Mixed-Input Residual Network for Air Temperature Forecasting by Low-Power Embedded Devices

*[1]Kayode P. Ayodele, [1]Faruk O. Adekola, [1]Abiodun A. Ogunseye, [1]Adedayo Abiona, [1]Olawale B. Akinwale, [1]Oladoyin E. Arewa,[1]Funmilayo B. Offiong, [1]Akinbode A. Olawole, and [2]Vincent O. Ajayi
[1]Department of Electronics & Electrical Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria
[2]Department of Meteorology and Climate Science, Federal University of Technology, Akure, Nigeria
kayodele@oauife.edu.ng|adekoladamilola4@gmail.com|aaogunseye@oauife.edu.ng|aoabiona@student.oauife.edu.ng|olawale.akinwale@googlemail.com|oearewa@student.oauife.edu.ng|phummykoya@gmail.com|alexolawole@gmail.com|vincentajayi@gmail.com

Received: 07-SEP-2023; Reviewed: 21-DEC-2023; Accepted: 03-FEB-2024
https://dx.doi.org/10.4314/fuoyejet.v9i1.2
ORIGINAL RESEARCH

Abstract— Accuracy and model compactness are essential requirements for weather forecasting models designed for operation on low-power embedded devices. This study developed Mixed-Input Residual Network (MIRNet), a compact temperature-forecasting deep neural network model. MIRNet integrates stacked bidirectional long short-term memory layers using concatenated 1-dimensional and 2-dimensional convolutional layers to improve model accuracy. MIRNet was trained and tested on two datasets: one, IfeData, comprising historical weather data from Ile-Ife, Nigeria and the other a standard weather forecasting dataset called the Jena dataset. Training was carried out using 100 epochs of data partitioned in the standard 80:20 ratio, with an adaptable learning rate strategy. The model was tested for Nth-hour-ahead prediction for $1 \leq N \leq 24$; where $N \in \mathbb{N}$ are natural numbers, and performance quantified using metrics such as mean absolute percentage error (MAPE) and mean square error (MSE). The model was also implemented on a Raspberry Pi 4 device with a 1.8 GHz 64-bit quad-core ARM Cortex-A72 processor. The model achieved a MSE of 1.00 x 10-3 on the IfeData dataset, and 1.23x10-4 on the Jena dataset for 1-hour ahead forecasting. This is currently the best verifiable result achieved on the Jena dataset by any prediction model globally. For Nth hour ahead forecasting, MIRNet achieved an MSE generally below 2.0x10-3 for all values of N on the standard Jena dataset. The MSE of MIRNet for N-sequential 1-hour ahead and single Nth hour predictions using the Jena dataset reveal quadratic and linear relationships with N respectively. The model compares favourably with existing models for multi-hour predictions. The developed model is compact and has good forecasting properties.

Keywords— Deep neural networks, embedded devices, IfeData dataset, MIRNet, temperature forecastings

1 INTRODUCTION

Accurate weather forecasting plays an important role in various industrial sectors such as energy management, agriculture, transportation, and disaster preparedness. Among the numerous weather parameters, temperature stands as one of the fundamental and highly influential factors. Accurate temperature forecasts enable decision-makers, businesses, and individuals to plan effectively, mitigate risks, and optimise resource allocation.

Over the years, advancements in computational techniques have transformed weather forecasting from a traditional, data-driven endeavour into a data-intensive scientific discipline (Kumar, 2023). The quality of weather forecasting is fundamentally dependent on the data on which the forecasting model is based. In this regard, the choice is usually between data gotten from in-situ or local instrumentation, versus data gotten from satellites. Each category offers a different trade-off between coverage area and local accuracy, with satellites offering more coverage than in-situ instruments, but being less accurate, in general, for any specific location.

This trade-off has implications on the prediction of very local or small-scale weather phenomena (Dirmeyer, 2003). Consequently, despite key advances in the qualities of global and regional weather models as well as available computational power to run them, there is still great interest in techniques for high-accuracy forecasting of weather parameters based solely on local measurements. Beside prediction accuracy, model compactness (to allow running the model on resource-limited platforms) is also a characteristic that is highly valued (Li et. al., 2020; Debelee and Ayano, 2022). For example, a nationwide network of low-cost weather stations in Nigeria named WeatherWAN is currently being prototyped by researchers under the Nigerian Communications Commision Research Innovations Grant. For sustainability and scalability of the weather stations, it is imperative that the network nodes being used are low-power, durable, and require minimal maintenance without compromising the accuracy of the system. The weather stations require accurate local weather data and make their predictions without recourse to external data sources or computational resources. Use cases such as this partly explain the recent attention paid to deep learning weather forecasting models, particularly those that can be run on low-power, low-resource embedded devices.

This paper introduces such a network, the Mixed-Input Residual Network (MIRNet), a temperature-forecasting framework that integrates residual blocks and bidirectional long short-term memory (BILSTM) networks, harnessing their capabilities to capture intricate temporal dependencies and enhance the modelling of

*Corresponding Author

Section B- ELECTRICAL/COMPUTER ENGINEERING & RELATED SCIENCES
Can be cited as:
Ayodele K.P., Adekola F. O., Ogunseye, A. A., Abiona A., Akinwale O. B., Arewa O. E., Offiong F. B., Olawale A. A. and Ajayi V. O. (2024).
Mixed-Input Residual Network for Air Temperature Forecasting by Low-Power Embedded Devices, FUOYE Journal of Engineering and Technology (FUOYE**JET), 9(1), 7-16.** https://dx.doi.org/10.4314/fuoyejet.v9i1.2

complex climate patterns. MIRNet utilises alternative residual paths for the variable of interest, as well as a multivariate representation of the full prevailing weather conditions over the last 48 hours.

However, different from the existing work, we take into consideration model compactness without compromising good forecasting accuracy in order to achieve a MIRNet system that significantly outperforms previous weather models. The system is demonstrated to run on a 5 W Broadcom BCM2711 device with a 1.8 GHz 64-bit quad-core ARM Cortex-A72 platform. For model compactness, the MIRNet makes use of an adaptable sampling rate, along with optimised energy usage and time-share utilisation of various subsystems based on anticipated weather sensing. MIRNet is a solution that we have developed as a weather forecasting tool that can be used on resource-constrained devices, thus making them usable in in-situ low-cost weather stations such as WeatherWAN.

The novelty in our research lies in the innovative integration of residual blocks and BiLSTM, aimed at addressing resource constraints. Additionally, our approach involves the effective fusion of single-dimensional and multidimensional inputs. This combination not only enhances the model's capacity to capture long-term dependencies and utilise skip connections but also ensures adaptability to diverse data formats. The subsequent sections of the paper will delve into the methodology, experimental setup, results and discussions, showcasing the effectiveness of the developed model.

Recent years have witnessed a surge of interest in deep learning techniques, particularly the class of recurrent neural networks (RNNs) called long short-term memory (LSTM) networks and convolutional neural networks (CNNs) for tackling temperature forecasting. CNNs are excellent for capturing spatial (two- or three-dimensional) patterns, while LSTMs are excellent for sequences or temporal dynamics. Haque, Tabassum and Hossain (2021) conducted a comparative study of six different deep network architectures: gated recurrent unit (GRU),convolutional neural network (CNN), simple recurrent neural network (SRN), long-short term memory (LSTM), and two hybrid models. The paper reported a RMSE value of 1.1729, 1.2010, 0.9764, 1.0270, 0.8575 and 1.0277 for SRN, GRU, LSTM, CNN, CNN-LSTM and GRU-LSTM respectively for 1-hour ahead prediction which shows the excellent temporal and spatial extraction capabilities of the hybrid CNN-LSTM approach.

One trend that has emerged in recent years is to utilise combinations of LSTMs and CNNs within the same model. This allows models to capture not just temporal dynamics, but inter-variable interactions. Hewage et al., (2021) proposed a novel compact data-driven weather forecasting model that utilises temporal modelling approaches of LSTM and temporal CNN. The performance of this model was compared with that of existing classical approaches, and it was shown that it out-performed selected classical models for efficient and accurate weather forecasting up to 12 hr. LSTM-CNN Integration was also explored in Hou et al., (2022) in which a hybrid LSTM-CNN model was used for hourly temperature prediction. In that study, the time series data dimensionality was reduced using CNN, while the long-term memory of the massive temperature time-series data was captured with LSTM.

In the SeriesNet model proposed by Shen et al., (2020), a CNN network was integrated with an LSTM network which learned holistic features and carry out the reduction of multi-conditional data dimensionality, and a dilated causal convolution network which attempts to learn different time intervals. This model was said to be able to learn multi-level and multi-range features from time series data, and had higher predictive accuracy in comparison to models using fixed time intervals. Moreover, SeriesNet adopted residual learning which is another relatively common technique to improve model performance. Residual blocks enhance the training of deep neural networks by mitigating the vanishing gradient problem and enabling the successful training of extremely deep architectures. Improvements in performance have been documented for variations of residual learning (He et al., 2016; Zhao et al., 2016; Huang et al., 2017).

A technique called stacking has also been investigated by numerous studies. Examples of this can be seen in Li et al. (2019) where a stacked LSTM was used to process temperature time series data and also carry out temperature prediction every half hour. The comparison of prediction and training with the two benchmark algorithms of deep neural network (DNN) and random forest (RF) on generated data under different sliding windows, found the stacked LSTM to perform better based on MSE, RMSE and MAE. Al Sadeque and Bui (2020) also provided evidence of benefits of stacking deep neural networks for improved forecasting accuracy.

Another variation on standard LSTMS that has been applied for temperature forecasting is BiLSTM. Bi-LSTM processes sequences in both backward and forward directions (in contrast to LSTM training exclusively in a forward manner) enabling the model to incorporate future information into the prediction, and thereby enhancing the performance (Graves and Schmidhuber, 2005). Liang et al., (2021) proposed a BiLSTM-based model called BL-FC network for temperature modelling and forecasting. BL-FC has four layers: the first layer is a Bi-LSTM layer, which learns features from continuous temperature data in both backward and forward directions; the other three layers are fully connected layers, the second and third layers additionally extract data features, and the last layer maps the final output of temperature prediction. The BL-FC model achieved values of 0.9761 and 0.7147 for MSE and MAE respectively for 4-hour ahead prediction.

2 METHODOLOGY

2.1. MIRNET DEVELOPMENT

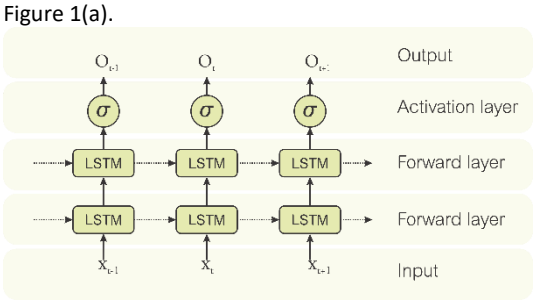
The following specifications were adopted in developing the MIRNet:

- Bidirectional LSTM
- Stacking
- Residual learning

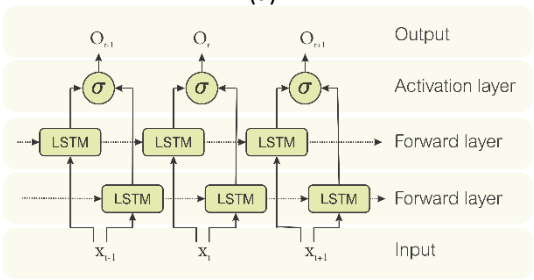
https://dx.doi.org/10.4314/fuoyejet.v9i1.2 engineering.fuoye.edu.ng/journal

- Combined 2D CNN
- Should run on embedded system

The first four specifications are features that have been used individually or collectively in previous high-performing models, while the last was necessitated by the specific use case for which MIRNet was developed: to be used for environmental variable prediction on the nodes of a network of low-powered automatic weather stations. Figure 1(b) shows the BiLSTM structure used for this work contrasted against the conventional LSTM structure in Figure 1(a).



(a)



(b)

Figure 1: Variants of the long short-term memory network structure (a) Conventional LSTM (b) BiLSTM structure

The relationship for a single LSTM cell h in the reverse direction is given by (Shuster and Paliwal, 1997):

$$\varepsilon_h^t = \sum_{h\varepsilon B} V_i^h . D_i^{t+1} \quad (1)$$

$$D_{og}^t = y_{og}^{t'} \cdot \sum_{h\varepsilon B} \left(\epsilon_h^t \cdot \tanh\left(s_h^t\right)\right) \quad (2)$$

$$\frac{\partial E^t}{\partial s_h^t} = \epsilon_h^t \cdot y_{og}^t \cdot \frac{\partial tanh\,(y_h^t)}{\partial y_h^t} + \frac{\partial E^{t+1}}{\partial s_h^{t+1}} \cdot y_h^{t+1} + D_{ig}^{t+1} . W_h^{ig} + D_h^{og} \cdot W_h^{og} \quad (3)$$

$$D_h^t = y_{ig}^t \cdot W_h^{t'} \cdot \frac{\partial E^t}{\partial s_h^t} \quad (4)$$

$$D_{fg}^t = y_{fg}^{t'} \cdot \sum_{h\epsilon B} \frac{\partial E^t}{\partial s_h^t} \cdot y_{og}^{t+1} \cdot \tanh\left(s_h^{t+1}\right) \quad (5)$$

$$D_{fg}^t = y_{ig}^{t'} \cdot \sum_{h\epsilon B} \left(\frac{\partial E^t}{\partial s_h^t} \cdot y_{og}^t\right) \quad (6)$$

Where ϵ_h^t is the back propagated output error on cell h at time t, V and W indicate weight matrices, and B denotes the group of cells. D is the error derivative for the cell's gates, and y is the output of each gate, where the subscripts ig, s, f_g, and o_g represent input gate, state, forget gate, and output gate, respectively. The state value of h is s_h, and the net output error at time t is E^t. According to the LSTM unit operation in the backward layer, BiLSTM can vary the parameters to reduce the forward layer propagated errors.

Residual learning (He et al., 2016) is a popular choice for countermeasures because of its straightforward approach and proven effectiveness. The desired mapping of the stacked layers in conventional DNN can be represented as

$$\phi(x) = \varphi(x) \quad (7)$$

where $\varphi(x)$ is the desired underlying mapping and $\phi(x)$ is the stacked nonlinear layers. For residual, we can learn another mapping, $\phi(x)$, defined as

$$\phi(x) = \varphi(x) - x \quad (8)$$

This way, the original mapping can be reformulated as

$$\phi(x) + x = \varphi(x) \quad (9)$$

This key adjustment in the network design shifts the focus from having the network learn the complete transformation from input to output, to simply learning the residual (or difference) mapping for each layer. When the mapping is close to being an identity function (meaning it does not change the input much), the network becomes adept at detecting subtle changes or perturbations. By stacking these residual blocks together, deep residual networks maintain or even improve forecasting accuracy without encountering diminishing returns.

In normal usage, LSTM networks typically contain multiple layers, working concurrently with activated versions of the output. A variant of this structure, motivated by the Multilevel Residual Network (MRN) in Zhang et al., (2018), was adopted in this study. For such a multilevel residual network with stacked layers and a total of 2n layers, the output, $\varphi_n(x)$ can be represented as:

$$\varphi_n(x) = \phi_n(\varphi_{n-1}(x)) + \sigma(\omega(\varphi_{n-1}(x))) + \sigma(\omega(x)) \quad (10)$$

where $\omega(x)$ denote the 1D CNN output for input x, $\sigma(\mathrm{x})$ represents an activated output for input x.

Concatenation was also employed to implement skip connections in ResNets. A repetitive concatenated structure such as that described in (Huang et al., 2017) was adopted, wherein the outputs of multiple stacked sub-networks were concatenated, and fed into subsequent sub-networks. In this network, the output of the jth stacked layer becomes

$$\varphi_j(\mathrm{x}) = \phi_j\left(\varphi_{j-1}(\mathrm{x})\right) \# \left(\left(\varphi_{j-1}(\mathrm{x})\right)\right) \# \left(\left(\varphi_{j-2}(\mathrm{x})\right)\right) \# \cdots \# \mathrm{x} \quad (11)$$

where # represents the concatenation operation.
Previous implementations of similar models all utilised univariate approaches, wherein only historical temperature readings are utilised as input vectors. They thus lack the inherent capability to effectively learn the spatial relationships within multivariate time series data. Multivariate time series data offers the advantage of capturing complex interactions between variables and dimensions, leading to more accurate modelling and valuable insights in time series prediction.

To address these limitations, this study introduces an approach combining the strengths of equations (10) and (11) while also enhancing the model by incorporating multivariate capability which offers the advantage of capturing complex interactions between variables and dimensions, leading to more accurate modelling performance. This was done by integrating 2-dimensional (2D) convolutional layers into the stacks.

Consequently, the ith stacked layer output of MIRNet can be expressed as the combination of equations (10) and (11) in addition to the stacked 2D layers for the multivariate processing:

https://dx.doi.org/10.4314/fuoyejet.v9i1.2 engineering.fuoye.edu.ng/journal

$$\varphi_i(x) = y_i \# \sigma(\omega(y_i)) \# (\varphi_{i-1}(x)) \# \sigma(\omega(\varphi_{i-1}(x)) \# x \# \nu(x_m)) \quad (12)$$

where x represents the input temperature feature, x_m denotes the multivariate input containing temperature and other correlated weather variables, $\varphi(x)$ is the desired mapping of stacked layers, $\phi(x)$ represents the pure stacked layers, $\sigma(z)$ denotes the activated output for input z, $\omega(z)$ denotes the 1D CNN output for input z and $\upsilon(x_m)$ denotes the output from stacked layers of 2D CNN for input x_m. Consequently, each stack within the iterative structure of MIRNet consists of elements shown in Figure 2. MIRNet consists of repetitions of the basic block in Figure 2.

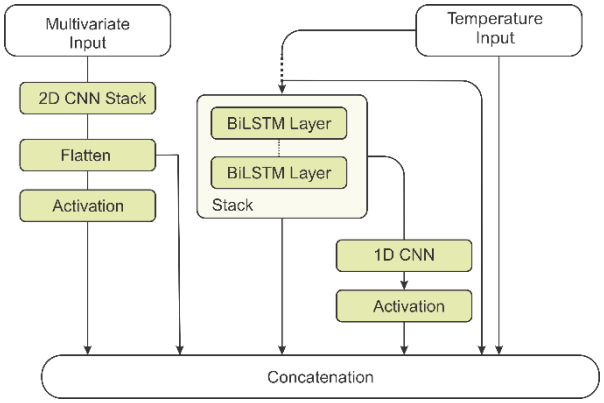


Figure 2: Basic block of MIRNet

The requirement of being run on an embedded system however places a constraint on how many of the blocks can be accommodated by MIRNet. Ultimately, by an iterative process, the optimal number of stacks was determined.

2.2. Training Datasets

Two training datasets are utilised in this study. The first dataset, which will be called the "IfeData" dataset in the rest of this paper, was obtained from Visual Crossing, an online platform that provides paid access to historical global weather data (accessible at https://www.visualcrossing.com/weather-data). Data for Ile-Ife, with specific GPS coordinates 7.49705N, 4.51688E, were downloaded for 1-hr interval from January 1, 2018 to December 31, 2022.

The downloaded data from Visual Crossing were stored in comma-separated columns, and contained variables such as data and time, sea level pressure (in millibars), temperature (in ◦C), dew point (dew) humidity denoting air moisture as a percentage, percentage cloud cover, solar radiation (in watts per square metre), precipitation (in mm), wind speed (km/h) and direction (in degrees), precipitation probability (as a percentage), UV Index (indicating the level of ultraviolet radiation in a scale from 0 to 10), vapour pressure, and related metrics such as "feels like temperature".

It is to be noted that Weather Crossing data are acquired primarily through satellites, and as previously stated, satellites may lack accuracy at high geographical resolutions. Consequently, a second dataset, Jena weather dataset, was also utilised for testing MIRNet. The Jena dataset (discussed below) has become a standard dataset for testing weather prediction models, which facilitates easy comparisons of different models by various research groups worldwide.

The Jena weather dataset originates from the meteorological station situated at the Max Planck Institute of Biogeochemistry in Jena, Germany. It encompasses a comprehensive compilation of 14 distinct features, each meticulously recorded at 10-minute intervals. This extensive data collection spanned eight years, starting on January 1, 2009, and concluding on December 31, 2016. This interval contains 420,551 timestamps, each associated with 14 weather parameters. The entire eight-year span was used in this study. The dataset is available at https://s3.amazonaws.com/keras-datasets/jena_climate_2009_2016.csv.zip for download.

The Jena dataset encompasses a range of meteorological information presented in comma-separated columns. This dataset includes essential data points such as Date and Time for temporal context. Pressure measurements are denoted in mbar, while temperature is conveyed both in degrees Celsius (C) and Kelvin (K). Tdew (degC) represents temperature in relation to humidity. The level of air saturation with water vapour is expressed as rh (%), and related metrics like saturated vapour pressure (mbar), actual vapour pressure (mbar), and vapour pressure deficit (mbar) are also included.
Additional meteorological parameters include specific humidity (g/kg) and water vapour concentration (mmol/mol). Airtightness is quantified in g/m3, while wind-related data comprises wind speed (m/s) and maximum wind speed (m/s), along with wind direction in degrees.

2.3. Experiments

2.3.1. Data Preprocessing

Correlation analyses were carried out to determine how many of the 14 IfeData variables and 14 Jena variables to use, with the correlation coefficients with temperature of all variables sorted as shown in Table 1. The cloud condition variable of the IfeData was excluded as it is a qualitative score. Variables that strongly correlate with temperature selected, using a cutoff of 0.6 for the IfeData dataset, and 0.8 for the Jena dataset. This resulted in the following input variables being extracted for IfeData: temp, vapour_pressure, feelslike, solarradiation, uvindex. Seven variables ('T (degC)','Tpot (K)','VPmax (mbar)','Tdew (degC)', 'VPact (mbar)','H2OC (mmol/mol)', 'sh (g/kg)') were extracted from the Jena dataset.

Table 1: Results of correlation analyses on IfeData and Jena datasets

IfeData dataset		Jena dataset	
Variable	Coefficient	Variable	Coefficient
temp	1.000000	T (degC)	1.0000
vapour_pressure	0.990656	Tpot(K)	0.9968
feelslike	0.949192	VPmax (mbar)	0.9511
solarradiation	0.657872	Tdew (degC)	0.8957
uvindex	0.654772	VPact (mbar)	0.8677
windspeed	0.272730	H2OC (mmol/mol)	0.8672
winddir	0.000487	sh (g/kg)	0.8668
precipprob	-0.054596	VPdef (mbar)	0.7617

https://dx.doi.org/10.4314/fuoyejet.v9i1.2 engineering.fuoye.edu.ng/journal

precip	-0.055685	max. wv (m/s)	-0.0040
dew	-0.064979	wv (m/s)	-0.0050
cloudcover	-0.135608	wd (deg)	0.0396
sealevelpressure	-0.259744	P (mbar)	-0.0454
humidity	-0.749907	rh (%)	-0.5721
Cloud conditions	Excluded	rho (g/m**3)	-0.9634

Standardisation techniques include min-max standardisation, z-score standardisation, arctan inverse tangent function standardisation, and log function standardisation; among these, z-score standardisation and min-max standardisation are the most effective at avoiding numerical issues for gradient update, facilitating learning rate adjustment, optimising the search trajectory, and speeding up the search for the best solution. For this study the min-max technique was utilised since the sample is less noisy and less contaminated. The standard expression for min-max , x^*, is given by (13):

$$x^* = \frac{x - \min}{\max - \min} \quad (13)$$

2.3.2. Model Training and Testing

MIRNet was thereafter trained on IfeData. A summary of key hyperparameters is presented in Table 2. The whole 43,824 hours of data comprising the IfeData dataset was partitioned into 80:20 splits for training and testing, with 25% of the training dataset being reserved for validation. The machine used for the training was a Hewlett-Packard Z640 workstation with a 20-core Intel Xeon ES-2630 v4 processor clocked at 2.20 GHz and 32 GB DDR RAM, along with 280 GB Intel Optane 900P SSD as swap memory, running Ubuntu operating system version 18.04.6 LTS and an Nvidia Quadro RTX 8000 graphical processing unit (GPU) with 48 GB GDDR6, and 4,608 Tensor cores.

Table 2: Hyperparameters and Test Settings

SN	Setting	Value
1	Batch size	100
2	Epochs	100
3	Training method	Adam
4	Loss function	MSE, MAE, MAPE
5	Activation Function	SELU
6	Input sequence length	48
7	Output sequence length	1
8	Training strategy	ReduceLROnPlateau

Performance metrics used in the literature for forecasting tasks include accuracy, mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), mean absolute percentage error (MAPE), R-squared (Coefficient of Determination) etc. Each of these measures have their strengths and weaknesses. Given that they are by far the most popular of the metrics used in literature, MAE, MSE, RMSE, and MAPE were all computed in assessing MIRNet performance on IfeData.

MAE indicates the average absolute difference between predicted and actual values, giving a measure of the average error magnitude without consideration for their direction. The MAE is given as

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|p_i - x_i| \quad (14)$$

where n is number of observations in the datasets, p_i is the predicted value for the ith data point generated by the model, and x_i is the ith datapoint in the dataset.

MSE is the average of the squared differences between actual and predicted values. It gives a measure of the average squared error, emphasising larger errors. The following was used to compute MSE:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(p_i - x_i)^2 \quad (15)$$

RMSE is the square root of the average of the squared differences between predicted and actual values. It gives a measure of the magnitude of errors while penalising larger errors more heavily. RMSE was computed in this study using:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - x_i)^2} \quad (16)$$

MAPE calculates the average percentage difference between predicted and actual values relative to actual values. It gives an estimate of the relative accuracy of the forecasts. MAPE gives insights into the proportional error in the forecasts and is useful for comparing accuracy across different data ranges. MAPE was computed using the following expression:

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{p_i - x_i}{x_i}\right| \times 100\% \quad (17)$$

All of the foregoing experimental steps, from data preprocessing, to training and testing, and evaluation, were repeated for the Jena dataset, and all metrics recorded.

MIRNet was designed from the ground up to be a single-step ahead model. However, its performance was evaluated for multi-step prediction. For this purpose, two approaches were tested. In one approach, the temperature at the Nth hour into the future was predicted using data from the 48 hours preceding the first predicted hour. N successive 1-step ahead predictions were then carried out, and each time, the most recent predicted result was added into the 48-point input vector in a first-in-first-out (FIFO) format. In the second approach, MIRNet was retrained solely on Jena data, while modifying the delta, D, between the hour labelled as the output, and the 48 hours forming the input. So, for example, for 1-step ahead prediction, D was 1. For true Nth-hour prediction, the training and testing were repeated a further 23 times, for values of D = 2,3, ... ,24.

2.3.3. Testing On A Cortex-A72 Processor

In order to verify the suitability of MIRNet for the target use case of low-power devices, it was tested on a selected

https://dx.doi.org/10.4314/fuoyejet.v9i1.2 engineering.fuoye.edu.ng/journal

node of the NCC WeatherWAN prototype within the network. The layout of the prototype network is shown in Figure 3. As illustrated in Figure 3, various battery and supercapacitor chemistries are used for energy storage. However, irrespective of the chemistry used, none of the node power supply units exceed 10 W rating. Furthermore, nodes are powered through energy harvested using solar panels, making energy usage optimization essential. Furthermore, ultracapacitors and lithium-based storage chemistries possess vastly different advantages and disadvantages, which have necessitated the development of an intelligent hybrid power system that uses data, including predicted weather conditions, to optimally switch between the different storage chemistries and sampling rates. Consequently, a device capable of running a prediction model is required, but must be one that can operate within the limited energy budgets.

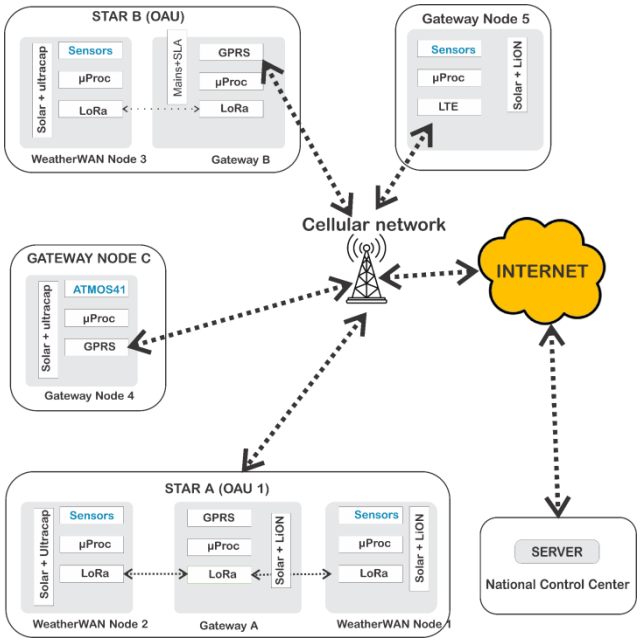


Figure 3: Network on which MIRNet was tested

A Raspberry pi 4 Block B device was adopted for this purpose. The device is built around a Broadcom BCM2711 system on chip, with a 1.8 GHz 64-bit quad-core ARM Cortex-A72 processor, and a rated power of 5 W. Normally, the lowest-power state of the device is about 1.48 W. However, through modifications made in the EEPROM's bootloader configuration, the "WAKE_ON_GPIO" parameter was disabled, and the "POWER_OFF_ON_HALT" parameter was enabled, bringing its consumption during its sleep state down to 0.1 W. MIRNet was loaded onto the device, and used to carry out both single-step and multi-step predictions. Purpose of these tests was not to determine the accuracy of the model (which were already known from the tests in Section 3.2 of this paper) but the temporal, and energy costs of computation.

3 Results And Discussion

3.1 Results

Figure 4 shows the variation of loss functions during the training of the MIRNet model using IfeData dataset. The performance of the model during training and testing were evaluated, and the results are presented in Table 3. It can be seen that the model converges very rapidly. So rapidly, in fact, that the chart has to use log vertical scaled to reveal more information. Multiple performance metrics (MSE, MAE, MAPE, RMSE) were extracted from the model training and testing in order to compare with previous results reported in literature. Wide variations are observed in the literature in terms of choice of evaluation metrics for forecasting tasks, and this is evident in the evaluation metrics for selected existing models in Table 4. The references listed in Table 4 show variations on a single-step prediction on temperature data.

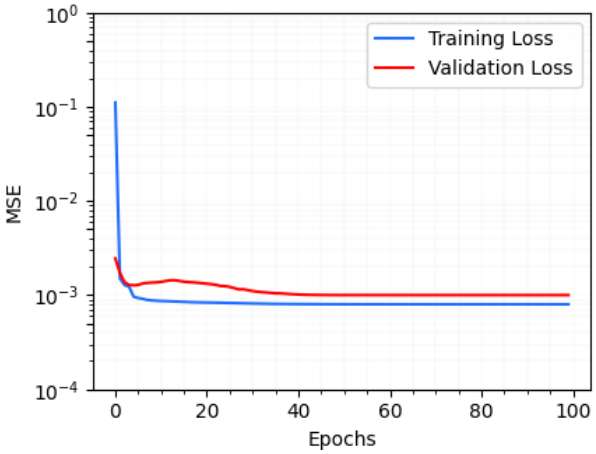


Figure 4: Training curved for the IfeData dataset, showing the variation of loss with epoch number

Given, however, the known dependence of model performance on dataset characteristics, the experiment was repeated using the standard Jena dataset. The loss variation curves for that training are shown in Figure 5, while the performance is presented using various metrics in Table 5. For comparison, Table 6 presents a list of published results carrying out the single-step task on the Jena dataset.

Finally, the results when MIRNet was used for N sequential 1-hour ahead, and true Nth-hour ahead prediction on the Jena dataset, are presented in Figure 6. While the single Nth-hour ahead has a reasonable linear relationship between performance (using MSE) and N, sequential 1-ahead predictions are more quadratic.

Table 3: Performance of the proposed MIRNet model for single-hour ahead forecasting on IfeData dataset

https://dx.doi.org/10.4314/fuoyejet.v9i1.2 engineering.fuoye.edu.ng/journal

Performance					
Validation			**Test**		
MSE ($x10^{-3}$)	MAE ($x10^{-2}$)	MAPE (%)	MSE ($x10^{-3}$)	MAE ($x10^{-2}$)	MAPE (%)
1.00	2.11	6.86	1.1611	2.40	9.01

The N sequential single-step ahead resulted in MSE that peaked around $8x10^{-3}$ for around 15 hours ahead prediction. However, the true N-ahead predicted generally stayed below $2x10^{-3}$. Some existing results from the literature, featuring multi-step prediction based on the same Jena dataset, are presented in Table 7, for comparison.

Table 4: Performance of selected existing models on single-step ahead forecast

Study	Input /Other Details	Performance
Abdel-Aal, (2004)	24 hours temperature for previous day	MAE: 0.93◦C
Smith et al., (2006)	Up to prior 24 hours of multiple variables	MAE: 0.53◦C
Smith et al. (2009)	Current and prior 24 h of data for multiple variables	MAE: 0.52◦C
Chevalier et al., (2011)	Up to prior 24 h for multiple variables	MAE: 0.51◦C
Li et al., (2019)	Global Temperature	MSE: 0.026 RMSE: 0.16
Sekertekin et al., (2021)	Temperature Predictions	MSE: 0.415 RMSE: 0.644
Hassani et al., (2018)	Yearly global 1-step	RMSE: 0.67
(Hossain et al., 2015)	Previous 96 h values of temperature, barometric pressure, humidity, wind speed	RMSE: 0.0138
Abubakar et al., (2016)	Rain, Pressure Wind Speed, Global Temperature, Relative Humidity.	RMSE:0.089
Curceac et al., (2019)	Hourly Temperature	RMSE: 0.7
Ortiz-García et al., (2012)	Relative humidity, Precipitation Pressure, Global radiation, Air temperature, Wind speed and Wind direction	RMSE: 0.61

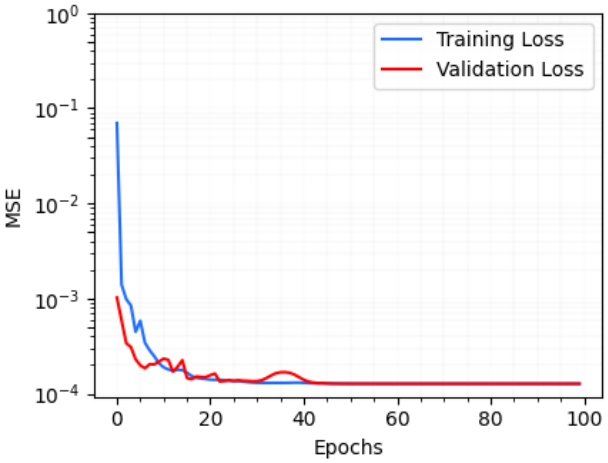


Figure 5: Training curved for the Jena dataset, showing the variation of loss with epoch number

Table 5: Performance of the proposed MIRNet model for single-hour ahead forecasting on Jena dataset

Performance					
Validation			**Test**		
MSE ($x10^{-4}$)	MAE ($x10^{-3}$)	MAPE (%)	MSE ($x10^{-4}$)	MAE ($x10^{-3}$)	MAPE (%)
1.23	7.62	1.31	1.23	7.69	1.36

Table 6: Performance of existing models on single-step ahead temperature forecast on Jena dataset

Study	Input /Other Details	Performance
Wang et al (2022)	72 hours of water vapour content, specific humidity, wind speed, and air pressure, and temperature	MSE: 2.12 x 10-5
Ko et al (2020)	Temperature	MSE: 1.8 x 10-4
Fang & Yuan (2019)	14 different quantities, such as air temperature, atmospheric pressure, humidity, and so on for single and multi-step temperature forecast	MSE: 0.018 MAPE: 3.56%

The power consumption of the Raspberry Pi 4 device during different states during the test were measured, and presented in Table 8. For the tests, the MIRNet, running on the Raspberry pi, was used for N-hours ahead predictions, where N=1,2,..,24. Since MIRNet only makes one prediction at a time, to predict N hours ahead, the model carried out N different y hours ahead for each N, where y = 1,...,N. The cumulative time required for prediction for all values of N up to 24 are presented in Figure 7. Finally, using the values in Table 8 and Figure 7, the average power required to run the Raspberry device such that it predicted H hours ahead, where H ∈{1,4,8,12,16,20,24} were calculated, and presented in Table 9.

https://dx.doi.org/10.4314/fuoyejet.v9i1.2 engineering.fuoye.edu.ng/journal

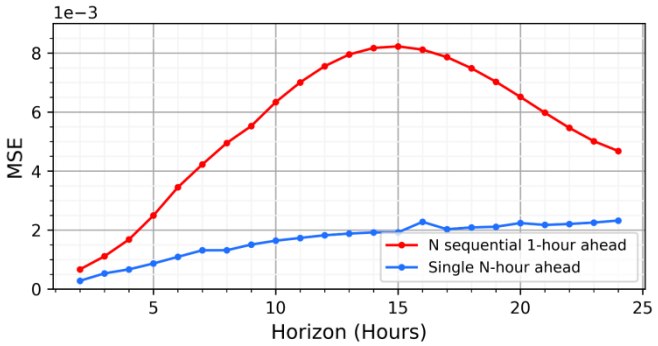


Figure 6: Performance of MIRNet model for N sequential 1-hour ahead, and single Nth-hour ahead prediction using the Jena dataset

Table 7: Results from existing multi-step models tested on the Jena dataset

Study	Details	Number of hours	Performance
Fang & Yuan (2019)	14 different quantities, such as air temperature, humidity, atmospheric pressure, and so on for single and multistep temperature forecast	10	MAE: 0.15 RMSE: 0.192
Dixon (2022)	20 steps of 14 different quantities (such as air temperature, atmospheric pressure, humidity, wind direction etc) for temperature predictions	10	MSE: 2.107 MAE: 1.097
Utku and Can (2022)	pressure, temperature, saturation vapour pressure, vapour pressure deficit, specific humidity, airtight, and wind speed as input for temperature prediction	12	MSE: 0.035 MAE: 0.126 RMSE: 0.189
Zhang et al (2021)	The temperature, pressure, and air density data of the past 24 hours as input	24	MSE: 0.000281
Li et al (2023)	21 meteorological indicators including, the amount of rain, humidity Temperature as the target, variable	48	MSE: 0.3216 MAE: 0.3433

Table 8: Power consumption for different states of the Raspberry Pi 4

SN	State	Average power consumption (W)
1	Boot	3.24
2	Idle	2.09
3	Model Loading	2.87
4	Prediction	3.33
6	Low-power mode	0.10

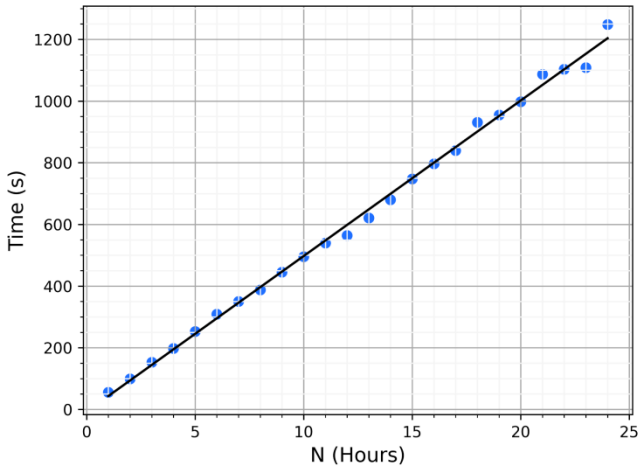


Figure 7: Time required to predict N total hours ahead on the Raspberry Pi device for 1≤N≤24

Table 9: Average power consumption by the Raspberry Pi device for selected number of hours ahead

H Hours Ahead (hours)	Average Power consumption (W)
1	0.152
4	0.279
8	0.450
12	0.608
16	0.816
20	0.997
24	1.221

3.2 Discussion

The immediate impression from the comparison of the IfeData performance in Table 3 with existing results in Table 4 is that MIRNet performed worse than existing models. There is always a suspicion, however, that the performance was held back by the quality of the IfeData dataset itself. A review of the IfeData records reveals a high of 56.1◦C on 18th October, 2022. Such a temperature was never recorded on the ground in Ile-Ife, and represented the sort of noisy data that is likely caused by the mode of data acquisition.

Comparison of MIRNet's single-step prediction on the Jena dataset with previous studies on the same dataset reveals a more objective view of the performance of the model. At first glance, MIRNet outperforms all but one (Wang et al., 2022) existing single-step model. A closer look at the Wang et al. model however raises a number of questions. While the reported performance from that paper would ordinarily be taken at face value, "Figure 8" of the paper is inconsistent with the claimed MSE. It can be seen at a glance that none of the three models in the figure were tracking the true temperature particularly well.

In order to investigate this further, the figure from Wang et al., (2022) was digitised using the PlotDigitizer online application available at https://plotdigitizer.com, after which the MSE for just that segment was calculated. Since the statistical properties of the Jena dataset are known, we were able to determine that the best Wang et al. model achieved an MSE of 4.3×10^{-4} for the segment. While it must be noted that a model does not necessarily have to perform equally well for all segments of data, a difference for any segment of over an

https://dx.doi.org/10.4314/fuoyejet.v9i1.2 engineering.fuoye.edu.ng/journal

order of magnitude (2.12×10^{-5} versus 4.3×10^{-4}) between the claimed MSE and the verified MSE must raise questions.

The particular Jena dataset points used in the above-mentioned Figure 8 of Wang et al. (2022) were determined using a simple algorithm. They turned out to be points in a 24-hour segment starting from the 50,095th hour of the Jena dataset. MIRNet was used to predict that same segment, and Figure 8 shows a comparison between MIRNet and Wang et al.'s predictions. MIRNet's superior performance is evident, in line with MIRNet's MSE of 1.2×10^{-4}. Finally, it should be noted that while the source code for the Wang et al. study is not available, the MIRNet code is available in its entirety online at https://github.com/Adedayo19/MIRNet. Based on this, it can be concluded that MIRNet provides the best verifiable single-step prediction performance for the Jena dataset in the literature till date.

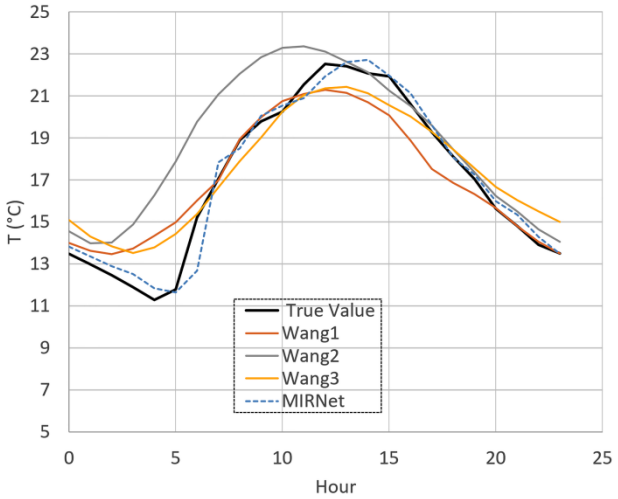


Figure 8: comparison of predictive performance of MIRNet versus those of the Wang paper for the 24-hour segment starting from hour 50,095

In terms of its multi-hour prediction capability, while MIRNet compares favourably with many existing models, it is handily bested by the Zhang et al (2021) paper. This was not unexpected. MIRNet was designed originally as a single-step predictor. When carrying out Nth-hour ahead prediction, MIRNet predicts the temperature for only the Nth hour ahead. In contrast, true N-hours ahead predictors predict the temperature not just for the Nth hour ahead, but for all the hours between the current time, and Nth hour ahead. In order to do this, MIRNet currently requires N separate predictions of y hours ahead, where $y \in \{1,...,N\}$.
This points to future improvements that need to be made on MIRNet. Although it has a passably good performance for multi-step prediction, it achieves this at the cost of excessive numbers of computations. The performance of MIRNet on the embedded device is satisfactory, as shown in Table 9. Even in the worst-case scenario where the prediction of the entire next 24 hours has to be repeated every hour, the average power consumed is 1.22 W, making it very feasible to run inference on the WeatherWAN network or similar low-power use cases.
.

4 Conclusion

MIRNet has been developed to be a compact framework mainly for the purpose of deployment on low-powered and computational resource-constrained embedded devices with limited storage/memory capacity, like Raspberry Pi, without sacrificing temperature forecasting accuracy. MIRNet was trained and verified with IfeData and Jena Datasets. The performance of the implemented framework promises improved forecasting accuracy as a single-step predictor with reasonable power consumption. A MSE of 0.000123 was achieved on the Jena dataset. This makes the system particularly attractive for deployment in low power, low cost weather station networks such as the WeatherWAN system.

Acknowledgement

This study was supported by the Nigerian Communications Commission through a 2020 "Telecommunications-based Research Innovations for Academics in Nigerian Tertiary Institutions" research grant. The Nvidia Quadro RTX 8000 GPU utilised for training all models in the study was kindly donated by the NVIDIA Corporation of Santa Clara, USA. The authors would like to thank Mr. Mujahid Yunus for his contributions, particularly during the testing using the Raspberry Pi device.

References

Abdel-Aal, R. E. (2004). Hourly temperature forecasting using abductive networks. Engineering Applications of Artificial Intelligence,17(5):543-556. https://doi.org/10.1016/j.engappai.2004.04.002

Abubakar, A., Chiroma, H., Zeki, A., & Uddin, M. (2016). Utilising key climate element variability for the prediction of future climate change using a support vector machine model. International Journal of Global Warming, 9(20):129-151. https://doi.org/10.1504/IJGW.2016.074952

Al Sadeque, Z. & Bui, F. M. (2020). A Deep Learning Approach to Predict Weather Data Using Cascaded LSTM Network. In IEEE Canadian Conference on Electrical and Computer Engineering. 1-5. https://doi.org/10.1109/CCECE47787.2020.9255716

Chevalier, R. F., Hoogenboom, G., McClendon, R. W., & Paz, J. A. (2011). Support vector regression with reduced training sets for air temperature prediction: A comparison with artificial neural networks. Neural Computing and Applications, 20(1): 151-159. https://doi.org/10.1007/s00521-010-0363-y

Curceac, S., Ternynck, C., Ouarda, T. B. M. J., Chebana, F., & Niang, S. D. (2019). Short-term air temperature forecasting using Nonparametric Functional Data Analysis and SARMA models. Environmental Modelling and Software, 111: 394-408. https://doi.org/10.1016/j.envsoft.2018.09.017.

Debelee, T. G., & Ayano, Y. M. (2022). Deep Learning Models for Audio Processing Applications Under Resource-Constrained Devices: A Survey. In Pan African Conference on Artificial Intelligence, 209-232. Cham: Springer Nature Switzerland.

Dirmeyer, P. A. (2003). The role of the land surface background state in climate predictability. Journal of Hydrometeorology, 4(3): 599-610.

Dixon, M. (2022). Industrial forecasting with exponentially smoothed recurrent neural networks,Technometrics, 64(1): 114-124. https://doi.org/10.1080/00401706.2021.1921035\

Fang, X., & Yuan, Z. (2019). Performance enhancing techniques for deep learning models in time series forecasting. Engineering Applications of Artificial Intelligence, 85: 533-542. https://doi.org/10.1016/j.engappai.2019.07.011

Graves, A. & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. In Proceedings of IEEE International Joint Conference on Neural Networks, 4:2047-2052. /home/kermorvant/refbase_files/2005/graves/1055_graves_schmidhuber2005.pdf

Haque, E., Tabassum, S., & Hossain, E. (2021). A Comparative Analysis of Deep Neural Networks for Hourly Temperature Forecasting. IEEE Access. 160646-160660. https://doi.org/10.1109/ACCESS.2021.3131533

Hassani, H., Silva, E. S., Gupta, R., & Das, S. (2018). Predicting global temperature anomaly: A definitive investigation using an ensemble of twelve competing forecasting models. Physica A: Statistical Mechanics and Its Applications, 509: 121-139. https://doi.org/10.1016/j.physa.2018.05.147

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 770–778. https://doi.org/10.1109/CVPR.2016.90

Hewage, P., Trovati, M., Pereira, E., & Behera, A. (2021). Deep learning-based effective fine-grained weather forecasting model. Pattern Analysis and Applications, 24(1): 343-366. https://doi.org/10.1007/s10044-020-00898-1

Hou, J., Wang, Y., Zhou, J., & Tian, Q. (2022). Prediction of hourly air temperature based on CNN–LSTM, Geomatics. Natural Hazards and Risk, 13(1): 1962-1986. https://doi.org/10.1080/19475705.2022.2102942

Hossain, M., Rekabdar, B., Louis, S. J., & Dascalu, S. (2015). Forecasting the weather of Nevada: A deep learning approach. In IEEE international joint conference on neural networks (IJCNN). 1-6. https://doi.org/10.1109/IJCNN.2015.7280812

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 4700-4708. https://doi.org/10.1109/CVPR.2017.243

Ko, M. S., Lee, K., Kim, J. K., Hong, C. W., Dong, Z. Y., & Hur, K. (2020). Deep concatenated residual network with bidirectional LSTM for one-hour-ahead wind power forecasting. IEEE Transactions on Sustainable Energy, 12(2):1321-1335. https://doi.org/10.1109/TSTE.2020.3043884

Li, C., Zhang, Y., & Zhao, G. (2019). Deep learning with long short-term memory networks for air temperature predictions. In International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM). 243-249. https://doi.org/10.1109/AIAM48774.2019.00056

Li, Y., Lu, X., Xiong, H., Tang, J., Su, J., Jin, B., & Dou, D. (2023). Towards Long-Term Time-Series Forecasting: Feature, Pattern, and Distribution. arXiv preprint arXiv:2301.02068. https://doi.org/10.48550/arXiv.2301.02068

Li, H. C., Deng, Z. Y., Chiang, H. H. (2020). Lightweight and Resource-Constrained Learning Network for Face Recognition with Performance Optimization. Sensors (Basel). 20(21): 6114. doi: 10.3390/s20216114. PMID: 33121101; PMCID: PMC7662273.

Liang, S., Wang, D., Wu, J., Wang, R., & Wang, R. (2021). Method of bidirectional LSTM modelling for the atmospheric temperature. Intelligent Automation and Soft Computing, 30(20): 701-714. https://doi.org/10.32604/iasc.2021.020010

Kumar, R. (2023). Data Mining Techniques for Weather Forecasting. *The Visual Computer*, *15*(5): 56-63.

Ortiz-García, E. G., Salcedo-Sanz, S., Casanova-Mateo, C., Paniagua-Tineo, A., & Portilla-Figueras, J. A. (2012). Accurate local very short-term temperature prediction based on synoptic situation Support Vector Regression banks. Atmospheric Research 107: 1- 8. https://doi.org/10.1016/j.atmosres.2011.10.013

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11): 2673-2681. https://doi.org/10.1109/78.650093

Sekertekin, A., Bilgili, M., Arslan, N., Yildirim, A., Celebi, K., & Ozbek, A. (2021). Short-term air temperature prediction by adaptive neuro-fuzzy inference system (ANFIS) and long short-term memory (LSTM) network. Meteorology and Atmospheric Physics, 133:943-959. https://doi.org/10.1061/(ASCE)IR.1943-4774.0000298

Shen, Z., Zhang, Y., Lu, J., Xu, J., & Xiao, G. (2020). A novel time series forecasting model with deep learning. Neurocomputing, 396: 302–313. https://doi.org/10.1016/j.neucom.2018.12.084

Smith, B. A., McClendon, R. W., & Hoogenboom, G. (2006). Improving air temperature prediction with artificial neural networks. International Journal of Computational Intelligence, 3(3):179-186.

Smith, B. A., Hoogenboom, G., & McClendon, R. W. (2009). Artificial neural networks for automated year-round temperature prediction, Computers and Electronics in Agriculture, 68(1): 52-61. https://doi.org/10.1016/j.compag.2009.04.003

Utku, A., & Can, U. (2023). An efficient hybrid weather prediction model based on deep learning. International Journal of Environmental Science and Technology. 1-14. https://doi.org/10.1007/s13762-023-05092-4

Wang, J., Lin, L., Teng, Z., & Zhang, Y. (2022). Multitask Learning Based on Improved Uncertainty Weighted Loss for Multi-Parameter Meteorological Data Prediction. Atmosphere, 13(6): 989. https://doi.org/10.3390/atmos13060989

Zhang, K., Sun, M., Han, T. X., Yuan, X., Guo, L., & Liu, T. (2018). Residual Networks of Residual Networks: Multilevel Residual Networks. IEEE Transactions on Circuits and Systems for Video Technology, 28(6), 1303–1314. https://doi.org/10.1109/TCSVT.2017.2654543

Zhang, F., Gao, X., Zhang, S., Wang, Q., & Lin, L. (2021). Atmospheric Environment Data Generation Method Based on Stacked LSTM-GRU. 15th IEEE International Conference on Electronic Measurement & Instruments (ICEMI). 17-26. https://doi.org/10.1109/icemi52946.2021.9679551

Zhao, L., Wang, J., Li, X., Tu, Z., & Zeng, W. (2016). On the connection of deep fusion to ensembling. *arXiv preprint arXiv:1611.07718*.