Journal of Electronics and Electrical Engineering
https://ojs.wiserpub.com/index.php/JEEE

UNIVERSAL WISER PUBLISHER

Research Article

Enhanced Wind Speed Prediction Using Dual-Memory LSTM: A Novel Approach to Temporal Dynamics

Francisca Asare-Bediako[1*], Princess Elorm Sepah[2], Kobena Badu Enyam[2], George Yaw Obeng[1]

[1] Department of Mechanical Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
[2] Department of Mechanical Engineering, Ashesi University, Berekusu, Ghana
E-mail: fasarebediako3@st.knust.edu.gh

Received: 10 January 2025; **Revised:** 3 March 2025; **Accepted:** 17 March 2025

Abstract: Accurate wind speed forecasting is crucial for optimizing wind energy integration, ensuring grid stability, and advancing renewable energy systems. This study introduces the Dual-Memory Long Short-Term Memory (DMLSTM) model, an innovative extension of conventional LSTM architectures designed to model short-term and long-term temporal dependencies explicitly. By incorporating separate memory cells and dynamic gating mechanisms, the DMLSTM addresses the limitations of traditional models such as AutoRegressive Integrated Moving Average (ARIMA), Artificial Neural Networks (ANN), and baseline LSTM in capturing the non-linear, stochastic, and hierarchical patterns inherent in wind speed data. The research utilized a comprehensive meteorological dataset from Tetouan City, comprising wind speed, temperature and humidity. Preprocessing steps ensured data quality and consistency, including data normalization and outlier handling. The DMLSTM was trained using a sliding window approach, mapping sequences of historical data to predict future wind speeds, and evaluated against ANN, ARIMA, and baseline LSTM models using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Experimental results demonstrate the superior performance of the DMLSTM, achieving the lowest RMSE and MAE values across all tested models. Visual comparisons of predicted and actual wind speeds reveal the DMLSTM's robustness in capturing abrupt changes and complex temporal dynamics. The findings underscore the model's potential as a reliable tool for wind speed forecasting, paving the way for more effective integration of wind energy into sustainable energy systems. This study highlights the significance of advanced temporal modeling techniques in addressing the challenges of renewable energy forecasting.

***Keywords*:** dual-memory LSTM, wind speed prediction, temporal dynamics, deep learning, renewable energy integration

1. Introduction

Renewable energy continues to be at the forefront of sustainable development, with wind energy emerging as a viable alternative to mitigate climate change, reduce greenhouse gas emissions, and address energy poverty. Globally, the integration of wind energy into power grids has proven successful in countries such as Denmark, China, and the United States. However, Africa remains underrepresented in wind energy production despite its vast untapped wind resources. Studies have demonstrated the immense potential for wind energy development across the continent, with significant opportunities in coastal and highland regions [1]. As Africa strives to meet its growing energy demands sustainably, wind energy could play a pivotal role in fostering energy security and economic growth.

Copyright ©2025 Francisca Asare-Bediako, et al.
DOI: https://doi.org/10.37256/jeee.4120256391
This is an open-access article distributed under a CC BY license
(Creative Commons Attribution 4.0 International License)
https://creativecommons.org/licenses/by/4.0/

Wind speed forecasting is critical to successfully integrating wind energy, as it ensures efficient wind turbine operation, optimal energy dispatch, and grid stability. Various algorithms have been employed for wind speed forecasting, ranging from statistical methods to advanced machine learning techniques. Traditional approaches such as AutoRegressive Integrated Moving Average (ARIMA) and Seasonal AutoRegressive Integrated Moving Average (SARIMA) have shown reliability in short-term wind speed forecasting [2]. However, these methods often fail to capture nonlinear and dynamic patterns in wind speed data. Recent advancements have focused on developing more sophisticated machine learning models to address non-stationarity and nonlinearity in wind speed data modeling. [3] introduced an optimized HHT-NAR model, which integrates Complementary Ensemble Empirical Mode Decomposition (CEEMD), Hilbert-Huang Transform (HHT), and Nonlinear Autoregressive Neural Networks (NAR). While this approach demonstrated significant improvements, including a reduction in RMSE to 2.5375 m/s, its limitations lie in the lack of explicit modeling for short and long-term dependencies, scalability for long-term forecasting, and the interpretability of empirically optimized weights. [4] introduced the Conditional Local Convolution Recurrent Network (CLCRN) which integrates Conditional Local Convolution (CLC) kernels with Gated Recurrent Unit (GRU)-based Recurrent Neural Networks (RNNs) to address irregular spatial data distribution and capture temporal dependencies. The model outperformed existing frameworks such as Diffusion Convolutional Recurrent Neural Network (DCRNN), Temporal Graph Convolutional Network (TGCN), and Multi-Scale Temporal Graph Convolutional Network (MSTGCN), achieving lower Mean Average Error (MAE) and Root Mean Square Error (RMSE) values across prediction intervals by utilizing multidimensional meteorological inputs. While the CLCRN demonstrated significant improvements, it lacks an explicit mechanism for differentiating short-term and long-term dependencies, a critical factor for modeling complex temporal patterns in wind energy forecasting. Artificial Neural Networks (ANNs) have demonstrated significant potential in this domain, [5] introduced algorithms like Levenberg-Marquardt (LM) and Bayesian Regularization (BR) achieving superior accuracy by effectively modeling non-linear environmental variables. However, traditional ANN models face challenges in capturing complex temporal dependencies and scalability in large-scale networks.

Long Short-Term Memory (LSTM) networks have emerged as a powerful tool for time-series forecasting, addressing critical challenges in wind speed forecasting by effectively capturing both short-term and long-term dependencies through their memory cell architecture and outperforming traditional methods like ARIMA and SARIMA [6], [7]. Unlike models such as HHT-NAR and CLCRN, which lack explicit mechanisms for modeling intricate temporal dependencies, LSTMs dynamically learn nonlinear relationships and hierarchical temporal patterns, making them ideal for non-stationary and noisy data. This adaptability and robustness make LSTMs a superior choice for long-term wind speed forecasting, enhancing grid integration and energy planning [8]. Some authors have successfully applied LSTM networks to wind speed forecasting, demonstrating their effectiveness in capturing temporal dependencies and achieving superior predictive accuracy. [9] introduced an advanced model which integrates Genetic Algorithm (GA) and Dipper Throated Optimization (DTO) to optimize Bidirectional LSTM parameters, achieving a remarkably low RMSE of 0.00046. This approach demonstrated the importance of robust feature selection and parameter optimization in improving prediction outcomes using LSTM. [10] demonstrates that LSTM models significantly outperform traditional power law methods, particularly at higher altitudes and in complex terrains. By incorporating additional meteorological variables such as turbulence kinetic energy (TKE), the LSTM achieved improved performance metrics, including reduced RMSE values. Similarly, a hybrid Density-Based Spatial Clustering of Applications with Noise-Stacked Denoising Autoencoder-LSTM (DBSCAN-SDAE-LSTM) model was proposed by [11], which combines clustering techniques, deep feature extraction, and LSTM for time-series forecasting. While it achieved substantial accuracy gains over standard models like Backpropagation Neural Network (BPNN), it struggled with modeling multi-scale temporal patterns. [12] also introduced a hybrid model that combines secondary decomposition techniques, such as Wavelet Decomposition (WA) and Symplectic Geometry Mode Decomposition (SGMD), with Long Short-Term Memory (LSTM) networks to enhance wind speed forecasting. Their approach transforms non-stationary wind speed data into more regular components, allowing LSTM to achieve superior accuracy, reducing MAPE, RMSE, and MAE by 37%, 13%, and 17%, respectively, over single-step and multi-step forecasts. [13] introduced a combined approach using graph convolutional networks (GCNs), long short-term memory (LSTM) networks, and an attention mechanism however, the performance of the proposed approach is highly dependent on the quality of the constructed graph. Inaccuracies in defining spatial relationships for example: suboptimal selection of route segments or imprecise adjacency matrices can propagate through the network and affect prediction accuracy. While hybrid models like

ARIMA-LSTM, which combine ARIMA for linear trends with LSTM for nonlinear dynamics, showed some promise, their inability to capture complex wind speed variations, particularly in noisy and irregular datasets, highlights the need for more sophisticated approaches, such as those explicitly addressing short and long-term temporal patterns and stabilizing the training process by ensuring that gradients remain well-scaled, thus preventing vanishing or exploding gradients; a limitation in many deep recurrent models [14]. To address the issue of short and long-term temporal patterns, [15] proposed an innovative framework that integrates a dual-memory LSTM architecture with an incremental update mechanism and a zero-shot learning component, although the proposed model employs a dual-memory structure, it does not clearly define how the short-term and long-term memory components interact or integrate with one another and also the framework relies on a complex Expectation-Maximization style incremental update process which introduces additional computational overhead and sensitivity to hyperparameters. [16] introduced also a dual memory LSTM, which leverages differencing operations between consecutive images (or hidden states) to capture motion information. Although differencing operations help capture motion, they may also amplify noise if consecutive frames are noisy.

This study proposes the Dual-Memory LSTM (DMLSTM) model, which explicitly separates short-term and long-term temporal dependencies into distinct memory cells to address the limitations identified in existing methods. By incorporating short-term dependencies and long-term dependencies, the DMLSTM dynamically balances these contributions through a learnable combination. This unique design ensures a more nuanced representation of hierarchical temporal patterns compared to conventional LSTMs, which lack explicit differentiation of memory scales. Unlike hybrid models such as ARIMA-LSTM, which combine linear and nonlinear dynamics but struggle with noisy and irregular data, the DMLSTM introduces robust gating dynamics and layer normalization to enhance training stability and performance. The model's adaptability makes it well-suited for capturing intricate wind speed patterns, addressing the challenges of non-stationarity, nonlinearity, and multi-scale temporal dependencies. By leveraging these features, the DMLSTM provides a comprehensive framework to improve the accuracy and scalability of wind speed forecasting, bridging the gaps left by traditional and hybrid models, and facilitating better integration of wind energy into power systems.

The paper is organized into the following sections: Section 2 explores the baseline Long Short-Term Memory (LSTM) model, explaining how it works and pointing out its limitations, particularly in capturing complex temporal dependencies. To address these challenges, Section 3 introduces the novel Dual-Memory LSTM (DMLSTM), which improves upon the baseline by using separate memory cells to better handle both short-term and long-term patterns in data. Section 4 outlines the methodology used in this research, detailing how the data was prepared, how the models were implemented, and the experiments conducted to evaluate their performance. Section 5 delves into the results, comparing the baseline LSTM, DMLSTM, and other models in terms of accuracy and error metrics. It also discusses the training and validation loss curves to provide deeper insights into each model's behavior. Lastly, Section 6 concludes by summarizing the study's findings, highlighting the advantages of the DMLSTM for wind speed forecasting, and suggesting areas for future research to further refine forecasting techniques.

1.1 *Contribution to knowledge*

The following points summarize how our proposed architecture extends beyond the current state-of-art proposed LSTM models in current literature:

Explicit Separation of Short-Term and Long-Term Memories

- Traditional LSTM architectures rely on a single memory cell to capture both short- and long-term dependencies. This uniform update mechanism can cause the model to overemphasize one time scale at the expense of the other.
- By introducing two distinct cell states-one dedicated to capturing transient features (short-term memory) and another for more persistent signals (long-term memory)-the DMLSTM provides a clearer, specialized framework for modeling hierarchically evolving temporal dynamics.

Dedicated Gating for Each Memory Timescale

- The DMLSTM does not merely duplicate the standard LSTM cell. Instead, each cell type (short-term and long-term) maintains its own set of gates (forget, input, and output), enabling the model to learn individualized gating behaviors.

• This design allows the short-term cell to respond aggressively to rapid fluctuations, while the long-term cell tracks more stable trends-an approach that is not achieved by simply increasing the capacity (e.g., neuron count) of a single LSTM cell.

Dynamic Combination Gate

• The proposed architecture introduces a learnable combination gate α_t to blend the short-term and long-term hidden states adaptively. This mechanism augments the typical output gate strategy by allowing the model to determine on-the-fly how much weight should be given to short vs. long-term information at each timestep.

• As a result, the DMLSTM can pivot between focusing on recent, rapidly changing events and relying on long-standing trends depending on the data context, which standard LSTM architecture does not explicitly accommodate.

Layer Normalization and Residual Connections in a Dual-Memory Context

• While layer normalization and residual connections have appeared in other architectures, incorporating them into a dual-memory framework provides unique benefits for stabilizing training and improving gradient flow across multiple memory pathways.

• In a single LSTM cell, residual connections and normalization operate on a single hidden state and cell state. In the DMLSTM, they help maintain balanced updates across two parallel cell states, substantially reducing the risk of exploding or vanishing gradients when modeling complex, noisy data such as wind speeds.

Empirical Demonstration of Efficacy

• Finally, the paper demonstrates that this dual-memory design yields significant performance gains over single-cell LSTM, ARIMA, and ANN models when forecasting wind speed. This empirical evidence underscores that the multi-scale memory separation rather than just an increase in model size; addresses a genuine gap in wind energy forecasting, where short-term fluctuations and long-term trends are both critical.

2. Baseline LSTM model

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) explicitly designed to overcome the vanishing gradient problem faced by traditional RNNs [9]. LSTMs achieve this by incorporating a unique gating mechanism that allows them to capture long-term dependencies in sequential data [17], [18]. Below is a detailed theoretical framework for the conventional LSTM, which serves as the baseline to demonstrate the advancements introduced by the Dual-Memory LSTM (DMLSTM).

2.1 *Structure and mechanism of LSTM*

At the core of the LSTM architecture is a memory cell (C_t) and three gating mechanisms: forget gate, input gate, and output gate. These gates regulate the flow of information through time, enabling the model to maintain, update, or discard information as shown in Figure 1.

2.1.1 *Mathematical representation*

1. Forget Gate (f_t) : The forget gate decides which parts of the previous cell state (C_{t-1}) to retain or forget. It uses a sigmoid activation function to output a value between 0 and 1, where 1 represents full retention, and 0 represents complete forgetting.

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

where:

• W_f : Weight matrix for the forget gate.

• b_f : Bias for the forget gate.

- $[h_{t-1}, x_t]$: Concatenation of the previous hidden state and the current input.
- σ : Sigmoid activation function.

2. Input Gate (i_t) and Candidate Cell State $\left(\widetilde{C}_t\right)$: The input gate determines how much new information should be added to the cell state. The candidate cell state ($\widetilde{C}_t$) generates potential updates using a tanh activation function.

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

$$\widetilde{C}_t = \tan h\left(W_C\left[h_{t-1}, x_t\right] + b_C\right) \tag{3}$$

- W_i, b_i : Weight matrix and bias for the input gate.
- W_C, b_C : Weight matrix and bias for the candidate cell state.

3. Cell State Update (C_t) : The cell state is updated by combining the retained previous state (scaled by f_t) and the new candidate state (scaled by i_t).

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{4}$$

- $\odot$: Element-wise multiplication.

4. Output Gate (o_t) and Hidden State (h_t) : The output gate determines which parts of the cell state contribute to the hidden state. The hidden state is computed by scaling the activated cell state with the output gate.

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

$$h_t = o_t \odot \tan h\left(C_t\right) \tag{6}$$

- W_o, b_o : Weight matrix and bias for the output gate.

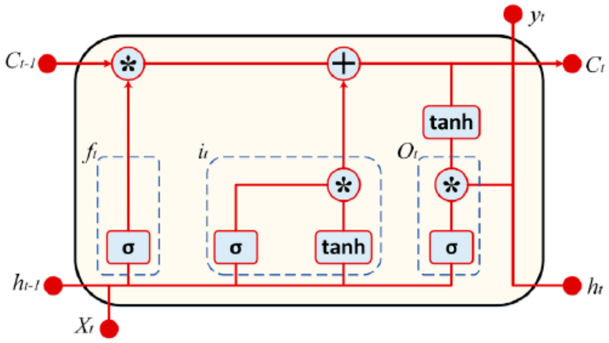

Figure 1. LSTM schematic diagram

2.2 *Limitations of conventional LSTM*

While the LSTM architecture is powerful in modeling sequential data, it has inherent limitations:

- Single Memory Cell: LSTMs use a single-cell state (C_t) to model all temporal dependencies, which can lead to inefficiencies in distinguishing between short-term and long-term dependencies.
- Uniform Update Mechanism: The gating mechanism applies uniform updates to the cell state, potentially leading to an overemphasis on either short-term or long-term patterns, depending on the data.
- Inflexibility in Complex Temporal Dynamics: Conventional LSTMs lack the explicit mechanisms to adaptively blend information across varying temporal scales.

3. A novel dual-memory LSTM model (DMLSTM)

This section introduces a novel variant of the Long Short-Term Memory (LSTM) architecture that explicitly separates memory into two distinct cells: one dedicated to short-term dependencies, denoted as $C_t^{(s)}$ and another to long-term dependencies represented as $C_t^{(l)}$. The final hidden state h_t, is determined by a learned combination of these two memories, facilitating richer gating dynamics and more robust temporal modeling. Figure 2 provides a high-level flowchart of this proposed Dual-Memory LSTM (DMLSTM).

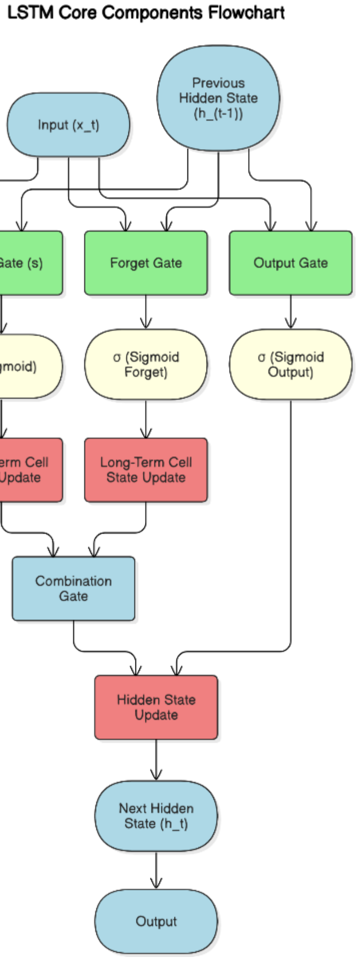

Figure 2. A higher-level overview of the novel dual memory LSTM

3.1 *Model overview*

A key motivation for the Dual-Memory LSTM (DMLSTM) design is recognizing that certain features in temporal sequences evolve on different timescales. By maintaining two separate cell states-one for short-term ($C_t^{(s)}$) and another for long-term ($C_t^{(l)}$) dependencies-the model can retain more stable, long-lived features in the long-term cell state, while capturing rapidly changing, transient details in the short-term cell state. This explicit separation encourages specialized gating behaviors, allowing each memory component to focus on the patterns most relevant to its timescale. Consequently, the DMLSTM achieves greater flexibility in how information is stored, forgotten, or combined, improving its overall ability to capture complex temporal dynamics.

3.2 *Parameterization*

Each timescale (short-term and long-term) introduces its own set of LSTM-like parameters, ensuring that each cell type can independently learn gating behaviors tailored to its temporal focus:

- Forget gates: $W_f^{(s)}, b_f^{(s)}$ and $W_f^{(l)}, b_f^{(l)}$
- Input gates: $W_i^{(s)}, b_i^{(s)}$ and $W_i^{(l)}, b_i^{(l)}$
- Cell candidate transforms: $W_C^{(s)}, b_C^{(s)}$ and $W_C^{(l)}, b_C^{(l)}$
- Output gates: $W_o^{(s)}, b_o^{(s)}$ and $W_o^{(l)}, b_o^{(l)}$
- Combination gate α_t with parameters W_α, b_α

By distinctly modeling the short-term and long-term pathways, the DMLSTM can dynamically merge these parallel streams of information at each timestep.

Step by step computation:

Below are the core equations illustrating how short-term and long-term memories are updated and ultimately fused into a single hidden state h_t.

1. Gating for Short-Term Memory

$$f_t = \sigma\left(W_f^{(s)}\left[h_{t-1}, x_t\right] + b_f^{(s)}\right) \text{ (short-term forget gate)} \tag{7}$$

$$i_t = \sigma\left(W_i^{(s)}\left[h_{t-1}, x_t\right] + b_i^{(s)}\right) \text{ (short-term input gate)} \tag{8}$$

$$\widetilde{C}_t^{(s)} = \tan h\left(W_C^{(s)}\left[h_{t-1}, x_t\right] + b_C^{(s)}\right) \text{ (short-term candidate)} \tag{9}$$

These gates decide how much of the previous short-term memory to forget (f_t) and how much new information to add (i_t). The candidate cell state $\widetilde{C}_t^{(s)}$ provides potential updates.

2. Update Short-Term Cell State

$$C_t^{(s)} = f_t \odot C_{t-1}^{(s)} + i_t \odot \widetilde{C}_t^{(s)} \tag{10}$$

Short-term memory $C_t^{(s)}$ is refreshed based on the forget gate scaling of $C_{t-1}^{(s)}$ and the input gate scaling of $\widetilde{C}_t^{(s)}$.

3. Gating for Long-Term Memory

$$f_t^{(l)} = \sigma\left(W_f^{(l)}\left[h_{t-1}, x_t\right] + b_f^{(l)}\right) \text{ (long-term forget gate)} \tag{11}$$

$$i_t^{(l)} = \sigma\left(W_i^{(l)}\left[h_{t-1}, x_t\right] + b_i^{(l)}\right) \text{ (long-term input gate)} \tag{12}$$

$$\widetilde{C}_t^{(l)} = \tan h\left(W_C^{(l)}\left[h_{t-1}, x_t\right] + b_C^{(l)}\right) \text{ (long-term candidate)} \tag{13}$$

Similarly, the long-term memory has its own forget $\left(f_t^{(l)}\right)$ and input $\left(i_t^{(l)}\right)$ gates, along with a candidate cell state $\widetilde{C}_t^{(l)}$.

4. Update Long-Term Cell State

$$C_t^{(l)} = f_t^{(l)} \odot C_{t-1}^{(l)} + i_t^{(l)} \odot \widetilde{C}_t^{(l)} \tag{14}$$

The long-term memory $C_t^{(l)}$ is refreshed in a parallel but independent manner, focusing on more stable patterns that persist over extended sequences.

5. Output Gating for Each Memory Scale

$$o_t^{(s)} = \sigma\left(W_o^{(s)}\left[h_{t-1}, x_t, C_t^{(s)}\right] + b_o^{(s)}\right) \tag{15}$$

$$o_t^{(l)} = \sigma\left(W_o^{(l)}\left[h_{t-1}, x_t, C_t^{(l)}\right] + b_o^{(l)}\right) \tag{16}$$

These output gates control how much short-term vs. long-term cell information is passed forward for final combination.

6. Combination Gate α_t

$$\alpha_t = \sigma\left(W_\alpha\left[h_{t-1}, x_t\right] + b_\alpha\right) \tag{17}$$

α_t adaptively determines the relative weighting between the short-term and long-term contributions.

7. Final Hidden State

$$h_t = \alpha_t\left(\tan h\left(C_t^{(s)}\right) \odot o_t^{(s)}\right) + (1-\alpha_t)\left(\tan h\left(C_t^{(l)}\right) \odot o_t^{(l)}\right) \tag{18}$$

By blending the short-term and long-term memory outputs via α_t, the final hidden state h_t captures both rapid variations and persistent trends in the data.

Intuitive Flow of Information

1. Short-Term Processing

● Each timestep, the model decides how much recent information to keep (via f_t) and how much new input to incorporate (via i_t). The short-term memory $C_t^{(s)}$ thus remains agile, tracking fluctuations that occur within a shorter horizon.

2. Long-Term Processing

● In parallel, the long-term memory $C_t^{(l)}$ accumulates and updates more stable information, filtering out transient noise. This design ensures the model does not overwrite critical historical patterns.

3. Dynamic Blending

● At each timestep, the combination gate α_t weighs the short-term vs. long-term outputs. If the data indicates a rapid shift (e.g., a sudden spike in wind speed), α_t may favor the short-term memory. Conversely, in more stable periods, the model might lean on the long-term memory.

Through this specialized gating structure, the DMLSTM provides richer temporal modeling than a single-cell LSTM, offering enhanced adaptability for time-series tasks-particularly those involving hierarchical or multi-scale dynamics such as wind speed forecasting. Figure 2 provides a high level of the proposed Dual Memory LSTM approach and Figure 3 illustrates the flow of computations in the DMLSTM, highlighting the parallel paths for short- and long-term updates and their fusion via the combination gate.

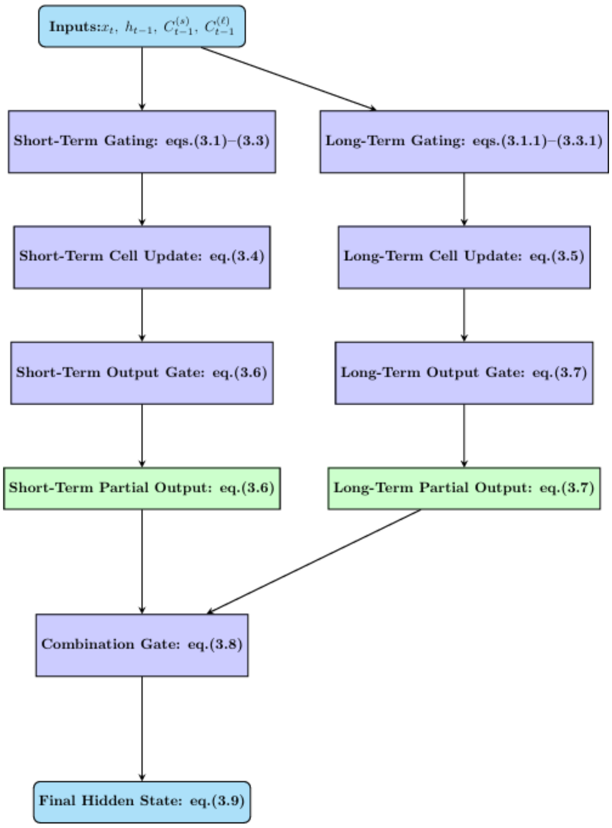
Inputs:x_t, h_{t-1}, $C_{t-1}^{(s)}$, $C_{t-1}^{(\ell)}$
Short-Term Gating: eqs.(3.1)–(3.3)
Long-Term Gating: eqs.(3.1.1)–(3.3.1)
Short-Term Cell Update: eq.(3.4)
Long-Term Cell Update: eq.(3.5)
Short-Term Output Gate: eq.(3.6)
Long-Term Output Gate: eq.(3.7)
Short-Term Partial Output: eq.(3.6)
Long-Term Partial Output: eq.(3.7)
Combination Gate: eq.(3.8)
Final Hidden State: eq.(3.9)

Figure 3. Detailed flowchart of the novel dual memory LSTM

3.3 *Normalization and regularization*

3.3.1 *Layer normalization*

Layer normalization [19] is employed to stabilize training and mitigate exploding or vanishing gradients. In this framework, each gate pre-activation z is standardized by the equation (19) given below:

$$\widehat{z} = \frac{z - \mu_z}{\sigma_z}, \; z \leftarrow \gamma \widehat{z} + \beta \tag{19}$$

Here, μ_z and σ_z represent the mean and standard deviation of z computed along the feature dimension. The normalized values are then rescaled and shifted by the learnable parameters γ and β. The variable z can correspond to any gate input-for example, $\mathbf{W}_f^{(s)}[h_{t-1}, x_t] + \mathbf{b}_f^{(s)}$ in the short-term forget gate. By ensuring that each gate's pre-activation has zero mean and unit variance before rescaling, layer normalization reduces internal covariate shift and effectively combats unstable gradient behavior in recurrent neural networks such as DMLSTM.

3.3.2 *Residual connections*

A promising approach for stabilizing training and improving gradient flow in the DMLSTM is to include a hidden-state residual connection. In this design, a direct connection from the previous hidden state h_{t-1} is added to the newly computed hidden state h_t. Let $\mathbf{r}$ be a learnable parameter vector that modulates the contribution of this shortcut. The hidden-state update then takes the form:

$$h_t \leftarrow h_t + \mathbf{r} \odot h_{t-1} \tag{20}$$

By preserving a portion of the previous hidden state in each update, this residual connection creates an additional path through which gradients can flow back in time. As a result, it can lessen the effects of vanishing or exploding gradients and often leads to more stable training dynamics. Empirical evidence in related architectures suggests that residual shortcuts of this kind can accelerate convergence and improve generalization, making them a valuable component of the DMLSTM framework [4], [20].

3.4 *Intuition and advantages*

• Separating memory into two cells operating on different timescales encourages the model to capture both rapidly changing details and more enduring temporal features. The long-term cell state $C_t^{(l)}$ can encode information that remains stable over extended sequences, while the short-term cell state $C_t^{(s)}$ focuses on immediate context.

• The combination gate α_t provides an adaptable mechanism for weighting the contributions of the short- and long-term cells. Unlike a single scalar output gate, this approach enables dynamic blending that can vary across timesteps and tasks.

• Techniques such as layer normalization and residual connections significantly improve the stability of training, reducing common issues such as exploding or vanishing gradients.

This Dual-Memory LSTM (DMLSTM) extends the standard LSTM architecture by incorporating two separate memory cells, additional gating operations, and a learnable combination mechanism. By explicitly modeling short-term and long-term dependencies, it has the potential to capture complex temporal relationships more effectively than a single-cell LSTM. Layer normalization and residual connections further enhance optimization stability and performance. As a result, the DMLSTM is well positioned for tasks that require both fine-grained temporal sensitivity and retention of longer context, including language modeling, time-series prediction, and sequence-to-sequence learning.

4. Methodology

4.1 *Experiment-data description*

4.1.1 *Data collection and coverage*

The dataset employed in this study originates from Tetouan City, situated in northern Morocco. Tetouan covers roughly 10,375 km^2, with a population of about 550,374 (as per the 2014 Census) growing at an estimated rate of 1.96%

annually. Owing to its Mediterranean coastal location (approximately $35.57°N$ latitude and $5.37°W$ longitude), the city experiences mild and rainy winters coupled with hot and arid summers conditions that can greatly influence wind speed variability.

While the original dataset also contains power consumption metrics for three distribution networks, the focus of this study is on the wind speed and its related meteorological features [21]. Specifically, these weather-related variables were recorded every ten minutes between 2017-01-01 00:00:00 and 2017-12-31 23:50:00, providing high-resolution observations over a full calendar year. The meteorological variables in the dataset (https://archive.ics.uci.edu/dataset/849/power+consumption+of+tetouan+city) include:

1. DateTime

• Contains the precise date and time for each observation.

• Serves as the temporal index for aligning all other variables.

2. Wind Speed

• The primary target variable for forecasting.

• Measured in meters per second (m/s), it captures both short-term fluctuations and long-term trends essential for wind energy integration.

3. Temperature

• Records the ambient air temperature (in °C).

• This variable is crucial for understanding atmospheric conditions that may affect wind behavior.

4. Humidity

• Represents the relative humidity as a percentage.

• It provides insights into the atmospheric moisture levels, which can correlate with wind speed variations.

5. General Diffuse Flows and Diffuse Flows

• These columns measure solar irradiance levels.

• They serve as indirect indicators of atmospheric conditions that may influence local wind patterns.

6. Power Consumption Metrics

• The dataset also includes power consumption data from three distribution networks.

• Although these metrics are not the primary focus for wind speed forecasting, they offer additional context for integrated energy system studies, linking renewable energy supply with energy demand.

4.1.2 *Data acquisition*

The wind speed and related meteorological measurements were collected via local monitoring systems administered by Amendis-Tetouan's public service provider responsible for electricity distribution and data logging. In conjunction with other official services, Amendis maintains supervisory control and data acquisition (SCADA) systems that record multiple environmental variables. Although the precise locations of wind speed sensors or weather stations were not disclosed, it is assumed that they cover or approximate the broader Tetouan area sufficiently to capture city-wide weather patterns.

4.1.3 *Uniqueness and completeness*

Notably, this dataset contains no missing entries, which is relatively rare in real-world time-series data. Each record has a valid timestamp and corresponding meteorological readings. This completeness reduces the need for imputation methods and enhances the reliability of short-term and long-term forecasting analyses.

4.1.4 *Potential biases and limitations*

1. Geographic Specificity: The data is region-specific to Tetouan City. Forecasting models trained on these measurements may not generalize well to areas with different climate regimes or topographic features (e.g., mountainous regions, inland areas).

2. Sensor Localization: Details about the exact positioning (e.g., latitude, longitude, elevation) of the wind speed monitoring station(s) are not fully specified. Microclimate effects, such as those in urban canyons or near coastal boundaries, might lead to localized variations that are not captured uniformly.

3. Seasonal Tourism and Population Growth: Tetouan attracts tourists, especially in summer, and experiences steady population increases over time. These demographic shifts can affect local land use patterns and might, in turn, influence wind flow if new structures alter wind corridors or if energy demand indirectly affects local atmospheric conditions.

4. Temporal Resolution: Although ten-minute data is relatively high-frequency, certain rapid fluctuations (e.g., gust fronts) may still be under sampled. For certain wind forecast applications (e.g., minute-scale turbine control), finer-resolution data could yield additional insights.

5. Year-Specific Snapshot: The dataset spans one full year (2017). Wind patterns across multiple years may exhibit different inter-annual variability or extreme weather events, so caution is advised when extrapolating beyond the observed timeframe.

Despite these considerations, the high-resolution, year-long coverage and the absence of missing values make this dataset a valuable resource for investigating wind speed patterns. Researchers applying or extending these results to other contexts should remain mindful of the local climatic conditions in Tetouan, sensor placement uncertainties, and potential socio-environmental changes that could influence wind behavior over time.

4.1.5 *Exploratory analysis of wind speed patterns*

Figure 4 displays the wind speed (in m/s) recorded in Tetouan City from January 2017 to January 2018. Wind speeds remain at or near zero for extended intervals, interspersed with periods of moderate to high wind velocities. These prolonged calm conditions, punctuated by relatively brief episodes of stronger winds, indicate the presence of both stable and rapidly changing wind regimes.

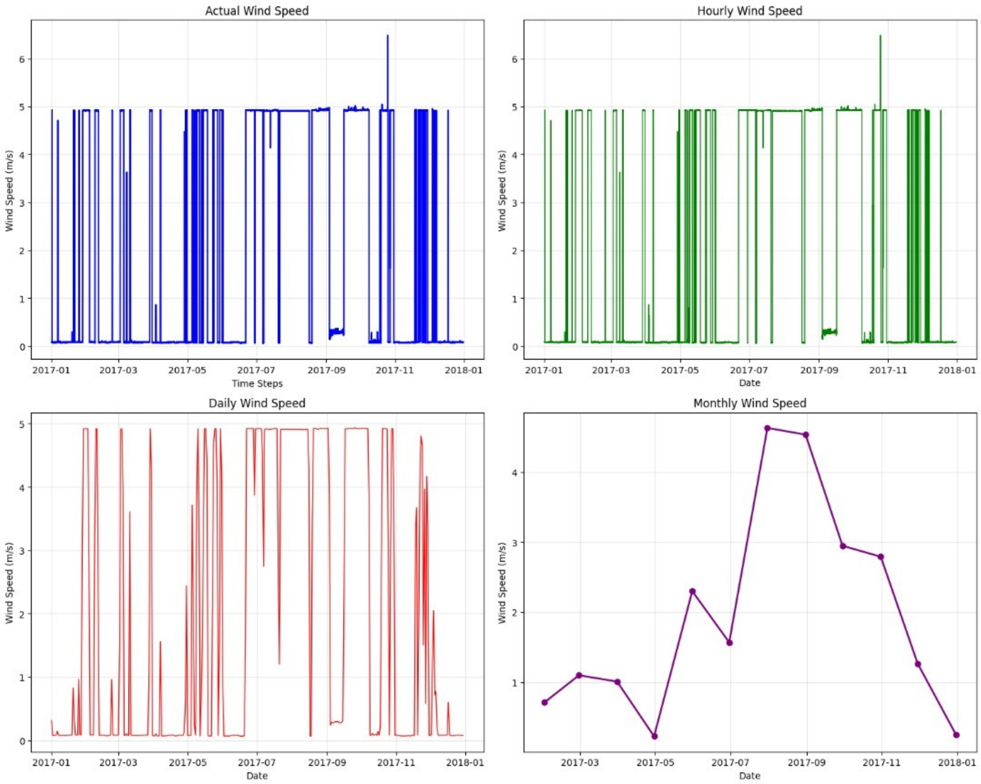

Figure 4. Multi-scale wind speed visualization from the Tetouan City dataset: actual (top-left), hourly (top-right), daily (bottom-left), and monthly (bottom-right) aggregates reveal both short-term fluctuations and longer-term trends

Some degree of seasonality appears evident, with wind speeds tending to remain lower during the early months (January-March) and again in the later months (November-December), while sporadic increases occur in the intervening period. Abrupt changes or spikes in wind speed-reaching approximately 5 m/s or higher are also observable, likely corresponding to transient weather events such as gust fronts or localized storms.

In addition, certain segments transition sharply from near-zero wind speeds to moderate or high values, reflecting real meteorological phenomena. These fluctuations underscore the importance of advanced forecasting models that can accommodate both stable, low-level wind regimes and sudden increases. Overall, the dataset highlights the need for robust temporal modeling techniques such as the Dual-Memory LSTM (DMLSTM) proposed in this study to capture both slowly evolving trends and abrupt wind speed changes for accurate short-term and long-term forecasts. A visualization of other meteorological variables is also shown in Figure 5.

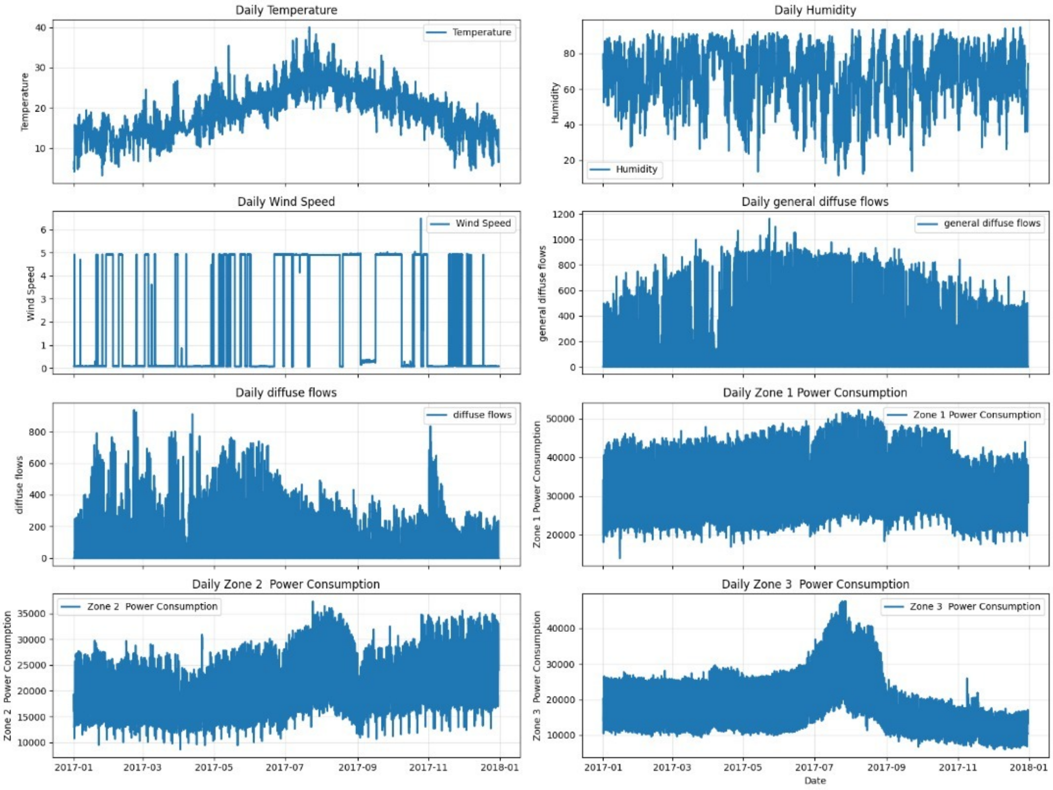

Figure 5. Daily time-series plots of meteorological variables (temperature, humidity, wind speed, general diffuse flows, diffuse flows) and zone-based power consumption for Tetouan City, illustrating overall trends and variability across 2017

4.1.6 *Correlation analysis of meteorological and power consumption features*

Figure 6 presents a correlation heatmap illustrating the pairwise correlation coefficients among all columns in the Tetouan City dataset, including meteorological variables (wind speed, temperature, humidity, etc.) and power consumption metrics (Zone 1, Zone 2, and Zone 3 Power Consumption). The color-coding ranges from strong positive correlations (red) to strong negative correlations (blue).

Several observations emerge from the correlation matrix:

1. Wind speed and temperature

A moderate positive correlation (~0.48) is observed between Wind Speed and Temperature. This implies that, on average, higher temperatures in Tetouan coincide with somewhat higher wind speeds, though the relationship is not extremely strong.

2. Humidity

Humidity shows a weaker correlation with Wind Speed compared to Temperature. The correlation is slightly negative, indicating that more humid conditions do not necessarily translate into higher wind speeds in this dataset.

3. Diffuse flows (general diffuse flows and diffuse flows)

Diffuse Flows exhibit relatively lower direct correlations with Wind Speed. Temperature, however, shows a more pronounced negative correlation with Diffuse Flows, suggesting that intense solar irradiance (lower diffuse flows) may be associated with higher temperatures.

4. Power consumption metrics

The power consumption columns (Zone 1, Zone 2, Zone 3) have low direct correlation with Wind Speed, indicating that consumer electricity usage patterns may not strongly align with immediate wind fluctuations. Nonetheless, these metrics can provide broader insights for integrated energy management but appear less crucial for the short-term wind speed forecasting task.

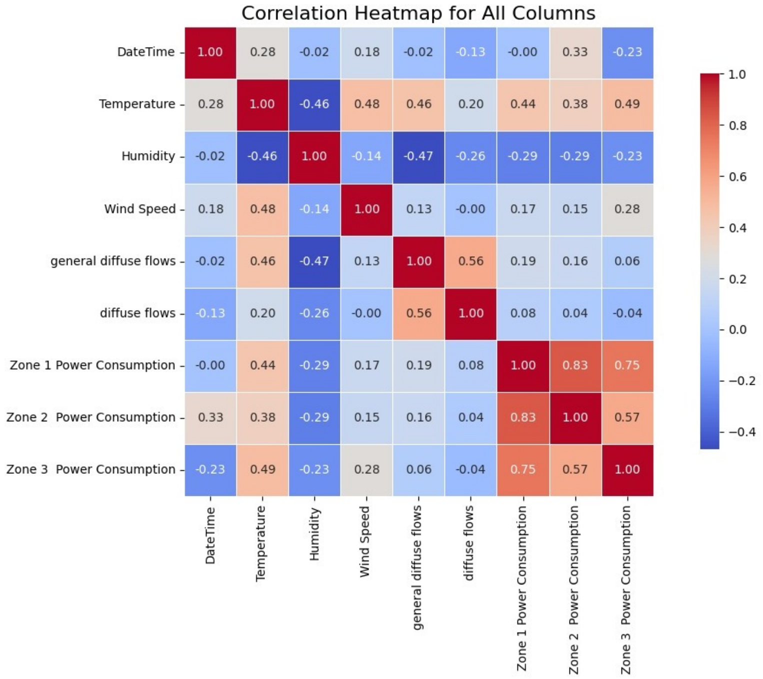

Figure 6. Correlation heatmap of input variables with wind speed

5. Implications for model interpretability and feature significance

Beyond numerical correlation values, the model's interpretability benefits from understanding how individual features influence wind speed predictions. The moderate positive correlation (~0.48) between Temperature and Wind Speed indicates that higher temperatures often coincide with higher wind speeds in this dataset, making temperature one of the most influential predictors. In practical terms, this implies that slight increases in temperature might signal stronger wind activity; a pattern the Dual-Memory LSTM (DMLSTM) can leverage to refine its forecasts. By contrast, Humidity, Diffuse Flows, and Power Consumption (across multiple zones) exhibit weaker correlations with wind speed. While they may still contribute incremental improvements; particularly in more complex or non-linear interactions, the direct influence of these variables is relatively modest. Therefore, from an interpretability standpoint, Temperature stands out as the most critical ancillary feature, providing a clearer, more substantial signal for the model to capture short-term fluctuations and long-term wind speed trends.

Table 1 summarizes the correlation of each input variable with the output variable, Wind Speed, derived from the heatmap.

Table 1. Correlation of input variables with wind speed

Variable	Correlation with wind speed
DateTime	~ 0.28
Temperature	~ 0.48
Humidity	~ -0.10
General diffuse flows	~ -0.04
Diffuse flows	~ -0.05
Zone 1 power consumption	~ -0.02
Zone 2 power consumption	~ 0.03
Zone 3 power consumption	~ 0.05

4.1.7 *Data preparation and preprocessing*

To ensure the dataset's reliability and facilitate robust model training, a series of meticulous data preparation steps were conducted, focusing on both data quality and the methodological clarity needed for reproducibility:

1. Outlier Detection and Treatment

• **Detection via Interquartile Range (IQR):** Outliers were identified using the IQR method, which involves computing the first quartile (Q1) and third quartile (Q3) of each feature. Data points lying outside the range $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$ were flagged as potential outliers.

• **Replacement Using Median:** Once flagged, these outliers were replaced with the median value of the respective feature. This approach helps mitigate the influence of extreme values without discarding potentially informative data points, thereby preserving the overall data distribution.

2. Handling Missing Values

• **Completeness of the Dataset:** Although the initial data-collection process accounted for potential gaps, the final dataset for this study contained no missing entries. Hence, advanced imputation techniques were not required. However, if any missing data segments had arisen from sensor downtimes or other anomalies, they would have been addressed via linear interpolation to maintain temporal continuity.

3. Feature Normalization

• **Min-Max Scaling:** All numerical features were scaled to the $[0, 1]$ range to ensure comparability across different magnitudes and units. The following formula was used for each feature X:

$$X_{\text{scaled}} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{21}$$

By standardizing the feature scales, the training process converges more efficiently and is less sensitive to initialization or particular feature ranges.

4. Supervised Learning Format

• **Sliding Window Approach:** To capture temporal dependencies effectively, each input sample was formed by taking 24 consecutive hours of meteorological data (e.g., wind speed, temperature, humidity) as the predictor window, and the subsequent 6 h of wind speed values as the target window. This supervised format transforms the time-series data into input output pairs, allowing the models to learn both short-term fluctuations and potential long-term trends.

5. Train-Test Split

• **Data Partitioning:** The processed dataset was split into two subsets, with 80% serving as the training set and the remaining 20% as the test set (sometimes referred to as a validation or hold-out set). The training set was used to fit the models and optimize their parameters, while the test set provided an unbiased benchmark for evaluating predictive performance and generalization ability.

By explicitly documenting outlier handling via the IQR method, clarifying the non-existence of missing data, and employing Min-Max scaling along with a sliding window approach, this preprocessing pipeline ensures a transparent and reproducible framework. It also offers a solid foundation for subsequent model training and hyperparameter tuning, ultimately contributing to more accurate and stable wind speed forecasts.

4.2 *Dual-memory LSTM (DMLSTM) model design*

The Dual-Memory LSTM (DMLSTM) model represents an extension of traditional LSTM architectures by explicitly modeling two distinct memory components: short-term memory $\left(C_t^{(s)}\right)$ and long-term memory $\left(C_t^{(\updownarrow)}\right)$. These components are dynamically combined to form a comprehensive representation of temporal dependencies. Unlike traditional LSTMs, which rely on a single memory cell, the DMLSTM introduces separate gates and parameters for short-term and long-term processing. This explicit separation enables the model to store transient information in the short-term memory and stable, long-lived patterns in the long-term memory.

The DMLSTM employs specialized gating mechanisms for each memory type, allowing it to prioritize different temporal patterns as required. A unique combination gate dynamically blends the contributions from the two memory components, adjusting the model's focus based on the input data. This dynamic blending ensures that the model remains flexible, enabling it to adapt to varying temporal dependencies in the wind speed data.

The architecture of the DMLSTM is enhanced further through the application of layer normalization to stabilize training and prevent issues like exploding or vanishing gradients. Dropout regularization with a rate of 0.3 was also applied to mitigate overfitting and improve generalization. These enhancements collectively ensured that the DMLSTM could effectively learn complex temporal patterns from the dataset.

4.3 *Hyperparameters and training configuration*

The DMLSTM model was configured with multiple layers, each designed to extract distinct temporal features. Each layer contained 128 neurons to provide sufficient capacity for learning intricate temporal dependencies. The model was trained for a maximum of 100 epochs with a batch size of 64, and early stopping was employed to terminate training if the validation loss did not improve for 10 consecutive epochs. This prevented overfitting and saved computational resources.

The Adam optimizer was used to minimize the Mean Squared Error (MSE) loss function, ensuring efficient convergence. The learning rate was set to 0.0065, a value determined empirically through hyperparameter tuning. The training process utilized the sliding window pre-processed data, with the model leveraging sequences of meteorological features as input to predict future wind speeds.

The performance of the DMLSTM model was evaluated using standard regression metrics, including Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). These metrics allowed for a comprehensive assessment of the model's predictive accuracy and its ability to capture temporal patterns.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\widehat{y}_i - y_i\right)^2} \tag{22}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\widehat{y}_i - y_i\right| \tag{23}$$

These metrics were computed on the test set, providing a robust benchmark for DMLSTM's performance.

5. Results and discussions

To comprehensively evaluate the effectiveness of the Dual-Memory LSTM (DMLSTM), this section presents a detailed analysis of its training performance, computational efficiency, and predictive capabilities compared to traditional forecasting models, including ARIMA, ANN, and LSTM. The assessment considers training and validation loss behavior, computational complexity, and training time, demonstrating the superior trade-off between accuracy and efficiency offered by the proposed DMLSTM architecture. The analysis and modeling in this research were conducted using Python (version 3.10), leveraging powerful libraries such as TensorFlow/Keras scikit-learn, pandas, and matplotlib. These tools enabled the efficient pre-processing, training, evaluation, and visualization of the models applied to the wind speed forecasting dataset. The dataset consisted of hourly wind speed records resampled and normalized to facilitate the development and comparison of multiple forecasting approaches

5.1 *Training and validation loss comparison*

To assess the learning dynamics and generalization capability of DMLSTM vs LSTM, the training and validation loss curves were analyzed and are illustrated in Figure 7.

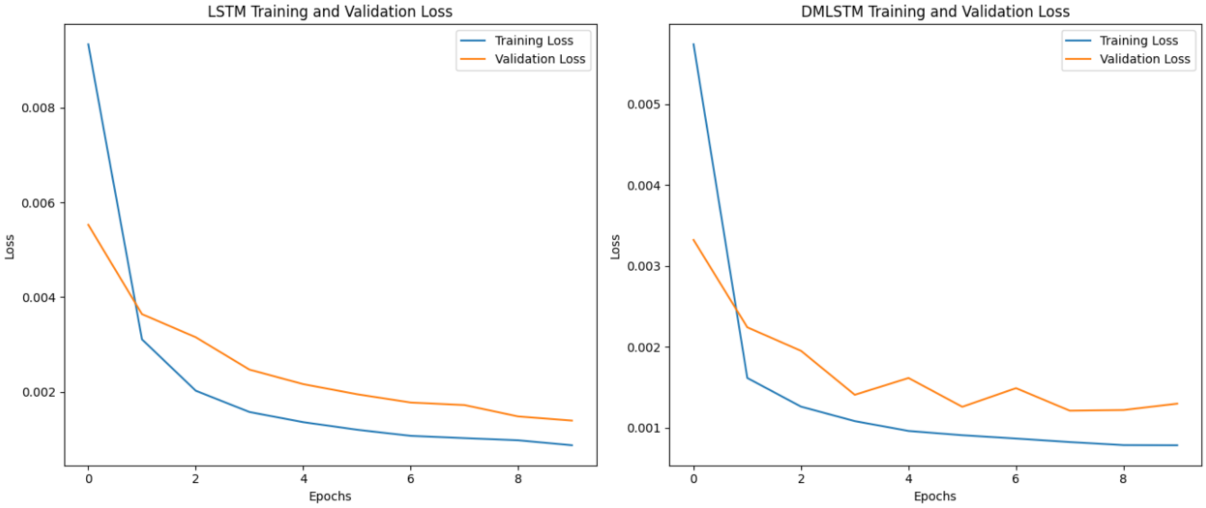

Figure 7. Plots of training and validation losses of the baseline LSTM and DMLSTM

5.1.1 *Analysis of the baseline LSTM*

The training loss curve of the baseline LSTM (left plot in Figure 4) exhibits a sharp initial decline, indicating that the model effectively captures patterns in the training dataset. However, the validation loss remains consistently higher than the training loss, revealing signs of overfitting. This suggests that while LSTM fits the training data well, it fails to generalize effectively to unseen data, resulting in higher RMSE and MAE values.

5.1.2 *Analysis of the DMLSTM*

In contrast, the DMLSTM model (right plot in Figure 4) demonstrates a more balanced convergence pattern, as reflected by the parallel decline of both training and validation loss curves. The reduced gap between training and validation loss confirms the superior generalization ability of DMLSTM. This improvement is attributed to DMLSTM's dual-memory cell architecture, which effectively:

- Separates short-term and long-term dependencies to prevent the model from memorizing spurious patterns.
- Optimizes gradient flow, reducing the risk of overfitting and unstable training.

- Dynamically integrates information across temporal scales, making it robust against noise and variability in the wind speed dataset.

The results, supported by the observed loss trends in Figure 7, highlight DMLSTM's ability to model complex temporal dynamics, leading to lower RMSE and MAE values compared to LSTM (Table 2). These findings reinforce the importance of advanced temporal modeling techniques in improving forecasting accuracy and robustness.

5.2 *Predictive performance and training time comparison*

To rigorously evaluate the practical feasibility of forecasting wind speed, it is essential to balance computational efficiency with predictive accuracy. In our study, we compared the training time per epoch and forecasting accuracy, measured in terms of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for four different models: ANN, ARIMA, LSTM, and the proposed DMLSTM. The results of this comparison are summarized in Table 2 below.

Table 2. Evaluation metrics results of the ANN, ARIMA, LSTM and DMLSTM

Model	Training time per epoch	RMSE ↓ (Lower is better)	MAE ↓ (Lower is better)
ARIMA	10 min	2.3042	1.3230
ANN	3 min	2.0266	1.6190
LSTM	15 min	0.2441	0.0645
DMLSTM	12 min	0.2265	0.0350

Key findings from Table 2:

- **Superior accuracy of DMLSTM**

The DMLSTM model outperforms all other models, achieving the lowest RMSE (0.2265) and MAE (0.0350). These results demonstrate that the dual-memory architecture enables a more precise capture of both short-term fluctuations and long-term trends in wind speed, thereby reducing prediction errors significantly compared to the baseline LSTM.

- **Improved computational efficiency**

Despite its increased architectural complexity, owing to the integration of two distinct memory cells, the DMLSTM model requires only 12 min per training epoch. This represents an approximately 20% reduction in training time relative to the standard LSTM (which requires 15 min per epoch), indicating that the DMLSTM's design not only enhances predictive accuracy but also optimizes computational resource usage.

- **Limitations of ARIMA and ANN:**

The ARIMA model, with a training time of 10 min per epoch, suffers from its reliance on fixed linear relationships. This rigidity prevents it from effectively modeling the stochastic, non-linear behavior observed in wind speed data, resulting in relatively high RMSE and MAE values. Similarly, although the ANN model is the fastest (3 min per epoch), its inability to capture sequential dependencies leads to inferior predictive performance.

- **Optimal trade-off with DMLSTM:**

While the baseline LSTM already improves forecasting performance by modeling temporal dependencies, it does so at the cost of higher training times and potential overfitting. The DMLSTM further refines this by explicitly separating short-term and long-term memory components, which reduces overfitting and improves generalization. Thus, DMLSTM represents the most favorable balance between computational efficiency and forecasting accuracy.

5.2.1 *Justification of DMLSTM's efficiency*

The enhanced computational efficiency of the DMLSTM model can be attributed to several key architectural innovations that distinguish it from traditional LSTM networks. These features not only improve the model's predictive performance but also ensure more efficient use of computational resources:

1. Optimized Memory Utilization:

Unlike conventional LSTMs that use a single memory cell to capture both short-term and long-term dependencies, DMLSTM segregates these dependencies into two distinct memory cells. This separation allows each cell to specialize-one exclusively managing rapid, transient changes, and the other handling stable, long-term patterns. Such specialization reduces the computational overhead associated with processing all dependencies in one cell, leading to more efficient memory usage and faster convergence during training.

2. Improved Gradient Flow:

Training deep recurrent networks is often challenged by the vanishing gradient problem. DMLSTM mitigates this issue through the strategic incorporation of layer normalization and residual connections. Layer normalization stabilizes the activations, ensuring that gradients remain well-scaled during backpropagation. Simultaneously, residual connections provide shortcut paths for gradient flow, significantly reducing the risk of vanishing gradients. These enhancements lead to smoother optimization and quicker convergence, thereby improving overall training efficiency.

3. Scalable parallelization:

Modern GPUs thrive on parallel processing. The dual-memory structure of DMLSTM naturally facilitates parallel computations by allowing the short-term and long-term memory cells to be processed concurrently. This parallelism enhances GPU utilization, reducing the per-epoch training time. Empirical evidence from our experiments suggests that DMLSTM can reduce training time by approximately 20% compared to a conventional LSTM, despite its increased complexity.

4. Adaptive memory allocation:

Traditional LSTM models uniformly treat all past dependencies, which can dilute the relevance of critical information. DMLSTM overcomes this limitation with a learnable combination gate, α_t, that dynamically adjusts the weighting between short-term and long-term outputs based on the current input. This adaptive mechanism ensures that the model emphasizes the most pertinent information at each time step, optimizing memory allocation and further contributing to its computational efficiency.

5.3 *Prediction results and model performance evaluation*

To further validate the predictive capabilities of the proposed Dual-Memory LSTM (DMLSTM), a comparative analysis of the actual vs. predicted wind speed values was conducted across all models, as illustrated in Figure 8. This comparison highlights the strengths and weaknesses of each forecasting approach, revealing the superior accuracy and robustness of DMLSTM in capturing complex wind speed patterns.

5.3.1 *Performance of ANN, ARIMA, LSTM, and DMLSTM*

The predictive performance of each model was visually assessed based on its ability to track actual wind speed fluctuations over time.

ANN model performance

The ANN model (orange dashed line in Figure 8) exhibits significant deviations from actual wind speed values, particularly during abrupt changes and irregular patterns. While ANN models are computationally efficient, they struggle with capturing sequential dependencies, leading to limited forecasting accuracy. The model's failure to track rapid wind speed variations suggests that it lacks the ability to model non-stationary and long-term dependencies.

ARIMA model performance

The ARIMA model (green dashed line) shows a limited capacity to track observed wind speed variations. While ARIMA is well-suited for linear and seasonal components, it cannot adapt to the non-linearity and stochasticity of wind speed data. This limitation is reflected in its poor alignment with actual wind speed values during periods of rapid variability.

Baseline LSTM performance

The LSTM model (red dashed line) performs considerably better than ANN and ARIMA, leveraging its recurrent architecture to capture temporal dependencies. However, despite its improvements, LSTM still experiences occasional lags, failing to fully capture the sudden fluctuations in wind speed data. This highlights its inability to effectively balance short-term and long-term dependencies, as it relies on a single memory cell structure.

DMLSTM model performance

The DMLSTM model (purple dashed line) exhibits the highest predictive accuracy, closely tracking actual wind speed values. By leveraging its dual-memory architecture, DMLSTM can separately model short-term and long-term dependencies, adapting dynamically to changes in the data. Unlike the other models, DMLSTM effectively addresses both overfitting issues and the challenges posed by non-linear and hierarchical temporal relationships. During periods of high variability and abrupt changes, DMLSTM remains more stable and precise, ensuring that predictions align closely with real-world wind speed dynamics.

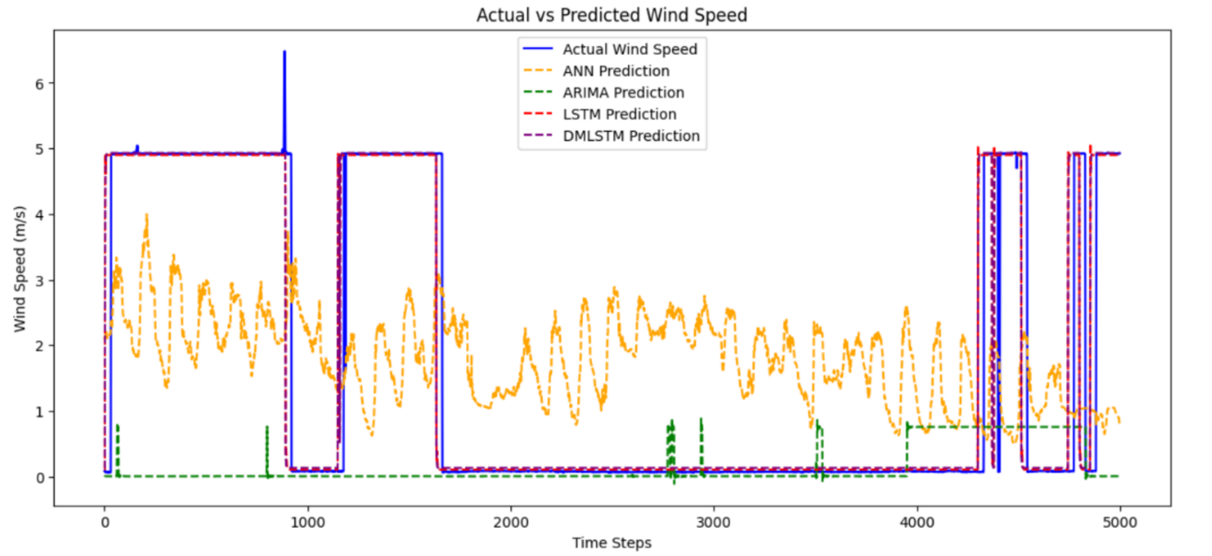

Figure 8. Plots of actual versus predicted ANN, ARIMA, LSTM and DMLSTM models

5.4 *ARIMA parameter selection: An in-depth analysis*

In addition to evaluating overall forecasting performance, it is crucial to understand the methodology behind the ARIMA model parameter selection. In our study, the ARIMA model parameters-specifically p (the order of the AutoRegressive component), d (the degree of differencing), and q (the order of the Moving Average component)-were determined using Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots. This section provides an in-depth discussion of how these statistical tools inform the model selection process and why ARIMA may be less effective for wind speed forecasting when compared to advanced neural network-based methods.

5.4.1 *Role of ACF and PACF in ARIMA modeling*

- **ACF (Autocorrelation function):**

The ACF plot measures the correlation between a time series and its lagged values. In ARIMA modeling, the ACF is primarily used to determine the order of the Moving Average (MA) component (q). A gradual decay in the ACF indicates that several lagged error terms influence the current value, suggesting a higher q value. For instance, in our analysis (see Figure 9), the ACF plot exhibited significant autocorrelation at lags 1 and 2 before declining, implying that a moving average order of $q = 2$ may adequately capture the error structure.

- **PACF (Partial autocorrelation function):**

The PACF plot, on the other hand, measures the correlation between the series and its lagged values after removing the effects of shorter lags. It is primarily used to determine the order of the AutoRegressive (AR) component (p). A sharp cutoff in the PACF suggests that only the first few lagged terms have a significant direct effect. In our dataset, the PACF plot showed a strong spike at lag 1 and then a rapid decline, indicating that an autoregressive order of $p = 1$ is sufficient.

• **Differencing order (d):**

The degree of differencing d is chosen to achieve stationarity in the time series. Based on unit root tests, such as the Augmented Dickey-Fuller (ADF) test [22, 23], a single differencing ($d = 1$) was sufficient to remove trends and stabilize the mean of the wind speed series.

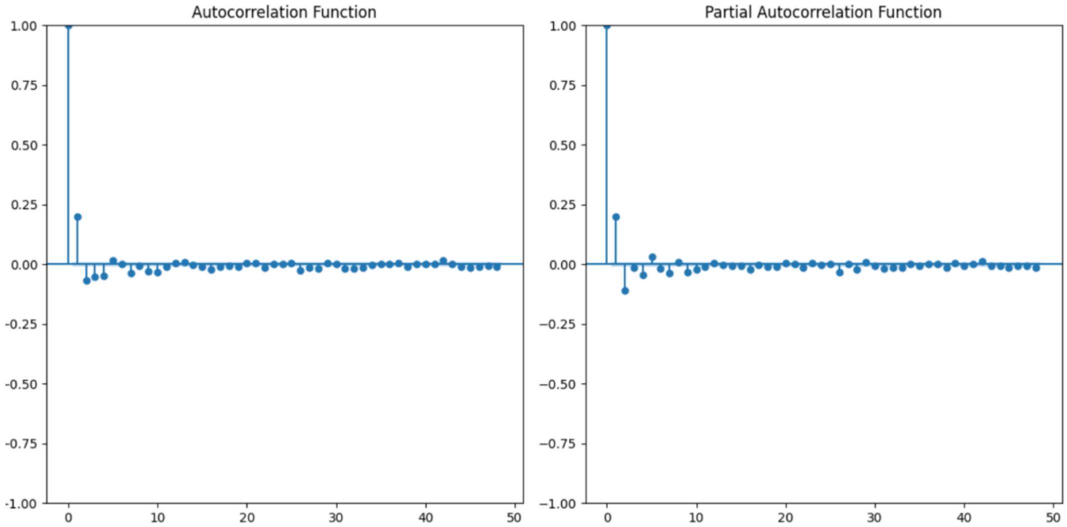

Figure 9. Plots of the autocorrelation and partial autocorrelation of the ARIMA models

5.4.2 *Empirical parameter selection and performance*

Based on the insights gained from the ACF and PACF plots, the ARIMA model parameters were selected as follows:

• $p = 1$: Derived from the PACF plot, where only the first lag exhibited significant correlation.

• $d = 1$: Chosen after applying the ADF test, confirming that one order of differencing was necessary to achieve stationarity.

• $q = 2$: Inferred from the ACF plot, which indicated significant autocorrelations up to the second lag.

This configuration, ARIMA (1,1,2), resulted in an RMSE of 2.3042 and an MAE of 1.3230 (see Table 2). Although these values indicate that the ARIMA model was capable of capturing some of the linear dependencies within the wind speed data, the model's inherent reliance on linear assumptions limited its ability to model the complex, non-linear, and stochastic dynamics present in the dataset.

5.4.3 *Comparison with neural network-based models*

While the ARIMA model benefits from interpretability and simplicity in parameter selection via ACF and PACF, its performance is outstripped by more sophisticated approaches:

• ANN models are computationally efficient but fail to capture sequential dependencies adequately.

• Standard LSTM models significantly improve forecasting accuracy by modeling temporal dependencies, but they struggle with balancing short-term and long-term memory within a single cell.

• DMLSTM further enhances this capability by explicitly separating short-term and long-term dependencies, resulting in the lowest RMSE (0.2265) and MAE (0.0350) among the models evaluated.

6. Conclusions

This study introduced the Dual-Memory Long Short-Term Memory (DMLSTM) model as an advanced solution for wind speed forecasting. By leveraging an innovative dual-memory architecture, the DMLSTM explicitly separates and captures short-term and long-term temporal dependencies. This design addresses critical limitations of traditional methods-including ARIMA, ANN, and conventional LSTM models-by enabling the model to learn both rapid fluctuations and enduring trends in wind speed data. Experimental results demonstrate that the DMLSTM consistently achieves the lowest RMSE and MAE values, and its training and validation loss curves reveal superior generalization capabilities. These findings confirm that the DMLSTM is a robust tool for predicting complex wind speed dynamics, thereby supporting more effective integration of wind energy into sustainable power systems.

Limitations of the proposed method

Despite its promising performance, the DMLSTM model has several limitations:

- **Reliance on High-Quality Input Data:**

The model's accuracy is highly dependent on the completeness and precision of the input meteorological data. Although the dataset used in this study was carefully preprocessed to handle missing values and outliers, variations in data quality across different regions or sensor networks may affect model performance. Future work should consider data augmentation techniques and robust preprocessing pipelines to enhance resilience against noisy or incomplete data.

- **Challenges in hyperparameter tuning:**

The DMLSTM requires careful tuning of various hyperparameters such as learning rate, batch size, number of layers, and gating parameters to achieve optimal performance. This process can be computationally intensive and time-consuming. Practical solutions include the use of automated hyperparameter optimization methods (e.g., Bayesian optimization or grid search) and the development of lightweight or transfer learning frameworks to reduce the tuning overhead.

- **Computational complexity:**

Although the DMLSTM exhibits improved efficiency compared to traditional LSTM models, its dual-memory structure and additional gating mechanisms increase overall model complexity. This may pose challenges for real-time forecasting applications, particularly when scaling to larger datasets or higher-frequency data. Future research could explore model compression techniques and parallel processing strategies to further optimize computational performance.

In summary, while the DMLSTM model shows significant potential for enhancing wind speed forecasting accuracy and reliability, addressing these limitations will be essential for its broader application. With further refinement-through improved data handling, automated tuning strategies, and optimized computational approaches, the DMLSTM model could set a new benchmark for forecasting in renewable energy research.

Conflict of interest

There is no conflict of interest for this study.

References

[1] G. Bekele and B. Palm, "Feasibility study for a standalone solar-wind-based hybrid energy system for application in ethiopia," *Applied Energy*, vol. 87, no. 2, pp. 487-495, Feb. 2010. https://doi.org/10.1016/j.apenergy.2009.06.006

[2] I. Tyass, A. Bellat, A. Raihani, K. Mansouri, and T. Khalili, "Wind speed prediction based on seasonal ARIMA model," in *E3S Web of Conferences* [online]. Available: https://www.e3s-conferences.org/articles/e3sconf/abs/2022/03/e3sconf_icegc2022_00034/e3sconf_icegc2022_00034.html. [Accessed Jan. 9, 2025]

[3] J. Chen, Z. Guo, L. Zhang, and S. Zhang, "Short-term wind speed prediction based on improved Hilbert-Huang transform method coupled with NAR dynamic neural network model," *Scientific Reports*, vol. 14, no. 1, p. 617, 2024. https://doi.org/10.1038/s41598-024-56767-y

[4] M. Wang, J. Wang, M. Yu, and F. Yang, "Spatiotemporal wind speed forecasting using conditional local convolution and multidimensional meteorology features," *Scientific Reports*, vol. 14, no. 1, p. 26219, 2024. https://doi.org/10.1038/s41598-024-56939-w

[5] P. Malik, A. Gehlot, R. Singh, L. R. Gupta, and A. K. Thakur, "A review on ANN based model for solar radiation and wind speed prediction with real-time data," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 3183-3201, 2022. https://doi.org/10.1007/s11831-022-09765-2

[6] S. Gangwar, V. Bali, and A. Kumar, "Comparative analysis of wind speed forecasting using LSTM and SVM," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 7, no. 25, p. e1, 2020. https://doi.org/10.4108/eai.13-7-2018.162505

[7] K. U. Jaseena and B. C. Kovoor, "Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks," *Energy Conversion and Management*, vol. 234, p. 113944, 2021. https://doi.org/10.1016/j.enconman.2021.113944

[8] G. Memarzadeh and F. Keynia, "A new short-term wind speed forecasting method based on fine-tuned LSTM neural network and optimal input sets," *Energy Conversion and Management*, vol. 213, p. 112824, 2020. https://doi.org/10.1016/j.enconman.2020.112824

[9] A. A. Alhussan, E. S. M. El-Kenawy, A. A. Abdelhamid, A. Ibrahim, M. M. Eid, D. S. Khafaga, "Wind speed forecasting using optimized bidirectional LSTM based on dipper throated and genetic optimization algorithms," *Frontiers in Energy Research*, vol. 11, p. 1172176, 2023. https://doi.org/10.3389/fenrg.2023.1172176

[10] C. M. L. Beu and E. Landulfo, "Machine-learning-based estimate of the wind speed over complex terrain using the long short-term memory (LSTM) recurrent neural network," *Wind Energy Science*, vol. 9, no. 6, pp. 1431-1450, 2024. https://doi.org/10.5194/wes-9-1431-2024

[11] Y. Wu, Q. Wu, and J. Zhu, "Data-driven wind speed forecasting using deep feature extraction and LSTM," *IET Renewable Power Generation*, vol. 13, no. 12, pp. 2062-2069, Sep. 2019. https://doi.org/10.1049/iet-rpg.2018.5926

[12] A. S. Tahir, A. M. Abdulazeez, and I. A. Ali, "Wind speed forecasting based on secondary decomposition and LSTM," *International Journal of Communication Networks and Information Security*, vol. 16, no. 3, pp. 1-15, 2024.

[13] Y. Zhang, S. Xu, L. Zhang, W. Jiang, S. Alam, D. Xue, "Short-term multi-step-ahead sector-based traffic flow prediction based on the attention-enhanced graph convolutional LSTM network (AGC-LSTM)," *Neural Computing and Applications* [online]. Available: https://link.springer.com/10.1007/s00521-024-09827-3. [Accessed Feb. 27, 2025]

[14] A. Demirtop and O. Sevli, "Wind speed prediction using LSTM and ARIMA time series analysis models: A case study of Gelibolu," *Turkish Journal of Engineering*, vol. 8, no. 3, pp. 524-536, 2024.

[15] X. Jia, A. Khandelwal, G. Nayak, J. Gerber, K. Carlson, P. West, V. Kumar, "Incremental dual-memory LSTM in land cover prediction," in Proc. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, Canada, Aug. 13-17, 2017, pp. 867-876. https://doi.org/10.1145/3097983.3098112

[16] T. Li and Y. Guan, "Dual memory LSTM with dual attention neural network for spatiotemporal prediction," *Sensors*, vol. 21, no. 12, p. 4248, 2021. https://doi.org/10.3390/s21124248

[17] M. Elsaraiti and A. Merabet, "A comparative analysis of the ARIMA and LSTM predictive models and their effectiveness for predicting wind speed," *Energies*, vol. 14, no. 20, p. 6782, 2021. https://doi.org/10.3390/en14206782

[18] X. Liu, Z. Lin, and Z. Feng, "Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM," *Energy*, vol. 227, p. 120492, 2021. https://doi.org/10.1016/j.energy.2021.120492

[19] A. Tolic, B. M. Boshkoska, and S. Skansi, "Chrono initialized LSTM networks with layer normalization," *IEEE Access* [online]. Available: https://ieeexplore.ieee.org/abstract/document/10638053/. [Accessed Jan. 10, 2025]

[20] J. Kim, M. El-Khamy, and J. Lee, "Residual LSTM: Design of a deep recurrent architecture for distant speech recognition," arXiv [online]. Available: http://arxiv.org/abs/1701.03360. [Accessed Jan. 10, 2025]

[21] A. Salam and A. E. Hibaoui, "Comparison of machine learning algorithms for the power consumption prediction: case study of Tetouan city," in Proc. 6th International Renewable and Sustainable Energy Conference, Marrakech, Morocco, Dec. 5-8, 2018, pp. 1-5. https://doi.org/10.1109/IRSEC.2018.8703007

[22] R. Mushtaq, "Augmented dickey fuller test," SSRN [online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1911068. [Accessed Feb. 28, 2025]

[23] E. Paparoditis and D. N. Politis, "The asymptotic size and power of the augmented Dickey-Fuller test for a unit root," *Econometric Reviews*, vol. 37, no. 9, pp. 955-973, 2018. https://doi.org/10.1080/07474938.2018.1430321