 and 64) in predicting wind speed at the Singida Wind Farm Site. The 14-year Singida Wind Farm Site daily wind speed dataset was first pre-processed by scaling (normalizing) it using Standard scaler and then split into training, validation and test sets before used to train and test the developed BiLSTM model which used previous 5 days wind speed values as input to predict the output (next day (6^{th} day) wind speed). The trained BiLSTM model with the optimal (best performing) batch size was then saved in .h5 format and integrated with a Gradio-based web App to provide a user interface for officials in the Singida region to predict daily wind speed at the Singida Wind Farm Site. The evaluation findings revealed that batch size has an impact on the prediction performance of the developed BiLSTM model, showing that the lower the batch size, the better the prediction performance of the BiLSTM model. The findings also revealed that, 16 is the optimal (best performing) batch size with Mean Absolute Error (MAE) score of 0.58, Root Mean Squared

Error (RMSE) score of 0.76 and R^2 score of 0.79, followed with a batch size of 32 (MAE score of 0.62, RMSE score of 0.79 and R^2 score of 0.75) and followed by a batch size of 64 (MAE score of 0.66, RMSE score of 0.81 and R^2 score of 0.72). This study recommends that Artificial Intelligence (AI) software developers and researchers use a batch size of 16 in BiLSTM models when forecasting wind speed at the Singida Wind Farm Site, as well as in environments and climates which resemble that of the Singida Wind Farm Site in Tanzania.

APA CITATION

Tende, I. G. (2025). Wind Speed Prediction Using BiLSTM Deep Learning Model and Comparable Batch Sizes of Training Data: A Case of Singida Wind Farm Site, Tanzania. *East African Journal of Engineering*, *8*(1), 255-271. https://doi.org/10.37284/eaje.8.1.3441

CHICAGO CITATION

Tende, Isakwisa Gaddy. 2025. "Wind Speed Prediction Using BiLSTM Deep Learning Model and Comparable Batch Sizes of Training Data: A Case of Singida Wind Farm Site, Tanzania". *East African Journal of Engineering* 8 (1), 255-271. https://doi.org/10.37284/eaje.8.1.3441.

HARVARD CITATION

Tende, I. G. (2025) "Wind Speed Prediction Using BiLSTM Deep Learning Model and Comparable Batch Sizes of Training Data: A Case of Singida Wind Farm Site, Tanzania", *East African Journal of Engineering*, 8(1), pp. 255-271. doi: 10.37284/eaje.8.1.3441.

IEEE CITATION

I. G., Tende "Wind Speed Prediction Using BiLSTM Deep Learning Model and Comparable Batch Sizes of Training Data: A Case of Singida Wind Farm Site, Tanzania" *EAJE*, vol. 8, no. 1, pp 255-271, Aug. 2025.

MLA CITATION

Tende, Isakwisa Gaddy. "Wind Speed Prediction Using BiLSTM Deep Learning Model and Comparable Batch Sizes of Training Data: A Case of Singida Wind Farm Site, Tanzania" *East African Journal of Engineering*, Vol. 8, no. 1, Aug. 2025, pp. 255-271, doi:10.37284/eaje.8.1.3441.

INTRODUCTION

Wind energy, a kind of renewable energy, has emerged as an efficient source of electric power around the world, mostly because of its huge potential to contribute to the national electric grids and its role in reducing carbon emissions (Jung, 2024; Skibko et al., 2022; Turi et al., 2022). A study by Jung (2024) revealed global installed capacity of wind energy nearly tripled from 2014 to 2024. Singida region, located in central Tanzania, has long been identified as a potential area for installing wind farms due to steady annual wind speed (Kumwenda and Mangara, 2024; Mangara and Kumwenda, 2023). Because of this, there is an ongoing construction of the Singida Wind Power Station inside the Singida Wind Farm Site (Singida WFS).

Accurate wind speed prediction is important when harvesting wind energy as it plays a significant role in managing wind farms and converting wind power into electric energy (Filik and Filik, 2017). Failure to accurately forecast wind speed can lead to poor harvest of wind power and low contribution to the national grid, and in the end, affect consumers such as manufacturing companies needing electricity to manufacture products and households needing electricity for basic needs, such as cooking and entertainment. To address this, it is important to have in place an effective Artificial Intelligence (AI) model which can accurately predict wind speed at Singida WFS, located in the Singida region, central Tanzania.

Bidirectional Long Short-Term Memory (BiLSTM) is one of the Deep Learning models which can be used to predict various time series variables. In a Deep Learning model, a batch size is an important hyperparameter as it is used to set the number of training data samples to be processed together before the weights of a Deep Learning model are updated, and it can have a significant impact on the

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

performance of the Deep Learning model (Hwang et al., 2024).

BiLSTM models with different batch sizes have been used in several studies to predict time series variables. García et al. (2024) used a BiLSTM model with a batch size of 32 to forecast the EURO/USD exchange rate, with results revealing that the BiLSTM model outperformed the LSTM model in prediction performance, achieving a Mean Absolute Error (MAE) score of 0.0189 compared to the MAE score of 0.0331 achieved by the LSTM model. Wang et al. (2023) used a BiLSTM model with a batch size of 20 to predict post-construction subsoil settlement under embankment in China, with the findings revealing that the BiLSTM model achieved a low Relative Error of 0.92%. Hao et al. (2022) used an Attention-based BiLSTM model with a batch size of 256 to predict atmospheric temperature in Beijing, China, with the findings revealing that the Attention-based BiLSTM model outperformed the BiLSTM model by achieving an MAE score of 0.013, which is 0.72% lower than that of the BiLSTM model. Meng et al. (2025) used an Inter-Attention based BiLSTM model with a batch size of 128 to predict oil well production in Sichuan, China, with the findings revealing Inter-Attention based BiLSTM model outperformed the BiLSTM model by achieving an MAE score of 0.0902 compared to an MAE score of 0.0965 achieved by the BiLSTM model. Zhao et al. (2020) used a BiLSTM model combined with a Convolutional Neural Network (CNN) and a batch size of 512 to predict the remaining useful life of engines, with the findings revealing their proposed CNN-BiLSTM hybrid model achieved a better performance with a Root Mean Squared Error (RMSE) score of 12.51 compared to the RMSE score of 15.32 achieved by the CNN model. Zhang et al. (2024) used a CNN-BiLSTM model with an Attention Mechanism and a batch size of 16 to forecast severe convective weather in China, with the findings revealing that the CNN-BiLSTM model outperformed human forecasts in predicting short-term heavy rainfall with a Threat Score (TS) of 0.463 compared to a TS of 0.401 achieved by human forecasts. Méndez et al. (2023) used a CNN-BiLSTM model with a batch size of 32 to predict long-term road traffic flow in Madrid, Spain, with the findings revealing MAE score of the proposed CNN-BiLSTM model was 2.8% smaller than that of the BiLSTM model. Lu et al. (2021) used a CNN-BiLSTM model with an Attention Mechanism and a batch size of 64 to predict stock prices for Shanghai Composite Index stock, with the findings revealing that the proposed CNN-BiLSTM model outperformed the BiLSTM model by achieving an MAE score of 21.952 compared to an MAE score of 23.409 achieved by the BiLSTM model. Bai et al. (2025) proposed a multivariate temperature prediction model based on CNN-BiLSTM and Random Forest and a batch size of 32 to predict temperature in Hunan, China, with the findings revealing the proposed method achieved better results than the current leading Dliner method by reducing the Mean Squared Error (MSE) score by 57.5%. Zhang et al. (2025) used a CNN-BiLSTM model with an Attention Mechanism and a batch size of 32 to forecast stock market volatility using mixed-frequency data, with the findings revealing that the CNN-BiLSTM model outperformed the BiLSTM model by achieving an MAE score of 0.2683 compared to an MAE score of 0.3737 achieved by the BiLSTM model.

Despite effective results from the reviewed BiLSTM-based studies, there is still a research gap on the impact of batch size on BiLSTM model performance, and hence, it is still unknown what the optimal batch size is to use in a BiLSTM model in the context of forecasting wind speed at Singida WFS, which has unique characteristics such as a small geographical area and unique climatic conditions. Therefore, there is a need to conduct a study to compare the performance of several batch sizes in the BiLSTM model. This is because an optimal batch size cannot just be selected, assuming it will have the best performance, as it can be evident from the reviewed literature that the choice

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

of BiLSTM batch size depends on the nature of the problem and the conditions being studied.

The objectives of this study are threefold: first, to develop a BiLSTM model for predicting wind speed at Singida WFS, second, to comparatively evaluate prediction performances of different batch sizes in the developed BiLSTM model and third, to develop a Web App and integrate it with the developed BiLSTM model configured with the optimal (best performing) batch size to help officials in the Singida region to forecast daily wind speed at Singida WFS. As a result, this study aims to answer one key research question: What is the impact of batch size on the performance of the BiLSTM model in predicting wind speed at Singida WFS? This study's findings will help to fill the existing research gap on performance comparison of BiLSTM model batch sizes in forecasting wind speed, especially in the context of Singida WFS in Tanzania.

MATERIALS AND METHODS

Study Area

The study area, Singida WFS, is located in the Singida district, Singida region, Tanzania, as shown in the blue polygon in *Figure 1*. The red ellipse shows the Singida Wind Power Station currently being constructed.

Figure 1: Singida WFS (blue polygon) and Singida Wind Power Station (red ellipse)

Research Design

This study uses an experimental research design by developing a BiLSTM Deep Learning model, training it using training and validation data and testing its performance to predict daily wind speed at Singida WFS using test data (never seen before data).

Research Approach

This study used a quantitative research approach by utilising quantitative data (14-year daily wind speed data at Singida WFS) to train and evaluate (test) the performance of the developed BiLSTM Deep Learning model in predicting daily wind speed at Singida WFS.

Data Collection and Analysis Methods

The study used secondary data by downloading daily wind speed data for Singida WFS from the European Reanalysis (ERA5 Land - 11km Daily) dataset available in the Google Earth Engine (GEE) cloud platform. On the other hand, this study utilised time series analysis of 14-year daily wind speed data at Singida WFS to train and test a BiLSTM Deep Learning model to predict daily

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

wind speed at Singida WFS. Also, Descriptive Statistics was used to analyse the pattern and trend of the 14-year daily wind speed data at Singida WFS and identify key indices such as mean, standard deviation, minimum and maximum values of wind speed.

Sampling Technique and Sample Size

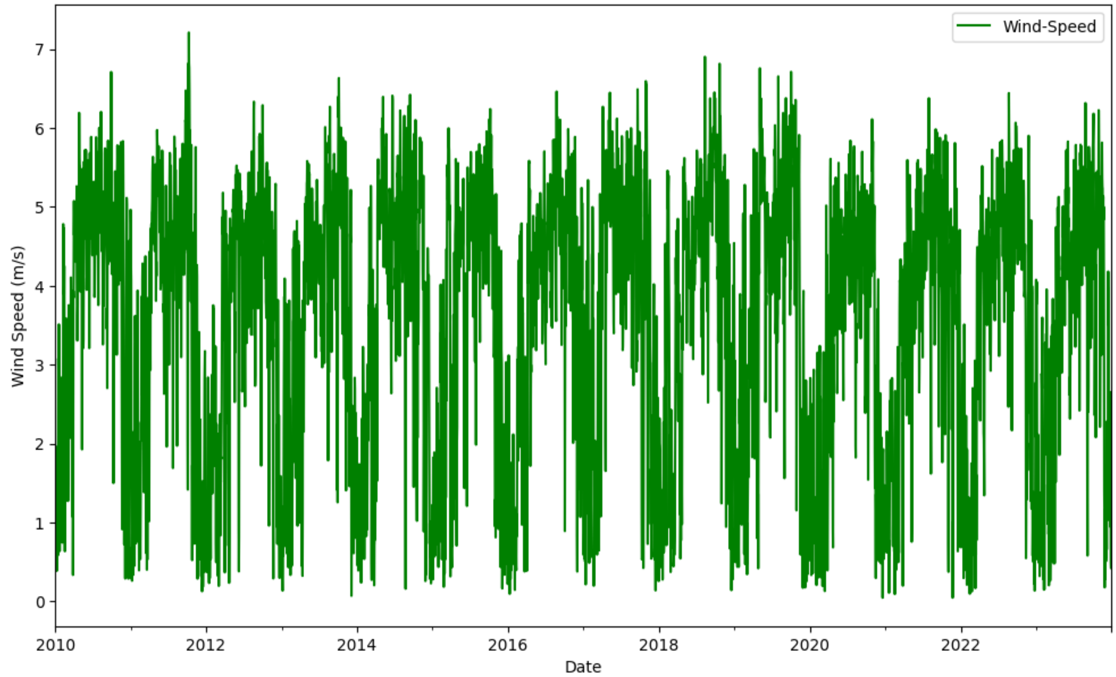
In this study, purposive sampling, a non-probability sampling approach, was used to select a single study area, Singida WFS. The study area was chosen because of the reliable availability of its historical wind speed data, its susceptibility to wind speed variability, as well as its potential in attracting the installation of wind farms and the construction of wind power stations due to its steady annual wind speed. A large sample size of temporal data points (daily wind speed data for a total of 14 years) ensured sufficient data was available for training and testing the developed BiLSTM model. Apart from that, choosing only a single study area enhances the focus of the developed BiLSTM model and reduces variability, which might happen if different climates or multiple study areas are selected. A sample size of only three batch sizes (16, 32 and 64) was chosen because of their widespread use in Deep Learning research (as evident in in studies by (Zhang et al., 2024; Radiuk, 2017; Lin, 2022; Abdulnabi et al., 2016) which used a batch size of 16, studies by (García et al., 2024; Méndez et al., 2023; Bai et al., 2025) which used a batch size of 32 and studies by (Lu et al., 2021; Zhang et al., 2025; Kandel et al., 2020; Oyedotun et al., 2023) which used a batch size of 64 and development, as well as because of their ability to represent a meaningful range from small to large batch size. A batch size of 16 is useful in capturing the effects of frequent weight updates and better generalisation, which is important for processing time series data like wind speed. A batch size of 32 serves as a commonly accepted default batch size in the literature, which balances prediction performance and computational efficiency, while a batch size of 64 reflects a larger batch size aimed at reducing training time. Limiting the study to these three batch sizes ensured methodological relevance, practical feasibility and meaningful performance comparison without incurring unnecessary computational costs.

Dataset

The European Reanalysis (ERA5 Land - 11km Daily) global dataset was used to download time-series daily wind speed data for Singida WFS (refer to the blue polygon in *Figure 1*). The downloaded wind speed data were limited to the enclosed area in the polygon. The wind speed data in CSV format for a total of 14 years (from January 1st, 2010 to December 31st, 2023) for Singida WFS were downloaded from Google Earth Engine (GEE), which hosts the ERA 5 Land – 11km Daily dataset. GEE is a cloud-based platform owned by Google (Tamiminia et al., 2020). The ClimateEngine application (Huntington et al., 2017) was used to download the CSV data from the GEE. *Figure 2* shows daily wind speed for Singida WFS for a duration of 14 years, from January 1st, 2010 to December 31st, 2023.

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

Figure 2. Daily Wind Speed at Singida WFS from January 1st, 2010 to December 31st, 2023



Pre-processing of Data

For input data to be fed into the BiLSTM model, they first need to be pre-processed. This section describes the steps used to pre-process data.

- Data Analysis**:** All of the daily wind speed data at Singida WFS for a period of 14 years were analysed. Descriptive Statistics analysis results (refer to *Table 1*) of the wind speed data revealed a count (total datapoints) of 5113, an average (Mean) value of 3.6003 m/s, a standard deviation (STD) value of 1.6571 m/s, a lowest (Min) value of 0.0473 m/s, first quartile (25%) value of 2.2287 m/s, second quartile (50%) value of 4.0796 m/s, third quartile (75%) value of 4.9083 m/s and a highest (Max) value of 7.2094 m/s.
- Data Normalisation (Scaling): The daily wind speed data at Singida WFS was normalised (scaled) by using Standard Scaler in order to enhance BiLSTM model training and convergence, as shown in equation (i), where X, μ, σ and X_S Represent actual, mean, standard deviation, and scaled wind speed values, respectively.

$$X_S = \frac{X-\mu}{\sigma} \qquad (i)$$

- Data Split: To properly train and test the BiLSTM model, the dataset needed to be split into a train, validation and test set. The train and validation sets are usually used during training of the BiLSTM model, while the test set (alternatively called unseen data) is usually used to test (evaluate) prediction performance of the BiLSTM model and measure its generalisation capability when fed with completely new data it has never seen before (Halpern-Wight et al., 2020; Doğan, 2021). For example, a pre-trained BiLSTM model exported and saved in .h5 format can be given the first 5 days of wind speed data, say wind speed data for July 1st, July 2nd, July 3rd, July 4th and July 5th as input data and asked to predict wind speed for July 6th as an output. The downloaded 14-year daily wind speed data at Singida WFS was split into a training set (the first 70%, from January 2010 to October,

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

2019), validation set (the next 15%, from October 2019 to November 2021) and test set (the last 15%, from November, 2021 to December, 2023).

- Input Features and Labels: Because the dataset contained only daily wind speed data, there was a need to create input features (inputs to the BiLSTM model) and their corresponding labels (output of the BiLSTM model) in order to train the BiLSTM model and make it learn how to map the inputs to the output. Each input-output pair consisted of a sequence of the previous 5 days' values of wind speed as inputs and the next day's (6th day) wind speed as output. Input-output pairs were created for all of the data in each of the three sets (train, validation and test sets).

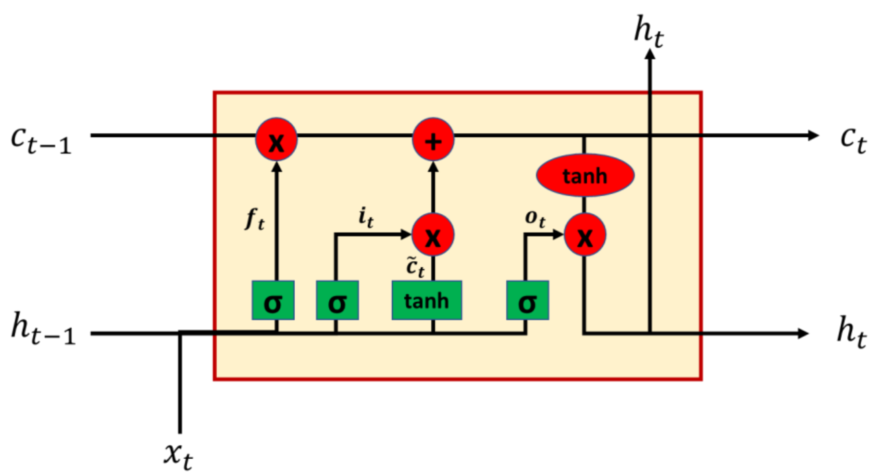
Table 1: Descriptive Statistics Analysis Results for 14-year Daily Wind Speed Data at Singida WFS

Index	Wind Speed (m/s)
Count	5113.0
Mean	3.6003
STD	1.6571
Min	0.0473
25%	2.2287
50%	4.0796
75%	4.9083
Max	7.2094

Architecture of the LSTM Unit

Long Short-Term Memory (LSTM) studied by Hochreiter and Schmidhuber (1997) is a special kind of Recurrent Neural Network (RNN), suitable and usually applied to process data arranged in sequence, like time series variables, including the wind speed. Classical RNNs are faced with a critical problem of vanishing gradients, where the RNN model fails to remember useful information from the past timesteps during RNN model training by backpropagation. To address the vanishing gradients problem, LSTM is designed to have the ability to retain useful information from past timesteps over many timesteps, making the useful past information instantly available whenever it needs to be utilised in the future. *Figure 3* shows the architecture of the LSTM unit.

Figure 3: Architecture of LSTM Unit



Article DOI: https://doi.org/10.37284/eaje.8.1.3441

The main components of the LSTM unit are the cell state. c_t and the three gates; forget gate f_t, input gate i_t and output gate o_t.

- Cell State: Cell c_t The role is to store information over timesteps. Forget gate and input are normally used to update the cell state by adding or removing information. Through this, the LSTM unit can either retain or forget information depending on whether it is relevant or not, allowing the LSTM unit to remember useful information over many timesteps.

- Forget Gate: Forget gate. f_t Role is to decide what information to discard (not keep) from the cell state. This is done through combining two inputs: the previous hidden state h_{t-1} and the current input x_t And then compute an output ranging between 0 and 1 for each component in the cell state, with an output of 0 indicating 'completely forget' and an output of 1 indicating 'completely retain'. The sigmoid neural network layer σ role (refer to equation (ii)) is to give an output ranging between 0 and 1 when fed with any input. Equation (iii) shows the forget gate with W_f and b_f Indicating weight matrices and bias vector parameters of the forget gate, respectively, both of which are learned during training of the LSTM unit.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (ii)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (iii)$$

- Input Gate: Input gate i_t The role is to decide which new information is added to the cell state. Two elements were part of the input gate: the sigmoid neural network layer σ and a *tanh* layer (refer to equation (iv) with *e* being Euler's number), whose role was creating a candidate vector. $\tilde{c}_t$ Whose values could be added to the cell state. Equations (v) and (vi) show the components of the input gate with W_i and b_i being weight matrices and bias vector parameters of the input gate, respectively, and W_c and b_c Being weight matrices and bias vector parameters of the candidate cell state, respectively, all of which are learned during training of the LSTM unit.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (iv)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (v)$$

$$\tilde{c}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (vi)$$

- Output Gate: The output gate o_t role is controlling the output of the LSTM unit. The output gate (refer to equation (vii)) does this by combining the previous hidden state h_{t-1} and the current input x_t and then decides which part of the cell state to output as the next hidden state, with W_o and b_o being weight matrices and bias vector parameters of the output gate, respectively, both of which are learned during training of the LSTM unit.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (vii)$$

At last, the cell state and hidden state need to be updated. Forget gates and input gates were used to update the cell state as shown in equation (viii) and then the updated cell state and the output gate were used to update the hidden state (LSTM unit output) as shown in equation (ix).

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (viii)$$

$$h_t = o_t * \tanh(c_t) \quad (ix)$$

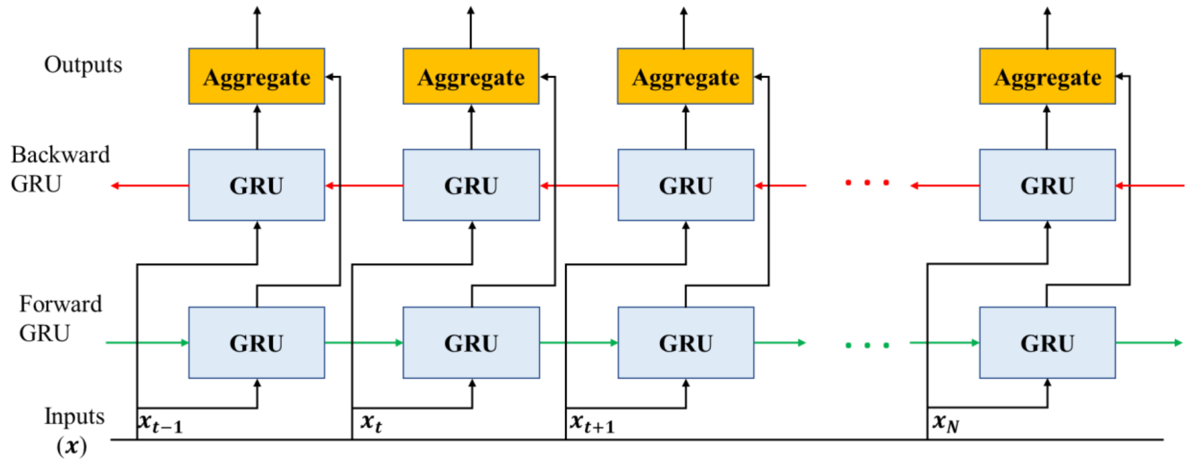
Architecture of BiLSTM Unit

The architecture of the BiLSTM unit (refer to *Figure 4*) is based on the LSTM unit architecture. The key difference between LSTM and BiLSTM units is the fact that, LSTM unit processes timeseries information in only one direction (forward direction) while the BiLSTM unit processes information in both forward and backwards directions. BiLSTM unit consists of two LSTM units: the Forward LSTM unit, whose role is

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

to process information in a forward direction, from the first timestep to the last timestep, and the Backwards LSTM unit, whose role is to process information in a backwards direction, from the last timestep to the first timestep. Finally, the outputs from the Forward LSTM unit and the Backwards LSTM unit are aggregated to produce a single output of the BiLSTM unit.

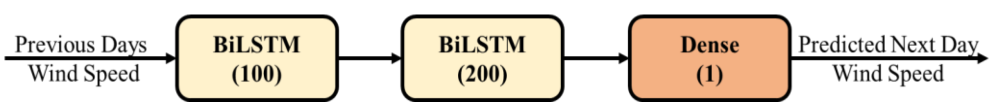
Figure 4: Architecture of BiLSTM Unit



Proposed BiLSTM Model

The proposed BiLSTM model is shown in *Figure 5*. The architecture of the proposed BiLSTM model consisted of two BiLSTM layers and one Dense layer. BiLSTM layer role is to learn the pattern of the input data (previous 5 days wind speed) and how to map it with the output (the next day (6th day) wind speed). The dense layer role is to output a single numerical value, which acts as the forecasted (predicted) wind speed. A total of 3 batch sizes, 16, 32 and 64, were chosen for this study. The selected batch sizes were alternately used during training of the BiLSTM model.

Figure 5: Proposed BiLSTM model



Loss Function and Performance Evaluation Metrics

Loss Function was used to assess the BiLSTM model's performance during training and check if the model is correctly learning how to map inputs to the output. It does this by measuring the error between the actual value y and the predicted value $\hat{y}$ and assess how accurately the model predicts the values which are close to the true (actual) values (Martin-Donas et al., 2018; Edalatifar et al., 2022). Mean Squared Error (MSE) (refer to equation (x)) is used as a Loss Function in this study. After being trained, the performance of the BiLSTM model needs to be evaluated (tested) on the test set (unseen data) in order to measure its ability to generalise on new data that it has never seen before. This study utilised three performance evaluation metrics. The first evaluation metric, Mean Absolute Error (MAE) (refer to equation (xi)), measured the mean absolute error between the actual value y and the predicted value $\hat{y}$, the lower the MAE score the better performance of the BiLSTM model (García et al., 2024; Hao et al., 2022). The second evaluation metric, Root Mean Squared Error (RMSE) (refer to equation (xii)), measured the root mean squared error between the actual value y and the predicted value $\hat{y}$, the lower the RMSE score, the better the

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

performance of the BiLSTM model (Zhao et al., 2020; Jiang et al., 2021). The third evaluation metric, R-Squared (R^2) (refer to equation (xiii), with $\bar{y}_i$ indicating the mean of actual values) measures the coefficient of determination between the actual value y and the predicted value $\hat{y}$, the higher the R^2 score, the better the performance of the BiLSTM model (Jiang et al., 2021; Michael et al., 2024).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (x)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (xi)$$

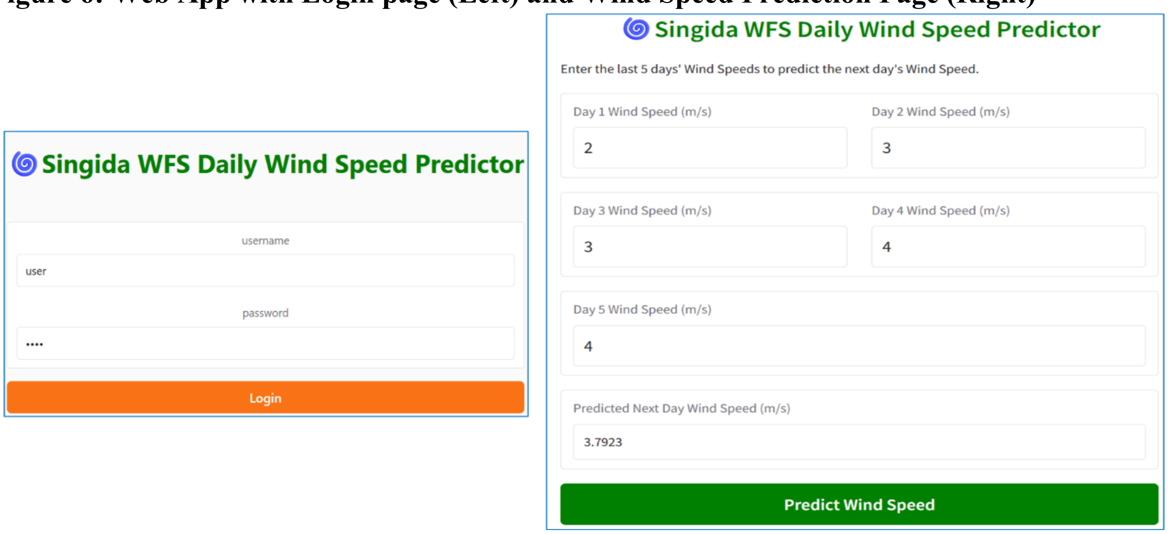
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (xii)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (xiii)$$

Web App

This study developed a Web App shown in *Figure 6* using the Gradio framework (Ferreira et al., 2024). Officials at Singida WFS can use Web browsers to access CSS (Cascaded Style Sheet) styled Web pages in the Web App, get authenticated, enter the 5 previous days' wind speed values and click the 'Predict' button to predict the next (6th) day wind speed. Afterwards, the Web App takes the entered parameters (previous 5 days wind speed values) and sends the entered parameters to the imported and pretrained BiLSTM model in .h5 format. The pretrained BiLSTM model takes the entered parameters as input, predicts the next day wind speed, and returns the predicted next day wind speed back to the Web App, which displays it back to the official.

Figure 6: Web App with Login page (Left) and Wind Speed Prediction Page (Right)



RESULTS

Computation Environment

The BiLSTM model was developed in IPython Notebook and all training and testing experiments of the BiLSTM model were conducted in Google Colab Cloud platform (Bisong, 2019) with the following allocation of runtime environment. System RAM of 12.7 GB and Hard Disk space of 107.7 GB. Various software libraries were used in the Google Colab environment, including TensorFlow, Keras, Scikit-Learn, Pandas, Numpy and MatplotLib.

Hyperparameters Tuning

Hyperparameters played a significant role in fine-tuning the Deep Learning models' performance during the training process. The BiLSTM Deep Learning model underwent several rounds of hyperparameter tuning, and finally, the following identical hyperparameters were selected for the BiLSTM model for each of the three batch sizes:

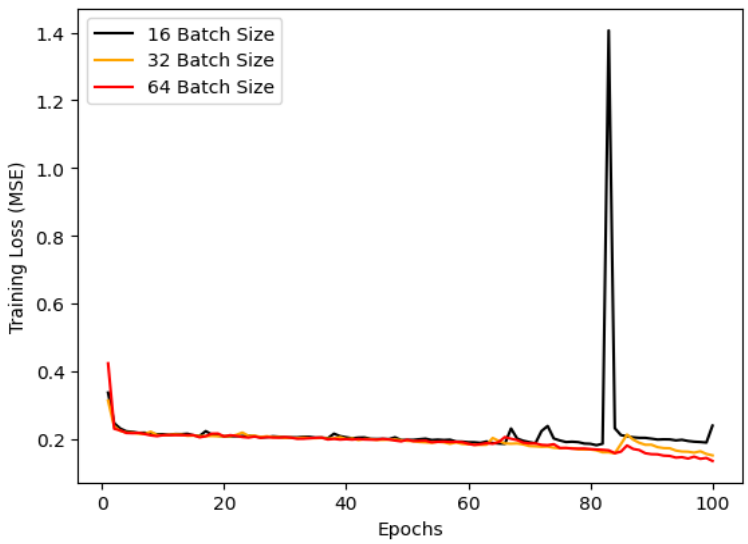
Article DOI: https://doi.org/10.37284/eaje.8.1.3441

two layers of BiLSTM units, output dimensionality of 100 for the first BiLSTM layer, output-dimensionality of 200 for the second BiLSTM layer, 1 unit in a Dense layer, learning rate of 0.01, Adam as an optimiser and 100 training epochs.

Training Results of BiLSTM Model

After finishing tuning the hyperparameters, a total of three experiments were conducted to train the BiLSTM model, using a different batch size in each experiment. After each training experiment, the trained BiLSTM model was saved in .h5 format for future inference in the performance testing (evaluation) phase. Therefore, there were three different saved versions of the trained BiLSTM model, one version for each batch size. *Figure 6* shows the training loss (MSE) of the BiLSTM model using three different batch sizes.

Figure 6: Training Loss (MSE) for BiLSTM Batch Sizes



Performance Evaluation Results of BiLSTM Model

Each version of the trained BiLSTM model was used to evaluate the performance of the corresponding batch size used in the BiLSTM model by evaluating the prediction performance of the BiLSTM model on a test set (the unseen data which have not been seen before by the BiLSTM model). Each version of the BiLSTM model was given test data as input and asked to forecast the wind speed. *Figure 7* shows actual wind speed against predicted wind speed at Singida WFS by three different versions of the BiLSTM model, with each version representing the BiLSTM model trained with a different batch size. *Table 2* shows performance evaluation results (Test MAE score, Test RMSE score, Test R^2 score and Training Time) for all three versions of the developed BiLSTM model. These findings revealed that, 16 was the best performing (optimal) batch size with MAE score of 0.58, RMSE score of 0.76 and R^2 score of 0.79, followed by a batch size of 32 (MAE score of 0.62, RMSE score of 0.79 and R^2 score of 0.75) and followed by batch size of 64 (MAE score of 0.66, RMSE score of 0.81 and R2 score of 0.72). These results implied that batch size had an impact on the performance of the BiLSTM model and that the lower the batch size, the better the performance of the BiLSTM model in predicting daily wind speed at Singida WFS. On the other hand, these results also revealed batch size had impact on computational efficiency of the BiLSTM model and that the lower the batch size the lower the computation efficiency (higher training time) with the results indicating batch sizes of 16, 32 and 64 required 16 minutes, 12 minutes and 7 minutes to train the BiLSTM model respectively.

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

Figure 7: Actual vs Predicted Wind Speed by BiLSTM Model Batch Sizes

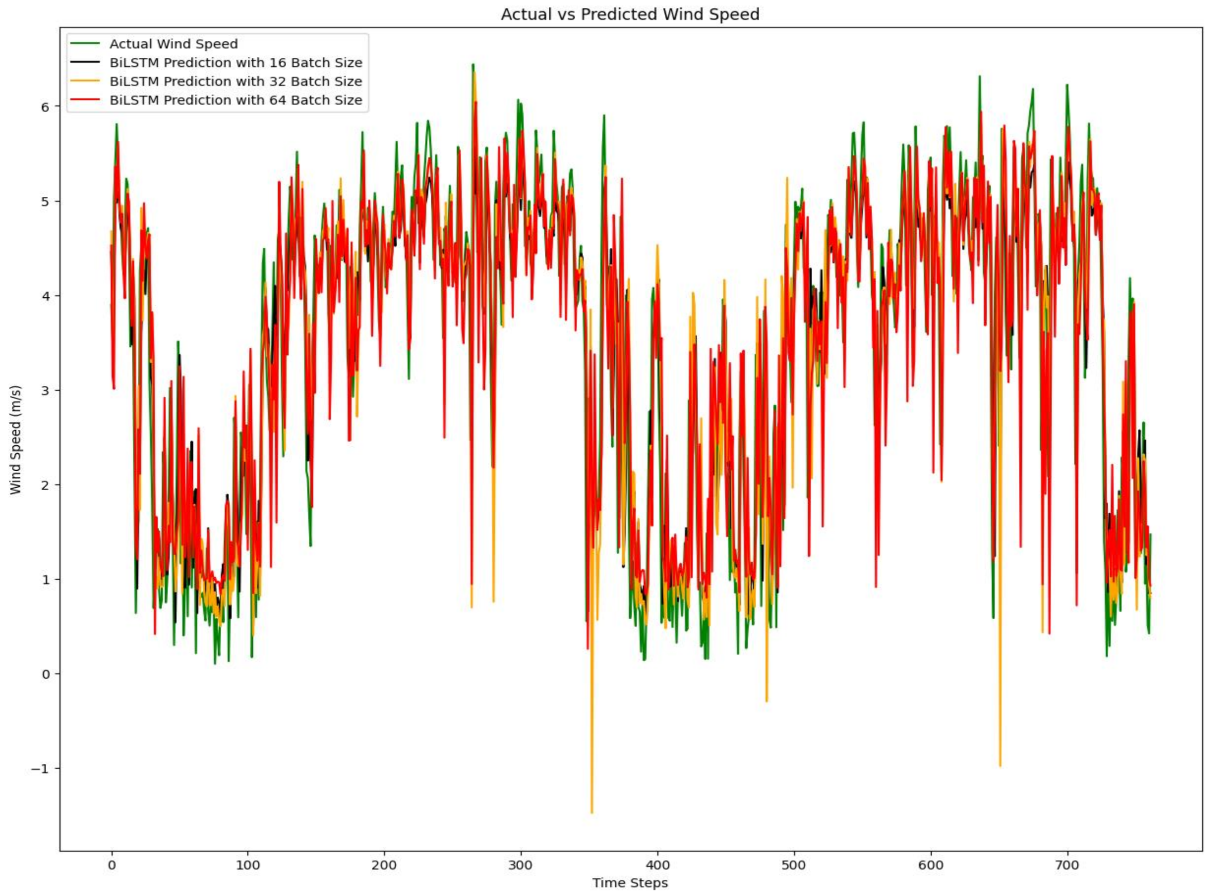


Table 2: Performance Evaluation Results of BiLSTM Model Batch Sizes

BiLSTM Batch Size	Test MAE	Test RMSE	Test R^2	Training Time (Minutes)
16	0.58	0.76	0.79	16
32	0.62	0.79	0.75	12
64	0.66	0.81	0.72	7

DISCUSSION

Difference in Prediction Performance between BiLSTM Model Batch Sizes

The findings reveal that a batch size of 16 achieved the best prediction performance (lowest test MAE and RMSE as well as highest R^2 scores), demonstrating its superior performance in learning and preserving wind speed data patterns in the BiLSTM model. The next best performing batch size was 32 with a slightly higher test MAE and RMSE scores, as well as a slightly lower test R^2 score. The last performing batch size was 64 with the highest test MAE and RMSE scores, as well as the lowest R^2 score. Best performance of 16 batch size can be attributed to the fact that it allows Deep Learning models to update model weights more frequently, which in turn helps to better capture local patterns and complex temporal dependencies in timeseries data (Zhang et al., 2024; Radiuk, 2017). The slightly lower performance of a 32 batch size can be attributed to the fact that the Deep Learning models still benefit from frequent updates, but now the gradient estimates are smoother (García et al., 2024; Méndez et al., 2023), possibly making Deep Learning models miss finer details in sequences of time series data. The lowest performance of 64 batch size can be attributed to the

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

fact that larger batch sizes produce very smooth gradient updates (Lu et al., 2021; Zhang et al., 2025), which can lead to slow learning and underfitting of Deep Learning models.

Difference in Computational Efficiency between BiLSTM Model Batch Sizes

During the training of the BiLSTM model, it was observed that batch size has a significant impact on the BiLSTM model training time. Smaller batch sizes (for instance, 16), while providing better prediction performance in terms of MAE, RMSE, and R^2 scores, required more training time per epoch. The reason behind this is that smaller batches result in more frequent model weight updates, which leads to slower convergence (Zhang et al., 2024). On the other hand, larger batch sizes (for instance, 64) reduced training time due to fewer model weight updates (Lu et al., 2021). These findings highlight a key trade-off between computational efficiency and predictive performance, which is important when deploying models in resource-constrained or time-sensitive environments. Despite having lowest computational efficiency compared to 32 and 64 batch sizes, this study prefers the highest prediction performance of 16 batch size over its lowest computational efficiency due to the fact that, after training the BiLSTM it is saved into .h5 format to ensure all future inferences in the Web App will use pretrained .h5 BiLSTM model without any model retraining. This reduces the impact of computational efficiency on the BiLSTM model. Also, 16 minutes required to train the BiLSTM model with a batch size of 16 is relatively attainable with free computational resources available in cloud platforms such as Google Colab.

Comparison with Findings from Literature

These findings suggest that, choice of batch size has an impact on the performance of BiLSTM models in prediction (forecasting) tasks of time series variables, and 16 is the optimal (best performing) batch size to use in BiLSTM models when predicting daily wind speed at Singida WFS, followed a 32 batch size, followed by 64 batch size. These findings align well with the findings from the literature, which also suggest that batch size has an impact on the prediction performance of Deep Learning models and that the best-performing batch size depends on the type of Deep Learning model used and the nature of the problem being addressed. This is evident in a study by Hao et al. (2022), which compared the performance of several batch sizes in predicting atmospheric temperature using a BiLSTM model and concluded that the optimal batch size was 256. Another evidence is found in a study by Hwang et al. (2024), which determined the optimal batch size after comparing the performance of several batch sizes of Deep Learning models in predicting traffic volume, revealing, the optimal batch size for the RNN model was 288 while the optimal batch size for the CNN model was 1440.

Practical Applications

This study provides a practical solution for daily wind speed prediction at Singida WFS and Singida Wind Power Station using a BiLSTM Deep Learning model, integrated with a user-friendly Gradio based Web App. The goal of this predictive model is to assist government officials, energy planners, and site engineers in making informed operational and strategic decisions by accurately forecasting wind speed and optimising wind power generation, properly scheduling turbine maintenance, ensuring grid stability and improving overall renewable energy planning in Tanzania. The Web App will enable users to input recent weather data and receive daily wind speed forecasts with ease. The Web App eliminates the need for technical expertise or access to complex tools to predict wind speed and thus can be adopted widely by energy agencies and policymakers. Furthermore, the Web App serves as a scalable framework that can be extended to other renewable energy sites with similar environmental and climatic conditions as Singida WFS and, as a result, contributes to a

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

data-driven approach for national energy planning and sustainability in Tanzania.

Theoretical Implications

These findings support well-established principles in Deep Learning, including the following:

- Stochastic Gradient Descent Behaviour: The smaller batch sizes approximate the behaviour of a pure stochastic gradient descent (SGD), which introduces higher variability in gradient updates. As a result, the optimiser can escape shallow local minima and better explore the optimisation landscape (Zhang et al., 2024). This is beneficial in complex models like BiLSTM.
- Tradeoff between Bias and Variance: The smaller batch sizes feed more noise into the learning process, as a result increasing the variance but lowering the bias. This can help Deep Learning models learn richer representations at the cost of slower convergence (Lin, 2022).
- Temporal Sensitivity: BiLSTM models capture forward and backwards dependencies in sequential timeseries data. Larger batch sizes might oversmooth gradients, making the BiLSTM model less sensitive to sudden changes or rare patterns in time series sequential data (García et al., 2024), which are critical in problems like wind speed prediction.
- Tradeoff between Generalisation and Computational Efficiency: Despite larger batch sizes being more computationally efficient, they may hinder model generalisation, especially in smaller or noisier datasets (Lu et al., 2021).

Major Contributions

This study has the following major contributions:

- Novel BiLSTM Model: In this study, a novel BiLSTM Deep Learning model has been developed, which uses the optimal 16 batch size to predict daily wind speed at Singida WFS. The developed BiLSTM model was trained and saved in .h5 format to facilitate future inference by the Gradio based Web App.
- Web App: This study developed a ready-to-use Gradio based Web App to help officials at Singida WFS and company officials at Singida Wind Power Station predict daily wind speed and take appropriate measures to efficiently manage the wind park.
- Pre-processed Dataset: This study pre-processed a 14-year Singida WFS daily wind speed dataset using several approaches, including data scaling, creation of input and output features and data splitting into train, validation, and test sets. Afterwards, the preprocessed dataset was saved in .pkl format, making it ready for importation and use by Deep Learning models. The preprocessed dataset will in future be shared on the GitHub cloud platform for use in AI research and development by anybody freely.
- Filling the Research Gap: The findings of this study will help to fill the existing research gap on the impact of batch sizes on the prediction performance of BiLSTM Deep Learning models, especially in the context of predicting daily wind speed at Singida WFS and environments resembling that of Singida WFS.

Study Limitations

Despite the findings of this study revealing strong effectiveness of the developed BiLSTM model in predicting wind speed at Singida WFS, there are several limitations of this study:

- Limited Geographic Area: This study is based on the Singida WFS and uses one dataset to train the BiLSTM model to predict wind speed. As a result, there is a limitation on the ability of the developed BiLSTM model to capture variable environmental and climatic conditions

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

across other geographic locations with different climates and environments.

- Operational Constraints Consideration: This study did not consider practical challenges and requirements of implementing the wind speed prediction system at Singida WFS, such as training of local staff or integration with existing wind management workflows and protocols. As a result, the actual deployment of the developed Web App might be impacted.

CONCLUSION

The study developed a BiLSTM Deep Learning model for predicting daily wind speed at Singida WFS and evaluated its prediction performance when implemented with three different batch sizes of training data. The findings reveal that a batch size of 16 is the best performing (optimal) batch size, achieving the lowest test MAE and RMSE scores and the highest R^2 score, followed by a batch size of 32 and followed by a batch size of 64. These results suggest choice of batch size has a direct impact on the performance of the BiLSTM Deep Learning model in predicting wind speed at Singida WFS and similar environments, with the results indicating that the lower the batch size, the better the prediction performance.

Recommendations

This study recommends a batch size of 16 as the optimal and practical batch size to use in BiLSTM Deep Learning models for predicting daily wind speed at Singida WFS and environments with similar climatic conditions.

Acknowledgements

The author would like to give thanks to the Dar es Salaam Institute of Technology for supporting this research.

REFERENCES

Abdulnabi, N. Z. T., & Altun, O. (2016). Batch size for training convolutional neural networks for sentence classification. Journal of Advances in Technology and Engineering Studies, 2(5), 156-163.

Bai, X., Zhang, L., Feng, Y., Yan, H., & Mi, Q. (2025). Multivariate temperature prediction model based on CNN-BiLSTM and RandomForest. *The Journal of Supercomputing*, 81(1), 162. DOI: https://doi.org/10.1007/s11227-024-06689-3

Bisong, E. (2019). Google Colaboratory. In Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners. Berkeley, CA: Apress.

Doğan, E. (2021). LSTM training set analysis and clustering model development for short-term traffic flow prediction. Neural Computing and Applications, 33(17), 11175-11188. DOI: https://doi.org/10.1007/s00521-020-05564-5

Edalatifar, M., Ghalambaz, M., Tavakoli, M. B., & Setoudeh, F. (2022). New loss functions to improve deep learning estimation of heat transfer. Neural Computing and Applications, 34(18), 15889-15906. DOI: https://doi.org/10.1007/s00521-022-07233-1

Ferreira, R., Canesche, M., Jamieson, P., Neto, O., & Nacif, J. A. (2024). Examples and tutorials on using Google Colab and Gradio to create online interactive student-learning modules. Computer Applications in Engineering Education, 32(4), e22729. DOI: https://doi.org/10.1002/cae.22729

Filik, Ü. B., & Filik, T. (2017). Wind speed prediction using artificial neural networks based on multiple local measurements in Eskisehir. *Energy Procedia*, 107, 264-269. DOI: https://doi.org/10.1016/j.egypro.2016.12.147

García, F., Guijarro, F., Oliver, J., & Tamošiūnienė, R. (2024). Foreign Exchange Forecasting Models: LSTM and BiLSTM Comparison.

Engineering Proceedings, 68(1), 19. DOI: https://doi.org/10.3390/engproc2024068019

Halpern-Wight, N., Konstantinou, M., Charalambides, A. G., & Reinders, A. (2020). Training and Testing of a Single-Layer LSTM Network for Near-Future Solar Forecasting. Applied Sciences, 10(17), 5873. DOI: https://doi.org/10.3390/app10175873

Hao, X., Liu, Y., Pei, L., Li, W., & Du, Y. (2022). Atmospheric Temperature Prediction Based on a BiLSTM-Attention Model. *Symmetry*, 14(11), 2470. DOI: https://doi.org/10.3390/sym14112470

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. Neural Comput.,9, 1735–1780

Huntington, J., Hegewisch, K., Daudert, B., Morton, C., Abatzoglou, J., McEvoy, D., & Erickson, T. (2017). Climate engine: Cloud computing and visualization of climate and remote sensing data for advanced natural resource monitoring and process understanding. Bulletin of the American Meteorological Society, 98(11), 2397-2410

Hwang, J.-S., Lee, S.-S., Gil, J.-W., & Lee, C.-K. (2024). Determination of Optimal Batch Size of Deep Learning Models with Time Series Data. *Sustainability*, 16(14), 5936. DOI: https://doi.org/10.3390/su16145936

Jiang, X., Wei, P., Luo, Y., & Li, Y. (2021). Air Pollutant Concentration Prediction Based on a CEEMDAN-FE-BiLSTM Model. Atmosphere, 12(11), 1452. DOI: https://doi.org/10.3390/atmos12111452

Jung, C. (2024). Recent Development and Future Perspective of Wind Power Generation. *Energies*, 17(21), 5391. DOI: https://doi.org/10.3390/en17215391

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT Express, 6(4), 312-315. DOI: https://doi.org/10.1016/j.icte.2020.04.010

Kumwenda, M. J., & Mangara, R. J. (2024). Assessment of Wind Speed Characteristics and Available Wind Power Potential for Electricity Generation in Tanzania. Tanzania *Journal of Engineering and Technology*, 43(4), 66-76. DOI: https://doi.org/10.52339/tjet.v43i4.1047

Lin, R. (2022, February). Analysis on the selection of the appropriate batch size in CNN neural network. In 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE) (pp. 106-109). IEEE. DOI: https://doi.org/10.1109/MLKE55170.2022.00026

Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, 33(10), 4741-4753. DOI: https://doi.org/10.1007/s00521-020-05532-z

Mangara, R. J., & Kumwenda, M. J. (2023). Wind Speed Forecasting Using Wavelet Analysis and Recurrent Artificial Neural Networks Based on Local Measurements in Singida Region, Tanzania. *Tanzania Journal of Science*, 49(3), 754-763. DOI: https://dx.doi.org/10.4314/tjs.v49i3.17

Martin-Donas, J. M., Gomez, A. M., Gonzalez, J. A., & Peinado, A. M. (2018). A deep learning loss function based on the perceptual evaluation of the speech quality. IEEE Signal Processing Letters, 25(11), 1680-1684. DOI: https://doi.org/ 10.1109/LSP.2018.2871419

Méndez, M., Merayo, M. G., & Núñez, M. (2023). Long-term traffic flow forecasting using a hybrid CNN-BiLSTM model. *Engineering Applications of Artificial Intelligence*, 121, 106041. DOI:

Article DOI: https://doi.org/10.37284/eaje.8.1.3441

https://doi.org/10.1016/j.engappai.2023.106041

Meng, X., Liu, X., Duan, H., Hu, Z., & Wang, M. (2025). Optimization of Oil Well Production Prediction Model Based on Inter-Attention and BiLSTM. *Electronics*, 14(5), 1004. DOI: https://doi.org/10.3390/electronics14051004

Michael, N. E., Bansal, R. C., Ismail, A. A. A., Elnady, A., & Hasan, S. (2024). A cohesive structure of Bi-directional long-short-term memory (BiLSTM)-GRU for predicting hourly solar radiation. Renewable Energy, 222, 119943. DOI: https://doi.org/10.1016/j.renene.2024.119943

Oyedotun, O. K., Papadopoulos, K., & Aouada, D. (2023). A new perspective for understanding generalization gap of deep neural networks trained with large batch sizes. Applied Intelligence, 53(12), 15621-15637. DOI: https://doi.org/10.1007/s10489-022-04230-8

Radiuk, P. (2017). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. Information Technology and Management Science, 20(1), 20-24.

Skibko, Z., Hołdyński, G., & Borusiewicz, A. (2022). Impact of Wind Power Plant Operation on Voltage Quality Parameters—Example from Poland. *Energies*, 15(15), 5573. DOI: https://doi.org/10.3390/en15155573
Tamiminia, H., Salehi, B., Mahdianpari, M., Quackenbush, L., Adeli, S., & Brisco, B. (2020). Google Earth Engine for geo-big data applications: A meta-analysis and systematic review. ISPRS Journal of Photogrammetry and Remote Sensing, 164, 152-170

Turi, J. A., Rosak-Szyrocka, J., Mansoor, M., Asif, H., Nazir, A., & Balsalobre-Lorente, D. (2022). Assessing Wind Energy Projects Potential in Pakistan: Challenges and Way Forward. Energies, 15(23), 9014. DOI: https://doi.org/10.3390/en15239014

Wang, L., Li, T., Wang, P., Liu, Z., & Zhang, Q. (2023). BiLSTM for Predicting Post-Construction Subsoil Settlement under Embankment: Advancing Sustainable Infrastructure. *Sustainability*, 15(20), 14708. DOI: https://doi.org/10.3390/su152014708

Zhang, J., Yin, M., Wang, P., & Gao, Z. (2024). A Method Based on Deep Learning for Severe Convective Weather Forecast: CNN-BiLSTM-AM (Version 1.0). *Atmosphere*, 15(10), 1229. DOI: https://doi.org/10.3390/atmos15101229

Zhang, Y., Zhang, T., & Hu, J. (2025). Forecasting Stock Market Volatility Using CNN-BiLSTM-Attention Model with Mixed-Frequency Data. Mathematics, 13(11), 1889. DOI: https://doi.org/10.3390/math13111889

Zhao, C., Huang, X., Li, Y., & Yousaf Iqbal, M. (2020). A Double-Channel Hybrid Deep Neural Network Based on CNN and BiLSTM for Remaining Useful Life Prediction. Sensors, 20(24), 7109. DOI: https://doi.org/10.3390/s20247109