

Multivariate time series weather forecasting model using integrated secondary decomposition and Self-Attentive spatio-temporal learning network

Nirmal Kumar M.¹, Subramanian P.^{2*} and Surendran R.³

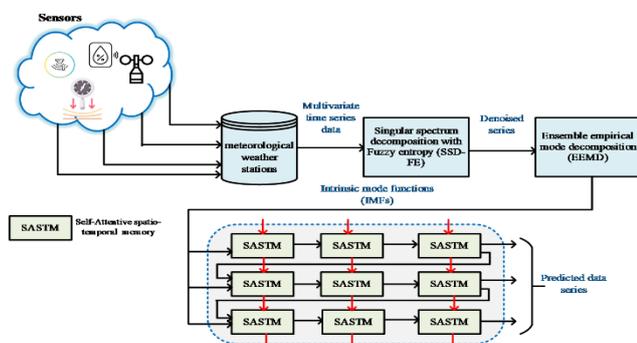
^{1,2,3}Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, 602105, India

Received: 23/05/2024, Accepted: 03/01/2025, Available online: 14/01/2025

*to whom all correspondence should be addressed: e-mail: subramanianp.sse@saveetha.com

<https://doi.org/10.30955/gnj.06195>

Graphical abstract



Abstract

Weather forecasting is an essential but complex task because of its substantial influence on human life and advanced atmospheric dynamics. In recent years, deep learning-based techniques have gain high attention for weather forecasting. Nevertheless, current forecasting models ignore the interdependent relationship between variables across regions and primarily examine temporal patterns of weather data. In this work, a new Multivariate time series weather forecasting model is proposed using integrated secondary decomposition and Self-Attentive spatio-temporal learning network (SASTLNet). Particularly, the weather data is filtered using a Singular Spectrum Decomposition with Fuzzy Entropy (SSD-FE) for removing the irrelevant components and screening the component containing vital information. Moreover, the spatiotemporal fluctuations in the meteorological variables are investigated using an enhanced empirical mode decomposition (E3MD) method and a SASTLNet model. The self-attention block of the SASTLNet model is discovered using ladder format to lower the processing costs while representing the temporal and spatial features through single memory cell. The simulation results prove that the suggested model can outperforms the baseline models in terms of efficacy and accuracy.

Keywords: Weather forecasting, spatio-temporal learning, secondary decomposition, self-attention, and Singular spectrum decomposition

1. Introduction

Weather forecasting (WF) is a challenging process because of the complex system of atmospheric motion and its considerable impact on human life. Fang *et al.* (2021) exposed the main goal of WF models is to predict variables including temperature, humidity, dew point, rainfall, and wind speed using historical data. Accurate weather forecasts are critical for many industries, including commerce (Wen, L. *et al.* (2024), Wang, Z. *et al.* (2024)), tourism (Liqin, W., and Yuan, Y. (2024)), sports, agriculture, mining, power generation (Yang *et al.* 2022), the food industry, airports (Periasamy S. *et al.* 2024), and naval activities. The WF methods can be widely classified as physical, statistical, intelligent, and hybrid methods. Mayer *et al.* (2023) and Zhang, *et al.* (2022) discussed the most popularly utilized physical technique is numerical weather prediction (NWP) model. It uses computer algorithms to solve a challenging set of nonlinear mathematical equations based on certain mathematical models in order to generate a forecast. Ren *et al.* (2021) implemented this approach to increases the financial cost and requires extensive computations.

In weather forecasting, statistical methods are crucial because they provide valuable tool for investigating the historical data, identifying trends, and generating forecasts. Jaseena, K. U *et al.* (2022) and Sharadga, *et al.* (2020) introduced statistical models are ARMA, ARIMA, and their derivatives. However, statistical approaches frequently require human interaction for parameter adjustment and assumptions derived from the historical data. It might not be suitable for new or varying weather patterns. In contrast, Dewitte *et al.* (2021) and Surendran R *et al.* (2021) developed the artificial intelligence models can continuously evolve and adapt to changing conditions, as they are capable of learning from new data and adjusting over time. Furthermore, it is not limited to

predetermined concepts and may discover more complicated patterns and correlations from data. Weather forecasting uses a number of well-known artificial intelligence models, such as autoencoders (AE), support vector machines (SVM), artificial neural networks (ANN), convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory networks (LSTM), and autoencoders (AE) from various researchers Donadio *et al.* (2021), Markovics *et al.* (2022), Xiao *et al.* (2021), Surendran R *et al.* (2023), Mani *et al.* (2023), Yadav *et al.* (2024).

The hybrid models combine two or more models to further improve forecasting model performance. The majority of the early models only used information at the time level and did not account for information at the space level (Venkatachalam, K *et al.* (2023), Dotse *et al.* (2024), Alqahtani *et al.* (2023)). Goel *et al.* (2023) and Christoforou, E *et al.* (2023) improved the temporal and geographical information incorporated into the models for increasing the accuracy of short-term weather forecasting. Spatio-temporal data shows universal characteristics of correlation and heterogeneity. When data are autocorrelated in both the temporal and spatial dimensions, it is referred to as correlation. Jin *et al.* (2023) explained the ability of spatiotemporal data to exhibit distinct patterns over a variety of temporal or spatial dimensions is known as heterogeneity. Recently, CNNs were used to forecast wind power and wind speed based on spatiotemporal features due to the advancement of deep learning methods (Liu *et al.* (2020)).

Weerakody *et al.* (2021) determined the intelligent approaches may be more sensitive to noisy data. Noise in the data can mask fundamental patterns and result in incorrect predictions. He Renfei *et al.* (2022) approached a decomposition boost prediction accuracy and enhance the signal-to-noise ratio by eliminating noise or random fluctuations. Therefore, a hybrid model that combines the best aspects of multiple models is needed to increase the accuracy of weather forecasting. Hybrid models based on the data decomposition approach aim to enhance forecasting performance through the reduction of stochastic disturbance of weather data series. These points motivate us to propose a new hybrid weather forecasting network based on decomposition and spatio-temporal learning methods. The research contribution of this research work are listed as follows:

- To introduce a new Multivariate time series WF model using integrated secondary decomposition and deep learning network.
- To integrate the concept of fuzzy entropy estimation into singular spectrum decomposition (SSD) algorithm. This will enable the screening of meteorological data as a component containing essential information and the removal of irrelevant components.
- To propose a new SASTLNet for capturing the spatiotemporal mutual dependence information at the same time. This SASTNNet contains a novel self-attention based spatiotemporal memory

(SASTM) cell to obtain global spatial contextual information by describing the connections between different regions using a self-attention block.

- To reduce the computational cost of self-attention mechanism by modelling a lightweight ladder self-attention block.

Studying a weather forecasting model in the Chinese context not only makes sense due to China's unique geographic and climatic features but also because of its ongoing technological innovations, large-scale infrastructure, environmental challenges, and critical role in global forecasting efforts. The model would be particularly important for addressing local weather phenomena, managing natural disasters, optimizing resource allocation, and contributing to China's broader sustainability and technological goals. By understanding how China develops and utilizes weather forecasting tools, we can gain valuable insights into improving prediction accuracy and disaster preparedness both within China and globally.

The rest of the paper is structured as follows: Section reviews the existing papers related to weather forecasting. Section 3 gives the description for the proposed weather forecasting model in detail. Section 4 investigate the effectiveness of the proposed model by conducting extensive simulation. Section 5 Concludes the proposed work with future research direction.

2. Literature survey

Meteorological analysis and weather forecasting play a major role in sustainable development to mitigate the damage caused by extreme events. The ability of LSTM to record long-term dependencies has led to substantial performance on numerous real-world applications. Surendran R *et al.* (2023) used LSTM based data-driven prediction framework for a WF application. Also, a Transductive LSTM (T-LSTM) was developed for exploiting the local knowledge from the time series data. In T-LSTM, the samples near the test point were thought to have a greater influence on model fitting. Also, a quadratic cost function was used for the regression problem. Bai *et al.* (2020) stated that node-specific patterns must be learned efficiently to avoid a predefined map during prediction. Also, an Adaptive Graph Convolutional Recurrent Network (AGConvRN) was introduced, which automatically seized detailed spatio-temporal correlations in data series. This was achieved through two modules: Node Adaptive Parameter Learning and Data Adaptive Graph Generation with RNN.

Wu *et al.* (2024) introduced a graph neural network (GNN) for multi-variable time series data (MVTGNN). This model extracted the one-way relations between variables via a graph learning unit. This unit could be united with external knowledge such as variable attributes seamlessly. Two additional layers including mix-hop propagation and a dilated inception layers were introduced for capturing the spatio-temporal dependences in the time series. Here the graph learning, graph convolution, and temporal

convolution blocks were collaboratively trained to form end-to-end framework.

One of the key markers for identifying climate change is variations in the earth's surface temperature. Suleman *et al.* (2022) introduced a new Spatial Feature Attention LSTM (SFA-LSTM) model for capturing the spatio-temporal interactions of several meteorological features. Accurate data forecasting was aided by significant spatial features and temporal interpretations of previous data that were directly connected to output features. The mutual influence of input features on the target feature was captured by the spatial feature attention. Here, the encoder-decoder structure allowed for learning temporal dependencies in the data through the use of LSTM layers during the encoder stage and spatial feature relations during the decoder stage. SFA-LSTM anticipates temperature through the simultaneous learning of the most significant time steps and meteorological factors.

Han *et al.* (2022) introduced an integrated wind speed forecasting framework on the basis of weather research and forecasting (WRF) model. Initially, the WRF model was used for obtaining the forecasted wind speed. Additional meteorological data were also collected from the various WRF fields. Moreover, the primary meteorological variables were chosen as the input series using the Pearson Correlation Coefficient (PCC) approach. Then the historical data and input series were decomposed into the appropriate intrinsic mode functions (IMFs) using the complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) approach. After that, a novel hybrid deep learning model was introduced that combined a CNN and a bidirectional LSTM (BiLSTM) for predicting the error and correcting the wind speed from WRF's deepest field. Here, the BiLSTM was enhanced using an attention and a grid search approach. At last, a validation case study was carried out to confirm that the suggested model works as intended.

Ma *et al.* (2023) developed a new Hierarchical Spatio-temporal Graph Neural Network (HST-GNN) for enabling precise prediction of several variables and stations over several time steps. HST-GNN incorporated a dynamic graph learning unit for constructing a self-learning hierarchical graph. It comprised a global graph that depicted regions and a local graph that recorded meteorological information for every area. This model captured a variety of long-term meteorological trends and hidden spatial relationships using a dilated inception and graph convolution as the foundation. An adaptive collaborative learning was also introduced for facilitating bidirectional information among the two-stage graphs.

Shenglin, M *et al.* (2024) analysis and demonstrates that spatiotemporal learning is essential for weather forecasting because of complex and dynamic characteristics of atmospheric events. Even though the existing deep learning algorithms are capable of producing accurate prediction results, there are still have certain issues. The existing T-LSTM does not capture spatial dependencies naturally. Also, GNNs have been the subject of numerous studies including GConvRN, MVTGNN and

HST-GNN. However, they do not take into account the long-range time correlation between the different time steps of the nodes. Furthermore, they are unable to effectively depict the complex spatial-temporal dependencies in the graph. Inaccurate predictions may result from random disturbances in the time series weather data. None of the existing models considered this issue before learning the spatiotemporal feature. To tackle these issue, this research introduced a new hybrid models based on the data decomposition approach. These models have the ability to forecast more accurately and are designed to remove stochastic disturbances from meterological variable series.

3. Proposed method

In this work, a new Multivariate time series weather forecasting model is proposed using integrated secondary decomposition and Self-Attentive spatio-temporal learning network (SD-SASTLNet). Figure 1 shows the complete architecture of the proposed weather forecasting model. Initially, the source data including temperature, humidity, wind speed, and pressure is acquired from weather forecasting datasets. After that, SSD-FE is introduced for decreasing the noise of the original multivariate series. Subsequently, an E3MD algorithm is used for decomposing the denoised series into their respective IMFs and residuals. Further, a new Self-Attentive spatio-temporal learning network (SASTLNet) is introduced for weather forecasting, which describes the temporal and spatial models via a unified self-attentive spatio-temporal memory (SASTM) cell. The SASTM architecture is constructed using Convolutional LSTM (C-LSTM) units as a base. In contrast to traditional methods that only use simple convolutions to extract spatial information, SASTM improves the C-LSTM by incorporating a self-attention module. This change significantly enhances the network's ability to represent global spatial data.

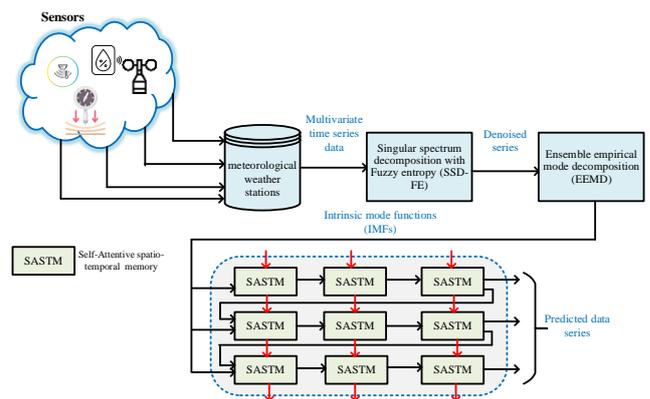


Figure 1. Architecture of the proposed weather forecasting model

3.1. Singular spectrum decomposition with fuzzy entropy

In this work, the original multivariate meteorological data is denoised and decomposed using SSD-FE. An innovative data analysis algorithm that has gained popularity recently is called singular spectrum analysis (SSA). Zou, F *et al.* (2024) is discussed the capability of separating weak fault features and reconstructing nonlinear time series.

Nonetheless, the artificial selection of the SSA embedding dimension is unavoidable, and the choice of parameters significantly impacts the decomposition outcomes. Santhanaraj R. K *et al.* (2023) address this problem, an adaptive signal analysis method namely Singular spectrum decomposition (SSD) is created. The weather data may contain noisy information because of the motion of the sensors. As a consequence, the suggested model screened and denoised the SSD decomposition findings using fuzzy entropy. The steps comprised in the suggested SSD-FE is described as follows:

Gather the original time series weather signal S_k .

Execute SSD for decomposing the sensed weather signal into k singular spectrum factors SSFs $S = \{s_1^k, s_2^k, \dots, s_L^k\}$.

Compute fuzzy entropy FE_k of every SSFs.

Select SSFs whose fuzzy entropy value exceeds the average value.

Use a soft threshold to reduce the noise of specific SSFs and then rebuild the signal. The threshold is computed using the following equation:

$$\gamma = \sigma \sqrt{2 \log_{10} L} \quad (1)$$

Every component s_{kj} ($j = 1, 2, \dots, L$) of SSF S_k is contracted utilizing the soft threshold:

$$\hat{s}_{kj} = \begin{cases} s_{kj} - \gamma, & s_{kj} \geq \gamma \\ 0, & |s_{kj}| < \gamma \\ s_{kj} + \gamma, & s_{kj} \leq -\gamma \end{cases} \quad (2)$$

The following paragraphs provide the detailed SSD solution process. Let $s(n)$ denotes a time-series weather data set with length L and embedding dimension is K . The construction of the trajectory matrix ($K \times L$) is given by $S = [s_1^T, s_2^T, \dots, s_K^T]$ SSD offers a useful way to flexibly determine the embedding dimension K using iterative process. SSD generates a residual component $r_i(n)$ and its power spectral density (PSD) using

$$(n) = s(n) - \sum_{u=1}^{i-1} r_u(n), \quad r_0(n) = s(n) \quad (3)$$

where i represents the iteration. If $i > 1$, the embedding dimension is rephrased as:

$$K = 1.2 \times \left(\frac{\hat{f}_s}{\hat{f}_{max}} \right) \quad (4)$$

Where \hat{f}_{max} is PSD peak, \hat{f}_s denotes the sampling frequency. Then, the singular value decomposition (SVD) is used to construct trajectory S as given below:

$$S = A \wedge B^T = [a_1, a_2, \dots, a_k] \begin{bmatrix} \varepsilon_1 & 0 & 0 & 0 & 0 \\ 0 & \varepsilon_2 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & 0 & \\ 0 & 0 & 0 & \varepsilon_k & 0 \end{bmatrix} \begin{bmatrix} b_1^T \\ b_1^T \\ \vdots \\ b_l^T \end{bmatrix} \quad (5)$$

Where $a_m \in \mathbb{R}^{K \times 1}$ represents m -th column vector of $B \in \mathbb{R}^{L \times L}$, $b_m \in \mathbb{R}^{L \times 1}$ defines the m -th column vector of $A \in \mathbb{R}^{K \times K}$, ε_m denotes the singular value matrix $\wedge \in \mathbb{R}^{K \times L}$. The trajectory matrix is defined as:

$$S = S_1 + S_2 + \dots + S_k \quad (6)$$

Where $S_m = \varepsilon_m a_m b_m$. The subsequent stage is the reconstruction of particular signals using these primary components. Lastly, the iteration's termination condition needs to be determined. The definition of the normalized mean square error (NMSE) between the residual component and the raw data is

$$NMSE^{(i)} = \frac{\sum_{j=1}^L (b^{(i+1)}(j))^2}{\sum_{j=1}^m (s(j))^2} \quad (7)$$

If NMSE is less than the specified threshold $T_h = 0.01$, it terminates the SSD. One can acquire the decomposition result as follows:

$$s(n) = \sum_{k=1}^z \tilde{u}^{(k)}(n) + b^{(z-1)}(n) \quad (8)$$

Where z denotes the number of SSFs and $\tilde{u}^{(k)}$ denotes the k -th SSF. Following the SSF calculation, the fuzzy entropy FE_k is calculated for every SSF value. Fuzzy membership function was introduced by fuzzy entropy in order to enhance traditional information entropy. The k dimension vector is created by considering the SSF series as $S = \{s_1, s_2, \dots, s_L\}$.

$$S_j^k = \{s(j), s(j+1), \dots, s(j+k-1)\} - \bar{s}(j) \quad (9)$$

$, j = 1, \dots, L - k + 1$

Where k represents the embedding dimension. The distance of S_j^k from its nearby vector S_i^k is detected as:

$$D_{ji}^k = D[S_j^k, S_i^k] = \max_{m \in (0, k-1)} |s_j^k(m) - s_i^k(m)|, \quad (10)$$

$i, j = 1 \sim \tilde{N} - k + 1, i \neq j$

The resemblance between S_j^k and S_i^k is computed using the fuzzy membership function as

$$S_{ji}^k = \mu(D_{ji}^k, g, w) = \exp\left(-\left(D_{ji}^k\right)^g / w\right) \quad (11)$$

where $\mu(D_{ji}^k, g, w)$ is denotes the fuzzy membership function, g and w denote the gradient and width respectively. Analogous to sample entropy, the average fuzzy similarity degree is provided as follows

$$\delta_g^k = \frac{1}{L-k} \sum_{i=1, j \neq i}^{L-k+1} S_{ji}^k \quad (12)$$

If two vectors are matched based on fuzzy probability then it is defined as:

$$\rho_g^k = \frac{1}{L-k+1} \sum_{j=1}^{L-k+1} \delta_g^k(j) \quad (13)$$

The fuzzy probability of $\{S_j^{k+1}\}$ is derived as

$$\rho_g^{k+1} = \frac{1}{(L-k)(L-k-1)} \sum_{j=1}^{L-k} \sum_{i=1, i \neq j}^{L-k} \delta_{ji}^{k+1} \quad (14)$$

The Fuzzy entropy (k, w) is expressed as

$$FE(k, w) = \lim_{n \rightarrow \infty} \left(-\ln(\rho_g^{k+1} / \rho_g^k) \right) \quad (15)$$

When L is limited, the fuzzy entropy is redrafted

$$FE(k, w, L) = \ln(\rho_g^{k+1} / \rho_g^k) \quad (16)$$

3.2. Enhanced ensemble empirical mode decomposition (E3MD)

The EMD is a dyadic filter that can separate the time series denoised weather signal $s'(n)$ into various types of time series components known as IMFs. The process continues until the residue parameter $r(n)$ is lesser than a predefined value of L significance or the residue parameter $r(n)$ becomes a monotonic function (Golyandina *et al.* (2020)). At last, the signal $s'(n)$ is stated as the addition of constituents with the last residue parameter $r(n)$. But the main problem of EMD is mode mixing. This work proposes an upgraded EMD to tackle the mode mixing issue in the EMD approach. This model receives the denoised weather time series data as input to decompose it into the residue $r(n)$ and IMFs. The residue $r(n)$ denotes the intrinsic signal inclination and is thought to have disconnected fluctuant elements. Figure 2 shows a flow chart for the proposed method.

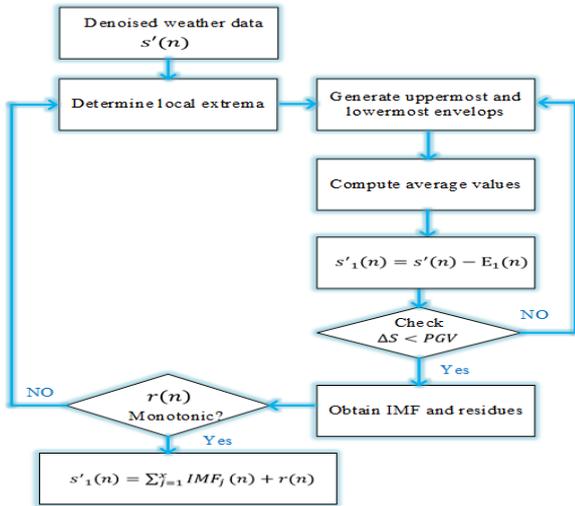


Figure 2. Flow chart of the E3MD approach

The following is a description of the decomposition process:

Determine the local peak of the signal $s'_1(n)$, and create the upper and lower envelopes via cubic spline interpolation. Then compute average of upper and lower envelopes $E_1(n)$:

$$s'_1(n) = s'(n) - E_1(n) \quad (17)$$

Use the standard deviation (ΔS) to decide whether to continue or halt the sifting procedure described above,

$$\Delta S = \sum_{n=0}^n \left| s'_j(n) - s'_{(j-1)}(n) \right|^2 / s'^2_{(j-1)}(n) \quad (18)$$

where J denotes the total iterations.

If ΔS is lesser than a predefined value, the procedure described above must be halted and compute IMF1

$$IMF_1 = s'_n(n) - E_n(n) \quad (19)$$

The residue $r(n)$ is the variances between IMF1 and signal $s'_1(n)$

$$r(n) = s'_n(n) - IMF_1 \quad (20)$$

Reiterate step 1 to step 4 till $r_x(n)$ turn out to be a monotonic function

$$r_j = r_{j-1} - IMF_j, j = 2, 3, \dots, x \quad (21)$$

Consequently, a residual that is isolated from $s'_1(n)$ and the number of IMFs are derived.

$$s'(n) = \sum_{j=1}^x IMF_j(n) + r(n) \quad (22)$$

The proposed model constructs a two-dimensional frequency matrix using IMFs and residual.

The constructed frequency matrix (F) was used as the input of the SASTLNet.

3.3. Self-Attentive spatio-temporal learning network (SASTLNet)

In this work, SASTLNet is proposed for extracting the spatiotemporal information concurrently. SASTLNet is developed specifically for short-term weather forecasting. Here, stacked RNN is used for describing spatiotemporal sequence forecasting problems as in Figure 3. The SASTM module is the fundamental unit of SASTLNet. By extracting extremely abstract information layer by layer and transferring them back to the values of the weather parameter, the SASTM may generate forecasting results.

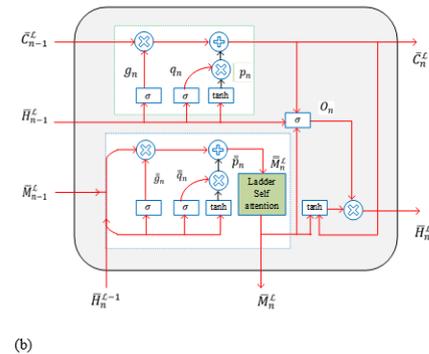
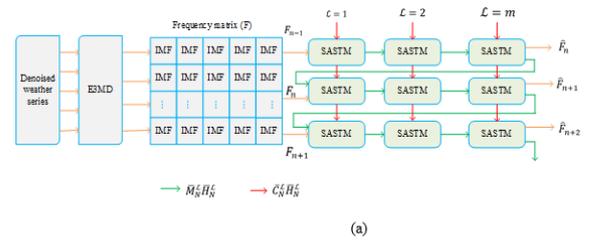


Figure 3. Realization of (a) SASTLNet (b) SASTM

The following expression illustrates the spatial dimension status conversion between the two-time steps.

$$\bar{M}_n^0 = \bar{M}_{n-1}^m \quad (23)$$

$$\bar{H}_n^0 = \bar{H}_{n-1}^m \quad (24)$$

Where \bar{M} denotes the spatial and temporal memory, \bar{H} represents the status of hidden layer, and m denotes the amount of stacked recurrent network layers. It should be noted that the spatiotemporal memory's starting value \bar{M}_n^0 is initiated with all zeros. For $2 \leq L \leq m$, the SASTM network with m -layer stack is developed using the following expression:

$$[\bar{H}_n^1, \bar{C}_n^1, \bar{M}_n^1] = \text{SASTM}_1(F_n, \bar{H}_{n-1}^1, \bar{C}_{n-1}^1, \bar{M}_n^0) \quad (25)$$

$$[\bar{H}_n^L, \bar{C}_n^L, \bar{M}_n^L] = \text{SASTM}_1(\bar{H}_{n-1}^{L-1}, \bar{H}_{n-1}^L, \bar{C}_{n-1}^L, \bar{M}_n^{L-1}) \quad (26)$$

Where hidden status \bar{H}_n^1 are tensors, \bar{C}_n^1 denote cell outcome and F_n represents input frequency matrix. Observe that the initial SASTM layer is designated as SASTM_1 . Figure 3 (b) depicts the proposed SASTM architecture. The SASTM is constructed using C-LSTM. It is utilized for capturing the required data in the temporal dimension. An additional self-attention module based on C-LSTM is introduced for capturing the long-term dependences of contextual data in the spatial dimension. Lastly, weather data is predicted by combining the data on temporal and spatial dimensions.

3.3.1. Temporal Learning

Initially, the convolutional gating unit in C-LSTM is utilized for obtaining the present changes in the weather data along the temporal domain based on the hidden status data \bar{H}_n^{L-1} . Subsequently, the proposed model updates the preceding temporal cell state \bar{C}_{n-1}^L to generate temporal unit \bar{C}_n^L as indicated by the green line container in Figure 3. The temporal data transfer procedure is expressed as follows:

$$p_n = \tanh(\omega_{fp} * \bar{H}_n^{L-1} + \omega_{hp} * \bar{H}_{n-1}^L + \beta_p) \quad (27)$$

$$q_n = \sigma(\omega_{fq} * \bar{H}_n^{L-1} + \omega_{hq} * \bar{H}_{n-1}^L + \beta_q) \quad (28)$$

$$g_n = \sigma(\omega_{fg} * \bar{H}_n^{L-1} + \omega_{hg} * \bar{H}_{n-1}^L + \beta_g) \quad (29)$$

$$\bar{C}_n^L = g_n \circ \bar{C}_{n-1}^L + q_n \circ p_n \quad (30)$$

where σ denotes the sigmoidal activation function, '*' and 'o' represent the convolution operation and the Hadamard operation correspondingly. p_n , q_n , and g_n define various tensors.

3.3.2. Spatial learning

Here, C-LSTM is utilized for extracting spatial changes in the weather data based on \bar{H}_n^{L-1} and the cell state \bar{M}_n^{L-1} . After that the spatial cell status \bar{M}_n^{L-1} of the preceding

layer is updated for generating a fresh spatial unit \bar{M}_n^L . Then the ladder self-attention (LSA) is used to give importance to the spatial data \bar{M}_n^L extracted by the C-LSTM operation. This allows the network to concentrate on the relationship between all data in the same setting as in blue dotted line container in Figure 3. The spatial data transfer procedure is expressed as follows:

$$\bar{p}_n = \tanh(\omega_{fp} * \bar{H}_n^{L-1} + \omega_{rp} * \bar{H}_n^{L-1} + \beta_p) \quad (31)$$

$$\bar{q}_n = \sigma(\omega_{fq} * \bar{H}_n^{L-1} + \omega_{rq} * \bar{H}_n^{L-1} + \beta_q) \quad (32)$$

$$\bar{g}_n = \sigma(\omega_{fg} * \bar{H}_n^{L-1} + \omega_{rg} * \bar{H}_n^{L-1} + \beta_g) \quad (33)$$

$$\bar{M}_n^L = \sigma(\omega_{fg} * \bar{H}_n^{L-1} + \omega_{rg} * \bar{H}_n^{L-1} + \beta_g) \quad (34)$$

$$= \sigma(\omega_{fg} * \bar{H}_n^{L-1} + \omega_{rg} * \bar{H}_n^{L-1} + \beta_g)^L$$

$$= \bar{g}_n \circ \bar{M}_n^{L-1} + \bar{q}_n \circ \bar{p}_n$$

$$\bar{M}_n^L = \text{LSA}(\bar{M}_n^L) \quad (35)$$

where \bar{g}_n , \bar{q}_n , and \bar{p}_n are tensors and $\text{LSA}(\cdot)$ denotes ladder self-attention operation.

3.3.3. Ladder self-attention block (LSA)

The proposed SASTNet utilizes LSA for greatly improving the system's capacity to represent global spatial information. The traditional self-attention modules are challenging to implement in edge computing devices like FPGAs that have limited memory and processing power. In order to lower the computational cost, a lightweight LSA block that models local self-attention in each branch is introduced. The LSA unit distributes the input feature map to various branches after dividing it into several equal sections along the channel dimension as in Figure 4 (a).

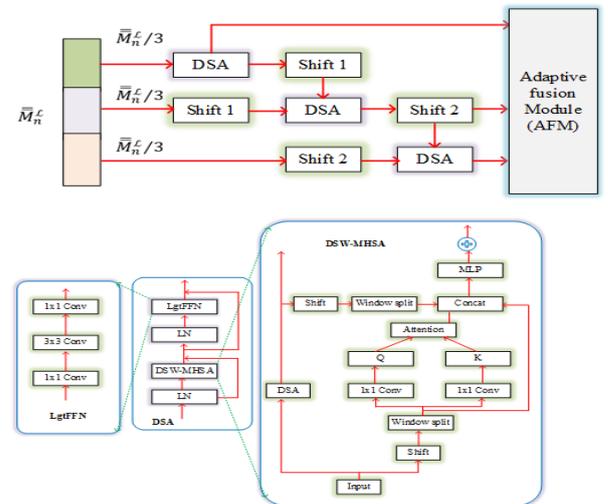


Figure 4. Self-attention block (a) LSA (b) DSA

Initially, the dynamic self-attention (DSA) with the dynamic self-windowed multi-head self-attention (DSW-MHSA) processes the input features in the 1st branch without using a shift process. The properties of the remaining two branches are changed in different ways to represent various local relations. DSW-MHSA performs 1×1 convolution on the input feature for yielding the query and key. This process is utilized for computing the resemblance between the feature points. The DSW-MHSA is used for modelling long-range dependences after delivering the output features for each branch. After modelling the local relations, the data controlled in the divided windows of the present branch is interacted with data external to the window in the subsequent branches.

This reduces the training time of the LSA unit and allows it to represent long-range relations between data in distinct windows with the aid of many branches. The LSA blocks are legitimately calculated as

$$\hat{O}_n = \text{DSW_MHSA} \left(\overset{=L}{M}_n, O_{n-1} \right) \quad (36)$$

$$\text{DSW}_{\text{MHSA}} \left(\overset{=L}{M}_n, O_{n-1} \right) = \text{softmax} \left(\frac{Q_{=L} K_{=L}}{\sqrt{c}} \right) O_{n-1} \quad (37)$$

$$\overset{=L}{M}_n = \text{LgtFFN} \left(\text{LN} \left(\hat{O}_n \right) \right) \quad (38)$$

$$\tilde{O} = \text{AFU} \left(O_n \right), \quad n=0,1 \quad (39)$$

The DSW-MHSA uses input data of n -th branch $\overset{=L}{M}_n$ and the output characteristics of the $(n-1)$ -th branch (O_{n-1}) to compute output (\hat{O}_n). Following the DSW-MHSA output computation, the Light FFN (LgtFFN) and layer norm (LN) are used for producing the outcome of the n -th branch (O_n). In the end, an adaptive fusion module (AFM) is created to produce the LSA block's output (\tilde{O}) based on the outcome of every branches. Here, LgtFFN is projecting the input with c_2 channels to a finer feature with $c_2/4$ channels by using a fully connected (FC) layer with c channels. After modeling local interactions with a depthwise convolution, the channels are restored using a pointwise convolution. At last, the AFM concatenates the outcome of every branch and transmitted to two FC layers to produce the weights for every data. The weights denote the significance of features from every branch.

3.4. Aggregation mechanism

The aggregation mechanism employs the communal output gate for smoothly merging the data stored in the temporal and spatial memories. The last hidden status relies on the spatiotemporal memory after the merging. The aggregation mechanism combines the memory data along both horizontal and vertical paths. After that, 1×1 convolutional layer is used for reducing the dimensionality. As a result, the size of hidden status \bar{H}_n^L is similar to the size of \bar{H}_n^L and \bar{H}_n^L . Here, the spatial and

temporal data is incorporated in an aggregate unit to improve the spatiotemporal weather series prediction. At last, the aggregation mechanism creates the final prediction for the subsequent SASTM unit using the subsequent formula:

$$\text{out}_n = \sigma \left(\begin{aligned} &\omega_{f0} * \bar{H}_n^{L-1} + \omega_{h0} * \bar{H}_n^L \\ &+ \omega_{v0} * C_n^L + \omega_{r0} * M_n^L + \beta_o \end{aligned} \right) \quad (40)$$

$$\bar{H}_n^L = \text{out}_n \tanh \left(\omega_{1 \times 1} * [C_n^L, M_n^L] \right) \quad (41)$$

Overall, the spatiotemporal information was retrieved simultaneously by the SASTLNet model. The SASTM module improves the global context information capturing capability and breakdowns the restriction of C-LSTM. Because C-LSTM captured local context information alone. As a result, it can improve weather forecasting ability.

4. Results and discussion

In this section, the proposed SASTLNet is validated by considering actual weather datasets from various regions. Initially, the required simulated parameters including datasets, evaluation metrics, comparison methods, and hyperparameter settings are outlined. Next, the simulated results are analysed in detail.

4.1. Dataset description

The suggested WF model is verified by collecting the data from <https://www.kaggle.com/datasets/selfishgene/historical-hourly-weather-data>. It contains the 4 meteorological variables including temperature, humidity, wind speed, and atmospheric pressure gathered from 6 cities of Israel. A 24-hour time step is used in the forecasting process. Ten percentage of the data are utilized for validation, eighty percentage are utilized for training, and the remaining data are utilized for testing in sequential order. **Table 1** provides a summary of the dataset. The temperature, wind speed, atmospheric pressure, and relative humidity of New York are displayed in Figure 5.

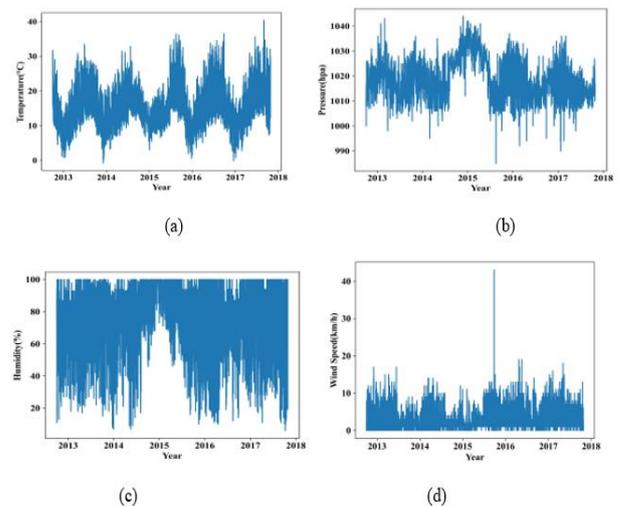


Figure 5. Visualizing the target data from dataset (a) Temperature (b) Pressure (c) Humidity (d) Wind speed

Table 1. Dataset details

Data	Description
Location	Israel
Time period	Feb. 2nd, 2012 to Oct. 28th, 2017
Time interval	1 hour
Meteorological variable	4
Weather station	6
Sample size	1850
Input length	48
Output length	24

4.2. Evaluation metrics

In this work, three widely used performance measures including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are used for validation. They can be measured using the following equations:

$$RMSE = \sqrt{\frac{1}{m} \sum_j (o_j - \hat{o}_j)^2} \quad (42)$$

$$MAE = \frac{1}{m} \sum_j |o_j - \hat{o}_j| \quad (43)$$

$$MAPE = \frac{1}{m} \sum_j \left| \frac{o_j - \hat{o}_j}{o_j} \right| \quad (44)$$

where O and \hat{O} denote the actual and forecasted data. m denote the total quantity of forecasted values. According to these measures, lower values are preferable for the best model.

4.3. Implementation details

The Python programming language has been used to simulate the suggested model. The L1 loss and L2 norm is used as the information loss and penalty term in the simulation respectively. The hyperparameter ρ should be considered before the penalty term for balancing the information loss and penalty terms. L1 and L2 losses are used to optimize the SASTLNet model. Training is done with an ADAM optimizer, with a weight decay of 0.025 and an initial learning rate of 0.001. 100 iterations are required to terminate the training procedure. Every iteration has a batch size of eight. The Nvidia Titan RTX GPUs are used for all experiments, which are built using Python 3.6 and PyTorch 1.2.0.

4.4. Evaluation of the proposed method

Initially, the performance of the proposed model (SSD-FE-E3MD-SASTLNet) is validated by comparing it with SASTLNet (core model), SSD-FE-SASTLNet, and SSD-FE-E3MD-STLNet (without self-attention) This analysis is carried out for emphasizing the significance of SSD-FE, E3MD and self-attention mechanism. The forecasting results of the proposed model and other models are shown in Figure 6. The actual and the predicted values of the weather data are indicated by solid blue lines and green lines respectively. The predicting errors of the models at each time point are shown by the brown histograms.

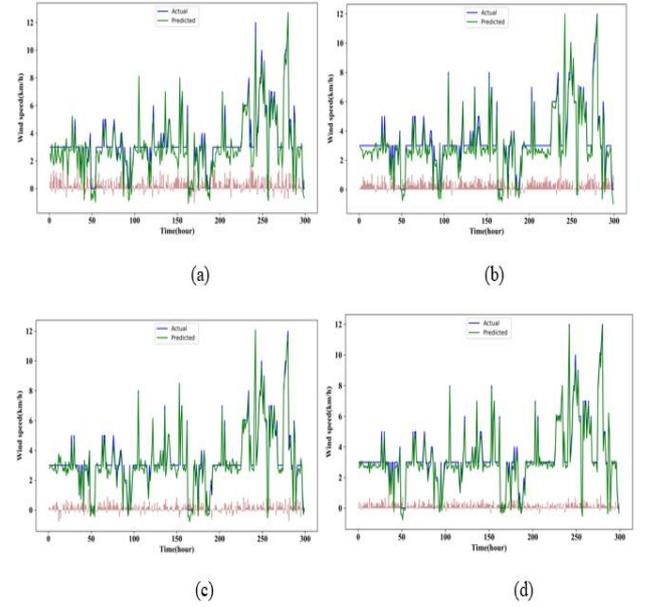


Figure 6. Forecasting results of Wind speed (a) SASTLNet, (b) SSD-FE-SASTLNet, (c) SSD-FE-E3MD-STLNet (without self-attention) and (d) SSD-FE-E3MD-SASTLNet (proposed)

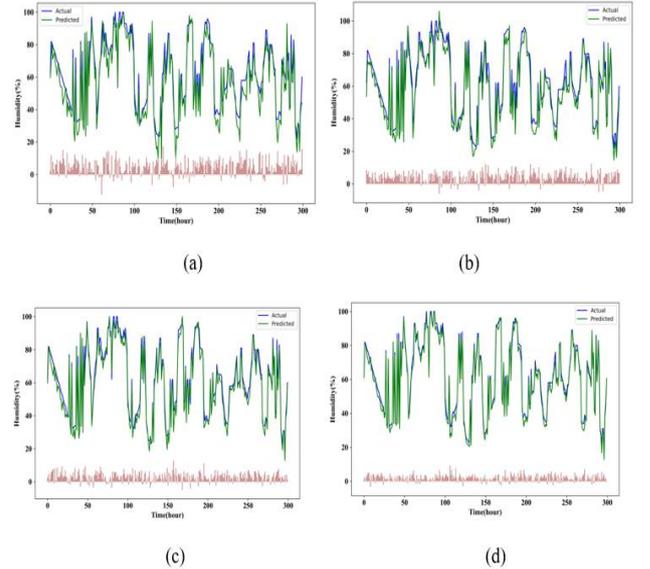


Figure 7. Forecasting results of Humidity (a) SASTLNet, (b) SSD-FE-SASTLNet, (c) SSD-FE-E3MD-STLNet (without self-attention) and (d) SSD-FE-E3MD-SASTLNet (proposed)

Table 2 presents the performance indicators such as MAE, RMSE, and MAPE for each variable and the average performance over all variables. Here, the forecasting results of SASTLNet, SSD-FE-SASTLNet, SSD-FE-E3MD-STLNet (without self-attention), and SSD-FE-E3MD-SASTLNet are compared. This table clearly shows that the predicting errors decrease when utilizing SSD-FE and E3MD decomposition. Consequently, the precision of weather forecasting is improved due to the use of multivariate data secondary decomposition techniques. Also, the SSD-FE-E3MD-SASTLNet model outperforms other models in terms of MAE and RMSE. This suggests that one may effectively capture the spatiotemporal properties of weather data through the addition of spatial dimension modeling on top of temporal dimension modeling with LSA.

Table 2. Ablation study

Methods	Metrics	Temperature	Humidity	Wind speed	Pressure	Average
SASTLNet	MAE	1.8432	6.9214	0.8264	3.8679	5.0194
	RMSE	2.9417	8.7631	1.1245	6.5267	6.2468
	MAPE (%)	6.76	12.14	29.34	0.3198	11.694
SSD-FE-SASTLNet	MAE	1.1497	6.2142	0.7849	3.1645	4.3481
	RMSE	2.1094	8.1955	0.9174	5.8364	5.9461
	MAPE (%)	5.64	11.68	28.88	0.2851	11.2481
SSD-FE-E3MD-STLNet	MAE	0.9032	5.3241	0.6719	2.7843	3.1457
	RMSE	1.9745	7.8491	0.8431	5.1248	4.5347
	MAPE (%)	4.57	10.21	26.94	0.2411	10.9423
SSD-FE-E3MD-SASTLNet	MAE	0.8712	4.3216	0.4231	1.2341	1.7125
	RMSE	0.9836	6.6314	0.6984	3.8467	3.0400
	MAPE (%)	3.94	9.83	25.31	0.142	9.8055

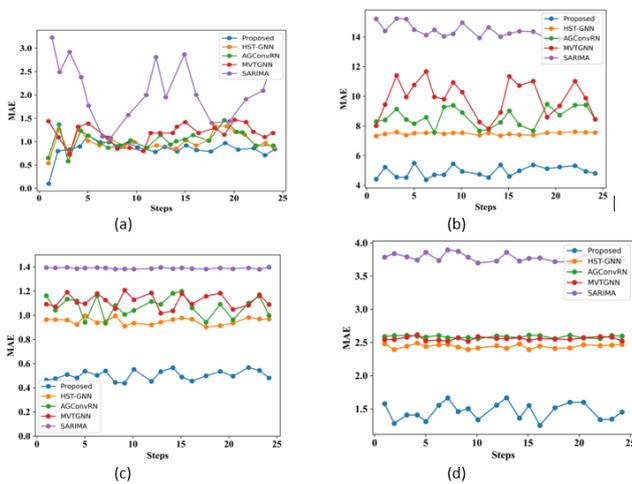


Figure 8. MAE performance analysis against baseline models (a) temperature (b) Humidity (c) Wind speed (d) Pressure

Next, the multi-time step forecasting capabilities of the baseline models and the suggested SASTLNet are thoroughly examined. Figure 8 presents a comparison between SASTLNet and other baseline models, including as SARIMA, AGConvRN [27], MVTGNN [28], and HST-GNN [31]. It shows the forecasting errors over the next 24 hours. It is observed that the HST-GNN network outperforms the conventional SARIMA model and all other models. Nevertheless, following a strong beginning, the performance of HST-GNN network is noticeably declining in terms of MAE values as compared to proposed model. Interestingly, the circle-shaped blue line surpassing the other lines indicates that SASTLNet consistently performed better than the other models and most of the multiple-time steps.

Figure 9 compare the MAPE and RMSE of the proposed model with SARIMA, AGConvRN, MVTGNN, convLSTM and HST-GNN for each weather forecasting variables including temperature, Humidity, wind speed and pressure. In comparison to other characteristics, this chart demonstrates that deep learning and conventional models do worse at forecasting humidity. Deep learning models have not effectively capturing all pertinent dependencies due to the complexity and nonlinearity of the humidity parameter. However, the techniques, such as spatio-

temporal modelling and ladder-based self-attention mechanisms. Also, its performance of humidity forecasting was improved by the suggested advances in deep learning provides reasonable performance improvements against all other meteorological variables.

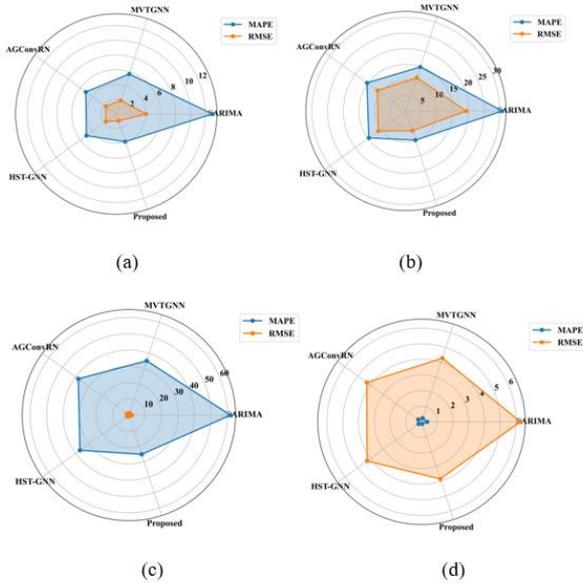


Figure 9. RMSE and MAE performance (a) temperature (b) Humidity (c) Wind speed (d) Pressure

The performance of the suggested weather prediction is compared with state-of-the-art methods in **Table 2**. Here, the traditional statistical univariate time series forecasting technique, i.e. Seasonal Autoregressive Integrated Moving Average (SARIMA) estimates future values by combining the autoregressive, differencing, moving average, and the three seasonal components. On the other hand, it may be difficult for them to cope with highly erratic or non-linear seasonal variations often seen in weather data, including sudden changes in pressure or temperature. Hence it achieves least performance as compared to deep learning models. Compared to conventional SARIMA, deep learning techniques like MVTGNN [28], AGConvRN [27], HST-GNN [31], and SASTLNet have a number of benefits. When it comes to managing the transition between

meteorological variables, MVTGNN and AGCRN use flat graph topologies. Hence, they perform worse than HST-GNN and SASTLNet. Furthermore, well pre-processed and organized spatiotemporal data such as graph representations of temporal sequences and spatial interactions are frequently needed for HST-GNNs. Model performance may be impacted by graph representations that are not properly structured. However, the proposed

model does not require such complex representation and uses lightweight ladder self-attention to learn the most significant information required for weather forecasting. Hence, it outperforms all other models by achieving average MAE of 1.7125, RMSE of 3.0400 and MAPE of 3.0400%.

Methods	Metrics	Temperature	Humidity	Wind speed	Pressure	Average
SARIMA	MAE	3.1419	15.8560	1.3799	3.6863	6.0160
	RMSE	4.0347	19.2929	1.7756	6.2767	7.8450
	MAPE (%)	12.98	30.76	61.85	0.36	26.49
MVTGNN	MAE	1.4483	7.5238	1.0138	2.5149	3.1252
	RMSE	1.9290	11.2421	1.4185	4.2898	4.7198
	MAPE (%)	5.69	14.87	35.13	0.25	13.99
AGConvRN	MAE	1.2644	7.5966	0.9105	2.5471	3.0770
	RMSE	1.7461	11.2421	1.4185	4.2898	4.7198
	MAPE (%)	5.07	15.37	38.00	0.25	14.67
HST-GNN	MAE	1.2551	7.2302	0.9018	2.3911	2.9446
	RMSE	1.7287	10.9434	1.3038	4.2666	4.5606
	MAPE (%)	4.98	14.66	36.73	0.23	14.15
SASTLNet	MAE	0.8712	4.3216	0.4231	1.2341	1.7125
	RMSE	0.9836	6.6314	0.6984	3.8467	3.0400
	MAPE (%)	3.94	9.83	25.31	0.142	9.8055

5. Conclusion

This study developed a unique hybrid weather forecasting model using a multivariate data secondary decomposition approach and deep learning algorithm to increase the precision and dependability of weather forecasting. The suggested model uses SSD-FE to filter out some noise components and extract significant data from a multivariate meteorological series. The denoised multivariate series is then broken down into corresponding IMFs and residuals using various frequencies by E3MD. SASTLNet then extracts characteristics related to correlation in both the spatial and temporal domains. In particular, the SASTM module makes sure that the memory states reflect the summarized spatial properties in the horizontal path and passed via the time states in the vertical path. A progressive shift approach is introduced to model long-range dependences using local self-attention on every branch and interrelating between these branches. This improves computation efficiency and expands the receptive field of the LSA unit. The findings indicate that the suggested model outperforms the other baseline models in terms of generalization and forecasting accuracy. The limitations of the work is could focus on optimizing the computational efficiency of the model, potentially through techniques like model pruning, parallelization, or leveraging specialized hardware like GPUs and TPUs to reduce inference time.

References

Alqahtani F., Abotaleb M., Subhi A.A., El-Kenawy E.S., Abdelhamid A.A. Alakkari K., Badr A., Al-Mahdawi H. K., Ibrahim A. and Kadi A. (2023). A hybrid deep learning model for rainfall in the wetlands of southern Iraq. *Modeling Earth Systems and Environment* **9**(4), 4295–4312.

- Bai L., Yao L., Li C., Wang X. and Wang C. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems* **33**, 17804–17815.
- Christoforou E., Emiris I.Z., Florakis A., Rizou D. and Zaharia S. (2023). Spatio-temporal deep learning for day-ahead wind speed forecasting relying on WRF predictions. *Energy Systems* **14**(2), 473–493.
- Dewitte S., Cornelis J.P., Müller R. and Munteanu A. Artificial intelligence revolutionises weather forecast, climate monitoring and decadal prediction. *Remote Sensing* **13**(16), 3209.
- Donadio L., Fang J. and Porté-Agel F. (2021). Numerical weather prediction and artificial neural network coupling for wind energy forecast. *Energies* **14**(2), 338.
- Dotse, Sam-Quarcoo, Isaac Larbi, Andrew Manoba Limantol, and Liyanage C. De Silva. A review of the application of hybrid machine learning models to improve rainfall prediction. *Modeling Earth Systems and Environment* **10**(1), 19–44.
- Fang W., Xue Q., Shen L. and Sheng V.S. (2021). Survey on the application of deep learning in extreme weather prediction. *Atmosphere* **12**(6), 661.
- Goel A., Goel A.M. and Kumar A. (2023). The role of artificial neural network and machine learning in utilizing spatial information. *Spatial Information Research* **31**(3), 275–285.
- Golyandina N. (2020). Particularities and commonalities of singular spectrum analysis as a method of time series analysis and signal processing. *Wiley Interdisciplinary Reviews: Computational Statistics* **12**(4), e1487.
- Han Y., Mi L., Shen L., Cai C.S., Liu Y., Li K. and Xu G. (2022). A short-term wind speed prediction method utilizing novel hybrid deep learning algorithms to correct numerical weather forecasting. *Applied Energy* **312**, 118777.
- He R., Zhang L. and Chew A.W.Z. (2022). Modeling and predicting rainfall time series using seasonal-trend decomposition and machine learning. *Knowledge-Based Systems* **251**, 109125.

- Jaseena K.U. and Kovoov B.C. (2022). Deterministic weather forecasting models based on intelligent predictors: A survey. *Journal of King Saud University-Computer and Information Sciences*, **34**, (6), 3393–3412.
- Jin G., Liang Y., Fang Y., Shao Z., Huang J., Zhang J. and Zheng Y. (2023). Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*.
- Liu Y., Qin H., Zhang Z., Pei S., Jiang Z., Feng Z. and Zhou J. (2020). Probabilistic spatiotemporal wind speed forecasting based on a variational Bayesian deep learning model. *Applied Energy* **260**, 114259.
- Liqin W., Ma S. and Lyu S. (2024). The influence of internet celebrity anchors' reputation on consumers' purchase intention in the context of digital economy: from the perspective of consumers' initial trust, *Applied Economics*.
- Mani J., Shaker H. and Jovanovic L. (2023). Sunspot occurrence forecasting with metaheuristic optimized recurrent neural networks. *Theoretical and Applied Computational Intelligence* **1**(1), 15–26.
- Markovics D. and Mayer M.J. (2022). Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction. *Renewable and Sustainable Energy Reviews* **161**, 112364.
- Ma M., Xie P., Teng F., Wang B., Ji S., Zhang J. and Li T. (2023). HiSTGNN: Hierarchical spatio-temporal graph neural network for weather forecasting. *Information Sciences* **648**, 119580
- Mayer M.J. and Yang D. (2023). Pairing ensemble numerical weather prediction with ensemble physical model chain for probabilistic photovoltaic power forecasting. *Renewable and Sustainable Energy Reviews* **175**, 113171.
- Periasamy S., Subramanian P. and Surendran R. (2024). An intelligent air quality monitoring system using quality indicators and transfer learning based lightweight recurrent network with skip connection.
- Ren X., Li X., Ren K., Song J., Xu Z., Deng K. and Wang X. (2021). Deep learning-based weather prediction: a survey. *Big Data Research* **23**, 100178.
- Santhanaraj R.K., Rajendran S., Romero C.A.T. and Murugaraj S.S. (2023), Internet of Things Enabled Energy Aware Metaheuristic Clustering for Real Time Disaster Management. *Computer Systems Science and Engineering*, **45**, 1561–1576.
- Sharadga H., Hajimirza S. and Balog R.S. (2020). Time series forecasting of solar power generation for large-scale photovoltaic plants. *Renewable Energy* **150**, 797–807.
- Shenglin M., Liqin W. and Yuan Y. (2024). Study on the coupled and coordinated development of tourism, urbanization and ecological environment in Shanxi Province, *Global NEST Journal*, **26**(4). <https://doi.org/10.30955/gnj.005907>.
- Surendran R., Tamilvizhi T. and Lakshmi S. (2021), Integrating the Meteorological Data into a Smart City Service Using Cloud of Things (CoT). In *Emerging Technologies in Computing: 4th EAI/IAER International Conference, iCETiC 2021, Virtual Event, August 18–19, 2021, Springer International Publishing*, **4**, 94–111.
- Surendran R., Alotaibi Y. and Subahi A.F. (2023), Lens- Oppositional Wild Geese Optimization Based Clustering Scheme for Wireless Sensor Networks Assists Real Time Disaster Management. *Computer Systems Science and Engineering*, **46**, 835–851.
- Surendran R., Alotaibi Y. and Subahi A.F. (2023), Wind Speed Prediction Using Chicken Swarm Optimization with Deep Learning Model. *Computer Systems Science and Engineering*, **46**, 3.
- Suleman M.A.R. and Shridevi S. (2022). Short-term weather forecasting using spatial feature attention based LSTM model. *IEEE Access* **10**, 82456–82468.
- Venkatachalam K., Trojovský P., Pamucar D., Bacanin N. and Simic V. (2023). DWFHs: An improved data-driven deep weather forecasting hybrid model using Transductive Long Short Term Memory (T-LSTM). *Expert systems with applications* **213**, 119270.
- Wang Z., Wu Q. and Ma S. (2024). Research on Carbon Emission Peaks in Large Energy Production Region in China —Based on the Open STIRPAT Model, *Global NEST Journal*, **26**(5).
- Weerakody P.B., Wong K.W., Wang G. and Ela W. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing* **441**, 161–178.
- Wen L., Ma S., Zhao G. and Liu H. (2024). The Impact of Environmental Regulation on the Regional Cross-Border E-Commerce Green Innovation: Based on System GMM and Threshold Effects Modeling. *Polish Journal of Environmental Studies*. <https://doi.org/10.15244/pjoes/187118>
- Wen L., Ma S., Wang C., Dong B. and Liu H. (2024). A Study of Green Strategy Choice and Behavioral Evolution of Consumers and Producers under the Double Subsidy Policy. *Polish Journal of Environmental Studies*.
- Wu Q., Jin Y. and Ma S. (2024). Impact of dual pilot policies for low-carbon and innovative cities on the high-quality development of urban economies, *Global NEST Journal*, **26**(9). <https://doi.org/10.30955/gnj.06307>.
- Xiao L., Wei S., Fulong J. and Wu Z. (2021). A self-adaptive kernel extreme learning machine for short-term wind speed forecasting. *Applied Soft Computing* **99**, 106917.
- Yadav H. and Thakkar A. (2024). NOA-LSTM: An efficient LSTM cell architecture for time series forecasting. *Expert Systems with Applications* **238**, 122333.
- Yang D., Wang W. and Hong T. (2022). A historical weather forecast dataset from the European Centre for Medium-Range Weather Forecasts (ECMWF) for energy forecasting. *Solar Energy* **232**, 263–274.
- Zhang G., Yang D., Galanis G. and Androulakis E. (2022). Solar forecasting with hourly updated numerical weather prediction. *Renewable and Sustainable Energy Reviews* **154**, 111768.
- Zou F., Ma S., Liu H., Gao T. and Li W. (2024). Do Technological Innovation and Environmental Regulation Reduce Carbon Dioxide Emissions? Evidence from China, *Global NEST Journal*, **26**(7). <https://doi.org/10.30955/gnj.06291>.