Journal of Data Science and Modeling, Vol. 3, No. 1, 1-28, December 2024

Research Manuscript

Applying the Modified Sinc Neural Network for Weather Forecasting

Ghasem Ahmadi*

Department of Mathematics, Payame Noor University, Tehran, Iran.

Recieved: 15/03/2025 Accepted: 17/07/2025

Abstract:

Accurate weather prediction plays a vital role in many sectors, such as agriculture, disaster preparedness, transportation systems, and urban planning. Traditional meteorological models face challenges in capturing complex atmospheric dynamics, leading to an increased reliance on artificial neural networks (ANNs) for improved forecasting accuracy. ANNs have been widely applied in meteorology due to their ability to model nonlinear relationships and temporal dependencies. Based on Sinc numerical methods, the modified Sinc neural network (MSNN) has been introduced recently. This model leverages the advantages of the Sinc function, such as smoothness and oscillatory behavior, while enhancing the ability to model nonlinear dependencies and temporal dynamics in environmental data. This work utilizes the MSNN for time series forecasting, with its parameters adjusted using a discrete-time online Lyapunov-based learning algorithm. The model is then applied to enhance weather forecasting. It is evaluated on datasets containing various meteorological components. The data used in this study pertains to the city of Khorramabad in Iran. The results show that, despite its simple structure, the MSNN demonstrates high efficiency in weather forecasting.

Keywords: Weather forecasting, time series forecasting, Sinc neural network.

Mathematics Subject Classification (2010): 68T07, 37M10.

*Corresponding Author: g.ahmadi@pnu.ac.ir

1. Introduction

Weather forecasting plays a crucial role in agriculture, disaster management, transportation, and daily life. Accurate predictions allow for better preparedness for extreme weather events such as storms, floods, and droughts, thereby reducing economic losses and protecting lives. Traditionally, numerical weather forecasting models have been the primary tools in this context, relying on physical and mathematical equations that describe atmospheric dynamics (Bauer *et al.* , 2015). Over the years, advancements in computational methods and data-driven models have significantly improved forecasting accuracy. However, traditional meteorological models often struggle with the complexity of atmospheric dynamics, require substantial computational resources, and face limitations in real-time applications (Ren and *et al.* , 2021). Consequently, there has been a growing shift among researchers toward leveraging machine learning methods—especially artificial neural networks (ANNs)—to enhance the accuracy and efficiency of weather forecasting.

ANNs are inspired by the human nervous system and consist of computational neurons connected via weighted edges (Gupta and *et al.* , 2003). By adjusting these weights using learning rules, ANNs can model and approximate complex nonlinear functions. The architecture of ANNs can vary widely, from simple feedforward networks—where information moves in one direction from input to output—to more sophisticated structures such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs). By harnessing these principles, ANNs have become fundamental components of modern machine learning and artificial intelligence, enabling the solution of problems once deemed unsolvable. Their capacity to learn from data and generalize to new, unseen scenarios makes them indispensable tools in today's world.

The use of ANNs in weather forecasting has attracted considerable interest because of their ability to capture complex nonlinear relationships within meteorological data (Ren and *et al.* , 2021). Unlike conventional statistical models, ANNs adapt to changing environmental conditions and uncover intricate patterns in historical weather data. Neural networks have demonstrated remarkable success in various meteorological applications, particularly in modeling nonlinear relationships in time series data.

One promising approach in the context of weather forecasting is the Sinc neural network (SNN), which employs Sinc functions as activation functions instead of traditional sigmoidal or rectified linear unit (ReLU) functions. The Sinc function has unique properties that make it well-suited for function approximation and signal processing tasks. Leveraging these characteristics, SNNs have shown improved performance in various applications.

While deep and complex neural networks have achieved remarkable results in many domains, simpler neural models remain highly relevant, especially when dealing with limited datasets (Goodfellow and *et al.* , 2016). Deep networks require vast amounts of data to generalize effectively; otherwise, they risk overfitting and poor performance on unseen data. In contrast, simpler models such as SNNs and traditional multilayer perceptrons (MLPs) offer better interpretability, faster training times, and reduced computational costs, making them more suitable when data is scarce. In many practical forecasting scenarios, especially in localized regions like Khorramabad, access to extensive, high-quality datasets is limited. Therefore, leveraging simpler yet efficient neural architectures can yield competitive results without the need for excessive computational resources or complex hyperparameter tuning.

The aim of this study is to assess the effectiveness of the modified Sinc neural network (MSNN) in forecasting weather components in Khorramabad. First, we outline the mathematical framework for time series forecasting using MSNN and employ a discrete-time Lyapunov-based learning algorithm to train the network. Then, we apply the proposed method to predict various weather elements.

The approach is evaluated using datasets that include multiple meteorological parameters such as minimum air temperature ($T_{\min}$), maximum air temperature ($T_{\max}$), minimum relative humidity ($RH_{\min}$), maximum relative humidity ($RH_{\max}$), wind speed (WS), wind direction (WD), rainfall (RA), and atmospheric pressure (QFE). Its performance is compared to that of an MLP, with a particular emphasis on forecasting accuracy and computational efficiency. The datasets used in this study were obtained from a meteorological station located in Khorramabad, Iran.

The contributions of this work are summarized as follows:

- The MSNN is employed within a systematic framework for time series forecasting.
- A discrete-time Lyapunov-based online learning algorithm is used to train the MSNN, providing stable and efficient convergence for nonlinear time series prediction.
- The MSNN is applied to forecast meteorological components (temperature, humidity, wind speed and direction, rainfall, and pressure) using real-world data from a station in Khorramabad, Iran—a region where high-resolution datasets are limited.
- Comparative analysis demonstrates that MSNN achieves better forecasting performance and computational efficiency compared to traditional MLP models. Additionally, the network shows acceptable performance and improved computational efficiency relative to LSTM networks.

- As a lightweight alternative to deep architectures, the MSNN offers a viable solution for localized and resource-constrained forecasting applications.

The remainder of this paper is structured as follows: related works are described in Section 2. Section 3 discusses the architecture of MSNN. Section 4 explains its application in time series prediction. Section 5 addresses the challenges of weather forecasting. Section 6 presents the implementation of MSNN for weather forecasting. Finally, Section 7 concludes the paper.

2. Related Works

The historical development of Sinc-based neural networks can be traced back to early research on wavelet networks and function approximation methods, where Sinc functions were recognized for their superior interpolation capabilities (Stenger , 1993). The Sinc function's inherent band-limited properties make it particularly effective in signal processing and function approximation tasks. Recent studies indicate that modifications to the SNN structure further enhance its predictive power, offering a viable alternative to conventional ANN models in meteorological forecasting (Ahmadi , 2024).

Research on ANNs employing the Sinc activation function remains limited but promising. Elwasif and Fausett (Elwasif and Fausett , 1996) utilized SNNs with a single input and output to approximate single-variable functions, demonstrating the function's efficacy in nonlinear approximation. More recent advances include Sinc-based convolutional neural networks applied to biomedical signal decoding (Borra *et al.* , 2020), showing robustness in processing time-dependent signals. Applications of SNNs span EEG-based brain-computer interface (BCI) motor imagery classification (Bria *et al.* , 2021), automatic speaker and age identification (Radha *et al.* , 2024), EEG motor imagery classification (Liu *et al.* , 2023), and human motion recognition (Biswas *et al.* , 2023), highlighting the broad adaptability of Sinc-based architectures. Recently, SNNs have also been used to address fractional optimal control problems (Heydari and Ahmadi , 2024).

To address inherent limitations of classical SNNs, Ahmadi (2024) proposed the MSNN, which refines the architecture to better capture nonlinear dependencies and temporal dynamics characteristic of environmental data. MSNN preserves the beneficial properties of the Sinc function—such as smoothness and band-limitation—while improving modeling flexibility and reducing computational costs. Its ability to incorporate frequency-domain information makes MSNN particularly suitable for time series forecasting and signal processing tasks relevant to weather prediction.

Neural networks have extensive applications in time series forecasting (Faruk

, 2010). Recently, deep learning models have been employed for multi-step-ahead time series forecasting (Chandra *et al.* , 2021). Novel research has also proposed hybrid approaches integrating dynamical systems, signal processing, and neural networks for time series forecasting (Azizi , 2024). Additionally, several modern neural architectures—such as transformer-based and attention-driven deep learning models—have been applied to time series, especially for weather forecasting. These methods have been reviewed in recent studies by Li and Eddie (2024), Kim et al. (2024), and Conti (2024).

The broader use of neural networks in weather forecasting has been extensively studied. Early implementations of neural networks in meteorology date back to the late 20th century, with models such as MLPs being used for temperature and precipitation forecasting (Hsieh and Tang , 1998), establishing neural networks as effective tools for meteorological time series. Recurrent neural networks (RNNs) and their gated variants, such as long short-term memory (LSTM) networks, have been widely used to capture temporal dependencies in weather data, yielding improved forecasts for components like rainfall and wind speed (Elsaraiti and Merabet , 2021; Ouma *et al.* , 2022). Convolutional neural networks (CNNs) have also been adapted for spatiotemporal weather prediction, especially in short-term rainfall nowcasting (Shi *et al.* , 2015).

Recent advancements in Iran have demonstrated the efficacy of neural network models for precipitation and hydrological forecasting. Taie Semiromi and Koch (2024) introduced a hybrid wavelet transform–artificial neural network–quantile mapping approach to downscale daily precipitation in the semi-arid Gharehsoo River Basin, achieving notable accuracy in capturing wet/dry spell dynamics under multiple scenarios. Moradpoor *et al.* (2023) implemented advanced AI models, including ANN frameworks, for spatiotemporal monthly rainfall modeling in Ilam Province, showing strong agreement between simulated and observed values. Barzegari Banadkooki *et al.* (2019) compared MLP and support vector machine (SVM) models optimized via a flow regime algorithm for precipitation forecasting, finding that MLP outperformed SVM, with quantified uncertainty estimates. Khoramabadi and Moradinia (2024) used a wavelet neural network to predict precipitation changes in the Aji-Chay watershed.

There are few studies specifically focused on weather forecasting in Khorramabad. Khosravi (2017) utilized fuzzy neural networks and genetic algorithms to forecast temperature in the city. Iranshahi *et al.* (2023) applied neural networks to analyze the effects of climate change on temperature and precipitation in Khorramabad. Veyskarami *et al.* (2024) used metaheuristic models to estimate the daily evaporation rate in Lorestan Province. More recently, the author has used neural networks for air pollution and wind speed forecasting in Khorramabad

(Ahmadi and Akbari , 2024; Ahmadi , 2024).

In summary, while deep learning dominates recent advances in weather forecasting, the MSNN presents a novel, computationally efficient approach well-suited to localized forecasting problems with limited data. Its frequency-domain integration and reduced parameter requirements position MSNN as a promising tool for improving meteorological time series predictions in Khorramabad and similar regions.

3. Modified Sinc Neural Network (MSNN)

This section focuses on the architecture and training methodology of the Modified Sinc Neural Network (MSNN).

3.1 Sinc Function

The MSNN employs the mathematically elegant Sinc basis functions as activation functions in the hidden layer. The Sinc function is commonly encountered in signal processing and Fourier transform theory and is sometimes referred to as the sampling function. In this study, we utilize the normalized Sinc function, defined as follows:

$$Sa(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0, \\ 1, & x = 0. \end{cases} \tag{3.1}$$

The Sinc function plays a significant role in various domains of applied sciences, including Sinc interpolation and Sinc-based numerical methods, yielding valuable results in solving nonlinear problems (Stenger , 2010). As the inverse Fourier transform of the rectangular function, it is fundamental in signal processing, enabling analysis of the relationship between signals in both time and frequency domains (Schanze , 1995).

This function exhibits a smooth, oscillatory behavior, alternating between positive and negative values. As the input approaches infinity, its output converges to zero. It is also central to the formulation of the Whittaker cardinal function, an essential concept in approximation theory.

Let

$$Sa_i(x) = Sa\left(\frac{1}{h}(x - ih)\right), \qquad i = -\infty, \dots, \infty, \tag{3.2}$$

be the i-th function in the collection of Sinc basis functions, where h is a positive real number. For a real-valued function f, the Whittaker cardinal function is defined as:

$$C(f,h)(x) = \sum_{i=-\infty}^{\infty} f(ih)Sa_i(x). \tag{3.3}$$

Notably, Equation (3.3) interpolates the function f at the discrete points ih for all integers i.

In the context of the complex ω-plane, the strip domain D_d is defined as (Eftekhari and Saadatmandi , 2021):

$$D_d = \{\omega = t + is : |s| < d\}. \tag{3.4}$$

When addressing problems on a subinterval I of $\mathbb{R}$, a conformal map ϕ is used such that $\phi(I) = \mathbb{R}$. Let ϕ be a conformal map, and let ζ denote its inverse, mapping the simply connected domain D—which contains the interval $(0, 1)$—onto D_d. Consequently, for the subinterval $I = (0, 1) = \zeta(\mathbb{R})$, where $\phi(0) = -\infty$ and $\phi(1) = +\infty$, the function $f(x)$ can be approximated by:

$$f(x) \approx \sum_{i=-N}^{N} f(x_i)Sa_i(\phi(x)), \tag{3.5}$$

where $x_i = \zeta(ih)$.

3.2 The Structure of MSNN

There is limited research focused on neural networks that utilize the Sinc activation function. A significant early contribution was made by Elwasif and Fausett in 1996, who used a single-input, single-output Sinc Neural Network (SNN) for function approximation with a single variable (Elwasif and Fausett , 1996). More recently, Heydari and Ahmadi extended the SNN framework to address fractional optimal control problems (Heydari and Ahmadi , 2024).

Building upon the SNN architectures presented in (Elwasif and Fausett , 1996; Heydari and Ahmadi , 2024) and incorporating the Sinc basis functions defined in equations (3.2), (3.3), and (3.5), the Modified Sinc Neural Network (MSNN) was introduced by Ahmadi (Ahmadi , 2024).

The standard SNN structure consists of three layers. Notably, the connections between the input and hidden layers do not involve trainable parameters; these weights are fixed, typically set to 1. Trainable parameters in the SNN include the weights between the hidden and output layers and the output biases.

MSNN extends this structure by introducing additional direct connections from the input layer to the output layer, each associated with trainable weights, as illustrated in Figure 1. These additional parameters enhance the network's flexibility and its capacity to model complex nonlinear relationships.

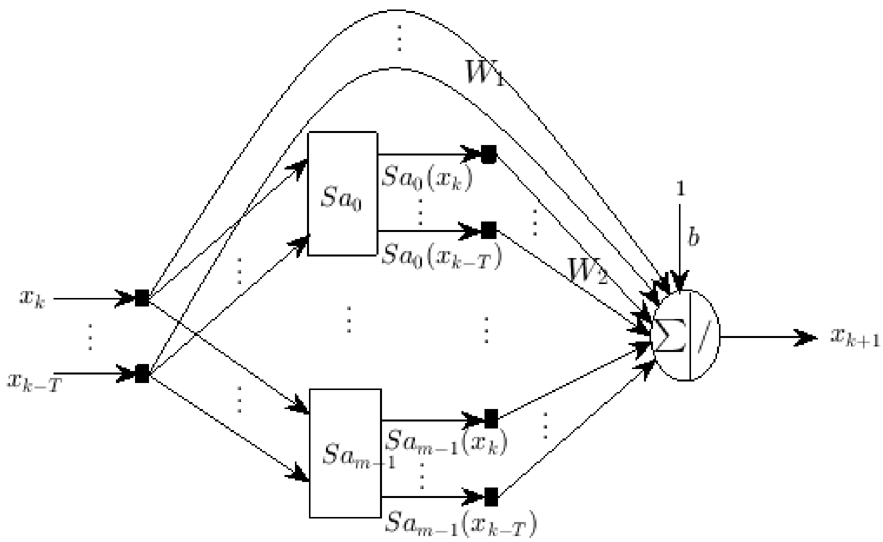

Figure 1: The structure of MSNN in time series prediction.

Let

$$X = [x_1, x_2, \ldots, x_n]^T, \quad Y = [y_1, y_2, \ldots, y_q]^T$$

denote the input and output vectors of the MSNN, respectively. Let $S_i(t)$, $(i = 0, 1, 2, \ldots, m-1)$ be the i-th function in the set of Sinc basis functions. Define:

$$\begin{aligned} Sa_i(X) &= [Sa_i(x_1), Sa_i(x_2), \ldots, Sa_i(x_n)]^T, \quad i = 0, 1, \ldots, m-1, \\ \mathbf{Sa}(X) &= \begin{bmatrix} Sa_0(X) \\ Sa_1(X) \\ \vdots \\ Sa_{m-1}(X) \end{bmatrix}. \end{aligned} \tag{3.6}$$

Let W_1 denote the weight matrix for the direct input-output connections, W_2 the weights from the hidden layer to the output, and $\mathbf{b}$ the bias vector for the output layer. Then:

$$\begin{aligned} Y &= W_1 X + W_2 \mathbf{Sa}(X) + \mathbf{b} \\ &= \begin{bmatrix} W_1 & W_2 & \mathbf{b} \end{bmatrix} \begin{bmatrix} X \\ \mathbf{Sa}(X) \\ 1 \end{bmatrix} \\ &= W\mathbf{S}(X), \end{aligned} \tag{3.7}$$

where

$$W = \begin{bmatrix} W_1 & W_2 & \mathbf{b} \end{bmatrix}, \quad \mathbf{S}(X) = \begin{bmatrix} X \\ \mathbf{Sa}(X) \\ 1 \end{bmatrix}. \tag{3.8}$$

Figure 1 illustrates the schematic of MSNN for time series prediction.

Remark 3.1. *The MSNN architecture is based on a set of Sinc basis functions, which differ from the standard activation functions (e.g., sigmoid, ReLU) commonly used in neural networks. These basis functions are particularly well-suited for capturing band-limited or oscillatory features found in real-world signals, such as those in meteorological and engineering contexts.*

Remark 3.2. *MSNN employs a shallow, single-hidden-layer architecture. Despite its simplicity, this structure provides sufficient representational capacity due to the richness of the Sinc basis, enabling effective approximation of smooth and periodic signals with computational efficiency and interpretability.*

Remark 3.3. *Each neuron in the MSNN is parameterized by a center and a scaling factor. The centers define the location of the Sinc basis functions, while the scaling factors determine their width, allowing the network to adaptively capture localized features in the input time series.*

3.3 Training of MSNN

By applying a suitable learning algorithm, the neural network parameters can be optimized and the trained MSNN can be used for time series forecasting. An online learning algorithm based on stochastic gradient descent has been proposed in (Ahmadi , 2024). In this study, we employ a discrete-time Lyapunov-based stability method to design an online training algorithm for MSNN.

In Lyapunov-based learning algorithms, parameter updates are guided by Lyapunov stability theory. These algorithms avoid local minima by using an energy function with a global minimum. A candidate Lyapunov function v_k is selected, and the parameters are updated to ensure $\Delta v_k < 0$. Consequently, the modeling error asymptotically converges to zero (Ahmadi and Teshnehlab , 2017).

Let Y_k be the output of the target system at time step k corresponding to input X_k, and let $\widehat{Y}_k$ be the output of the MSNN. The modeling error is then defined as:

$$E_k = Y_k - \widehat{Y}_k. \tag{3.9}$$

Following the approach in (Ahmadi , 2022), the learning algorithm is given by:

$$\Delta W_k = \Gamma E_k \mathbf{S}(X_k)^T, \tag{3.10}$$

where Γ is the learning rate matrix.

Assume that the MSNN with optimal parameters $W_\star$ satisfies:

$$Y_k = W_\star \mathbf{S}(X_k). \tag{3.11}$$

Let W_k be the current estimate of $W_\star$. Then:

$$\begin{aligned} \widehat{Y}_k &= W_k \mathbf{S}(X_k), &(3.12)\\ E_k &= W_\star \mathbf{S}(X_k) - W_k \mathbf{S}(X_k) = \widetilde{W}_k \mathbf{S}(X_k), &(3.13)\end{aligned}$$

where $\widetilde{W}_k = W_\star - W_k$ is the weight error.

Theorem 3.4. *Suppose the MSNN parameters are updated using equation (3.10) and that:*

$$\lambda_{\max}(\Gamma) \max_k \|\mathbf{S}(X_k)\|^2 \leq 2. \tag{3.14}$$

Then, the modeling error E_k converges to zero.

Proof. Let

$$v_k = \operatorname{tr}\left(\widetilde{W}_k^T \Gamma^{-1} \widetilde{W}_k\right)$$

be a candidate Lyapunov function. Then:

$$\begin{aligned} \Delta v_k &= v_{k+1} - v_k \\ &= 2\operatorname{tr}\left(\widetilde{W}_k^T \Gamma^{-1} \Delta \widetilde{W}_k\right) + \operatorname{tr}\left(\Delta \widetilde{W}_k^T \Gamma^{-1} \Delta \widetilde{W}_k\right) \\ &= -2\operatorname{tr}\left(\widetilde{W}_k^T E_k \mathbf{S}(X_k)^T\right) + \operatorname{tr}\left(\mathbf{S}(X_k) E_k^T \Gamma E_k \mathbf{S}(X_k)^T\right) \\ &= -2\operatorname{tr}\left(E_k E_k^T\right) + E_k^T \Gamma E_k \|\mathbf{S}(X_k)\|^2 \\ &\leq -2\|E_k\|^2 + \lambda_{\max}(\Gamma)\|E_k\|^2 \|\mathbf{S}(X_k)\|^2 \\ &\leq \left(\lambda_{\max}(\Gamma) \max_k \|\mathbf{S}(X_k)\|^2 - 2\right) \|E_k\|^2. \end{aligned} \tag{3.15}$$

According to condition (3.14), we have $\Delta v_k < 0$. Following the argument in Theorem 1 of (Ahmadi , 2022), it follows that

$$\lim_{k \to \infty} E_k = 0.$$

□

4. Time Series Forecasting by MSNN

Time series forecasting involves estimating future values by analyzing previously recorded data points. This technique is widely utilized in domains such as finance, meteorology, and supply chain management. In recent years, artificial neural networks (ANNs) have been increasingly employed to enhance prediction accuracy.

A time series is represented by $\{x_k\}_{k=0}^{\infty}$, where x_k is the observation at time index k. We aim to forecast x_{k+1} based on the previous values:

$$\begin{aligned} x_{k+1} &= f(x_k, x_{k-1}, \ldots, x_{k-T}) \\ &= f(X_k), \end{aligned} \tag{4.16}$$

with $X_k = [x_k, x_{k-1}, \ldots, x_{k-T}]^T$. Let A be a Hurwitz matrix (i.e., its eigenvalues lie within the unit circle). Adding and subtracting AX_k yields:

$$x_{k+1} = AX_k + g(X_k), \tag{4.17}$$

where $g(X_k) = f(X_k) - AX_k$. We approximate g using MSNN and equation (3.7):

$$x_{k+1} = AX_k + W_\star \mathbf{S}(X_k), \tag{4.18}$$

where $W_\star$ is the optimal parameter matrix. The parametric MSNN model becomes:

$$\widehat{x}_{k+1} = AX_k + \widehat{W}_k \mathbf{S}(X_k), \tag{4.19}$$

with $\widehat{x}_{k+1}$ as the forecast and $\widehat{W}_k$ the current estimate of $W_\star$ at time k.

We emphasize the strategic use of this shallow MSNN architecture. Unlike deep learning models that demand extensive data and computation, MSNNs are ideal when data are limited—as is typical at regional weather stations. Benefits include:

- Reduced risk of overfitting with fewer training samples.
- Computational efficiency and real-time training suitability in resource-constrained environments.
- Higher interpretability and stability than deep architectures, which is advantageous in sensitive domains like environmental monitoring.
- A balanced solution offering modeling power, training simplicity, and predictive performance for regional forecasting with modest data.

5. Weather Forecasting

Weather forecasting is essential across numerous sectors. While traditional models face challenges in adaptability and accuracy, ANNs provide a powerful alternative for capturing complex, nonlinear environmental dependencies.

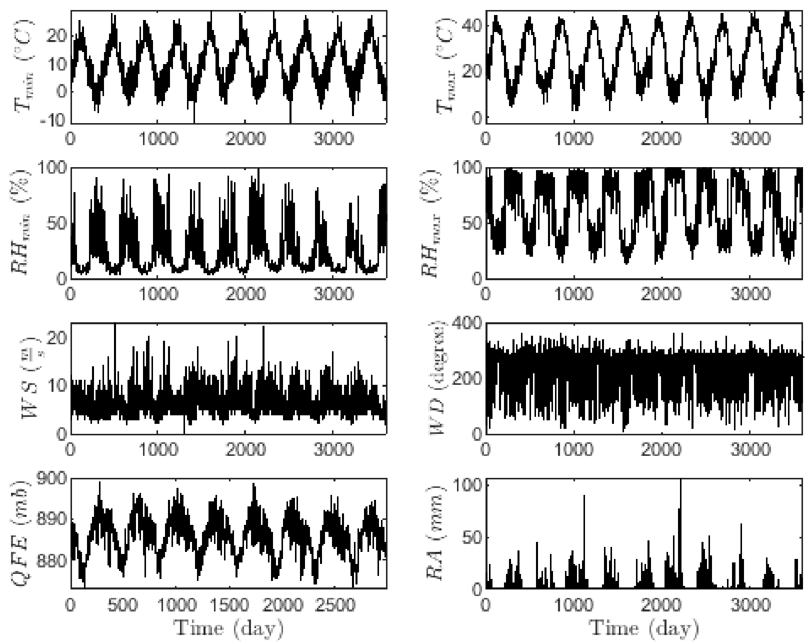

Figure 2: Daily weather records in Khorramabad from early 1392 to end of 1401 (solar Hijri calendar).

This section describes the MSNN-based weather forecasting process. We adopt a Knowledge Discovery in Databases (KDD) approach, which involves extracting actionable insights from raw weather data (Cordova *et al.* , 2021). This method supports multi-day-ahead predictions, aiding urban and resource management in Khorramabad.

The KDD pipeline comprises:

1. **Problem Definition:** We explore weather forecasting in Khorramabad (population ∼400,000), using MSNN to predict key meteorological variables.

2. **Data Collection:** Daily observations from early 1392 to end of 1401 (solar Hijri) include $T_{\min}, T_{\max}, RH_{\min}, RH_{\max}, WS, WD, RA, QFE$, as visualized in Figures 2 and 3.

3. **Data Preprocessing:** Missing values are handled via interpolation. A Butterworth filter smooths fluctuations (see Figure 4). Data are normalized to $[\alpha, \beta]$ as:
$$y_k = \alpha + (\beta - \alpha)\frac{x_k - \min\{x_k\}}{\max\{x_k\} - \min\{x_k\}},$$
where α, β are set empirically to preserve distribution characteristics.

4. **Model Training:** The MSNN is trained online using the Lyapunov-based algorithm detailed in Section 3. Hyperparameters are tuned empirically for optimal performance.

5. **Evaluation:** We assess MSNN performance against MLP and LSTM models using Mean Squared Error (MSE), Mean Absolute Error (MAE), and CPU runtime.

6. Weather Forecasting Using the MSNN

In this section, we employ the MSNN architecture to forecast meteorological variables using data from Khorramabad, Iran. The input vector for the MSNN is defined as
$$X_k = [x_k, x_{k-1}, x_{k-2}]^T.$$
In the result tables, n_h denotes the number of hidden neurons, and h is defined in equation (3.2). The optimal number of neurons is determined via grid search, balancing validation error and model complexity. We evaluated the model across a range of neuron counts and selected the configuration yielding the best predictive performance.

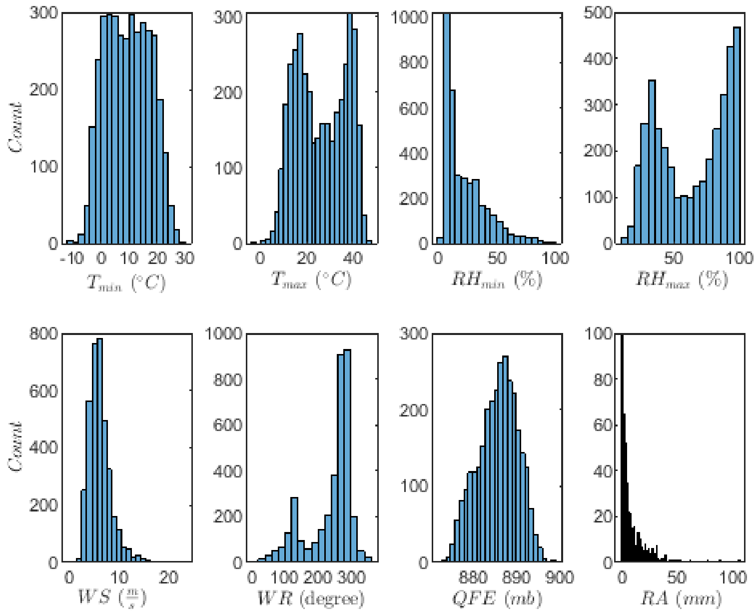

Figure 3: Histograms of daily weather data in Khorramabad (early 1392–late 1401, solar Hijri).

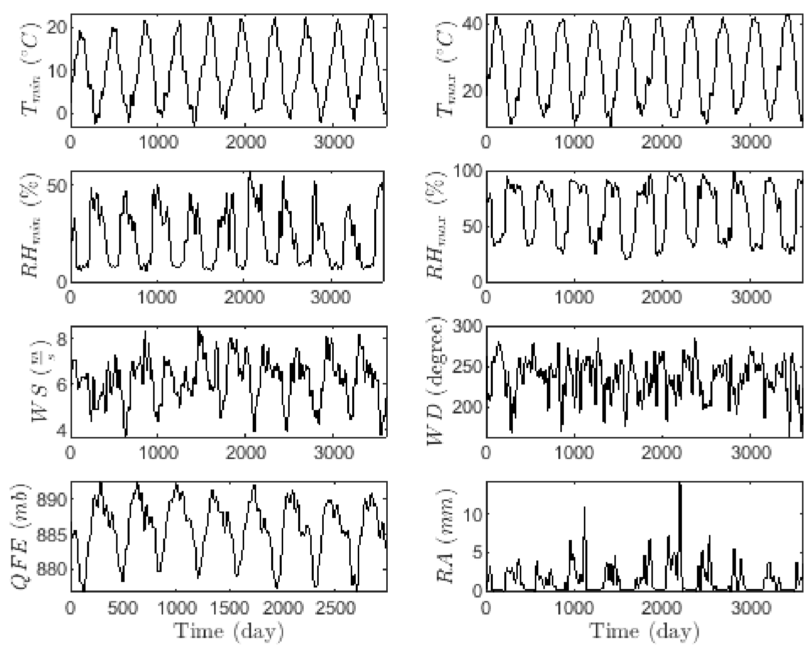

Figure 4: Preprocessed daily weather data after interpolation and filtering.

The dataset was partitioned chronologically to preserve the temporal dependencies inherent in time series data. Specifically, the first 86% of the samples were allocated for training, and the remaining 14% for testing.

The performance of the MSNN is compared against two widely used models: Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) networks. The MLP comprises three layers: an input layer, a single hidden layer with a hyperbolic tangent activation function, and an output layer with a linear activation function suitable for regression tasks. It is trained using the same online Lyapunov-based learning algorithm as the MSNN. The input vector for the MLP is identical to that of the MSNN:

$$X_k = [x_k, x_{k-1}, x_{k-2}]^T.$$

The LSTM model includes one hidden layer with 50 memory cells. Training is performed using the Adam optimizer with a learning rate of 0.005. To prevent overfitting, early stopping is applied with a patience of 20 epochs.

6.1 Forecasting of $T_{\min}$

The minimum daily temperature ($T_{\min}$) is forecasted using the three neural architectures: MLP, LSTM, and MSNN. In the MLP, the trainable parameters are initialized randomly within the interval $[-0.05, 0.05]$. For the MSNN, all adjustable parameters are initialized to zero, reflecting the differences in structural design and parameter sensitivity between the two networks.

Detailed hyperparameter configurations and corresponding performance metrics for each model are summarized in Table 1. Additionally, Figure 5 provides a comparative visualization of the forecasting results for $T_{\min}$ using the MLP and MSNN models, offering a clear depiction of their relative accuracy and generalization capabilities.

Table 1: MSEs of neural models in the forecasting of $T_{\min}$ in Khorramabad.

Model	n_h	α	β	A	Γ	Epochs	CPU time	MSE	MAE
MLP	500	-1	1	-25	$0.02I_{500}$	1	5.769	0.1826	0.4002
LSTM	50	0	1	-	0.005	50	166.822	0.1221	0.3162
MSNN	34	-5	5	-20	$0.1I_{34}$	1	4.806	**0.0565**	**0.2111**

6.2 Forecasting of $T_{\max}$

The prediction of $T_{\max}$ is performed using three distinct neural network architectures: MLP, LSTM, and MSNN. In the MLP model, the trainable parameters are initialized with random values within the range $[-5, 5]$, whereas in the MSNN, all

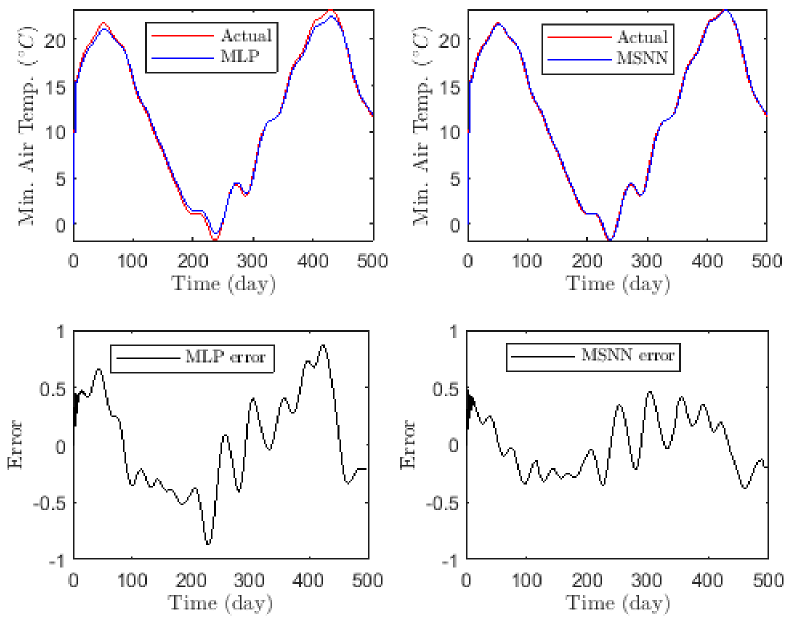

Figure 5: Implementation of MLP and MSNN to forecast the $T_{\min}$ in Khorramabad.

Table 2: MSEs of neural models in the forecasting of $T_{\max}$ in Khorramabad.

Model	n_h	α	β	A	Γ	Epochs	CPU time	MSE	MAE
MLP	85	-1	1	-35	$0.005I_{85}$	1	4.356	0.4018	0.5040
LSTM	50	0	1	-	0.005	20	121.703	0.1668	**0.3324**
MSNN	34	-3	3	-40	$0.1I_{34}$	1	4.765	**0.1118**	0.3900

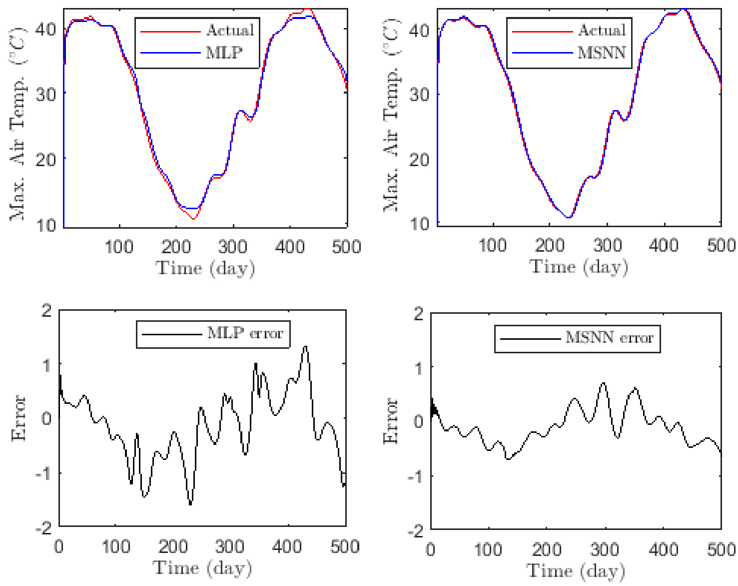

Figure 6: Implementation of MLP and MSNN to forecast the $T_{\max}$ in Khorramabad.

adjustable parameters are initialized to zero. Additional hyperparameters, along with the corresponding model outcomes, are provided in Table 2. Furthermore, a comparative visualization of the $T_{\max}$ forecasting results using MLP and MSNN is presented in Figure 6, offering a clear illustration of their performance differences.

6.3 Forecasting of $RH_{\min}$

The prediction of $RH_{\min}$ is performed using three distinct neural network architectures: MLP, LSTM, and MSNN. In the MLP model, the trainable parameters are initialized with random values within the range $[-5.5, 5.5]$, whereas in the MSNN, the adjustable parameters are initialized within the range $[-0.005, 0.005]$. Additional hyperparameters, along with the corresponding model outcomes, are provided in Table 3. Furthermore, a comparative visualization of the $RH_{\min}$ fore-

Table 3: MSEs of neural models in the forecasting of $RH_{\min}$ in Khorramabad.

Model	n_h	α	β	A	Γ	Epochs	CPU time	MSE	MAE
MLP	70	-1	1	-40	$0.01I_{70}$	1	3.795	0.6649	0.6457
LSTM	50	0	1	-	0.005	20	115.102	0.5006	0.5000
MSNN	34	-6	6	-40	$0.2I_{34}$	1	5.850	**0.2482**	**0.3599**

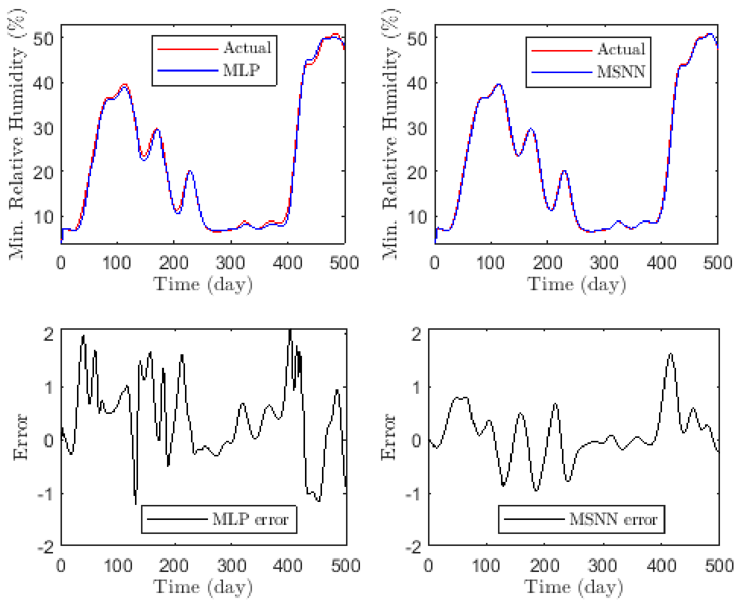

Figure 7: Implementation of MLP and MSNN to forecast the $RH_{\min}$ in Khorramabad.

casting results using MLP and MSNN is presented in Figure 7, offering a clear illustration of their performance differences.

6.4 Forecasting of $RH_{\max}$

The prediction of $RH_{\max}$ is performed using three distinct neural network architectures: MLP, LSTM, and MSNN. In the MLP model, the trainable parameters are initialized with random values within the range $[-5, 5]$, whereas in the MSNN, the adjustable parameters are initialized within the range $[-0.005, 0.005]$. Additional hyperparameters, along with the corresponding model outcomes, are provided in Table 4. Furthermore, a comparative visualization of the $RH_{\max}$ forecasting results using MLP and MSNN is presented in Figure 8, offering a clear illustration of their performance differences.

Table 4: MSEs of neural models in the forecasting of $RH_{\max}$ in Khorramabad.

Model	n_h	α	β	A	Γ	Epochs	CPU time	MSE	MAE
MLP	70	-1	1	-40	$0.01I_{70}$	1	5.835	1.5118	0.9599
LSTM	50	0	1	-	0.005	20	138.719	1.3567	0.9756
MSNN	34	-5	5	-40	$0.2I_{34}$	1	6.717	**1.1308**	**0.6815**

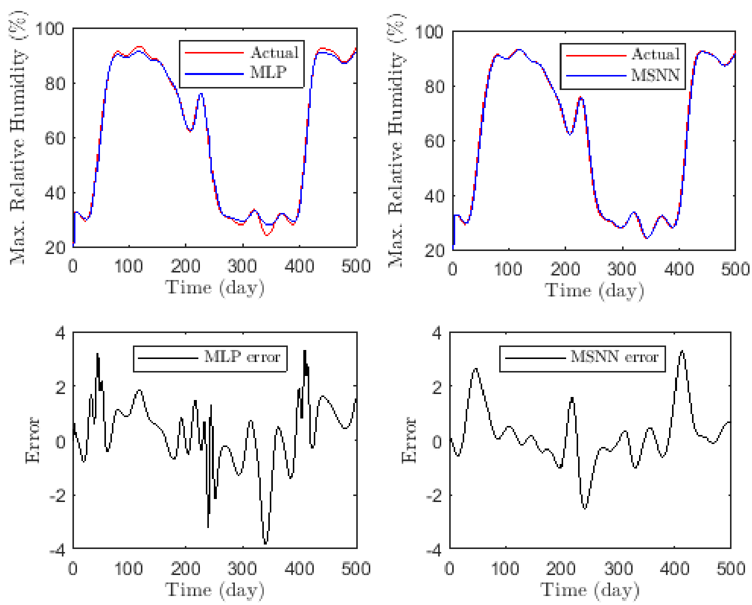

Figure 8: Implementation of MLP and MSNN to forecast the $RH_{\max}$ in Khorramabad.

6.5 Forecasting of WS

The prediction of WS is carried out using three distinct neural network architectures: MLP, LSTM, and MSNN. In the MLP model, the trainable parameters are initialized with random values within the range $[-0.025, 0.025]$, while the initialization range for the adjustable parameters in the MSNN is set to $[-0.005, 0.005]$. Additional hyperparameters, along with the corresponding model outcomes, are detailed in Table 5. Furthermore, a comparative visualization of the WS forecasting results using MLP and MSNN is presented in Figure 9, providing a clear illustration of their performance differences.

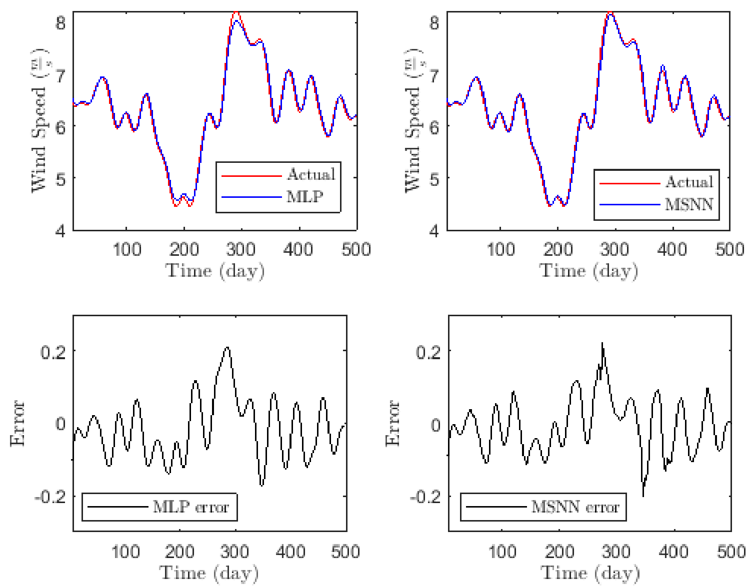

Figure 9: Implementation of MLP and MSNN to forecast the WS in Khorramabad.

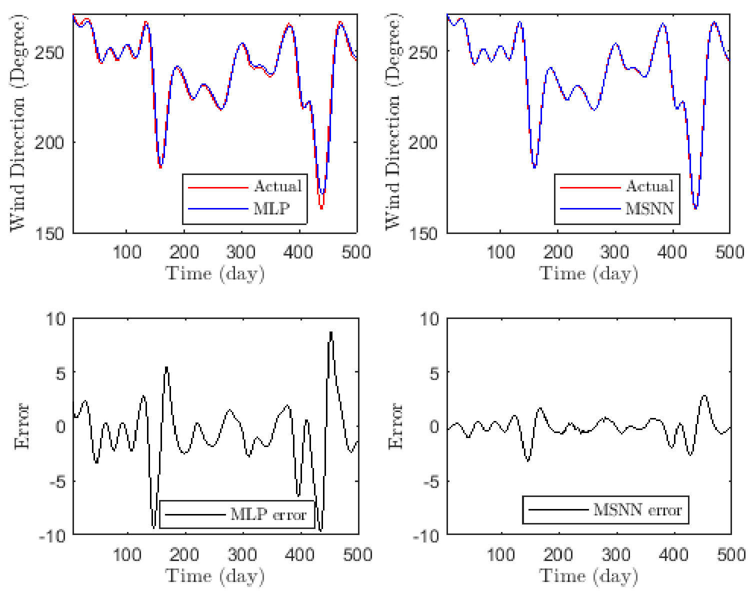

Figure 10: Implementation of MLP and MSNN to forecast the WD in Khorramabad.

Table 5: MSEs of neural models in the forecasting of WS in Khorramabad.

Model	n_h	α	β	A	Γ	Epochs	CPU time	MSE	MAE
MLP	200	-1	1	-40	$0.03I_{200}$	1	3.003	0.0070	0.0883
LSTM	50	0	1	-	0.005	20	112.837	**0.0041**	**0.0510**
MSNN	34	-2	2	-40	$0.2I_{34}$	1	4.707	0.0052	0.0590

Table 6: MSEs of neural models in the forecasting of WD in Khorramabad.

Model	n_h	α	β	A	Γ	Epochs	CPU time	MSE	MAE
MLP	200	-1	1	-40	$0.01I_{200}$	1	6.932	9.3070	2.1910
LSTM	50	0	1	-	0.005	20	107.096	3.3324	1.2178
MSNN	34	-8	8	-40	$0.2I_{34}$	1	5.551	**0.9326**	**0.6690**

6.6 Forecasting of WD

The prediction of WD is carried out using three distinct neural network architectures: MLP, LSTM, and MSNN. In the MLP model, the trainable parameters are initialized with random values within the range $[-0.005, 0.005]$, while the initialization range for the adjustable parameters in the MSNN is set to $[-0.005, 0.005]$. Additional hyperparameters, along with the corresponding model outcomes, are detailed in Table 6. Furthermore, a comparative visualization of the WD forecasting results using MLP and MSNN is presented in Figure 10, providing a clear illustration of their performance differences.

6.7 Forecasting of RA

Table 7: MSEs of neural models in the forecasting of RA in Khorramabad.

Model	n_h	α	β	A	Γ	Epochs	CPU time	MSE	MAE
MLP	85	-1	1	-40	$0.03I_{85}$	1	4.919	0.0303	0.0887
LSTM	50	0	1	-	0.005	20	109.473	0.0315	0.0936
MSNN	34	-6	6	-40	$0.2I_{34}$	1	6.768	**0.0160**	**0.0599**

The prediction of RA is carried out using three distinct neural network architectures: MLP, LSTM, and MSNN. In the MLP model, the trainable parameters are initialized with random values within the range $[-10, 10]$, while the initialization range for the adjustable parameters in the MSNN is set to 0. Additional hyperparameters, along with the corresponding model outcomes, are detailed in Table 7. Furthermore, a comparative visualization of the RA forecasting results using MLP and MSNN is presented in Figure 11, providing a clear illustration of their performance differences.

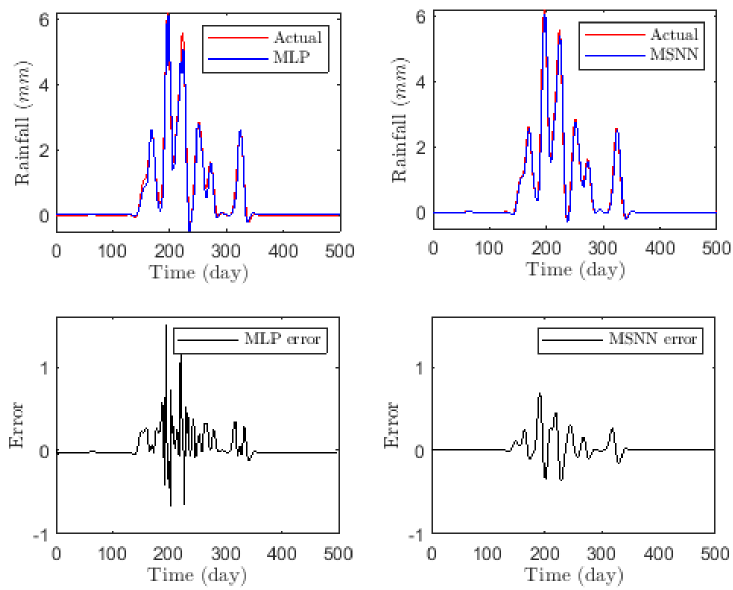

Figure 11: Implementation of MLP and MSNN to forecast the RA in Khorramabad.

6.8 Forecasting of QFE

Table 8: MSEs of neural models in the forecasting of QFE in Khorramabad.

Model	n_h	α	β	A	Γ	Epochs	CPU time	MSE	MAE
MLP	200	-1	1	-40	$0.01I_{200}$	1	6.268	0.0729	0.1980
LSTM	30	0	1	-	0.005	50	123.598	0.0471	**0.1633**
MSNN	34	-3	3	-40	$0.1I_{34}$	1	3.909	**0.0468**	0.1708

The prediction of QFE is carried out using three distinct neural network architectures: MLP, LSTM, and MSNN. In the MLP model, the trainable parameters are initialized with random values within the range $[-0.35, 0.35]$, while the initialization range for the adjustable parameters in the MSNN is set to 0. Additional hyperparameters, along with the corresponding model outcomes, are detailed in Table 8. Furthermore, a comparative visualization of the QFE forecasting results using MLP and MSNN is presented in Figure 12, providing a clear illustration of their performance differences.

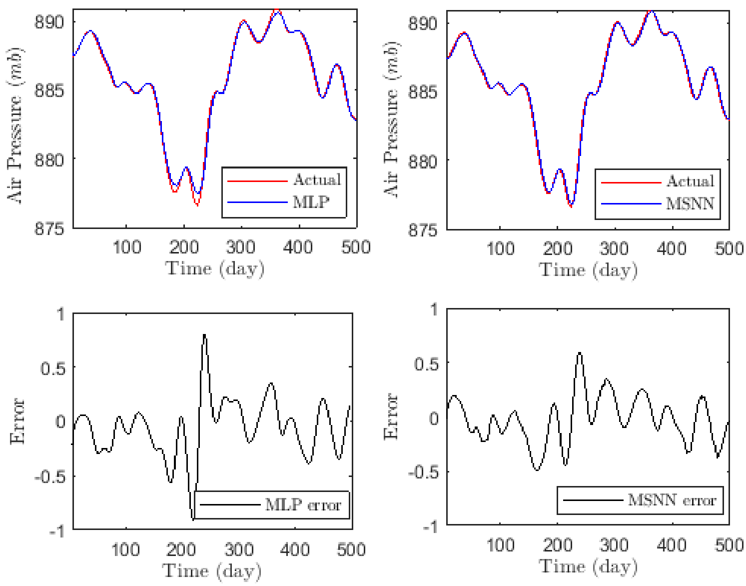

Figure 12: Implementation of MLP and MSNN to forecast the QFE in Khorramabad.

6.9 Discussion

The analysis reveals that the MSNN outperforms the MLP in forecasting all the weather components. This is particularly remarkable given that the MSNN has a relatively simpler architecture and fewer trainable parameters compared to the MLP. This advantage, which is related to the capabilities of the Sinc basis in nonlinear function approximation, is critical for real-time and edge-based meteorological applications. The superior performance of the MSNN underscores its efficiency and effectiveness in this specific forecasting task, making it a promising approach for future applications in temperature prediction.

Our experimental results demonstrate that the MSNN outperforms LSTM in terms of forecasting accuracy (measured using MSE) across most cases. Moreover, the computational efficiency of MSNN is significantly higher; its training and inference require much less CPU time compared to LSTM. This performance gain is largely due to MSNN's shallow structure.

Importantly, MSNN is particularly advantageous when dealing with time series data that exhibit periodic or oscillatory behavior. In such cases, MSNN provides a fast, interpretable, and efficient solution. On the other hand, if the time series contains complex, long-term temporal dependencies, or if multi-step ahead forecasting is required, LSTM is typically the more suitable choice due to its inherent capability to retain and process temporal context over extended horizons.

7. Conclusion

This study emphasizes the significant challenges associated with weather forecasting in Khorramabad, Iran, and showcases the innovative application of the MSNN for predicting weather elements. Despite its relatively straightforward architecture and fewer trainable parameters, the MSNN has outperformed traditional models, demonstrating remarkable accuracy and efficiency. By combining advanced modeling techniques with real-time learning algorithms, this research establishes a robust framework that can support informed decision-making in urban management and disaster preparedness.

Looking to the future, there is considerable potential for further exploration and enhancement. Expanding the datasets to include more diverse and comprehensive meteorological data, multivariate extensions, and integrating cutting-edge technologies such as deep learning architectures and hybrid modeling approaches could significantly improve the precision and reliability of weather predictions.

Acknowledgment

The author of this article expresses profound thanks to the managers and employees of the meteorological organization in Khorramabad for their exceptional collaboration and assistance in supplying the crucial data utilized in this research.

References

Ahmadi, G. (2024), Using a modified Sinc neural network to identify the chaotic systems with an application in wind speed forecasting. *J. Math. Model.*, **12(4)**, 733-752.

Ahmadi, G. (2022), Stochastic gradient-based hyperbolic orthogonal neural networks for nonlinear dynamic systems identification. *J. Math. Model.*, **10(3)**, 529-547.

Ahmadi, G. and Akbari, Z. (2024), Investigating air pollutants in Lorestan province and predicting their concentration using multi-layer neural network with stable online training (case studies: Khorramabad and Poldakhter). *Nivar*, **48(126-127)**, 109-126.

Ahmadi, G. and Dehghandar, M. (2023), Chaotic time series prediction using rough-neural networks. *Math. Interdisc. Res.*, **8(2)**, 71-92.

Ahmadi, G., Teshnehlab, M. (2017), Designing and implementation of stable sinusoidal rough-neural identifier. *IEEE Trans. Neural Netw. Learn. Syst.*, **28**(**8**), 1774-1786.

Azizi, S. P. (2024) A dynamical system decomposition of signal approach to learning with application in time series prediction. • DOI: 10.1016/j.physd.2024.134203

Barzegari Banadkooki, F., Ehteram, M., Najah Ahmed, A., Fai, C.F., Afan, H.A., Ridwam, W.M., Sefelnasr, A., and El-Shafie, A. (2019). Precipitation forecasting using multilayer neural network and support vector machine optimization based on flow regime algorithm taking into account uncertainties of soft computing models. *Sustainability*, **11**(**23**), 6681.

Bauer, P., Thorpe, A. and Brunet, G. (2015), The quiet revolution of numerical weather prediction. *Nature*, **525**, 47–55.

Biswas, S., Ayna, C. O., Gurbuz, S. Z., and Gurbuz, A. C. (2023), CV-SincNet: Learning complex Sinc filters from raw radar data for computationally efficient human motion recognition, *IEEE Trans. Radar Syst.*, **1**, 493-504.

Borra, D., Fantozzi, S., and Magosso, E. (2020), Interpretable and lightweight convolutional neural network for EEG decoding: Application to movement execution and imagination, *Neural Netw.* **129**, 55-74.

Bria, A., Marrocco, C., and Tortorella, F. (2021), Sinc-based convolutional neural networks for EEG-BCI-based motor imagery classification. in: A.D. Bimbo, et al. *Pattern Recognition. ICPR International Workshops and Challenges, ICPR Lecture Notes in Computer Science*, vol **12661**, Springer, Cham.

Chandra, R., Goyal, S., and Gupta, R. (2021), Evaluation of deep learning models for multi-step ahead time series prediction. *IEEE Access*, **9**, 83105–83123.

Conti, S. (2024), Artificial intelligence for weather forecasting. *Nat. Rev. Electr. Eng.*, **1**(**8**).

Cordova, C. H., Portocarrero, M.N.L., Salas, R., Torres, R., Rodrigues, P.C., and López-Gonzales, J.L. (2021), Air quality assessment and pollution forecasting using artificial neural networks in Metropolitan Lima-Peru. *Sci Rep.*, **11**, 24232.

Eftekhari, A., Saadatmandi, A. (2021), DE Sinc-collocation method for solving a class of second-order nonlinear BVPs, *Math. interdiscip. res.*, **6**(**1**), 11-22.

Elsaraiti, M., and Merabet, A. (2021), Application of long-short-term-memory recurrent neural networks to forecast wind speed. *Appl. Sci.* **11**, 2387.

Elwasif, W., Fausett, L. V. (1996), Function approximation using a sinc neural network, in *Proceedings of the SPIE*, 690-701.

Faruk, D. O. (2010), A hybrid neural network and ARIMA model for water quality time series prediction, *Engineering Applications of Artificial Intelligence*, **23**, 586-594.

Goodfellow, I., Bengio, Y., and Courville, A. (2016), *Deep Learning.* MIT Press.

Gupta, M. M., Jin, L., Homma, N. (2003), *Static and dynamic neural networks: From fundamentals to advanced theory*, John Wiley & Sons, New Jersey.

Heydari Dastjerdi, R., Ahmadi, G. (2024), Designing the sinc neural networks to solve the fractional optimal control problem. *Iran. j. numer. anal. optim.*, **14**(**4**), 1016-1036.

Hsieh, W. W. and Tang, B. (1998), Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bulletin of the American Meteorological Society*, **79**(**9**), 1855-1870.

Iranshahi, M., Ebrahimi, B., Yousefi, H., and Moridi, A. (2023) Investigating the Effects of Climate Change on Temperature and Precipitation Using Neural Network and CMIP6 (Case Study: Aleshtar and Khorramabad Stations). *Water and Irrigation Management*, **12**(**4**), 821-845.

Kim, J., Kim, H., Kim, H. G., Lee, D., Yoon, S. (2024), A Comprehensive Survey of Time Series Forecasting: Architectural Diversity and Open Challenges. *arXiv*:2411.05793.

Khoramabadi, F., and Moradinia, S. F. (2024), The prediction of precipitation changes in the Aji Chay watershed using CMIP6 models and the wavelet neural network. *Journal of Water and Climate Change*, **15**(**5**), 2141–2161.

Khosravi, S. (2017). Forecasting Temperature using Fuzzy Neural Networks and Genetic Algorithms (Case Study Lorestan Province, Khorramabad City). *International Journal of Environmental and Science Education*, **12**(**5**), 993-1006.

Li, W., and Law, K.L.E. (2024). Deep Learning Models for Time Series Forecasting: A Review, *IEEE Access*, **12**, 92306-92327.

Liu, K., Yang, M., Xing, X., Yu, Z., and Wu, W. (2023), SincMSNet: A Sinc filter convolutional neural network for EEG motor imagery classification. *J. Neural Eng.*, **20**(**5**), 056024.

Moradpoor, N., Najarchi, M., and Mirhoseini Hezave, S. M. (2023), Spatiotemporal analysis of long term rainfall in semi arid area using artificial intelligence models (case study: Ilam Province, Iran). *Water*, **15**(**19**), 3521.

Ouma, Y. O., Cheruyot, R., and Wachera, A. N. (2022), Rainfall and runoff time-series trend analysis using LSTM recurrent neural network and wavelet neural network with satellite-based meteorological data: case study of Nzoia hydrologic basin. *Complex Intell. Syst.* **8**, 213–236.

Parcollet, T., Morchid, M., and Linares, G. (2020), E2E-SINCNET: Toward fully end-to-end speech recognition. *ICASSP*, Barcelone, Spain, hal-02484600.

Radha, K., Bansal, M., Pachori, R. B. (2024), Automatic speaker and age identification of children from raw speech using sincNet over ERB scale. *Speech Commun.*, **159**, 103069.

Ren, X., Li, X., Ren, K., Song, J., Xu, K., Deng, K., and Wang, X. (2021), Deep learning-based weather prediction: A survey, *Big Data Research*, **23**, 100178.

Schanze, T. (1995), Sinc interpolation of discrete periodic signals, *IEEE Trans. Signal Process.*, **43**(**6**), 1502-1503.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K. and Woo W.-C. (2015), Convolutional LSTM Network: A machine learning approach for precipitation nowcasting. *arXiv*: 1506.04214.

Stenger, F. (2010), *Handbook of Sinc Numerical Methods*, CRC Press, Inc., USA.

Taie Semiromi, M., Koch, M. (2024), Statistical downscaling of precipitation in northwestern Iran using a hybrid model of discrete wavelet transform, artificial neural networks, and quantile mapping. *Theor. Appl. Climatol.* **155**, 6591–6621.

Stenger, F. (1993), *Numerical methods based on Sinc and analytic functions*, Springer-Verlag, New York.

Veyskarami, I., Dehghani, R., Chamanpira, R. (2024), Evaluation of meta-heuristic models in estimating daily evaporation rate (Case study: Kakareza basin - Lorestan province), *Journal of Climate Research*, **1403**(**57**), 197-209.