eng

MDPI

Article

Hybrid LSTM-CNN Model with Temporal Feature Engineering and Genetic Algorithm Optimization for Temperature Forecasting

Farrukh Hafeez [1], Zeeshan Ahmad Arfeen [2], Touqeer Ahmed Jumani [3,*], Muhammad I. Masud [4], Nasser Alkhaldi [1], Ameer Azhar [1], Mohammed Aman [5] and Mehreen Kausar Azam [6]

[1] Department of Electrical Engineering, Jubail Industrial College, Al Jubail 35718, Saudi Arabia; farrukhhafeez82@hotmail.com (F.H.)
[2] Department of Electrical Engineering, The Islamia University of Bahawalpur (IUB), Bahawalpur 63100, Pakistan; zeeshan.arfeen@iub.edu.pk
[3] Department of Electrical Engineering and Computer Science, College of Engineering, A'Sharqiyah University, Ibra 400, Oman
[4] Department of Electrical Engineering, College of Engineering, University of Business and Technology, Jeddah 21361, Saudi Arabia; m.masud@ubt.edu.sa
[5] Department of Industrial Engineering, College of Engineering, University of Business and Technology, Jeddah 21361, Saudi Arabia; m.aman@ubt.edu.sa
[6] Department of Industrial Manufacturing & Engineering, Pakistan Navy Engineering College, National University of Sciences and Technology (NUST), Karachi 75350, Pakistan
* Correspondence: touqeer.ahmed@asu.edu.om

Abstract

The accurate temperature forecasting system provides essential benefits for managing outdoor activities, controlling electricity consumption, and ensuring public health and safety in areas with extreme heat. The researchers developed a hybrid Long Short-Term Memory–Convolutional Neural Network (LSTM–CNN) model that uses daily time-series data from Makkah, Saudi Arabia, to enhance short-term temperature prediction results. The forecasting task is defined as daily multi-step prediction, generating 1-day, 3-day, and 6-day ahead temperature forecasts. The proposed model combines LSTM networks to capture long-term temporal dependencies and CNN to extract short-term variations. The system uses temporal features, lag features, and rolling statistical features to improve data representation, while Genetic Algorithm (GA) optimization handles the selection of model hyperparameters. The framework uses ten-fold cross-validation to test its performance, ensuring consistent performance across all testing scenarios. The results demonstrate strong predictive accuracy, with the GA-optimized model achieving a Mean Absolute Error (MAE) of 0.55 °C for 1-day forecasts and 1.28 °C for 6-day forecasts, with R^2 values reaching up to 0.98. The proposed model outperformed Autoregressive Integrated Moving Average (ARIMA), LSTM, and Transformer models during benchmark tests, providing better forecasting results across various time intervals. These findings indicate that the proposed model demonstrates accurate and reliable temperature forecasting performance for arid to semi-arid climatic conditions.

Keywords: time series forecasting; temperature prediction; hybrid deep learning; LSTM–CNN; genetic algorithm optimization; temporal feature engineering; meteorological data; extreme climate

Check for updates

Academic Editors: Akos Odry, Juvenal Rodriguez-Resendiz, Marco Antonio Aceves-Fernandez and José Manuel Álvarez-Alvarado

Received: 4 March 2026
Revised: 2 May 2026
Accepted: 4 May 2026
Published: 8 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Accurate forecasting of temperature is essential in addressing the effects of extreme weather conditions, particularly in areas such as Makkah, Saudi Arabia, where tempera-

tures change drastically [1]. Temperatures in Makkah exceed 45 °C during the summer months. Forecasting temperatures enables authorities to plan heat-mitigation measures and public safety strategies and to ensure proper preparation for harsh weather conditions, as heat-related illnesses are significant concerns in such environments. The challenges in making accurate forecasts are far from trivial, and temperature forecasting is a well-established field of study [2]. When combined with extreme heat, temperature variability makes forecasting difficult and increases safety risks in outdoor environments. Makkah experiences harsh summer temperatures exceeding 45 °C, with additional increases during daytime hours. Forecasting extreme heat episodes is essential for ensuring preparation in logistics and safety measures, since high levels of heat cause heat-related stress, dehydration, and heatstroke [3]. Predicting when and where these extremes are most likely to occur allows preventive measures to be implemented in advance. Traditional models of temperature forecasting are generally based on less complex environments and thus do not capture the full complexity of environmental dynamics [4]. Another challenge in temperature forecasting is understanding how different meteorological variables interact. Temperature is affected by three different time scales, which include daily temperature changes, seasonal variations, and specific weather conditions such as wind speed, humidity, solar radiation, and air pressure. Traditional models tend to overlook these interdependencies, which leads to inaccurate forecasting results [5]. In many regions, temperature is related to humidity, and high temperatures usually align with lower humidity in dry desert climates such as that of Makkah. A valid forecast should consider this relationship to avoid incorrect predictions [4].

Physics-based, statistical, and neural network/deep learning models are often used for weather forecasting. These approaches calculate future temperature based on physical characteristics such as solar irradiation, wind speed, humidity, precipitation, and cloud cover [5]. Physics-based models use sensor measurements from many sources to calculate temperature, which varies greatly by location. These models perform better for long-term temperature time-series forecasts than for short-term projections. Autoregressive Integrated Moving Average (ARIMA) uses time-series analysis to anticipate long-term changes over daily and monthly time horizons [6]. ARIMA is a popular linear statistical method for time-series forecasting and regression analysis. ARIMA's inability to adequately capture strong seasonality is one of its main weaknesses, since temperature exhibits strong seasonal patterns [7].

Recent work has applied hybrid deep learning with optimization in AIoT systems, including two-stage models for industrial failure prediction [8] and particle swarm optimization–based frameworks for leakage current classification [9]. These studies further support the effectiveness of combining deep learning with optimization techniques. Researchers have investigated CNN–LSTM approaches for temperature forecasting due to their progress in recent studies. CNN–LSTM models show strong performance in time-series studies as well as environmental prediction tasks, which has led researchers to apply them to temperature forecasting problems [10]. For instance, Recent work has applied hybrid deep learning with optimization in AIoT systems, including two-stage models for industrial failure prediction [11] and particle swarm optimization–based frameworks for leakage current classification [12]. These studies further support the effectiveness of combining deep learning with optimization techniques.

Gong et al. [13] proposed a CNN–LSTM model for temperature forecasting, where CNN extracts spatial features and LSTM captures temporal dependencies. The model achieved better prediction outcomes through its use of advanced network systems because it worked with weather data. The research focused on architectural design while it excluded advanced feature engineering methods that included lag and rolling statistics. Uluocak and

Bilgili [14] developed LSTM–CNN and GRU–CNN hybrid models for daily air temperature forecasting. The study found that hybrid models performed better than single deep learning models when they needed to identify non-linear temperature changes. Optimal use of genetic algorithms is required for optimization of hyperparameters, but unfortunately was not covered in this study. Wang and Wang [15] introduced a GA-optimized CNN–LSTM hybrid model for weather prediction. The forecasting accuracy improved when genetic algorithms worked together with model parameter selection methods. The study focused on optimization methods but did not test which feature engineering techniques provide the best model performance results.

Zhang et al. [16] developed a framework based on CNN and LSTM which utilizes spatial and temporal feature extraction to forecast energy system temperature changes. The model showed better ability to model intricate temperature changes than other models. The research studies one particular field and its results do not apply to situations with extremely cold climate conditions. Yasavoli et al. [17] created a hybrid deep learning system which combines CNN and recurrent networks to predict future weather temperatures. The model produced dependable forecasting results when tested on different datasets. However, the study does not provide a complete assessment of feature importance and does not include optimization methods such as genetic algorithms. Yu et al. [18] suggested a GA-CNN-LSTM hybrid model for the prediction of humidity and temperature, where CNN extracts local features and LSTM captures temporal dependencies, while GA optimizes hyperparameters. The model demonstrated improved prediction accuracy compared to conventional approaches. However, the study focuses on model optimization but lacks detailed feature engineering, including temporal, lag, and rolling statistical features. Çınarer [19] developed a hybrid deep learning model which combines CNN with LSTM and stacking ensemble methods to predict global temperature changes. The model uses CNN to extract features from data while LSTM processes temporal information. The approach requires more computational resources but does not incorporate advanced feature engineering methods, which reduces both interpretability and efficiency of the system.

Research on CNN–LSTM temperature forecasting has progressed in developing new forecasting systems but still requires improved feature extraction methods, as most studies focus primarily on model architectures. The integration of temporal, lag, and rolling statistical features remains limited. Additionally, the interaction between genetic algorithms and advanced feature engineering has not been fully explored. Most studies evaluate their methods under moderate conditions, while performance in extreme weather conditions remains largely untested. Therefore, a comprehensive framework that combines hybrid modeling with feature engineering and optimization methods is required.

To address these limitations, this study proposes a CNN–LSTM framework enhanced with temporal, lag, and rolling statistical feature engineering and optimized using GA for adaptive hyperparameter tuning to improve temperature forecasting accuracy under extreme climatic conditions.

The research presents its main contributions which include three major findings as follows:

- The hybrid LSTM–CNN framework is developed to forecast daily temperatures for multiple future time periods by capturing both long-term temporal patterns and short-term variations.
- A comprehensive temporal feature engineering method is introduced, which uses lag features and rolling statistical measures to enhance temperature predictions for daily time-series data.
- The system applies GA-based optimization for automatic hyperparameter tuning, which results in improved predictive performance and enhanced model generalization capabilities.

- The proposed framework demonstrates better forecasting accuracy than baseline models when tested on daily meteorological data from extreme climatic conditions.

The paper has three main sections that organize its content. The development of the hybrid framework and the GA optimization is explained in the methodology section. The results and discussion section presents performance results across different forecasting periods. The paper ends with important findings and recommendations for upcoming research work.

2. Research Methodology

The research presents a hybrid method which combines LSTM and CNN models that have been enhanced by GA optimization and temporal feature engineering work. The temperature time-series data collection process begins with researchers obtaining environmental data from Makkah, which they will use for their study. The initial data preprocessing phase begins with missing value replacement, which the team performs before they start to scale, select features, and extract temporal features, lag features, and rolling statistical features, which will enhance model performance. The model design uses LSTM to predict long-term trends and applies CNN to identify short-term patterns in the data. The hybrid model uses GA to conduct hyperparameter optimization. The research uses GA as an external optimization tool for the LSTM–CNN model, which is demonstrated in Figure 1. GA creates candidate hyperparameter sets (which include learning rate, batch size, and number of layers) that researchers test by training the LSTM–CNN model. The prediction error results from MAE and RMSE calculations are used by the fitness function. The GA selection process uses its crossover and mutation operations to select optimal hyperparameter settings through its selection process. The final LSTM–CNN model is trained using GA-optimized hyperparameters, which result in better forecasting accuracy and generalization performance. The model evaluation process uses standard metrics, which include MAE, RMSE, and R^2, to assess performance across multiple forecasting time frames. The study employs its forecasting task through a multi-step-ahead prediction framework, which uses daily time-series data to forecast temperature values at 1-day, 3-day, and 6-day ahead horizons.

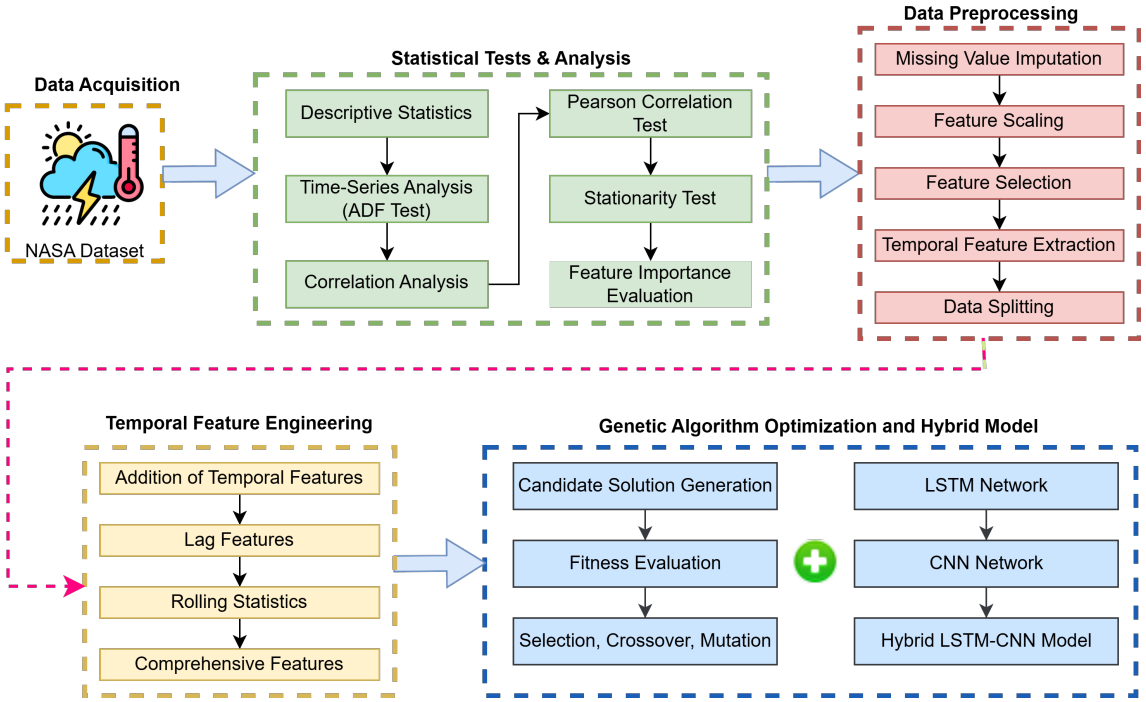


Figure 1. Proposed framework showing data preprocessing, feature engineering, and GA-based hyperparameter optimization integrated with the hybrid LSTM–CNN model.

2.1. Dataset Description

The research uses a dataset which NASA provided through its website to analyze temperature time-series data and environmental data for Makkah, which spans from May 1995 until May 2024. The dataset is based on daily observations and includes various meteorological parameters such as maximum and minimum temperatures at 2 m above ground level, surface shortwave radiation, relative humidity, and wind speed, among others. The dataset also allows analysis of long-term temperature trends, including local variations in maximums, minimums, and averages, along with their associated environmental factors such as humidity, wind speed, and radiation, which are indispensable for predicting complex temperature patterns. The dataset provides the maximum daily temperatures recorded (T2M_MAX) for Makkah between May 1995 and March 2024. The time series displays major seasonal temperature changes, which result in temperature peaks that reach above 40 °C during the months with the highest temperatures.

The variables include maximum temperature at 2 m (T2M_MAX), minimum temperature (T2M_MIN), average temperature (T2M), relative humidity (RH2M), wind speed (WS2M), surface shortwave radiation (ALL-SKY_SFC_SW_DWN), and surface pressure (PS). The maximum temperature in Makkah generally ranges between 19.17 and 48.66 °C, and the minimum temperature (T2M_MIN) ranges from 7.36 to 32.58 °C. The T2M temperature stays within the limits that extend from 13.29 °C to 39.50 °C. The relative humidity levels start at 5.48% and reach a peak of 84.09%, while wind speeds range from 1.13 m/s to 7.77 m/s. These variables essentially highlight the variation of weather, which characterizes extreme temperatures and comparatively lower humidity, typical of this region. Table 1 summarizes the main statistical properties of the dataset.

Table 1. Descriptive statistics of key meteorological variables used in the study.

Statistic	T2M_MAX (°C)	T2M_MIN (°C)	T2M (°C)	RH2M (%)	WS2M (m/s)
Count	10624	10624	10624	10624	10624
Mean	36.77	22.72	28.95	34.09	3.02
Std. Dev.	5.68	5.02	5.27	15.01	0.65
Min	19.17	7.36	13.29	5.48	1.13
25th Percentile	32.04	18.67	24.40	21.82	2.58
50th Percentile	37.64	23.59	29.88	31.51	2.92
75th Percentile	41.44	27.09	33.51	44.70	3.34
Max	48.66	32.58	39.50	84.09	7.77

The highest annual temperatures can be used to estimate Makkah's temperature distribution. This indicates that fairly high levels of variability exist, with some years displaying extreme peaks above 45 °C, especially during the hot months. The capability to capture these seasonal fluctuations makes the dataset very suitable for modeling temperature patterns. Figure 2 presents the maximum temperature measurements, showing annual variation.

Exploring relationships using correlation matrices is very useful for determining the relationships among variables in meteorological datasets. The correlation matrix, presented in Figure 3, shows strong relationships among some meteorological variables. The T2M_MAX represents the target variable and is not included as an input feature; the correlation analysis is used solely to understand relationships among variables and guide feature selection. The data can be used extensively in future temperature predictions, especially as it shows strong correlations between T2M_MAX and T2M_MIN, and between

T2M_MAX and T2M, which relate maximum and minimum temperatures with average temperature, respectively, suggesting predictable temperature patterns. A negative correlation between temperature and RH2M (humidity) indicates that higher temperatures are associated with lower humidity, which is typically observed in arid areas like Makkah. The dataset incorporates other important variables such as radiation and wind speed. Radiation (ALLSKY_SFC_SW_DWN) shows a strong positive correlation with temperature, while wind speed is not significantly correlated with temperature but still provides useful information about local weather conditions.

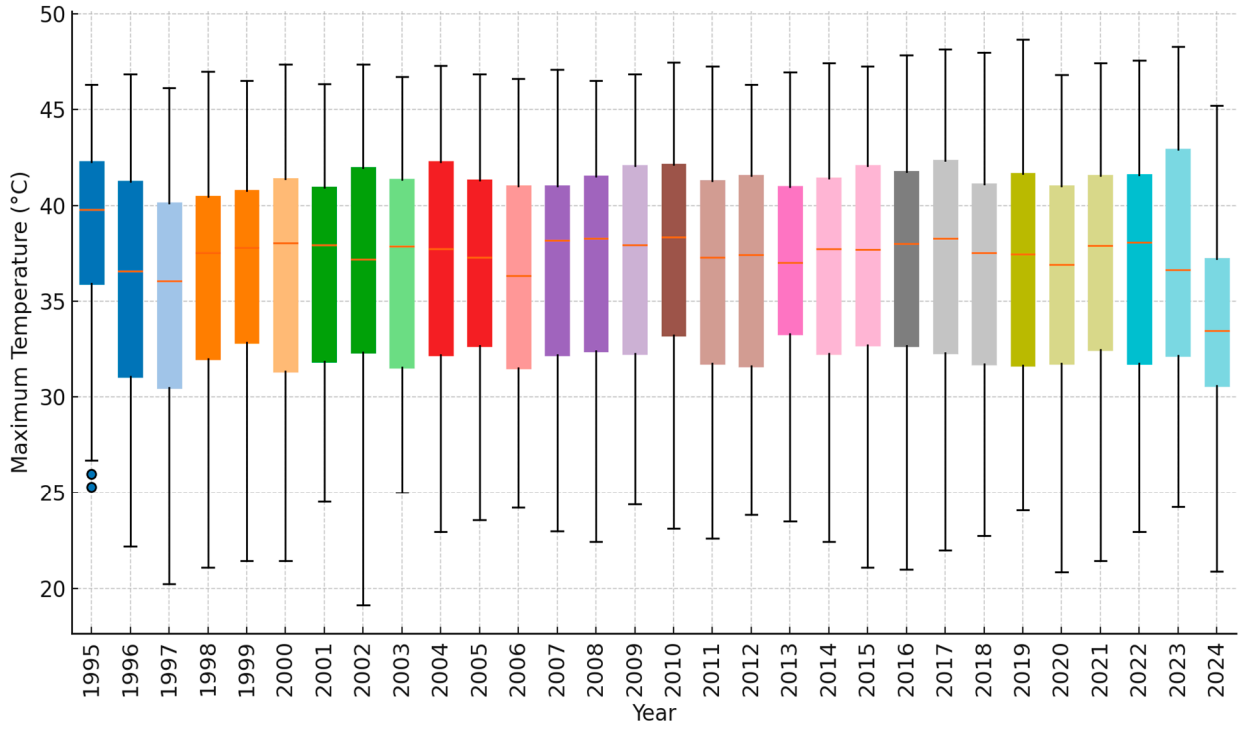


Figure 2. Analysis of maximum temperature trends in Makkah (1995–2024).

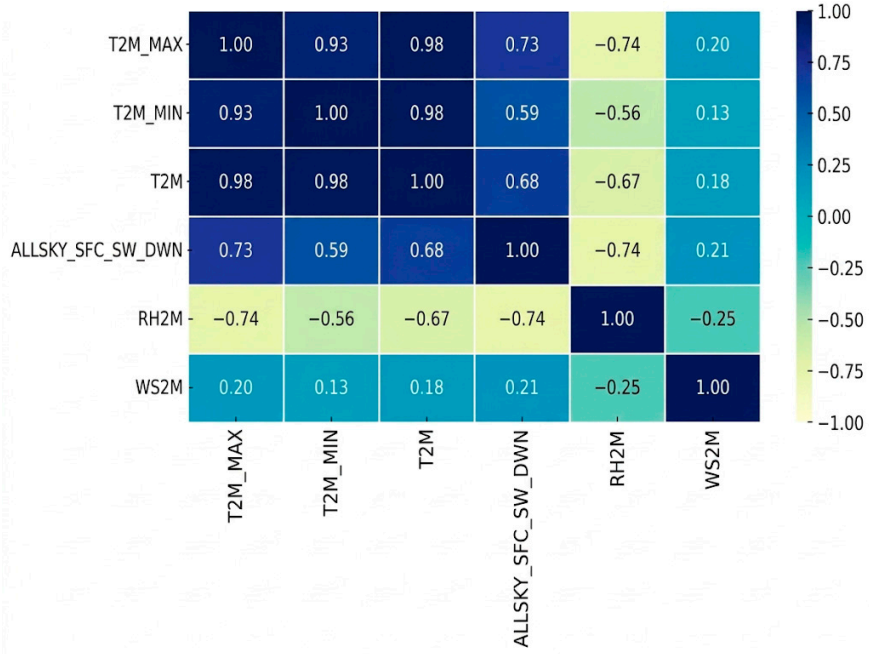


Figure 3. Correlation matrix highlighting inter-relationships among key meteorological variables.

The statistical tests on the dataset are used to determine its capacity for time-series modeling, which would yield precise temperature predictions and to study the relationships between major meteorological elements. The Augmented Dickey–Fuller (ADF) test is conducted to assess the stationarity of the T2M_MAX (maximum temperature) series. The results in Table 2 indicate that the T2M_MAX series is stationary, with a test statistic of −8.14, significantly lower than the critical values at the 1%, 5%, and 10% levels. The null hypothesis of non-stationarity is rejected, with a p-value of 1.03×10^{-12}, thus confirming that the temperature data can be used for forecasting without differencing.

Table 2. Augmented Dickey–Fuller (ADF) test results for T2M_MAX series.

Statistic	Value
ADF Test Statistic	−8.14
p-value	1.03×10^{-12}
Number of Lags	38
Number of Observations	10,585
Critical Value (1%)	−3.43
Critical Value (5%)	−2.86
Critical Value (10%)	−2.57

Pearson correlation analysis of T2M_MAX against other meteorological factors, such as RH2M (humidity) and ALLSKY_SFC_SW_DWN (surface shortwave radiation), showed a strong negative correlation of −0.74 between T2M_MAX and RH2M, indicating that an increase in maximum temperature corresponds to a decrease in relative humidity. A strong positive correlation of 0.73 between T2M_MAX and ALLSKY_SFC_SW_DWN indicates that increases in temperature correspond to increases in surface radiation levels. Both correlations were significant at $p < 0.001$, thereby confirming that the findings are statistically significant. The Pearson correlation coefficients between T2M_MAX and the selected variables are presented in Table 3.

Table 3. Pearson correlation coefficients between T2M_MAX and key variables.

Variable	Correlation with T2M_MAX
T2M_MIN	0.93
T2M	0.98
ALLSKY_SFC_SW_DWN	0.73
RH2M	−0.74
WS2M	0.20

2.2. Data Preprocessing

Data preprocessing is crucial to ensure proper preparation of the dataset for effective temperature forecasting. Handling missing data is the first step, where missing values are filled using the mean of the respective columns. Since the proportion of missing values in the dataset is very small, mean imputation is adopted as a simple and efficient approach; however, it is acknowledged that more advanced time-series imputation methods (e.g., interpolation) could better preserve temporal dependencies and may be explored in future work. Figure 4 depicts the data preprocessing workflow. This ensures that there are no missing values in the data and maintains the integrity of the dataset for modeling.

The process of feature scaling establishes a standard measurement system which allows for the comparison of all data features through two methods, normalization and standardization to zero mean and unit variance. Domain expertise, together with Pearson correlation analysis, establishes the feature selection method. The study selects T2M_MIN, T2M, RH2M, and ALLSKY_SFC_SW_DWN as temperature change predictors because

these variables exhibit strong correlation with the target temperature (T2M_MAX). The study removes WS2M because it exhibits weak correlation, which helps to decrease noise while enhancing model performance. The study selects variables through Pearson correlation analysis, which shows that variables with stronger absolute correlation values are selected as part of the final feature set that includes T2M_MIN, T2M, RH2M, and ALLSKY_SFC_SW_DWN. The process of temporal feature extraction creates time-based features from the datetime column through the extraction of day-of-the-week and month information, which identifies seasonal and periodic patterns. The research creates two separate dataset divisions, which include training data and testing data. The steps for data preprocessing establish data quality standards which create consistent data patterns that allow for precise temperature prediction. The temperature-related variables T2M, T2M_MIN, and T2M_MAX show strong correlations because these relationships represent the natural physical connections that exist among meteorological data. The study uses lagged features of all data elements, which include $t-1$ and $t-2$, to restrict future temperature predictions to information that existed before that time. Therefore, the model does not rely on contemporaneous variables, and the forecasting task remains non-trivial. Unlike linear models, deep learning architectures such as LSTM–CNN are less sensitive to multicollinearity, as they learn nonlinear feature representations rather than estimating independent coefficients. Formal multicollinearity diagnostics such as Variance Inflation Factor (VIF) are typically more relevant for linear regression models and were therefore not explicitly applied in this study. Furthermore, the non-trivial nature of the forecasting task is evidenced by the increase in prediction error across longer horizons (1-day to 6-day), indicating that the model is learning meaningful temporal patterns rather than relying on near-identical variables. The dataset, comprising 10,624 daily observations (May 1995–March 2024), is partitioned chronologically into training (70%), validation (15%), and testing (15%) sets, where earlier observations are used for training and later observations are reserved for validation and testing to prevent data leakage.

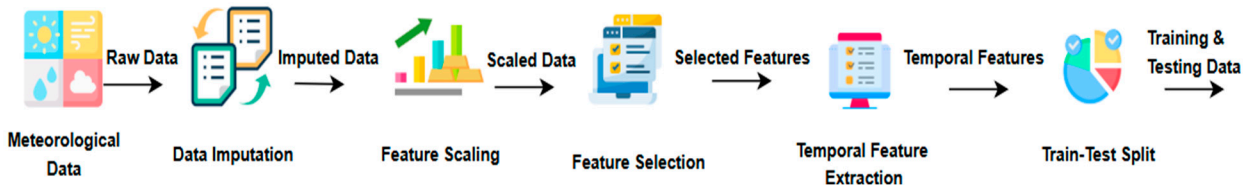


Figure 4. Data-processing workflow.

2.3. Temporal Feature Engineering

Temporal feature engineering was applied to a temperature forecasting model to improve accuracy. The process began with preprocessing the raw temperature data, then extended the feature set to capture long-term trends and short-term fluctuations.

The baseline model used only the preprocessed temperature signal, where temperature at time t is denoted as $T(t)$. The equation for the basic model is:

$$T(t) = f(T(t-1),\ T(t-2), \ldots, T(t-n)) \quad (1)$$

Temporal features are added to capture seasonal patterns. Since the dataset uses daily intervals, these features include td (day of the week) and tm (month of the year).The model is updated as:

$$T(t) = f(T(t-1),\ T(t-2), \ldots, T(t-n),\ t_d\ t_m) \quad (2)$$

The system uses rolling statistical features to track long-term trends while reducing short-term variations. The system uses rolling mean ($\mu(t,w)$) and rolling standard deviation ($\sigma(t,w)$) which both require a window size of (w) for their calculations.

$$\mu(t, w) = \frac{1}{w} \sum_{i=t-w+1}^{t} T(i) \tag{3}$$

$$\sigma(t, w) = \sqrt{\frac{1}{w} \sum_{i=t-w+1}^{t} (T(i) - \mu(t, w))^2} \tag{4}$$

Finally, incorporating lag features, temporal features, and rolling statistics, the complete model formulation is given by

$$T(t) = f(T(t-1), T(t-2), \ldots, T(t-n),\ t_d, t_m, \mu(t, w), \sigma(t, w)) \tag{5}$$

The model performs superior temperature forecasts because it can identify seasonal weather patterns and time-based weather relationships which require both lagging indicators and current time data and historical weather patterns. The study adopts a lag order of 7 (n), which allows the model to use data from the previous seven days as input to track short-term temporal patterns. The rolling window size (w) is set to 3, which is applied to calculate rolling mean and standard deviation features that help to decrease short-term fluctuations. These parameters were selected based on empirical considerations and domain knowledge of daily temperature patterns. Lag and rolling statistical features are computed for all selected input variables, including T2M_MIN, RH2M, and ALLSKY_SFC_SW_DWN, ensuring that both individual variable dynamics and inter-variable relationships are captured. To ensure a realistic daily forecasting setup and avoid data leakage, all input features are constructed exclusively from historical observations. Specifically, lagged values of meteorological variables (e.g., T2M(t − 1), T2M_MIN(t − 1), RH2M(t − 1), ALLSKY_SFC_SW_DWN(t − 1)) are used as model inputs, while the target variable corresponds to future daily values of T2M_MAX at forecasting horizons of t + 1 (1-day), t + 3 (3-day), and t + 6 (6-day). No same-day (t) measurements are included in the input feature set. This ensures a strictly causal and operationally valid forecasting framework.

2.4. Genetic Algorithm for Hyper-Parameter Optimization of the Hybrid Model

The hybrid model is trained after hyperparameter optimization using a Genetic Algorithm (GA). This process identifies the best combination of hyperparameters for effective model initialization and improved forecasting performance. The key hyperparameters of the system include learning rate η and batch size b and number of layers L as demonstrated in Figure 1. The system generates candidate solutions through the creation of a population which consists of different hyperparameter combinations for each candidate solution. The evaluation process for each candidate uses prediction error metrics which include MAE and RMSE as the fitness criteria.

$$(X) = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{6}$$

The selection process of this function prefers candidates who have achieved lower error values. The best performing candidates proceed to the reproduction process which produces new solutions through crossover between two parent solutions. Crossover combines two parent solutions X_1 and X_2 into offspring $X_{\text{offspring}}$, represented as:

$$X_{\text{offspring}} = \left(\eta_{\text{offspring}}, b_{\text{offspring}}, L_{\text{offspring}}\right) \tag{7}$$

After crossover, mutation slightly alters the offspring's hyperparameters by applying a small perturbation drawn from a normal distribution, represented as:

$$\eta_{\text{mutated}} = \eta_{\text{offspring}} + \Delta\eta,\ \Delta\eta \sim \mathcal{N}\left(0, \sigma_{\eta}^{2}\right) \quad (8)$$

After optimization, the hybrid model is trained using the selected hyperparameters. GA searches within predefined ranges for learning rate, batch size and number of layers, selected based on empirical experimentation and standard deep learning practices to ensure stable training and efficient convergence.

The GA runs with 20 individuals per population across 30 generations. A crossover rate of 0.8 is applied to combine candidate solutions, while a mutation rate of 0.1 is used to introduce diversity into the population. The hyperparameter search space includes the learning rate (0.0001–0.01), batch size (16–128), and the number of layers (1–3). As summarized in Table 4, the GA iteratively explores the defined search space and selects the optimal set of hyperparameters, which are then used to train the proposed hybrid model.

Table 4. Genetic Algorithm hyperparameters and search space.

Parameter	Value/Range
Population Size	20
Number of Generations	30
Crossover Rate	0.8
Mutation Rate	0.1
Learning Rate	0.0001 to 0.01
Batch Size	16 to 128
Number of Layers	1 to 3

The architectural configuration used in the model (e.g., 64 LSTM units, 32 CNN filters, kernel size = 3, as shown in Figure 5) was determined prior to GA optimization based on preliminary experimentation and kept fixed during the optimization process. This design choice limits the search space to key training hyperparameters while maintaining computational efficiency. The GA configuration parameters (population size, number of generations, crossover and mutation rates) were selected based on empirical considerations and commonly adopted practices. The optimal hyperparameter configuration identified by GA was used to train the final model.

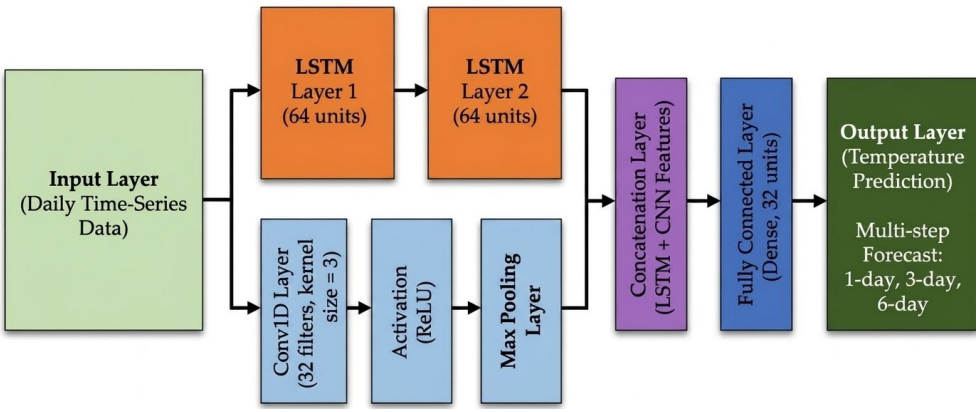


Figure 5. Hybrid LSTM–CNN architecture for temperature forecasting.

However, the GA-based optimization introduces additional computational overhead compared to baseline training. In the present configuration, the GA evaluates 600 candidate models (20 individuals × 30 generations), where each candidate requires full model training.

Experiments were conducted on a standard GPU-enabled environment (e.g., NVIDIA GPU with typical CPU and RAM support), and the total training time for a complete GA run is significantly higher than a single baseline model training due to repeated evaluations. Compared to fixed hyperparameter training and Random Search, the GA approach requires greater computational time because of its iterative population-based search strategy. This cost is incurred only during the offline training phase. Once the optimal hyperparameters are identified, the final model is trained once and can be efficiently deployed for real-time temperature forecasting without additional computational overhead.

2.5. Proposed Hybrid Model

The proposed hybrid model combines an LSTM network and CNN for temperature prediction. The LSTM component captures long-term dependencies and seasonal trends, while the CNN captures local patterns and short-term variations. Thus, the model can represent both long-term trends and short-term fluctuations in temperature data. The LSTM network models temporal dependencies, with its output denoted as y_{LSTM}. The LSTM operations are defined as:

$$f_t = \sigma\Big(W_f[h_{t-1}, x_t] + b_f\Big) \tag{9}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{10}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{11}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{12}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{13}$$

where:

- f_t, i_t , o_t are the forget, input, and output gates, respectively;
- C_t is the memory cell state;
- h_t is the hidden state at time t;
- W and b represent the weight matrices and bias vectors;
- $\sigma(\cdot)$ and $\tanh(\cdot)$ denote the sigmoid and hyperbolic tangent activation functions, respectively.

The CNN component extracts the local features from the temperature data to catch the fluctuations over a brief period. The output $y_{\{CNN\}}$ from the CNN layer is computed as:

$$y_i = \sum(x_j \cdot w_{ij}) + b_i \tag{14}$$

where:

- y_i output from the ith feature map;
- x_j input data;
- w_{ij} weights of the filter;
- b_i bias term.

The final temperature prediction $y_{\{Hybrid\}}$ is obtained by combining outputs of the LSTM and CNN.

$$y_{\text{Hybrid}} = f(y_{\text{LSTM}}, y_{\text{CNN}}) \tag{15}$$

Both the LSTM and CNN branches operate on the same input sequence structured as (T × F), where T is the number of time steps and F is the number of input features. The LSTM branch captures long-term temporal dependencies, while the CNN branch extracts short-term local patterns from the same input. The function f(·) in Equation (15)

is implemented as a concatenation of the outputs from the LSTM and CNN branches, followed by a fully connected (dense) layer for final prediction. The LSTM branch uses 64 hidden units, while the CNN branch applies 32 filters with a kernel size of 3, followed by pooling and flattening. The outputs of both branches are concatenated and passed to dense layers, as shown in Figure 5.

The total number of trainable parameters in the model is on the order of tens of thousands, depending on the configuration.

The architecture of the proposed hybrid LSTM–CNN model is illustrated in Figure 5. The model takes multivariate daily time-series data as input, where T represents the number of time steps and F represents the number of features. The architecture consists of two parallel branches. The first branch uses two stacked LSTM layers with 64 units each. The first LSTM layer returns sequences to capture temporal dependencies in the data. The second branch applies a one-dimensional convolution (Conv1D) layer with 32 filters and a kernel size of 3. This is followed by a ReLU activation function and a max-pooling layer (pool size = 2) to extract local patterns. The resulting features are then flattened. The outputs from both branches are combined using a concatenation layer. The merged features are passed through a dense layer with 32 units and ReLU activation, followed by an output layer that generates multi-step forecasts for 1-day, 3-day, and 6-day horizons.

2.6. Model Training and Evaluation

The model training stage is crucial in the research process, with the hybrid model trained using the processed dataset, which contains temporal lag and rolling statistics features. To start the process, the model is fed with input data, from which the LSTM network learns the long-term dependencies of the temperature time series. In addition, the local patterns and short-term variations are learned by the CNN network. The benefit of hybridizing is achieved by optimizing both types of architecture in a model since such hybridization would allow capturing both global and local temperature patterns. During training, the model experiences many rounds of forward propagation and backpropagation to minimize the loss function. The learning rate, batch size, and epochs are adjusted depending on how the model is fine-tuned. There can be early stopping techniques to control the overfitting of the model and preserve the generalization aspect. In the next phase, the hyperparameters will be fine-tuned using the GA, which maximizes the efficiency of the model for predictive purposes. For analysis, the temperature predictions will be compared with the historical information learned by the model. The model will be evaluated using various data sets—the training, validation, and testing datasets.

2.7. Performance Evaluation

The evaluation of how well the hybrid model performs includes various performance metrics for forecasting accuracy analysis. These metrics include Mean Absolute Error (*MAE*), Root Mean Squared Error (*RMSE*), and the coefficient of determination (R^2), which are used to evaluate the model's prediction performance. Based on the evaluation, temperature forecasts are evaluated for their accuracy and reliability. The model is tested against LSTM, CNN, and traditional statistical baselines on real-world temperature forecasting tasks

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{16}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{17}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{18}$$

where:

- n = total number of observations;
- y_i = actual (observed) value;
- $\hat{y}_i$ = predicted value;
- $\bar{y}$ = mean of the observed values.

3. Result and Discussion

The research required multiple experiments to assess forecast performance using various model configurations and different feature engineering approaches and all available evaluation metrics [14]. The study assessed accuracy for different forecasting periods through three feature types temporal features and lag features and rolling features before using cross-validation to verify the results. Daily errors for 1-day, 3-day, and 6-day predictions show how performance shifts across horizons. Table 5 presents fold-wise results obtained using 10-fold cross-validation prior to final hyperparameter optimization. The time-series-aware cross-validation approach uses data splits which maintain the original time sequence through TimeSeriesSplit function to enable model testing with upcoming data that was not part of the training process. Random shuffling is not applied.

Table 5. Forecasting errors (MAE/RMSE/R^2) across 1-day, 3-day, 6-day horizons.

Fold	MAE (°C)			RMSE (°C)			R^2		
	1-Day	3-Day	6-Day	1-Day	3-Day	6-Day	1-Day	3-Day	6-Day
1	0.78	1.05	1.31	0.88	1.14	1.42	0.96	0.92	0.89
2	0.85	1.12	1.38	0.94	1.20	1.50	0.95	0.91	0.87
3	0.81	1.08	1.29	0.90	1.16	1.40	0.96	0.92	0.90
4	0.89	1.15	1.42	0.97	1.23	1.55	0.94	0.90	0.86
5	0.92	1.20	1.48	1.00	1.28	1.60	0.93	0.89	0.85
6	0.86	1.10	1.36	0.95	1.18	1.48	0.94	0.91	0.88
7	0.83	1.07	1.33	0.92	1.16	1.45	0.95	0.91	0.88
8	0.88	1.14	1.40	0.97	1.22	1.52	0.94	0.90	0.86
9	0.80	1.06	1.30	0.89	1.15	1.42	0.96	0.92	0.89
10	0.87	1.13	1.37	0.96	1.21	1.49	0.94	0.90	0.87

For instance, among the shorter horizons, the MAEs for the 1-day ahead forecast are the lowest, ranging approximately from 0.78 °C to 0.92 °C across folds. With 3-day forecasts, higher errors are observed, with MAE values ranging from about 1.05 °C to 1.20 °C. For 6-day forecasts, larger errors are observed, with MAE values reaching up to approximately 1.48 °C. The model performs better at shorter forecasting horizons, where temperature patterns are more stable, while errors increase for longer horizons due to higher variability and uncertainty. Nevertheless, the model maintains reasonably good performance across all forecasting horizons.

The addition of temporal features, such as day of the week and month of the year, brings a considerable improvement in model accuracy, with an MAE of 0.92 °C and RMSE of 1.10 °C. Lag features improve the capture of historical temperature data but still result in slightly higher error, with an MAE of 0.98 °C and RMSE of 1.18 °C. Rolling statistics enable the model to smooth out fluctuations over time, thus improving performance, with error measures of an MAE of 0.90 °C and RMSE of 1.08 °C. A further reduction in error occurs when combining temporal features with lag features, reducing the MAE to 0.85 °C

and RMSE to 1.02 °C. Combining temporal features with rolling statistics further improves performance, bringing the MAE down to 0.82 °C and the RMSE to 0.98 °C. Finally, applying all features, including temporal, lag, and rolling statistics, achieves the lowest observed error, with an MAE of 0.78 °C and an RMSE of 0.88 °C. The results are summarized in Table 6.

Table 6. Feature-set impact on accuracy.

Feature Set	MAE (°C)	RMSE (°C)	R^2	Improvement in MAE (%) vs. Baseline
Original Features Only	1.15	1.40	0.82	0%
With Temporal Features	0.92	1.10	0.88	20.0%
With Lag Features	0.98	1.18	0.86	14.8%
With Rolling Statistics	0.90	1.08	0.89	21.7%
With Temporal + Lag Features	0.85	1.02	0.91	26.1%
With Temporal + Rolling Statistics	0.82	0.98	0.93	28.7%
All Features (Temporal, Lag, Rolling)	0.78	0.88	0.96	32.2%

Table 7 presents the comparative performance of the baseline hybrid model (without optimization), Random Search, and GA across different forecasting horizons. The baseline model represents the performance without any hyperparameter tuning, while Random Search is employed as a conventional hyperparameter optimization method to provide a fair comparison with the proposed GA approach. In this study, Random Search samples hyperparameters such as learning rate (0.0001–0.01), batch size (16–128), and the number of layers (1–3) from predefined ranges over a fixed number of iterations. Each configuration is evaluated using prediction error metrics, including MAE, RMSE, and R^2. Table 7 shows the progressive improvement from baseline to GA-optimized model.

Table 7. Optimization across forecast horizons.

Optimization	Horizon 1-Day			Horizon 3-Day			Horizon 6-Day		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
Without Optimization	0.88	1.00	0.91	1.15	1.28	0.87	1.45	1.55	0.83
Random Search	0.72	0.82	0.94	0.98	1.10	0.90	1.30	1.40	0.86
GA	0.55	0.62	0.98	0.75	0.80	0.96	1.28	1.35	0.95

The results show that Random Search improves performance over the baseline model by reducing MAE and RMSE and increasing R^2 across all forecasting horizons. GA achieves the best overall performance with the lowest error values and highest R^2. Random Search samples the search space without feedback between trials, while GA refines candidates in each generation using evolutionary operations, which drives it closer to optimal hyperparameters.

All models were trained on the same dataset with identical preprocessing steps and feature sets, including temporal, lag, and rolling features. The data splits were also kept identical, and 10-fold cross-validation was applied to all models. For the deep learning models (LSTM, CNN, GRU, and Transformer), hyperparameters were tuned within similar ranges to maintain a balanced comparison. The model development process used standard techniques to optimize the performance of both Random Forest and XGBoost machine learning models. The statistical models ARIMA and Prophet required the use of their standard model configurations for their implementation. The research tested all baseline

models by using the same experimental framework and dataset which had been established for the study. The evaluation process for all models took place under identical evaluation conditions. The test environment between the two models remains unchanged so that performance differences can be accurately attributed to the actual model design. The benchmark results presented in Table 8 represent the 1-day forecasting period, which used the GA-optimized model for forecasting. All models receive a fair comparison because the same input features are applied to all models, which allows for controlled testing between them, despite simpler models lacking the capacity to process complex feature sets.

Table 8. Benchmark comparison with other forecasting models.

Model	MAE (°C)	RMSE (°C)	R^2	MAE Improvement (%)	RMSE Improvement (%)	R^2 Improvement (%)
ARIMA	1.50	1.80	0.83	63.33%	65.56%	18.07%
CNN	1.30	1.60	0.87	57.69%	61.25%	12.64%
LSTM	1.20	1.50	0.89	54.17%	58.67%	10.11%
Random Forest	1.20	1.50	0.88	54.17%	58.67%	11.36%
Prophet	1.10	1.40	0.90	50.00%	55.71%	8.89%
GRU	1.00	1.30	0.92	45.00%	52.31%	6.52%
XGBoost	1.00	1.30	0.92	45.00%	52.31%	6.52%
Transformer	0.75	1.00	0.94	26.67%	38.00%	4.26%
Proposed Hybrid	0.55	0.62	0.98	*	*	*

* The proposed model is used as the reference baseline for improvement calculation.

The comparative benchmark reveals that the proposed model achieves the best performance, with an MAE of 0.55 °C and an RMSE of 0.62 °C, surpassing all other models. Although the LSTM model performs well, with an MAE of 1.2 °C and an RMSE of 1.5 °C, it does not rival the proposed model in capturing local patterns. The CNN model slightly trails behind, with an MAE of 1.3 °C and an RMSE of 1.6 °C, as it does not effectively model sequential dependencies. The MAE and RMSE for the GRU model are 1.0 °C and 1.3 °C, respectively, making it a faster alternative to LSTM, although still inferior to the proposed hybrid approach. The Transformer model achieves good performance, with an MAE of 0.75 °C and an RMSE of 1.0 °C, benefiting from the self-attention mechanism; however, its performance remains inferior to the proposed hybrid model. XGBoost and Random Forest also show good performance, with MAE values of 1.0 °C and 1.2 °C, respectively, although both fail to capture the complex sequential patterns of temperature data. ARIMA and Prophet rank lowest in performance, with MAE values of 1.5 °C and 1.1 °C, respectively, as these models struggle with non-linear dependencies and complex temporal relationships. Furthermore, the relative improvement analysis presented in Table 8 (1-day forecasting horizon using the GA-optimized model) shows that the proposed model achieves substantial error reduction, with up to 63.33% improvement in MAE and 65.56% improvement in RMSE compared to traditional models such as ARIMA, along with consistent improvements over all baseline methods. These results indicate that deep learning models are more effective at capturing complex patterns within time series, particularly in temperature forecasting. The selected benchmark models include widely used classical and deep learning approaches, including Transformer-based architectures, providing a representative comparison framework.

The 1-day forecasting, shown in Figure 6 achieves strong performance, with R^2 values reaching approximately 0.96 and low MAE and RMSE values, reflecting a high degree of

accuracy. At the 3-day horizon, shown in Figure 7, forecast accuracy begins to decline, with R^2 values ranging between approximately 0.89 and 0.92, and increases in MAE and RMSE reflecting the growing uncertainty. The performance declines further for the 6-day forecast, shown in Figure 8, with R^2 values ranging between approximately 0.85 and 0.90, where MAE and RMSE values are highest, indicating greater difficulty in longer-term forecasting.

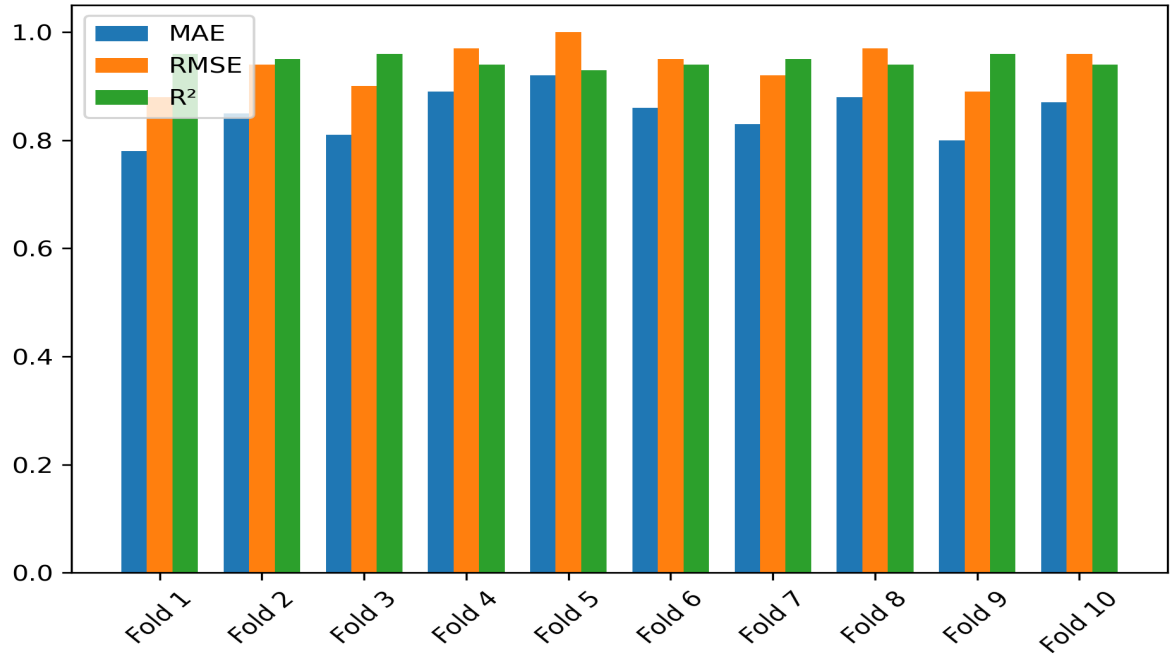


Figure 6. One-day forecasting performance.

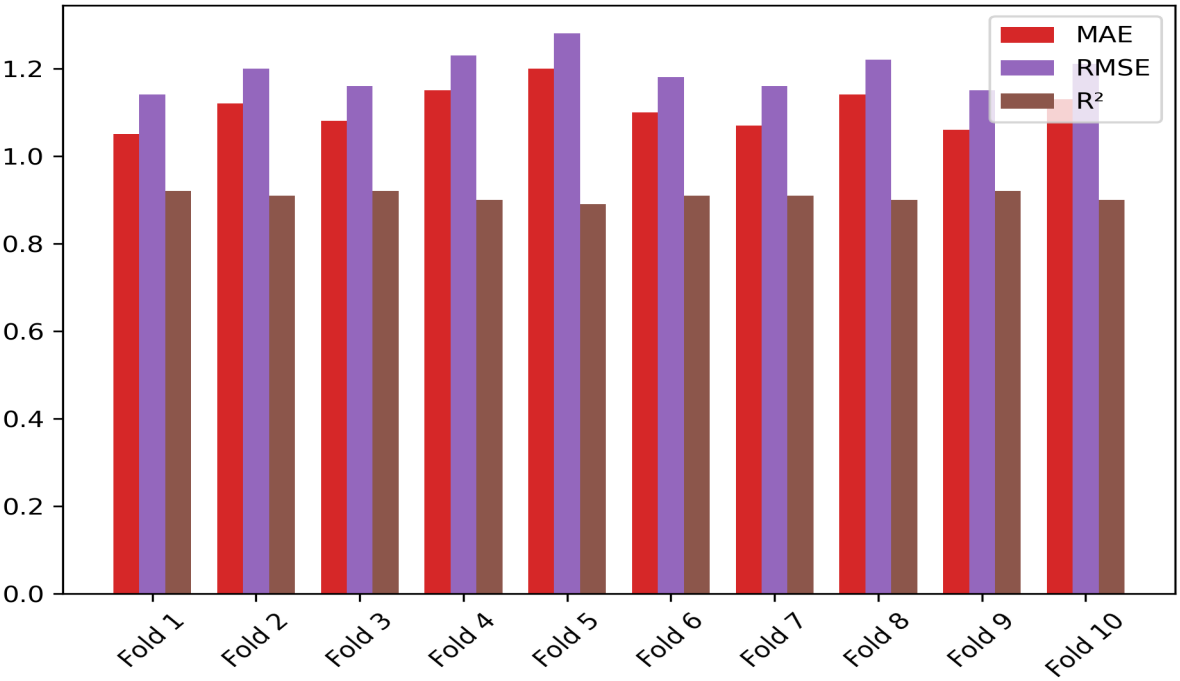


Figure 7. Three-day forecasting performance.

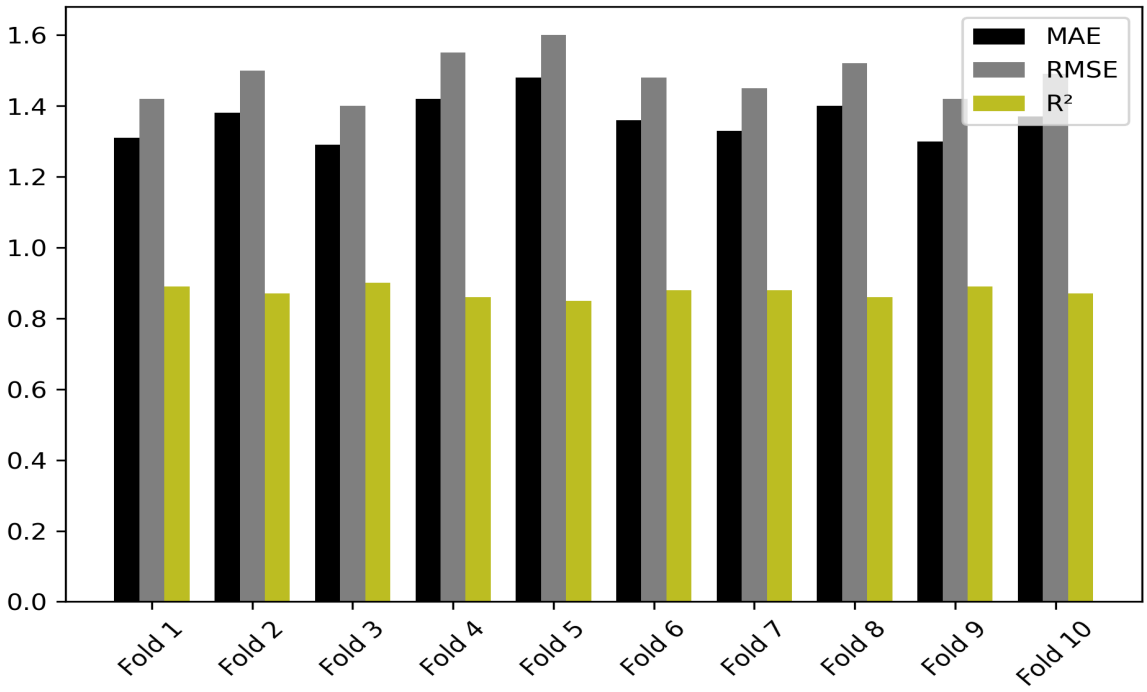


Figure 8. Six-day forecasting performance.

The correlation analysis for the LSTM, CNN, and hybrid models determines how effectively each predicts temperature values compared to the actual values. Figures 9–11 show the correlation between predicted and observed temperature values for the LSTM, CNN, and hybrid models. The hybrid model shows the highest correlation among the models, with predicted values closely matching the true values, thereby supporting the benefit of combining both temporal and spatial attributes. In contrast, the LSTM model shows moderate correlation, reflecting its ability to capture sequential patterns. The CNN captures local patterns and helps represent the dataset's structure. The gap in model performance shows that combining architectures with different strengths improves deep learning results.

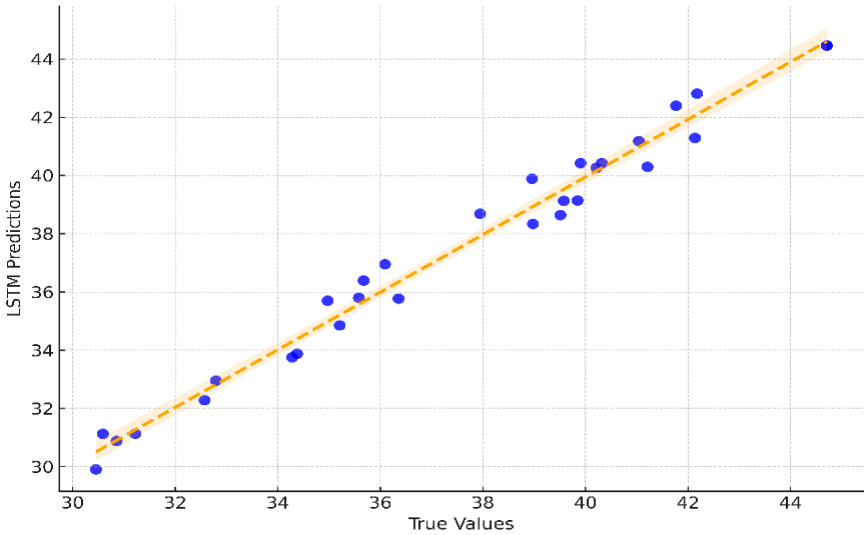


Figure 9. LSTMM model: predicted and observed temperature values.

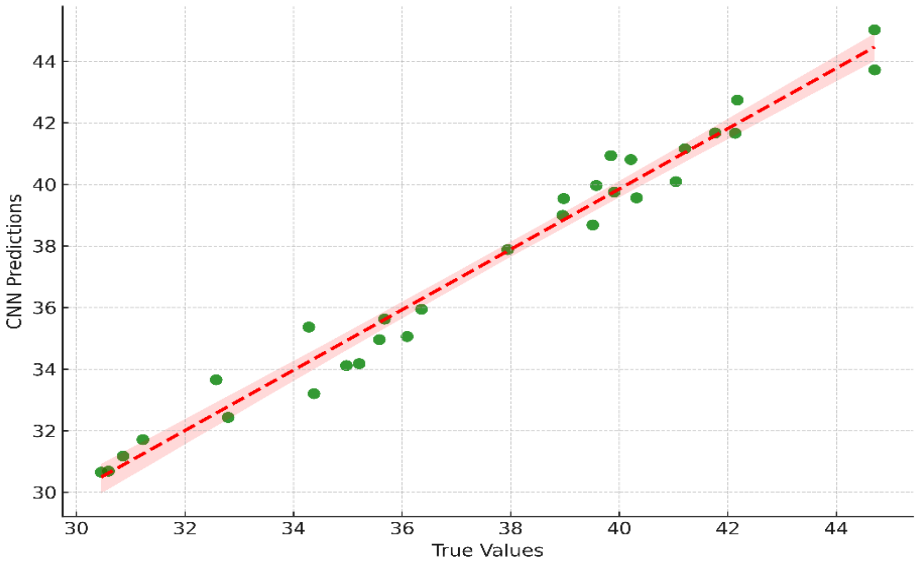


Figure 10. CNN model: predicted and observed temperature values.

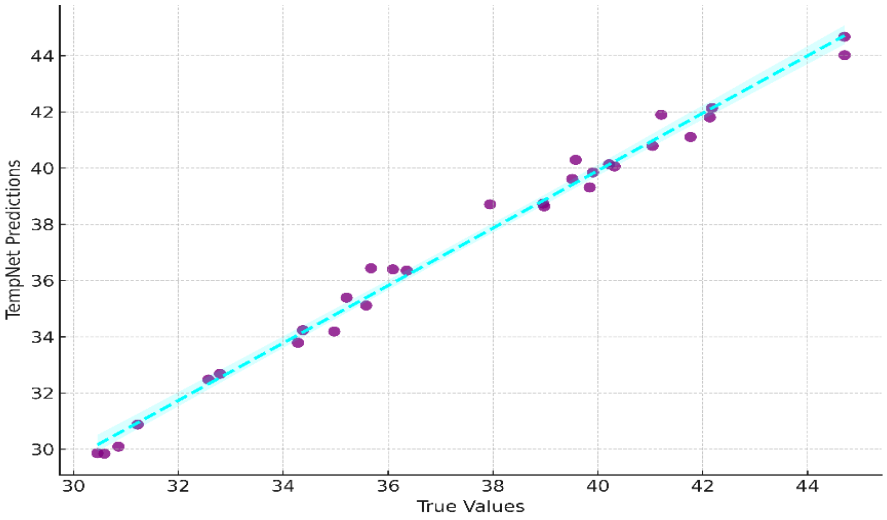


Figure 11. Relation between predicted and observed temperature values.

The temperature patterns for the complete duration of the study are shown in Figures 12 and 13 which display both actual measurements and forecasted estimates. The blue line denotes the actual recorded temperatures, while the red dashed line indicates the predicted values obtained from the training data. The model training process enables temperature pattern detection which researchers use to assess model accuracy through predicted results comparison with actual measurements. The model accurately represents all temperature patterns which occur during the yearly cycle and throughout its 21-year evaluation period. The model demonstrates accurate historical data matching which enables it to project future temperature patterns. The model illustrates decreased prediction accuracy during the 2015 to 2024 period yet it maintains its operational capacity for both training and testing procedures which demonstrates its ability to forecast temperature variations.

The forecast error variation which is measured by MAE shows its yearly changes through Figure 14. The graph compares MAE values across all years to assess model performance over time. The color gradient demonstrates error distribution, which shows that areas with larger errors exceed the threshold 1.2 value, thus showing major errors in red color. The threshold for temperature forecasting practices defines significant errors as any temperature error which exceeds 1.0 °C range in meteorological models. The model

shows temperature variation success through its ability to maintain errors below 1.0 °C limit because most errors fall within that range.

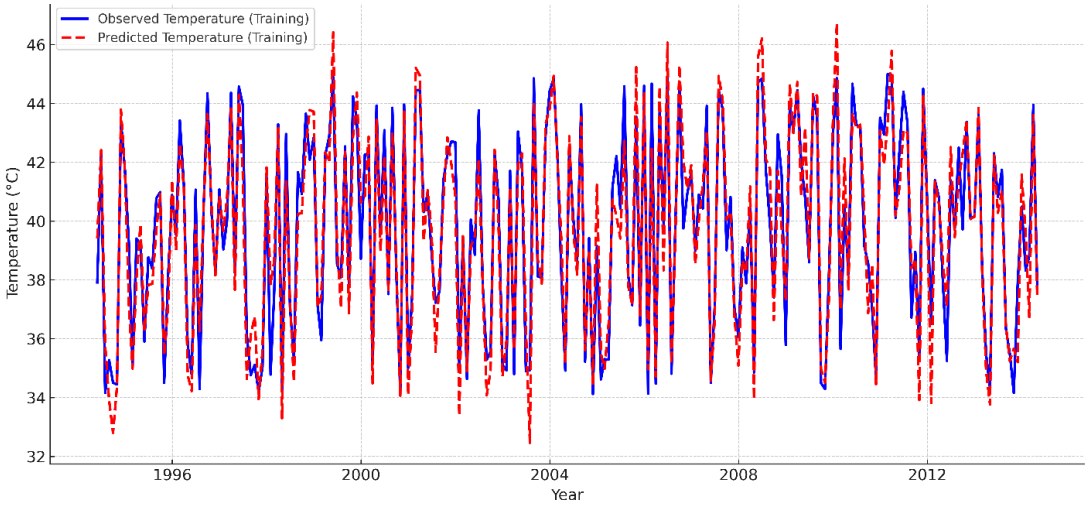


Figure 12. Observed and predicted temperatures during the training phase.

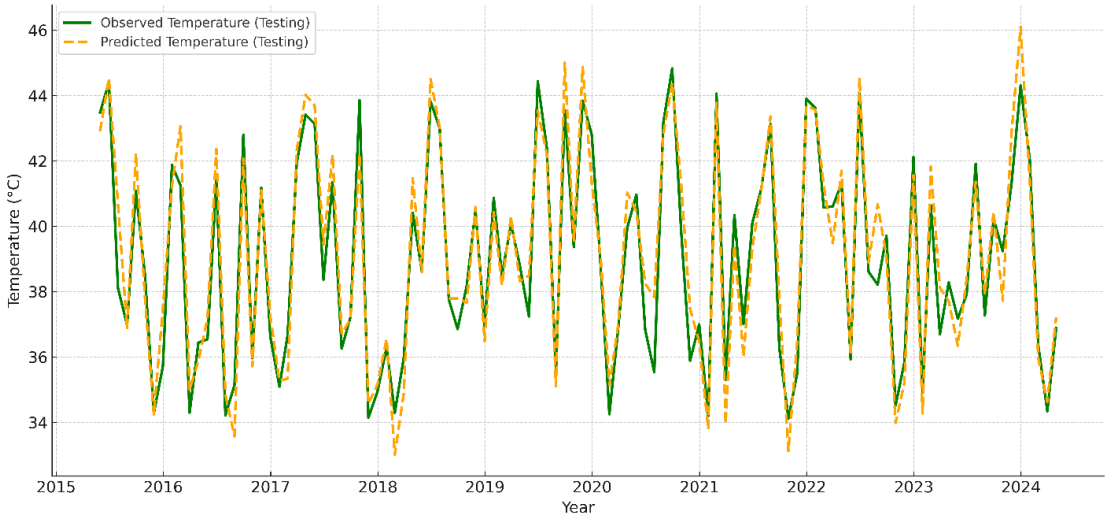


Figure 13. Observed and predicted temperatures during the testing phase.

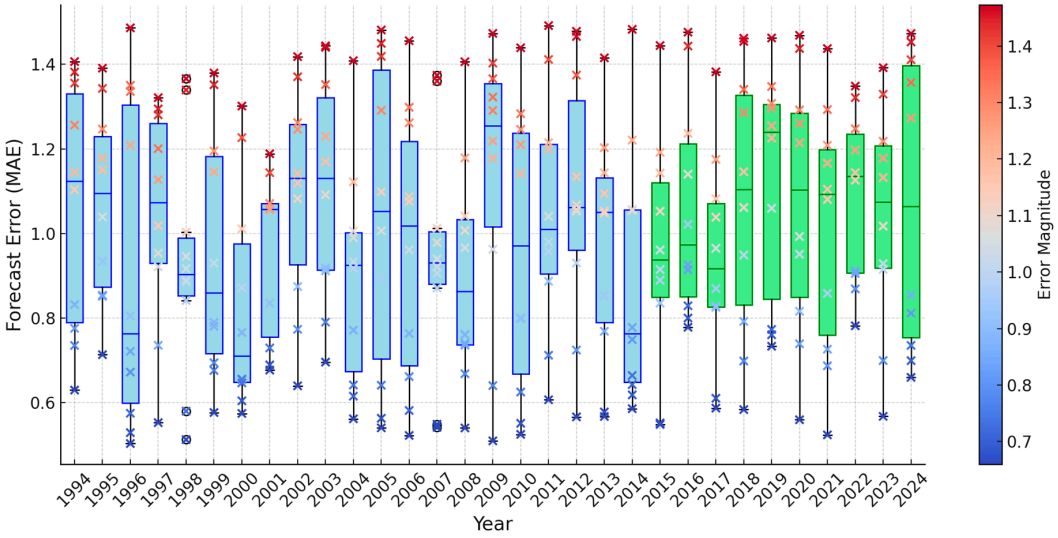


Figure 14. Year-wise forecast error distribution with gradient representation of MAE magnitude.

The analysis presented in Figure 15 uses Pearson correlation coefficients to evaluate the relationship between each variable and the target variable T2M_MAX instead of using model-based feature importance methods such as SHAP or permutation importance. The correlation analysis indicates that T2M_MIN exhibits the strongest relationship with the target variable T2M_MAX, followed by RH2M and ALLSKY_SFC_SW_DWN (surface shortwave radiation), which are essential environmental factors that affect temperature changes. The WS2M variable demonstrates low correlation values, which indicate it has a weaker connection to temperature changes. The findings establish that physical variables which directly relate to the target variable produce greater effects on temperature change forecasts. The figure displays T2M_MAX as the prediction target, which serves as a self-correlation baseline but does not function as an input feature, thus preventing any data leakage. Figure 15 demonstrates how variables relate to each other while showing the significance of features which models derive from their input data.

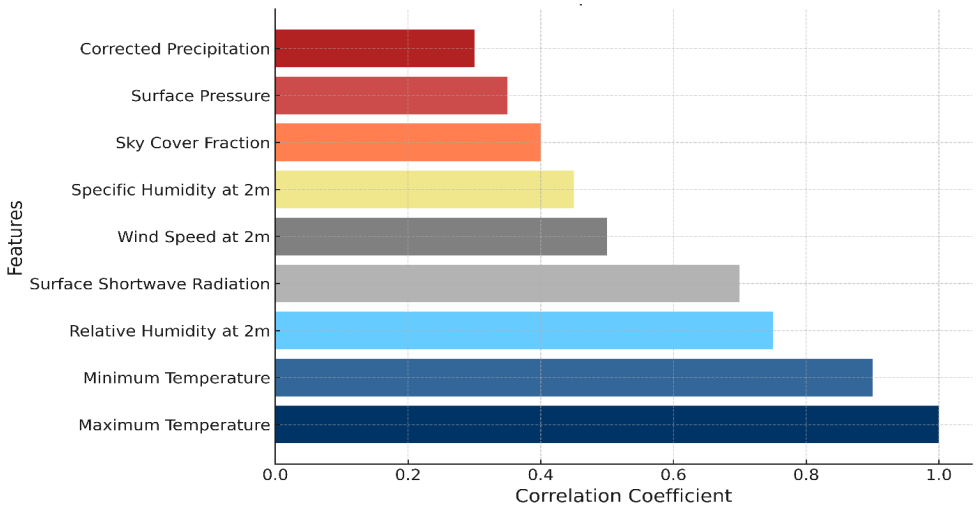


Figure 15. Feature Correlation Analysis with Target Variable (T2M_MAX).

The results demonstrate that the hybrid LSTM–CNN model successfully captures temporal dependencies and local temperature variations through its hybrid design. The cross-validation results show consistent performance across different test folds, which demonstrates strong generalization ability of the model. The prediction errors increase as the forecasting period lengthens because multiple steps in the forecasting process lead to greater uncertainty accumulation. The year-wise error distribution analysis in Figure 14 shows that specific time periods experience higher MAE values, particularly during extreme summer conditions and in recent years starting from 2015. Climate variability and extreme temperature events, together with meteorological pattern shifts, create non-stationary conditions which make forecasting tasks more difficult. The model demonstrates its highest performance during stable environmental conditions, yet its accuracy decreases during times of sudden environmental changes. The proposed model shows better performance than previous research results. Existing methods experience increased forecasting errors because they combine standalone models and hybrid architectures without complete feature engineering together with optimization techniques. The proposed framework demonstrates superior predictive ability through its lower MAE of 0.55 °C for 1-day forecasting and its higher R^2 values which reach 0.98.

The hybrid LSTM–CNN model achieves superior performance because it simultaneously models various data aspects. The LSTM component captures long-term temporal dependencies and seasonal trends while the CNN component extracts short-term local patterns and fluctuations. The model learns temperature dynamics through its complemen-

tary learning mechanism which improves complex temperature dynamics representation compared to standalone models. The present evaluation uses Makkah data from a hyper-arid region which has stable seasonal weather patterns. The climatic conditions of this environment affect the performance results which differ from results obtained in regions with more climatic variability. NASA POWER reanalysis data, which derives from models, shows less atmospheric fluctuations than ground-based measurements, leading to potential adverse effects on measurement accuracy. The validation process will extend to various climatic regions through the addition of station-based datasets which will test the proposed framework's ability to function across different environments.

4. Conclusions

The research shows that the hybrid LSTM–CNN model which the study introduced works effectively for predicting temperatures over multiple time intervals. The model demonstrates high predictive accuracy through its 1-day forecasting performance which achieves an MAE value of 0.55 °C and demonstrates consistent prediction results across all validation testing. The combination of temporal features and lag features and rolling features together with GA-based optimization leads to better forecasting accuracy compared to the basic forecasting methods. The model is not without limitations. GA optimization introduces computational overhead, which may be challenging in resource-limited settings. In addition, the evaluation is limited to a single hyper-arid region (Makkah), and further validation across different climatic conditions is required to assess generalizability. Overall, the proposed approach provides a reliable framework for temperature forecasting in arid climatic conditions.

Author Contributions: Conceptualization, F.H. and Z.A.A.; methodology, F.H. and T.A.J.; software, A.A.; validation, F.H., Z.A.A., M.I.M. and T.A.J.; formal analysis, F.H., T.A.J. and M.A.; investigation, M.I.M., T.A.J. and M.A.; resources, N.A.; data curation, M.K.A.; writing—original draft preparation, F.H.; writing—review and editing, Z.A.A., T.A.J. and M.A.; visualization, A.A.; supervision, F.H.; project administration, F.H.; funding acquisition, T.A.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study are publicly available from the NASA POWER database. The full dataset and the pre-processed/feature-engineered data are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jayagopal, P.; Muthukumaran, V.; Koti, M.S.; Kumar, S.S.; Rajendran, S.; Mathivanan, S.K. Weather-based maize yield forecast in Saudi Arabia using statistical analysis and machine learning. *Acta Geophys.* **2022**, *70*, 2901–2916. [CrossRef]
2. Alghamdi, A.S. Climatology and changes in temperature seasonality in the Arabian Peninsula. *Atmosphere* **2024**, *15*, 26. [CrossRef]
3. Miky, Y.; Al Shouny, A.; Abdallah, A. Studying the impact of urban management strategies and spatiotemporal dynamics of LULC on land surface temperature and SUHI formation in Jeddah, Saudi Arabia. *Sustainability* **2023**, *15*, 15316. [CrossRef]
4. Alomar, M.K.; Khaleel, F.; Aljumaily, M.M.; Masood, A.; Razali, S.F.M.; AlSaadi, M.A.; Al-Ansari, N.; Hameed, M.M. Data-driven models for atmospheric air temperature forecasting at a continental climate region. *PLoS ONE* **2022**, *17*, e0277079. [CrossRef] [PubMed]
5. Huva, R.; Song, G.; Zhong, X.; Zhao, Y. Comprehensive physics testing and adaptive weather research and forecasting physics for day-ahead solar forecasting. *Meteorol. Appl.* **2021**, *28*, e2017. [CrossRef]
6. Kontopoulou, V.I.; Panagopoulos, A.D.; Kakkos, I.; Matsopoulos, G.K. A review of ARIMA vs. machine learning approaches for time series forecasting in data-driven networks. *Future Internet* **2023**, *15*, 255. [CrossRef]
7. Sirisha, U.M.; Belavagi, M.C.; Attigeri, G. Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: A comparison. *IEEE Access* **2022**, *10*, 124715–124727. [CrossRef]

8. Da, T.N.; Cho, M.Y.; Thanh, P.N. Hourly load prediction based feature selection scheme and hybrid CNN-LSTM method for building's smart solar microgrid. *Expert Syst.* **2024**, *41*, e13539. [CrossRef]
9. Nguyen, T.P. AIoT-based indoor air quality prediction for building using enhanced metaheuristic algorithm and hybrid deep learning. *J. Build. Eng.* **2025**, *105*, 112448. [CrossRef]
10. Nguyen-Da, T.; Nguyen-Thanh, P.; Cho, M.Y. Real-time AIoT anomaly detection for industrial diesel generator based on efficient deep learning CNN-LSTM in Industry 4.0. *Internet Things* **2024**, *27*, 101280. [CrossRef]
11. Nguyen, D.T.; Nguyen, T.P.; Cho, M.Y. Online prognostic failure AIoT system for industrial generators maintenance service based two-stage deep learning algorithm. *Control Eng. Pract.* **2025**, *157*, 106263. [CrossRef]
12. Nguyen, T.P.; Cho, M.Y. A cloud-based leakage current classification system for high-voltage insulators with improved particle swarm optimization and hybrid deep learning technique. *Eng. Appl. Artif. Intell.* **2025**, *143*, 109987. [CrossRef]
13. Gong, Y.; Zhang, Y.; Wang, F.; Lee, C.-H. Deep learning for weather forecasting: A CNN-LSTM hybrid model for predicting historical temperature data. *arXiv* **2024**, arXiv:2410.14963. [CrossRef]
14. Uluocak, I.; Bilgili, M. Daily air temperature forecasting using LSTM-CNN and GRU-CNN models. *Acta Geophys.* **2024**, *72*, 2107–2126. [CrossRef]
15. Wang, Z.; Wang, L. Optimization of convolutional long short-term memory hybrid neural network model based on genetic algorithm for weather prediction. In Proceedings of the ACM International Conference, Dalian, China, 24–26 September 2021; pp. 2488–2494. [CrossRef]
16. Zhang, W.; Zhou, H.; Bao, X.; Cui, H. Outlet water temperature prediction of energy pile based on spatial-temporal feature extraction through CNN-LSTM hybrid model. *Energy* **2023**, *264*, 126190. [CrossRef]
17. Yasavoli, B.; Habibirad, A.; Javanshiri, Z. A hybrid deep learning model in predicting weather temperature. *Earth Sci. Inform.* **2025**, *18*, 461. [CrossRef]
18. Yu, Y.; Wang, S.; Ou, H.; Wang, Q.; Quan, Y. GA-CNN-LSTM-based temperature and humidity prediction method. In Proceedings of the IFEEA 2024, Shenzhen, China, 22–24 November 2024; pp. 250–254. [CrossRef]
19. Çınarer, G. Hybrid deep learning and stacking ensemble model for time series-based global temperature forecasting. *Electronics* **2025**, *14*, 3213. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.