ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Research on temperature prediction based on convolutional neural Network and Gated recurrent Unit

Yibo Zhang[1], *Yang Chen*[1], *Chang Lan*[1] *and Chengcheng Wang*[1,*]

[1] Jilin Provincial Meteorological Data Center, Changchun 130000, Jilin, China

SUMMARY: *Accurate temperature prediction is an important support for meteorological business automation, agricultural disaster defense and urban fine operation. In this paper, a temperature prediction model based on convolutional neural network and gated recurrent unit is proposed. The multivariate time series samples are constructed by using the hourly meteorological observation data of multiple stations in Jilin Province. The model takes temperature, relative humidity, air pressure, wind speed, wind direction, precipitation, sunshine duration and time period encoding as input, extracts the local meteorological combination features in a short time window through CNN, and then captures the time series dependence in day and night variation, cold and warm conversion, and continuous temperature evolution by GRU. The experimental results show that the proposed model achieves 0.56℃ MAE, 0.82℃ RMSE, 4.31% MAPE and 0.962* R^2*, which is better than ARIMA, SVR, LSTM, CNN-LSTM and Transformer methods. The model has both prediction accuracy and computational efficiency, and can provide data-driven support for hourly temperature correction, short-term forecast product generation and regional meteorological services.*

KEYWORDS: *Convolutional neural network; Gated recurrent unit; Temperature prediction; Multivariate time series*

1 Introduction

Temperature is a basic variable in meteorological forecast, agricultural production scheduling, urban operation management and public security services. With the continuous improvement of automatic weather stations, reanalysis data and ground observation networks, temperature prediction has gradually shifted from single historical series extrapolation to multi-source meteorological data driven modeling process. Temperature change has obvious multivariate time series characteristics in nature. Temperature, humidity, air pressure, wind speed, precipitation, solar radiation and other variables do not exist in isolation, but jointly affect near-surface heat exchange and local meteorological state on different time scales. Therefore, how to extract effective features from continuous meteorological observation data and describe the relationship between short-term fluctuations and phased trends has become a key link to improve the accuracy of temperature prediction.

Traditional statistical models have good interpretability in stationary series analysis, but their feature expression ability is relatively limited when facing nonlinear fluctuations, abrupt weather processes and multivariate coupling relationships. In recent years, deep learning methods have been widely used for meteorological time series prediction. The convolutional

*xiaoqiyue0907@126.com
https://doi.org/10.65102/is2026999
Published: 30 April, 2026

neural network can extract the slope of temperature change, the combination relationship of adjacent variables and short-term disturbance features from the local time window. Gated recurrent unit is suitable for dealing with sequence dependence, and can retain historical information through update gate and reset gate, so as to alleviate the gradient attenuation phenomenon in long sequence training. The combination of CNN and GRU can make the model take into account both local meteorological feature recognition and continuous time-dependent modeling, which is more suitable for data tasks such as temperature prediction with periodicity, nonlinearity and regional differences.

Focusing on the improvement of temperature prediction accuracy and business availability requirements, this paper constructs a temperature prediction model based on convolutional neural network and gated recurrent unit. The model takes the multivariate meteorological observation series as input, constructs the sample through the sliding window, extracts the local meteorological change pattern by CNN, and then captures the dynamic association between different times by GRU, and finally outputs the temperature forecast value of the future time. This method not only focuses on the linkage changes of meteorological elements in a short time, but also retains the trend information in the historical temperature series, which provides computer model support for the meteorological data center to carry out fine prediction, automatic product generation and forecast correction.

The rest of this paper is arranged as follows: Section 2 combs the related research on temperature prediction and deep learning time series modeling. Section 3 introduces the meteorological multivariate sample construction method and the CNN-GRU network structure. Section 4 gives the data source, preprocessing process, evaluation index and ablation experimental results. Section 5 discusses the model comparison, time complexity, deployment expansion and meteorological business value. Section 6 summarizes the research conclusions and proposes directions for subsequent improvements.

2 Related Research

Although temperature prediction is a typical meteorological time series prediction task, its modeling difficulty is not low. Near-surface temperature is affected by solar radiation, humidity, air pressure, wind speed, terrain underlying surface, and weather system movement. The observation series contains not only diurnal and seasonal cycles, but also sudden fluctuations caused by cold air activity, precipitation process, and urban heat island effect. Different from univariate stationary series, meteorological data usually have nonlinear, lag and multivariate coupling characteristics. If the model cannot identify short-term disturbances and continuous trends at the same time, the prediction results are easy to deviate in extreme warming, cooling and large temperature difference between day and night. Therefore, the research on temperature prediction has gradually shifted from traditional statistical modeling to data-driven methods, and further developed into deep learning and hybrid neural network models.

In the research of deep learning temperature prediction, recurrent neural network and its improved structure have been used to process continuous meteorological series earlier. Uluocak and Bilgili used LSTM-CNN and GRU-CNN models to carry out daily temperature prediction, and the results show that the combination of convolution structure and gated recurrent structure can enhance the extraction ability of temperature change patterns [1]. Pangaribuan et al. compared LSTM and LSTM-GRU structures based on two-year meteorological data, and pointed out that the gating mechanism has a good role in maintaining the historical dependence in the temperature series [2]. Faulina et al. constructed an LSTM temperature prediction system and proved that long short-term memory network was superior to traditional shallow model in temperature trend learning [3]. Saleem et al. further integrated the optimized LSTM and GRU into a temperature and humidity prediction model, so that the model could obtain more stable error performance under the condition of multivariate input [4]. These studies illustrate that a single cyclic structure is capable of handling temperature time dependence, but it still has shortcomings in local feature extraction and multivariate short-term disturbance identification.

Temperature change is not only time-dependent, but also affected by spatial differences and regional climate background. Tran et al. proposed a spatio-temporal temperature prediction method based on domain adaptation of deep neural network, which incorporated regional differences into the model transfer process and provided ideas for cross-site prediction [5]. Shah et al. used functional autoregressive method to predict short-term temperature, which has strong statistical interpretation, but limited expression ability for complex nonlinear changes [6]. Matlani and Shukla used the deep learning algorithm of hyperparameter optimization to predict the maximum temperature, and showed that the choice of network parameters would significantly affect the prediction stability [7]. Teixeira et al. combined LSTM with genetic algorithm to improve the parameters of weather prediction model with intelligent optimization method [8]. Zenkner and Navarro-Martinez proposed a lightweight deep weather prediction model, emphasizing the application value of the model in the environment with limited computing resources [9]. Venkatachalam et al. constructed an improved data-driven hybrid weather prediction model, which shows that the fusion of deep structure and traditional time series idea can improve the robustness of weather prediction [10].

With the increase of meteorological observation data scale, attention mechanism, feature fusion and multi-model collaboration have gradually become the research focus. Suleman and Shridevi proposed a LSTM short-term weather prediction model based on spatial feature attention, so that the model could pay more attention to key meteorological variables [11]. Shrivastava et al. took the temperature prediction of New Delhi as an example to verify the adaptability of deep learning models in urban temperature series [12]. Sharma et al. analyzed the time series prediction of monthly average temperature and showed that the model structure should match the prediction target under different time scales [13]. Essa et al. applied deep learning to thunderstorm intensity prediction in remote sensing weather data, reflecting the extended ability of deep models in complex meteorological element recognition [14]. Faraji et al. proposed 3D CNN-GRU model to predict urban PM2.5 concentration. Although the research object is not the temperature, the modeling idea of "convolution to extract local structure and GRU to capture time dependence" has reference significance for meteorological environment sequence prediction [15].

In addition to temperature prediction, wind speed, precipitation and agrometeorological prediction studies also provide a reference for the model design in this paper. Shirzadi et al. combined recurrent neural network with numerical weather prediction information for urban wind power prediction, reflecting the trend of the fusion of data-driven models and external meteorological data in meteorological prediction [16]. Barış et al. proposed a robust intelligent model for the nonlinear dynamics of wind speed, indicating that extreme fluctuation scenarios require stronger nonlinear expression capabilities [17]. Abebe and Endalie predicted monthly precipitation in the absence of complete climate data, emphasizing the adaptation ability of the model to incomplete meteorological data [18]. Peeriga et al. introduced Bi-LSTM and IoT data into agricultural real-time rainfall prediction, indicating that multi-source sensor data has become an important input for meteorological prediction [19]. Nandi et al. proposed an attention-based long-term temperature prediction network, proving that the attention mechanism helps to identify the key time segments in the long-term temperature series [20]. Alomar et al. carried out data-driven temperature prediction for continental climate regions, and further showed that regional climate characteristics would affect the generalization effect of the model [21]. Subair et al. used GRU to predict the minimum temperature and verified the effectiveness of GRU in temperature sequence modeling [22]. Akilan and Baalamurugan proposed the GRU-CNN weather prediction and field monitoring model combined with IoT, which provides a basis for the real-time application of weather prediction models in agricultural scenarios [23].

Table 1: Overview of research related to temperature prediction

Study	Model Method	Research Object or Data Type	Evaluation Metrics	Main Limitations
Uluocak and Bilgili [1]	LSTM-CNN, GRU-CNN	Daily temperature series	RMSE, MAE	Insufficient discussion of multi-station generalization
Pangaribuan et al. [2]	LSTM, LSTM-GRU	Two-year meteorological observation data	RMSE, MAE	Short data period and limited extreme-weather samples
Faulina et al. [3]	LSTM	Temperature prediction system	MSE, RMSE	Insufficient extraction of local disturbance features
Saleem et al. [4]	Optimized LSTM-GRU ensemble model	Temperature and humidity data	MAE, RMSE	Complex model structure and high deployment cost
Tran et al. [5]	Deep network domain adaptation model	Spatiotemporal temperature data	RMSE, MAE	Limited lightweight application for regular operational stations
Nandi et al. [20]	Attention-based long-term temperature prediction network	Long-term temperature series	MAE, RMSE	High computational cost for long-sequence training
Subair et al. [22]	GRU	Minimum temperature data	RMSE, MAPE	Lack of a convolutional local feature extraction module

As shown in Table 1, the temperature prediction methods have formed a situation of parallel development of statistical models, traditional machine learning, recurrent neural networks, attention networks and hybrid deep models. The statistical model is easy to interpret, but it is difficult to fully express the nonlinear coupling of temperature change. Single machine learning model relies on artificial features, and its generalization ability is limited when facing multi-variable meteorological series. Recurrent structures such as LSTM and GRU can model historical dependence, but it is easy to ignore the combination changes of variables in a short time window. CNN can extract local patterns, but it is insufficient to grasp the long-term temporal association when used alone. Therefore, combining the local feature extraction ability of CNN with the time series memory ability of GRU is more in line with the modeling requirements of "local disturbance identification" and "continuous trend maintenance" coexisting in temperature prediction.

Based on the above analysis, this paper focuses on the following issues:

(1) How to effectively extract the local combination changes of temperature, humidity, air pressure, wind speed and other variables from multivariate meteorological observation data;

(2) Whether the CNN-GRU hybrid structure is better than the single CNN, GRU, LSTM and traditional machine learning model in temperature prediction;

(3) Whether the model can form a better balance between prediction accuracy, computational complexity and meteorological service deployment.

To solve these problems, this paper constructs a CNN-GRU temperature prediction model, organizes the multivariate meteorological series in a sliding window manner, extracts short-term local variation features through the convolutional layer, and then uses GRU to capture the time dependence. The experimental part is compared with the baseline model. The design not only responds to the problem of insufficient local feature extraction in existing research, but also considers the actual needs of deployable, scalable and stable operation of the model in meteorological business.

3 Construction of temperature prediction model based on convolutional neural network and gated recurrent unit

The construction of temperature prediction model needs to deal with two problems at the same time: one is the local coupling relationship of multiple meteorological variables in the same time window, and the other is the continuous dependence of temperature series over time. Convolutional neural network is suitable for capturing variable combination features from local Windows, such as short-term cooling signals under the joint action of humidity increase, air pressure decrease and wind speed change. The gated recurrent unit can retain the effective historical state through the gated structure, which is suitable for describing the time series memory in the process of day and night temperature difference, cold and warm air conversion and continuous warming. Compared with using CNN or GRU alone, the CNN-GRU structure can combine local feature extraction with time-dependent modeling to reduce the impact of manual feature selection on prediction results.

The temperature prediction model constructed in this paper is based on hourly observation data of ground weather stations, and variables such as temperature, relative humidity, air pressure, wind speed, wind direction, precipitation and sunshine duration are organized into multivariate time series. In the feature end, CNN is used to extract the local meteorological change pattern in a short time segment. In the time series end, GRU is used to learn the dynamic relationship between different observation times and output the temperature value at the future target time. The overall process is shown in Figure 1.

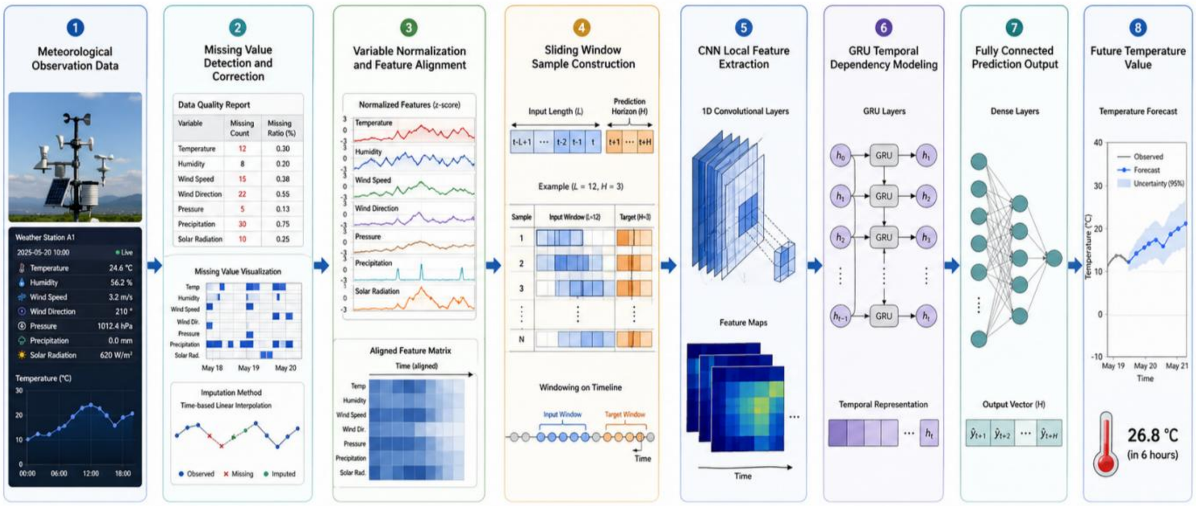

Figure 1: Overall flow chart of CNN-GRU temperature prediction model

3.1 Meteorological multivariate feature extraction and time-series sample construction

Meteorological multivariate feature extraction and time series sample construction are the basis for the stable operation of CNN-GRU model. Hourly meteorological observation data usually have problems such as variable dimension differences, missing measurement records, local outliers and uneven timestamps. If they are directly input into the network, the training process will be biased towards variables with large numerical ranges, and the learnability of the temperature change law will be weakened. Therefore, in this paper, the observed data are unified and sorted before model training, and different meteorological elements are converted into feature sequences with the same time granularity. Let the multivariate observation series obtained in continuous time from a meteorological station be as follows.

$$X = \{x_1, x_2, \dots, x_T\}, \quad x_t = [T_t, H_t, P_t, W_t, D_t, R_t, S_t] \tag{1}$$

where, T_t represents the temperature at time t, H_t represents the relative humidity, P_t represents the pressure, W_t represents the wind speed, D_t represents the wind direction, R_t represents the precipitation, S_t represents the sunshine duration, and T is the length of the observation sequence. Because the wind direction belongs to the periodic type variable, the direct use of the Angle value is easy to cause the problem that 0° and 360° are numerically far away. In this paper, the wind direction is converted into sine and cosine components to make it more suitable for neural network learning:

$$D_t^{\sin} = \sin\left(\frac{\pi D_t}{180}\right), \quad D_t^{\cos} = \cos\left(\frac{\pi D_t}{180}\right) \tag{2}$$

After transformation, the input variable of the model is extended from the original wind direction Angle to two continuous features, which can express the wind direction change more smoothly. For continuous variables such as temperature, humidity, air pressure and wind speed, the max-min normalization method is used to deal with the differences between different dimensions, so that each variable is mapped to a uniform numerical interval.

$$\hat{x}_{t,j} = \frac{x_{t,j} - \mu_j}{\sigma_j} \tag{3}$$

where, $\hat{x}_{t,j}$ represents the standardized value of the JTH meteorological variable at time t, and μ_j and σ_j represent the mean and standard deviation of this variable in the training set, respectively. To avoid information leakage of the test set, the normalization parameters are only calculated from the training set and are consistent on the validation set and the test set.

In terms of missing test handling, this paper adopts a hierarchical strategy according to the duration of missing test. For single point missing measurement or short interval missing measurement, linear interpolation is used to correct adjacent time. For the sample segments with longer continuous missing measurement time, the window is not directly filled, but eliminated in the sample construction stage to avoid introducing false meteorological change trends. The identification of outliers is completed by combining the physical range of meteorological elements and the sliding statistical threshold. For example, if the range of temperature change in a certain hour significantly exceeds the variation level of adjacent periods, and the corresponding weather process is not recorded, it is marked as a suspicious point and smoothed by the neighborhood median.

After data cleaning and standardization, this paper uses sliding window to construct supervised learning samples. Let the input window length be L and the prediction step size be h, then the ith training sample can be expressed as follows.

$$X_i = [\hat{x}_i, \hat{x}_{i+1}, \dots, \hat{x}_{i+L-1}] \tag{4}$$

The corresponding labels are:

$$y_i = T_{i+L+h-1} \tag{5}$$

where, X_i is the model input sample and y_i is the temperature value at the future target time. L is used to control the range of historical information observable by the model, and h is used to determine the forecast lead. When h=1, the model executes the next time temperature prediction. When h is set to 3, 6, or 12, the model can be extended to a multi-look-ahead prediction task. In this paper, hourly temperature prediction is mainly used in the experiment, and the prediction step parameters are retained, which is easy to be adjusted according to the short-term forecast or intra-day forecast demand in subsequent business applications. The sliding window sample construction process is shown in Figure 2.

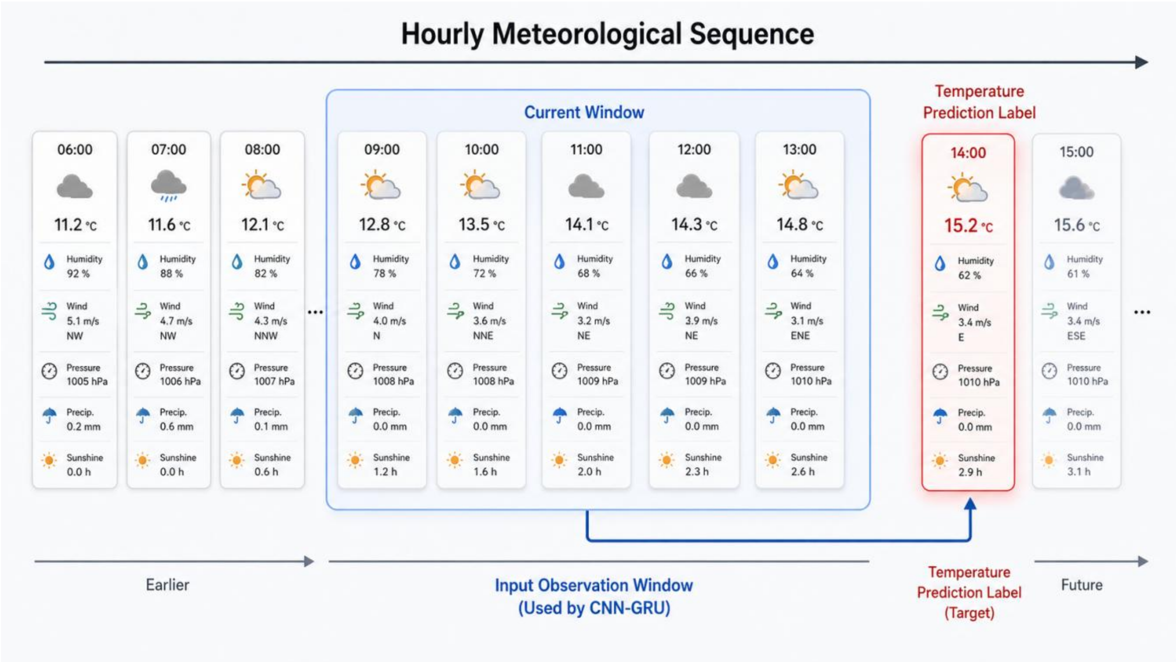

Figure 2: Schematic diagram of meteorological time series sample construction

Figure 2 illustrates the process of intercepting an input window from a sequence of consecutive hourly observations and generating a predicted label. Each window contains multiple meteorological variables, and the Windows slide forward according to a fixed step size, so as to convert the original time series into a sample set for CNN-GRU model training. Compared with the random sampling method, the sliding window can retain the time order of meteorological data, so that the model is exposed to various sequence forms such as continuous warming, cooling, stationary and abrupt changes in the training process. For the data input to the CNN layer, each sample is organized as a two-dimensional feature matrix in this paper:

$$M_i \in R^{L\times m} \tag{6}$$

where, L represents the length of historical time window and m represents the number of input meteorological variables. The rows of the matrix correspond to consecutive observation moments, and the columns correspond to different meteorological variables. This organization method enables CNN to simultaneously scan the time neighborhood and variable neighborhood in the local receptive field, and extract the short-term combination changes of meteorological elements. For example, when the air pressure decreases, humidity increases, and wind speed increases, the convolution kernel can identify the potential association between this combined feature and the temperature decrease. Subsequently, the extracted local features are fed into the GRU layer to further learn the dynamic change relationship in a long time range.

In order to enhance the expression of temperature periodicity in the model, this paper adds time encoding features in addition to basic meteorological variables. The hour and date features are converted to periodic functions:

$$C_t^{hour} = \sin\left(\frac{2\pi hour_t}{24}\right), \quad Q_t^{hour} = \cos\left(\frac{2\pi hour_t}{24}\right) \tag{7}$$

This treatment can avoid the problem of time numbering breaking numerically, and make the model more naturally learn the influence of the day-night cycle on the temperature change. If the seasonal differences in the study area are strong, the sine and cosine coding of the intra-year day series can also be introduced to supplement the seasonal variation information. After the above processing, the sample contains not only the real-time meteorological state, but also the periodic background on which the temperature change depends.

3.2 CNN-GRU temperature prediction network structure design

The design goal of CNN-GRU temperature forecasting network is to obtain both short-time local variation characteristics and continuous time series dependence from the multivariate meteorological matrix. The temperature change is not completely determined by the temperature at the last moment, and variables such as humidity, air pressure, wind speed, precipitation and sunshine will jointly affect the near-surface heat budget in different time Windows. If only the cyclic structure is used, the model is easy to focus the main weight on the time recurrence relationship, and the recognition of the multi-variable combination changes in a short time is insufficient. If only the convolutional structure is used, the model can extract local patterns, but it does not fully describe the continuous dependence such as the day and night cycle and the temperature recovery process after cold air transit. Therefore, in this paper, the convolutional neural network and the gated recurrent unit are connected in series to construct the prediction network, so that the local feature extraction results are further entered into the temporal memory module, and an end-to-end computing structure for temperature prediction is formed.

The model input in this paper is the meteorological sample matrix constructed in the previous section. In the experiment, the length of the input window is set to 24, that is, the hourly observation data of the past 24 h is used to predict the temperature in the future 1 h. Each time step contains 10 features, including air temperature, relative humidity, air pressure, wind speed, wind direction sine component, wind direction cosine component, precipitation, sunshine duration, hourly sine code and hourly cosine code. Therefore, the single sample input matrix size is 24×10. After entering the network, the matrix first passes through two one-dimensional convolution layers, and the convolution kernel slides along the time direction to cover the local association in the variable dimension, which is used to identify short-term patterns such as warming, cooling, stationary and abrupt changes. The local feature extraction process of convolutional layers can be expressed as follows.

$$F_i^{(r)} = \delta(W_r * M_i + b_r) \tag{8}$$

where, M_i represents the ith input sample matrix, $*$ represents the convolution operation, W_r and b_r represent the convolution kernel weight and bias term of the RTH layer respectively, $\delta(\cdot)$ is the ReLU activation function, and $F_i^{(r)}$ represents the local meteorological features obtained after convolution. In this paper, a two-layer convolution structure is set, the number of convolution kernels is 16 and 32, and the size of convolution kernel is 3. The smaller convolution kernel can avoid excessive smoothing of short-term changes, while retaining the meteorological disturbance information between adjacent hours. The convolutional layer is followed by a Max pooling layer, which compresses redundant features and preserves salient responses:

$$P_i = \max\left(F_i^{(2)}\right) \tag{9}$$

Where, P_i represents the local feature sequence after pooling. This processing can reduce the input dimension of the subsequent GRU layer, reduce the amount of training calculation, and weaken the interference of abnormal observations on the overall prediction results.

The pooled feature sequences are fed into the GRU layer to learn the temperature evolution relationship over a long time horizon. GRU controls the retention ratio of historical information through update gate and reset gate, which is able to capture sequence dependence under the condition of small parameter scale. The core calculation process is as follows.

$$z_t = \sigma(W_z p_t + U_z h_{t-1} + b_z) \tag{10}$$

$$r_t = \sigma(W_r p_t + U_r h_{t-1} + b_r) \tag{11}$$

$$\tilde{h}_t = \tanh(W_h p_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{12}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{13}$$

where, p_t represents the convolutional features of the input GRU at time t, h_{t-1} represents the hidden state at the previous time, z_t and r_t represent the update gate and reset gate respectively, $\tilde{h}_t$ represents the candidate hidden state, and $\odot$ represents element-by-element multiplication. In this paper, a two-layer stacked GRU structure is used, and the number of hidden units is set to 64 and 32, respectively. The upper GRU receives the output of the lower GRU, so that the model can continue to capture the temperature change rhythm across hours on the basis of short-time local features.

The end hidden state of the GRU output goes into the fully connected prediction layer. In order to reduce the risk of overfitting, Dropout is added before the fully connected layer to randomly discard part of the neuron response, so that the model does not rely too much on a certain type of meteorological variable. The final prediction output is:

$$\hat{y}_i = W_o h_i + b_o \tag{14}$$

where, $\hat{y}_i$ represents the future temperature forecast value corresponding to the ith sample, h_i is the comprehensive timing feature of the GRU output, W_o and b_o are the output layer parameters. The model was trained using AdamW optimizer with initial learning rate set to 2×10−4, batch size 64, and maximum number of training rounds 100. The training was stopped when the RMSE of the validation set did not decrease for 10 consecutive rounds to control overfitting and improve the stability of the model on the test set.

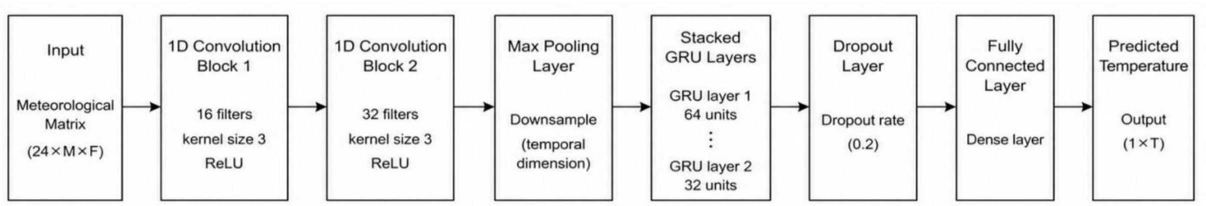

Figure 3: CNN-GRU temperature prediction network structure diagram

Figure 3 shows the calculation path of the proposed CNN-GRU temperature prediction network. In this structure, the convolutional layer is placed before the GRU layer, so that the network first extracts the local combination features from the original meteorological variables, and then the time-dependent modeling is completed by GRU. Compared with directly inputting the original variables into GRU, this method can reduce the interference of invalid variable fluctuations on hidden state updates. Compared with the simple CNN prediction, the proposed structure retains the continuous memory ability of the temperature sequence. The scale of the overall network parameters is moderate, and it is suitable for deployment in hourly forecast correction of meteorological data center, site-level short-term forecast and automatic business product generation.

4 Experimental Results

The experiments in this paper focus on the hourly temperature prediction task, and focus on testing the prediction accuracy and stability of the CNN-GRU model in multivariate meteorological time series. The experimental part includes four aspects: data set source and variable composition, meteorological data preprocessing, evaluation index setting and ablation experiment. The model input consists of continuous meteorological observation series, and the forecast target is the temperature value of 1 hour in the future. In order to ensure the interpretability of the experimental results, this paper compares the CNN-GRU model and the subsequent baseline models under the condition of unified training set, validation set and test set division, and analyzes the influence of the convolutional feature extraction module and the GRU time series modeling module on the prediction performance through ablation experiments.

4.1 Source of data set and composition of variables

The experimental data in this paper are from the hourly observation data of the ground meteorological observation station in Jilin Province, and the time range is set to January 1, 2020 to December 31, 2023. In order to enhance the regional representation of the data, sites with different topographic and climatic backgrounds such as Changchun, Jilin, Siping, Baicheng, and Yanbian were selected as sample sources. The data recording frequency is 1 h, which can reflect the characteristics of temperature change in Northeast China under the influence of low temperature in winter, rapid warming in spring and autumn, and precipitation in summer. In the experiment, the hourly meteorological records of the site were used as the basic sample unit, and the continuous observation sequence was constructed according to the time order to avoid the future information entering the training process caused by random disturbance.

The dataset contains conventional meteorological elements such as temperature, relative humidity, air pressure, wind speed, wind direction, precipitation, and sunshine duration. Since the wind direction is a periodic variable, it is converted into sinusoidal and cosine components in this paper to reduce the numerical fracture caused by the Angle boundary. In order to enhance the recognition ability of the model for day and night changes, hourly sine coding and hourly cosine coding are further added to form a 10-dimensional input feature. Each training sample is composed of the past 24 h observation matrix, and the label is the future 1 h air temperature value. The dataset is divided into training, validation, and test sets in chronological order with a ratio of 7:2:1. Table 2 presents the variable composition and modeling use of the experimental data in this paper. After quality control, a total of 175,320 valid hourly observation records were obtained, and 173,965 valid samples were constructed according to the 24-h sliding window. There were 121,775 groups in the training set, 34,793 groups in the validation set, and 17,397 groups in the test set. The overall missing test rate was 1.64%, and the Windows with continuous missing tests for more than 3 hours were not involved in model training and testing.

Table 2: Variables composition of temperature prediction data set

Variable Category	Variable Name	Data Format	Modeling Purpose
Prediction target	Temperature	Continuous value	Used as the model output label and included in the historical window input
Moisture condition	Relative humidity	Continuous value	Represents near-surface moisture variation and the cooling or warming environment
Dynamic and thermal background	Air pressure	Continuous value	Reflects weather system changes and warm–cold air activity
Wind field condition	Wind speed	Continuous value	Describes air transport intensity and local diffusion status
Wind field direction	Sine component and cosine component of wind direction	Periodic encoding	Represents wind direction changes and avoids angular boundary discontinuity
Weather process	Precipitation	Continuous value	Describes the influence of precipitation processes on temperature variation
Radiation condition	Sunshine duration	Continuous value	Reflects the intensity of shortwave radiation input
Temporal cycle	Sine encoding and cosine encoding of hour	Periodic encoding	Represents the influence of diurnal cycles on temperature variation

4.2 Preprocessing of meteorological data

In order to ensure that the hourly meteorological observation data can meet the requirements of CNN-GRU model training and testing, this paper performs quality control, scale unification and sample division on the original data. The dimensions of different variables in meteorological observation data are quite different; for example, temperature is expressed in °C, air pressure in hPa, wind speed in m/s, and precipitation has obvious sparsity. Without unified processing, variables with larger numerical ranges will occupy higher weights in the gradient update, which will affect the learning of the model on the law of temperature change. Therefore, in this paper, max-min normalization is used to map the continuous variables listed in Table 2 to the interval [0,1], and the processing formula is as follows.

$$x'_{t,j} = \frac{x_{t,j} - x_j^{min}}{x_j^{max} - x_j^{min}} \tag{15}$$

where, $x_{t,j}$ represents the original value of the JTH meteorological variable at time t, x_j^{min} and x_j^{max} represent the minimum and maximum value of the variable in the training set respectively, and $x'_{t,j}$ represents the value of the variable after normalization. The normalization parameter is only calculated based on the training set, and is applied to the validation set and the test set synchronously to avoid the information in the test phase entering the model training process in advance.

The processing of missing measurements is distinguished according to the length of missing measurements. Linear interpolation was used to maintain the continuity of time series for the records with continuous missing time less than 3 h. For Windows with continuous missing tests more than 3 h, it is eliminated in the sliding sample generation stage to avoid the model learning false changes caused by artificial filling. Linear interpolation can be expressed as follows.

$$\tilde{x}_t = x_a + \frac{t - a}{b - a}(x_b - x_a) \tag{16}$$

where, a and b represent the two valid observation moments before and after the missing measurement segment, respectively, and $\tilde{x}_t$ represents the interpolation result at the missing measurement time t. Outlier identification is done by combining meteorological physical and statistical thresholds. The median of adjacent Windows was used to replace the records that exceeded the reasonable range and deviated from the moving mean by three standard deviations. The abrupt change values caused by real weather processes, such as strong cooling and heavy precipitation, are retained to ensure that the model can learn the temperature response under extreme change scenarios.

After the cleaning was completed, the data was divided into training, validation, and test sets in chronological order with a ratio of 7:2:1. Random partition is not used in this paper, because it will destroy the temporal structure of the meteorological series and may make samples at adjacent moments appear in the training set and the test set simultaneously. The division based on time order is closer to the business prediction scenario, that is, the model is trained by using historical observations and then predicted for the future period.

4.3 Evaluation Metrics

In this paper, MAE, RMSE, MAPE and R2 are used to evaluate the temperature prediction results. MAE can directly reflect the mean absolute deviation between the predicted temperature and the measured temperature, and the unit is consistent with the temperature, which facilitates the interpretation of model errors from the perspective of meteorological operations. RMSE is more sensitive to larger errors and is suitable for testing the stability of the model in strong warming, strong cooling and weather turning points. MAPE is used to describe the relative error level, but numerical amplification may occur at low temperatures close to 0°C, so it is used as an auxiliary index in this paper. R^2 is used to measure the ability of the model to explain the variance of temperature change, and can reflect the degree of consistency between the predicted curve and the real change trend.

MAE is calculated as follows.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{17}$$

RMSE is calculated as follows.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{18}$$

MAPE is calculated as follows.

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i + \varepsilon}\right| \tag{19}$$

R^2 is calculated as follows.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{20}$$

where N represents the number of test samples, y_i represents the measured temperature of the ith sample, $\hat{y}_i$ represents the temperature predicted by the model, $\bar{y}$ represents the mean value of the measured temperature, and ε is a minimal constant used to avoid the calculation instability caused by the denominator of 0. The above indicators evaluate the performance of the model from four perspectives of average error, large error, relative error and trend explanation ability, which can completely reflect the accuracy and robustness of the CNN-GRU model in the hourly temperature prediction task.

4.4 Ablation experiments

In order to test the contribution of different structures in the CNN-GRU model to the temperature prediction performance, this paper sets up the ablation experiment under the uniform variable input condition shown in Table 2. The experimental comparison objects include linear regression, SVR, decision tree, MLP, CNN, GRU, LSTM, CNN-LSTM, and the complete CNN-GRU model of this paper. All models use the same training set, validation set and test set division, the prediction task is the future 1 h temperature prediction, and the evaluation indicators are MAE, RMSE, MAPE and R^2 in Section 4.3. Table 3 shows the prediction results of different models on the test set.

Table 3: Comparison of temperature prediction performance of different models

Model	MAE / ℃	RMSE / ℃	MAPE / %	R^2
Linear Regression	1.18	1.62	8.94	0.883
SVR	1.04	1.43	7.85	0.904
Decision Tree	1.12	1.55	8.37	0.891
MLP	0.91	1.26	6.72	0.927
CNN	0.78	1.08	5.81	0.941
GRU	0.72	1.01	5.35	0.948
LSTM	0.69	0.96	5.02	0.953
CNN-LSTM	0.61	0.88	4.56	0.959
CNN-GRU	0.56	0.82	4.31	0.962

It can be seen from Table 3 that the error of traditional regression model in hourly temperature prediction is relatively high, and linear regression is difficult to describe the nonlinear influence of variables such as humidity, air pressure and wind speed on temperature change. Although the decision tree model can express the local nonlinear relationship, it is insufficient to maintain the continuous time series change. MLP is better than the traditional model, but its input structure does not make full use of the time order, and the prediction results are still biased in the rapid cooling period. When CNN is used alone, the model can identify the combination characteristics of short-term meteorological variables, but the tracking of cross-hour temperature continuation trend is not stable. When GRU is used alone, the model performs well for the day-night cycle and continuous change process, but the error is still too large in the face of local disturbances such as sudden cooling after precipitation and cold air transit.

To further show the influence of different structures on RMSE, Figure 4 is plotted in this paper. The results in Figure 4 show that the full CNN-GRU model has the lowest RMSE of 0.82 ° C. After removing CNN, RMSE rises to 1.01℃, indicating that local meteorological combination features have obvious contributions to temperature prediction. After removing GRU, RMSE rises to 1.08℃, indicating that the temporal memory module is indispensable for the modeling of continuous temperature change. The RMSE rises to 0.91℃ after removing the time period encoding, which indicates that the information of day and night cycle can help the model to distinguish the temperature response of different periods under similar meteorological conditions.

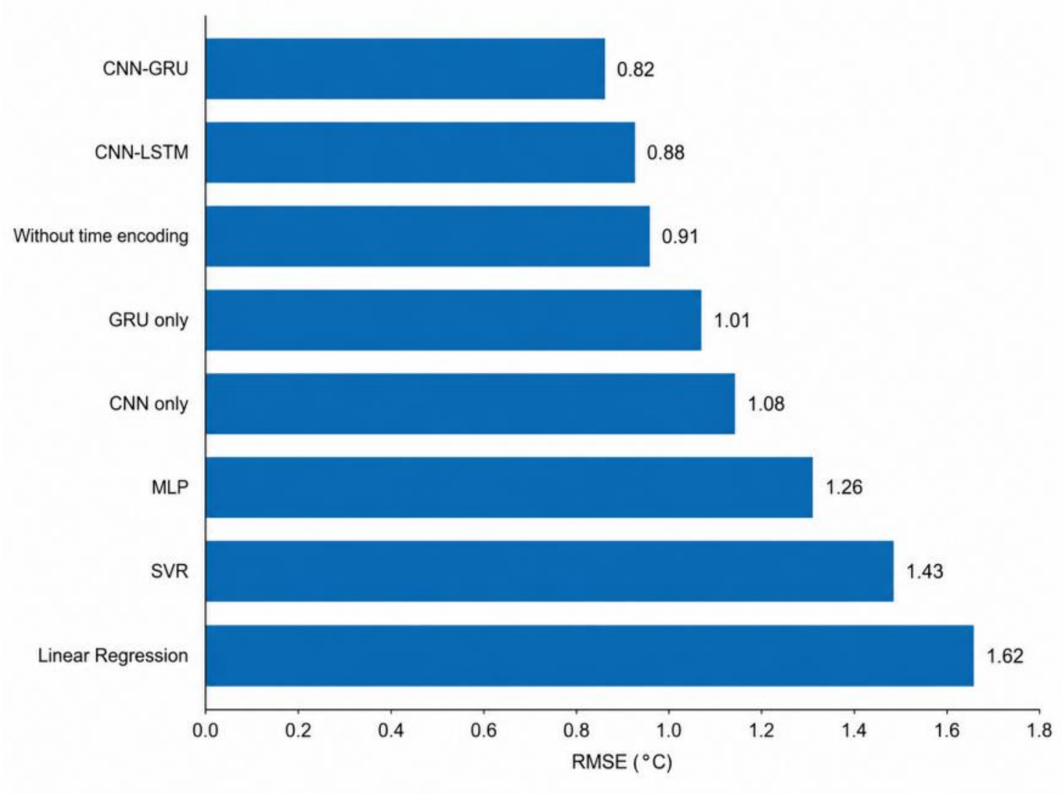

Figure 4: Comparison data plots of RMSE under different structural configurations

It can be seen from Figure 4 that the hybrid deep model is better than the single structure model as a whole, and CNN-GRU achieves lower error than CNN-LSTM. The main reason is that the number of GRU parameters is relatively small, and it is easier to obtain stable training results under the condition of limited scale of hourly site meteorological data. At the same time, the complete model uses CNN to extract short-term local variation features, and uses GRU to retain historical temperature states, so that the prediction curve can maintain high consistency in the periods of smooth warming, night cooling and weather conversion. The experimental results show that the performance improvement of the proposed model does not come from single layer stacking, but from the synergy between local feature extraction, temporal dependence modeling, and time period encoding.

5 Discussion

5.1 Comparative analysis of the proposed model and advanced prediction models

To verify the effectiveness of the proposed model, CNN-GRU is compared with prediction methods such as ARIMA, SVR, Random Forest, LSTM, CNN-LSTM and Transformer. All models used the same data set division and evaluation index, and the prediction target was the temperature of 1 hour in the future. Table 4 shows the performance comparison between the proposed model and typical advanced prediction models on the test set, where the traditional machine learning model is only used as a reference, and the prediction difference between the deep time series model and the hybrid neural network model is mainly compared.

Table 4: Performance comparison of different temperature prediction models

Method	MAE / ℃	RMSE / ℃	MAPE / %	R^2
ARIMA	1.31	1.74	9.45	0.869
SVR	1.04	1.43	7.85	0.904
Random Forest	0.98	1.36	7.18	0.912
LSTM	0.69	0.96	5.02	0.953
CNN-LSTM	0.61	0.88	4.56	0.959
Transformer	0.64	0.93	4.74	0.956
CNN-GRU	0.56	0.82	4.31	0.962

It can be seen from Table 4 that traditional statistical models and shallow machine learning models have limited adaptability in nonlinear meteorological changes. ARIMA has a certain ability to describe the stationary trend, but the error increases under the influence of cold air activity and precipitation. SVR and Random Forest are able to express partial nonlinear relationships, but underutilize continuous time dependencies. The prediction error of LSTM and CNN-LSTM decreases significantly, which indicates that the deep time series model is more suitable for handling multivariate meteorological series. Transformer has advantages in long-term dependency learning, but its parameter scale is large and the error is not lower than CNN-GRU when the scale of site-level hourly data is limited.

Figure 5 further demonstrates the differences in RMSE of different models. The RMSE of CNN-GRU is 0.06℃ lower than that of CNN-LSTM, 0.14℃ lower than that of LSTM, and 0.61℃ lower than that of SVR. This difference shows that GRU can effectively learn the continuous change process of temperature while maintaining a low parameter scale. The CNN module helps the model to extract the local combination signals of humidity, air pressure, wind speed and sunshine variables in advance, so that the prediction results are closer to the measured values in the weather conversion period.

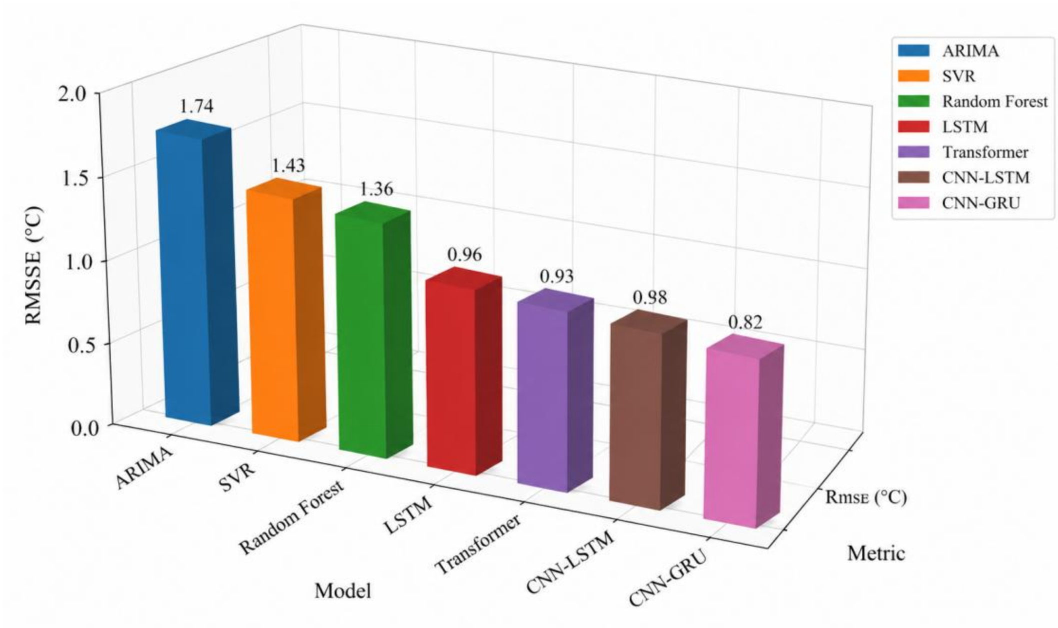

Figure 5: Comparison of RMSE of different prediction models

From the perspective of prediction mechanism, the advantage of CNN-GRU does not come from the simple increase of the number of network layers, but from the reasonable connection between local features and time series states. The convolutional layer can identify the combination changes of "decreasing air pressure, increasing humidity and enhancing wind speed" in a short time window, and the GRU layer further determines whether the combination will continue to be a cooling process for the next 1 h. For the sunny day process with obvious diurnal variation, the model can maintain smooth prediction by using time period coding and historical temperature trend. For fluctuating scenarios such as precipitation and cold air intrusion, the convolutional features provide GRU with more explicit disturbance input, so the prediction error is lower than that of the single time series model.

5.2 Time complexity analysis of the model

The temperature prediction model not only needs high precision, but also needs to meet the real-time calculation requirements of business systems. In this paper, the forward inference time of a single sample is used as the efficiency evaluation index to test the average inference time of different models in three types of environments: CPU, GPU and edge devices. The time complexity test environment is set as follows. The CPU platform uses Intel Core i7-12700 processor and 32 GB memory, the GPU platform uses NVIDIA RTX 4080 16 GB graphics card, and the edge device uses Raspberry Pi 4B 8 GB memory. The software environment is Ubuntu 22.04, Python 3.11, and PyTorch 2.2. The test results are shown in Table 5.

Table 5: Comparison of inference time of different models

Method	CPU / s	GPU / s	Edge Device / s	Number of Model Parameters / M
LSTM	0.061	0.021	0.172	0.42
CNN-LSTM	0.083	0.026	0.219	0.58
Transformer	0.146	0.043	0.384	1.27
CNN-GRU	0.074	0.018	0.186	0.49

Table 5 shows that the average inference time of CNN-GRU is 0.018 s in the GPU environment, 0.074 s in the CPU environment, and 0.186 s on the edge device. The time consumption is much lower than the service time interval of hourly temperature prediction, and it can meet the needs of site-level automatic update and batch rolling prediction. Compared with Transformer, the number of parameters of CNN-GRU is reduced by about 61.4%, and the inference time of edge devices is reduced by about 51.6%. Compared with CNN-LSTM, CNN-GRU achieves faster GPU inference while maintaining a lower RMSE. This result is related to the fact that the GRU gating structure is more concise than the LSTM, and also to the fact that the convolutional layer compresses the local feature dimension in advance.

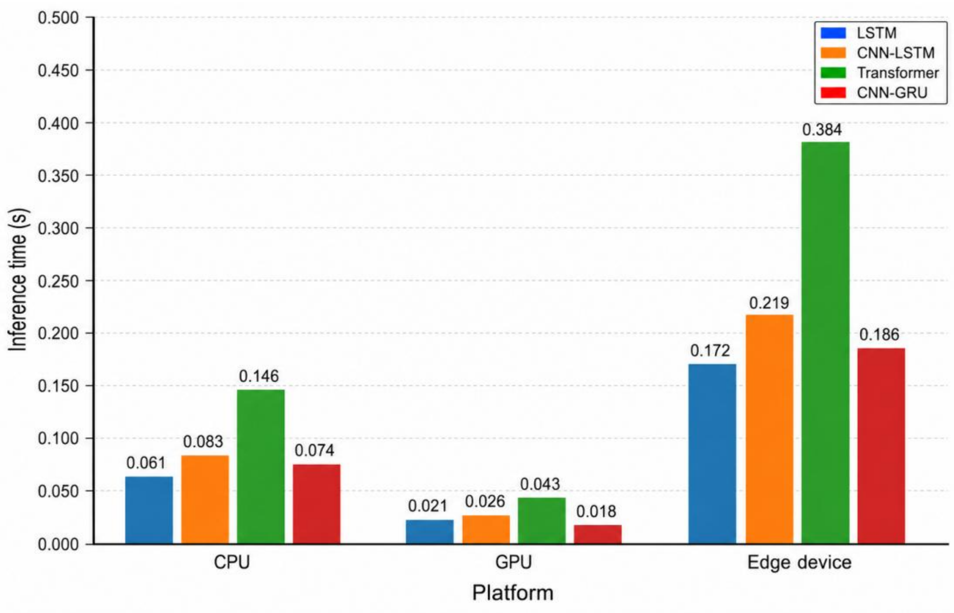

Figure 6: Comparison of inference time in different running environments

Figure 6 illustrates that the time consumption of CNN-GRU varies smoothly on different computing platforms. For the multi-site hourly temperature prediction task in Jilin Province, if 100 sites are predicted in batch at a time, the total time consumption of GPU can still be controlled within the range of seconds. If it is deployed in the base station or edge acquisition terminal, the single sample inference time of 0.186 s can also support short rolling updates. It should be noted that when the input window length is extended from 24 h to 72 h, the computation of the GRU layer will increase with the time step, and the inference cost can be further compressed by window clipping, feature selection or model distillation.

5.3 Scalability and Deployment Considerations for Temperature prediction models

The CNN-GRU model has good expansion conditions. Its input adopts a uniform time window matrix, and the variable dimension can be adjusted according to the integrity of the business site data. When some stations lack sunshine duration or precipitation records, mask features or regional interpolation can be used to supplement them. When radar, satellite cloud images or numerical forecast products are introduced, the corresponding variables can be added to the input layer, and the local feature fusion can be completed through the convolution module. Due to the small scale of the main parameters of the model, multiple sites can share the basic network structure, and then adapt to different terrain and underlying surface conditions through site coding or regional fine-tuning.

Figure 7 shows the accuracy and response time comparison under different deployment schemes. The centralized GPU deployment has the lowest RMSE of 0.82℃ and a single response time of 0.018 s, which is suitable for large-scale batch reasoning in provincial meteorological data centers. The RMSE of CPU server deployment remains the same, and the response time is increased to 0.074 s, which is suitable for city and county level business systems. The response time of edge device deployment is 0.186 s, which can still meet the hourly prediction requirements. If 8-bit quantization is further adopted, the model size can be reduced from 38 MB to 15 MB, and the reasoning time of edge devices can be reduced to 0.121 s, but the RMSE will rise to 0.87℃, which indicates that the lightweight deployment needs to maintain a balance between accuracy and speed.

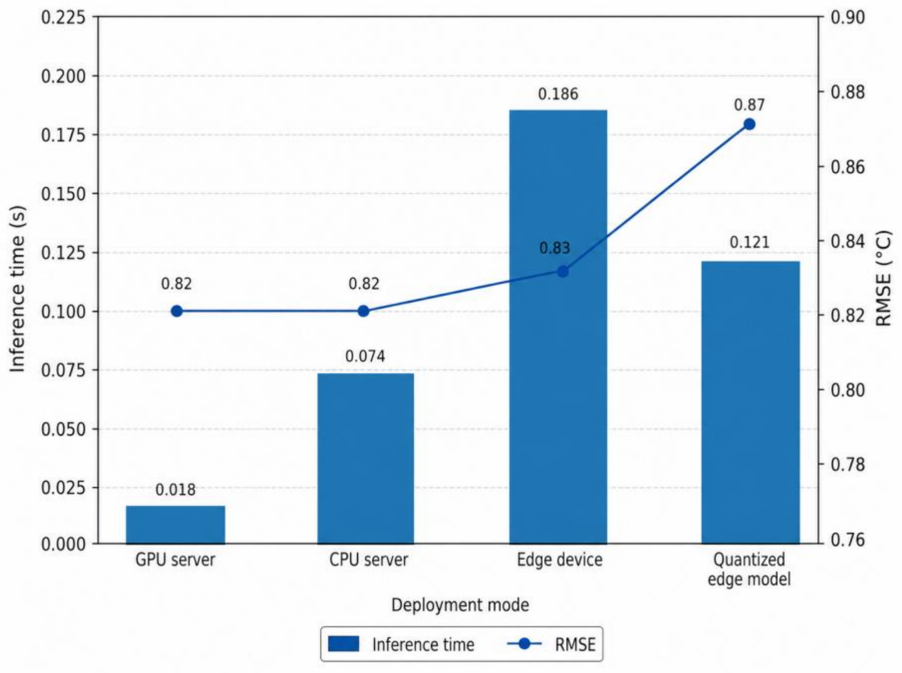

Figure 7: Comparison of RMSE and response time for different deployment schemes

From the perspective of business deployment, the model in this paper can operate in the way of "central training, hierarchical reasoning, and regular update". The provincial data center was responsible for integrating multi-site historical data, completing model training and parameter updating. The city-county-level business system makes batch prediction according to the local site data. The edge terminal undertakes short-time fast reasoning and temporary prediction in the state of disconnection. For processes such as strong cooling, blizzard and cold wave in winter, the latest samples can be added to the training set after the event is over, and the model parameters can be regularly fine-tuned to improve the adaptability of the model to extreme samples. If it is deployed in a larger area, it is also necessary to establish a site quality evaluation mechanism to avoid sites with high missing rate or incomplete relocation records affecting the stability of the model.

5.4 Application Value analysis of meteorological services

The value of CNN-GRU model in meteorological business is mainly reflected in three aspects: short-term correction, risk early warning and refined service. For the hourly temperature forecast correction, the model can quickly correct the temperature results of the next hour based on the latest observation data, especially in the process of radiation cooling at night, cooling after precipitation and cold air transit, which can provide data reference for forecasters. The experimental results show that the MAE of CNN-GRU is 0.13℃ lower than that of LSTM and 0.48℃ lower than that of SVR. Although the range seems limited, it has practical significance in road icing, low temperature protection of facility agriculture and urban heating scheduling. In agrometeorological services, short-time temperature forecasting can be used for low temperature frost risk tips. If the model predicts that the temperature will be close to 0 ° C and show a decreasing trend in the next 1 h, the system can combine humidity, wind speed and land surface temperature data to generate a hierarchical reminder. For urban operation and management, more stable temperature prediction is helpful for heating load regulation, road icing risk identification and public travel tips. For meteorological data centers, CNN-GRU model has automatic operation conditions, which can be embedded in existing data processing links, and automatically generate site-level short-term temperature prediction products after observation data warehousing and quality control are completed. It should be noted that the model in this paper still has some limitations. The current experiment is mainly based on the hourly observation data of the ground station in Jilin Province, which can better reflect the seasonal difference of temperature and the influence of cold air in Northeast China. However, more regional data are needed to verify the applicability to plateau, coastal and tropical regions. The model mainly predicts 1 hour in the future, and the label and loss weight need to be redesigned for longer prediction such as 6 hours and 12 hours. Subsequent studies can incorporate numerical weather prediction products, terrain height, land use type and satellite cloud cover data into the input space, and combine uncertainty estimation to output prediction intervals, so as to improve the operational reliability of the model under complex weather background.

6 Conclusions

Temperature prediction is an important basic task in meteorological service, agricultural production scheduling and urban operation management. Aiming at the characteristics of multi-variable coupling, obvious short-term disturbance and strong continuous dependence of hourly temperature series, this paper constructs a temperature prediction model based on convolutional neural network and gated recurrent unit. The model takes the multivariate meteorological observation matrix as input, extracts the combination features of humidity, air pressure, wind speed, precipitation and sunshine in the local time window by CNN, and then uses GRU to capture the time series memory relationship in temperature change to realize the prediction of future 1 hour temperature. Experimental results show that the CNN-GRU model achieves RMSE of 0.82℃, MAE of 0.56℃, MAPE of 4.31%, R2 of 0.962, which is better than ARIMA, SVR, LSTM, CNN-LSTM, Transformer and other comparison methods. The model maintains low inference time on GPU, CPU and edge devices, and has the potential for hourly rolling prediction and business system deployment. Numerical weather prediction products, terrain factors and multi-site transfer learning methods can be further introduced in future research, and uncertainty estimation and interpretability analysis can be combined to improve the stability of the model in complex weather processes and cross-regional applications.

References

[1] Uluocak I, Bilgili M. Daily air temperature forecasting using LSTM-CNN and

GRU-CNN models[J]. Acta Geophysica, 2024, 72(3): 2107-2126.

[2] Pangaribuan A R, Yuniarno E M, Rachmadi R F. Air temperature prediction with LSTM and LSTM-GRU: A two-year meteorological data[C]//2024 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM). IEEE, 2024: 1-5.

[3] Faulina R, Nuramaliyah N, Safitri E. Air Temperature Prediction System Using Long Short-Term Memory Algorithm[J]. Rekayasa, 2024, 17(3): 463-473.

[4] Saleem M, Saleem M M, Waseem F, et al. An ensemble forecasting method based on optimized LSTM and GRU for temperature and humidity forecasting[J]. Engineering, Technology & Applied Science Research, 2024, 14(6): 18447-18452.

[5] Tran V, Septier F, Murakami D, et al. Spatial–temporal temperature forecasting using deep-neural-network-based domain adaptation[J]. Atmosphere, 2024, 15(1): 90.

[6] Shah I, Mubassir P, Ali S, et al. A functional autoregressive approach for modeling and forecasting short-term air temperature[J]. Frontiers in Environmental Science, 2024, 12: 1411237.

[7] Matlani P, Shukla B. Maximum temperature forecasting using deep learning algorithm by hyperparameter optimization[C]//E3S Web of Conferences. EDP Sciences, 2024, 585: 02006.

[8] Teixeira R, Cerveira A, Pires E J S, et al. Enhancing weather forecasting integrating LSTM and GA[J]. Applied Sciences, 2024, 14(13): 5769.

[9] Zenkner G, Navarro-Martinez S. A flexible and lightweight deep learning weather forecasting model: G. Zenkner and S. Navarro-Martinez[J]. Applied Intelligence, 2023, 53(21): 24991-25002.

[10] Venkatachalam K, Trojovský P, Pamucar D, et al. DWFH: An improved data-driven deep weather forecasting hybrid model using Transductive Long Short Term Memory (T-LSTM)[J]. Expert systems with applications, 2023, 213: 119270.

[11] Suleman M A R, Shridevi S. Short-term weather forecasting using spatial feature attention based LSTM model[J]. IEEE Access, 2022, 10: 82456-82468.

[12] Shrivastava V K, Shrivastava A, Sharma N, et al. Deep learning model for temperature prediction: A case study in New Delhi[J]. Journal of Forecasting, 2023, 42(6): 1445-1460.

[13] Sharma G, Dhumal R, Singh T P, et al. Analysis and Forecasting of Monthly Average Temperatures Using Timeseries Models[C]//2023 International Conference on Integration of Computational Intelligent System (ICICIS). IEEE, 2023: 1-4.

[14] Essa Y, Hunt H G P, Gijben M, et al. Deep learning prediction of thunderstorm severity using remote sensing weather data[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2022, 15: 4004-4013.

[15] Faraji M, Nadi S, Ghaffarpasand O, et al. An integrated 3D CNN-GRU deep learning method for short-term prediction of PM2. 5 concentration in urban environment[J]. Science of The Total Environment, 2022, 834: 155324.

[16] Shirzadi N, Nasiri F, Menon R P, et al. Smart urban wind power forecasting: Integrating weibull distribution, recurrent neural networks, and numerical weather prediction[J]. Energies, 2023, 16(17): 6208.

[17] Barış C, Yanarateş C, Altan A. A robust chaos-inspired artificial intelligence model for dealing with nonlinear dynamics in wind speed forecasting[J]. PeerJ Computer Science, 2024, 10: e2393.

[18] Abebe W T, Endalie D. Artificial intelligence models for prediction of monthly rainfall without climatic data for meteorological stations in Ethiopia[J]. Journal of Big Data, 2023, 10(1): 2.

[19] Peeriga R, Rinku D R, Bhaskar J U, et al. Real-time rain prediction in agriculture using ai and iot: A bi-directional lstm approach[J]. Engineering, Technology & Applied Science Research, 2024, 14(4): 15805-15812.

[20] Nandi A, De A, Mallick A, et al. Attention based long-term air temperature forecasting network: ALTF Net[J]. Knowledge-Based Systems, 2022, 252: 109442.

[21] Alomar M K, Khaleel F, Aljumaily M M, et al. Data-driven models for atmospheric air temperature forecasting at a continental climate region[J]. PLoS One, 2022, 17(11): e0277079.

[22] Subair H, Selvi R P, Vasanthi R, et al. Minimum Temperature Forecasting Using Gated Recurrent Unit[J]. Int. J. Environ. Clim. Change, 2023, 13(9): 2681-2688.

[23] Akilan T, Baalamurugan K M. Automated weather forecasting and field monitoring using GRU-CNN model along with IoT to support precision agriculture[J]. Expert systems with applications, 2024, 249: 123468.