Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

STAM-LSGRU: a spatiotemporal radar echo extrapolation algorithm with edge computing for short-term forecasting

Hailang Cheng[1], Mengmeng Cui[1,2*] and Yuzhe Shi[1]

Abstract

With the advent of Mobile Edge Computing (MEC), shifting data processing from cloud centers to the network edge presents an advanced computational paradigm for addressing latency-sensitive applications. Specifically, in radar systems, the real-time processing and prediction of radar echo data pose significant challenges in dynamic and resource-constrained environments. MEC, by processing data near its source, not only significantly reduces communication latency and enhances bandwidth utilization but also diminishes the necessity of transmitting large volumes of data to the cloud, which is crucial for improving the timeliness and efficiency of radar data processing. To meet this demand, this paper proposes a model that integrates a spatiotemporal Attention Module (STAM) with a Long Short-Term Memory Gated Recurrent Unit (ST-ConvLSGRU) to enhance the accuracy of radar echo prediction while leveraging the advantages of MEC. STAM, by extending the spatiotemporal receptive field of the prediction units, effectively captures key inter-frame motion information, while optimizations to the convolutional structure and loss function further boost the model's predictive performance. Experimental results demonstrate that our approach significantly improves the accuracy of short-term weather forecasting in a mobile edge computing environment, showcasing an efficient and practical solution for processing radar echo data under dynamic, resource-limited conditions.

Keywords Radar echo extrapolation, Mobile edge computing, Deep learning, Spatiotemporal

Introduction

The precision and timeliness of weather forecasting are crucial for addressing extreme weather events, agricultural production, aviation safety, and many other domains. Radar echo extrapolation, as a vital weather prediction technique, provides essential information on short-term weather changes. However, the effectiveness of this method greatly depends on the accurate capture and analysis of spatiotemporal features in radar data [1, 2].

Traditional radar extrapolation methods primarily rely on linear or simple mathematical models to predict weather patterns, which often perform poorly in handling complex weather systems [3]. With the advancement of MEC and Artificial Intelligence (AI) technologies, new solutions have emerged for radar echo extrapolation. The low-latency characteristic of MEC allows for rapid processing of data near its point of origin, while AI, particularly deep learning technologies, demonstrate immense potential in analyzing large-scale, complex datasets [4, 5].

Firstly, MEC plays a pivotal role in processing radar data. Traditionally, radar data required transmission to remote servers for processing and analysis, which was not only time-consuming but could also lead to data delays [6]. MEC significantly reduces data transmission time by providing computational resources near the data source, thus accelerating data processing [7]. This type

*Correspondence:
Mengmeng Cui
cuimengmeng@nuist.edu.cn
[1] School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2] Jiangsu Province Engineering Research Center of Advanced Computing and Intelligent Services, Nanjing 210044, China

of near-source processing is particularly well-suited for weather forecasting, as it necessitates rapid response and real-time analysis [8]. Secondly, AI technologies, especially machine learning and deep learning, have proven highly effective in interpreting radar data and enhancing forecast accuracy. Deep learning models can learn from historical weather data and predict future changes in weather patterns [9]. These models are particularly adept at handling large volumes of radar data and extracting meaningful insights, aiding meteorologists in making more accurate predictions [10].

In recent years, there has been growing interest in developing algorithms to infer radar echoes beyond the instrument range and to forecast the evolution of echoes over time [11–14]. Currently, weather forecasting methods can be broadly categorized into two main approaches: numerical weather prediction (NWP) methods and radar echo extrapolation. NWP methods utilize fluid dynamics and thermodynamic laws to simulate the physical processes of the lower atmosphere, providing predictions based on complex physical state equations and supercomputers [15]. While NWP methods offer valuable insights, they face challenges such as prediction delays, low resolution, and limitations in forecasting sudden severe weather events [16, 17]. On the other hand, radar echo extrapolation methods, such as artificial neural networks [18], support vector machines [19], and decision trees [20], leverage radar data to understand the relationship between radar echoes and other variables, enabling the prediction of future weather conditions. These data-driven methods have gained attention in recent years due to the availability of large amounts of historical data and have shown superior performance in various fields [21, 22].

The purpose of radar echo extrapolation is to predict future radar echo maps for a specific area based on previously observed radar echoes. This prediction task poses significant challenges as it requires spatiotemporal modeling of radar data to accommodate high resolution, thereby rendering it a spatiotemporal forecasting problem. Convolutional Neural Networks and Recurrent Neural Networks have been extensively utilized in such spatiotemporal forecasting tasks [23, 24]. However, existing models still confront challenges in handling high spatiotemporal resolution and complex non-stationary information, particularly in the context of convection formation and dissipation. Moreover, theoretical models for radar echo extrapolation are capable of generating prediction sequences of any length. Yet, as the prediction length increases, error accumulation can lead to image blurriness and loss of details.

In this context, the integration of MEC and AI offers a new perspective for radar echo extrapolation. The core advantage of MEC lies in its low-latency characteristics, enabling rapid processing near the data generation point, which is particularly crucial for real-time radar data analysis. This capability for rapid response, coupled with advanced abilities in handling and analyzing large-scale complex datasets, provides robust support for enhancing the efficiency and accuracy of radar data processing. Therefore, this paper proposes a STAM-LSGRU network to address key challenges in radar echo extrapolation, including error accumulation and effective extraction of high-order non-stationary information. The main contributions of this paper can be summarized as follows:

- By designing STAM, this model achieves long-term prediction in MEC environments and effectively captures global spatiotemporal dependencies, significantly reducing error accumulation during the prediction process.
- A predictive RNN unit is devised, integrating the Inception network structure, which effectively captures high-order non-stationary information by employing multi-scale layers and receptive fields, thereby enhancing the model's prediction accuracy.
- By incorporating Critical Success Index (CSI) and Heidke Skill Score (HSS) evaluation metrics, improvements are made to the loss function, reducing the ambiguity and distortion of the prediction results, and enhancing predictive performance in heavy rainfall regions.

The remainder of this article is organized as follows: Related Work, Methodology, Experiments, and Conclusion. The Related Work section provides an overview of previous studies, highlighting existing methods and findings in the field. The Methodology section details the theoretical frameworks and techniques used in the research, outlining the design of the proposed model. The Experiments section introduces the experimental setup, dataset description, and obtained results, offering empirical validation and comparison. Finally, the Conclusion section summarizes the contributions and proposes directions for future research.

Related work

The MEC with AI technologies demonstrates significant potential in fields such as radar echo extrapolation and weather forecasting. As an emerging computational paradigm, MEC shifts computational tasks from the cloud to the network edge, achieving low-latency and high-efficiency data processing. For instance, zhou [25] noted that 'Edge Intelligence' is a product of the convergence of MEC and AI. This concept aims to provide superior solutions for key issues in edge computing. It also explores how to establish AI models on edge devices, including model training and inference processes. This approach exemplifies the innovative strides being made in combining AI with edge computing, to optimize computational efficiency and enhance the capabilities of edge devices in processing complex tasks.

MEC technology enhances data processing efficiency by providing computational resources at the network edge, thereby significantly reducing latency and bringing processing closer to the data source. The integration of AI technologies further elevates MEC's data processing capabilities. AI algorithms, particularly deep learning models, have demonstrated exceptional performance in areas like image recognition, pattern detection, and predictive analytics. Al-Habob and Dobre [26] explored the symbiotic relationship between MEC and AI, highlighting AI's critical role in the MEC offloading process, such as resource management and scheduling. Huang [27] proposed an infrastructure for executing machine learning tasks on MEC servers, assisted by Reconfigurable Intelligent Surfaces. Deng [28] discussed the role of AI with software orchestration and hardware acceleration in reducing edge computing latency. Yazid [29] provided a comprehensive review of Unmanned Aerial Vehicles (UAVs) in the application of MEC and AI, exploring their role in enhancing the efficiency of IoT applications. Dahmane [30] introduced a blockchain-based AI paradigm for secure implementation of UAVs in MEC. Wang [31] surveyed the convergence of MEC, Metaverse, 6G wireless communications, AI, and blockchain, and their impact on modern applications. Chakraborty and Sukapuram [32] examined the application of MEC in urban informatics, emphasizing its contribution to the development of smart cities.

Recently, deep learning-based radar echo extrapolation models have been proposed that are more accurate than traditional methods. In 2015, Shi [21] introduced the Convolutional Long-Short Term Memory (ConvLSTM) model for precipitation nowcasting. This model is designed to handle time series data with spatial structures and uses convolutional operations instead of the Hadamard product in the FC-LSTM [33]. In 2016, Shi [34] improved the ConvGRU model and proposed the TrajGRU model, which can dynamically learn the network recursive structure. Wang [35, 36] proposed PredRNN and PredRNN++ models based on ConvLSTM. The LSTM unit was rebuilt by the team, and the Spatiotemporal LSTM (ST-LSTM) unit was created, allowing memory state to propagate in both vertical and horizontal directions and no longer being restricted to each individual LSTM unit. They developed the Gradient Highway Units (GHU), which were added between the first and second layers of the model at each time step, and the Causal LSTM unit in subsequent research. This greatly shortened the gradient propagation path and solved the issue of information loss in long-term predictions. Due to the relatively simple state transition function and the ineffective differential signal processing in the majority of RNNs used for spatiotemporal prediction, it is challenging for the model to learn complicated spatiotemporal changes. They proposed the MIM structure [37], which is used to extract stationary features and non-stationary features (MIM-S layer and MIM-N layer, respectively). The model achieved better performance on radar datasets. Lin [38] proposed the Sa-ConvLSTM, which added a self-attention mechanism at the output end of ConvLSTM. By using an additional memory unit M and a self-attentive feature aggregation mechanism, it computed pairwise similarity scores to fuse the previous features that contain the global spatial receptive field. Wu [39] has made further advances in the utilization of spatiotemporal information by proposing the MotionRNN architecture and designing the MotionGRU unit. This unit can model transient changes and motion trends in a unified manner, and a new motion highway has been introduced, which significantly enhances the ability to predict variable motion and avoids the issue of vanishing motion when stacking multiple prediction models. Chang [40] proposed a spatial-temporal residual prediction model applicable for high-resolution video prediction. The model employs a spatial-temporal encoding-decoding scheme to capture complex motion information in high-resolution videos. Jin [41] proposed a novel spatiotemporal graph neural network model called BGGRU, which integrates spatial and temporal information to explore the temporal patterns and spatial propagation effects of time series, aiming to enhance prediction accuracy. However, in the above methods, the global spatiotemporal dependencies of radar echoes have not been fully explored. This paper analyzes existing spatiotemporal prediction models and proposes an STAM module that addresses the issue of error accumulation. Additionally, the convolutional structure and loss function of the basic unit are improved, resulting in more accurate predictions of high-echo regions at different scales.

Methodology

The task of radar echo extrapolation aims to learn the mapping from input sequences to a latent space. To achieve this objective, we constructed a convolutional recurrent neural network, STAM-LSGRU. As shown in Fig. 1, its recurrent connections bestow it with memory functionality, enabling the capture and storage of previously inputted information. This memory capability allows the network to consider the information of the entire sequence comprehensively, rather than being

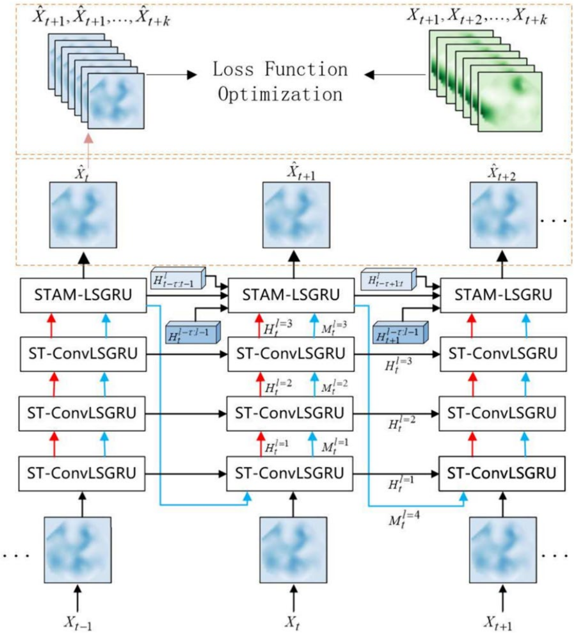


Fig. 1 The overall framework of STAM-LSGRU

limited to the inputs at the current timestep. Within STAM-LSGRU, this memory mechanism is crucial, allowing the network to effectively learn patterns and regularities within sequence data, thereby achieving the mapping from input sequences to latent space. By stacking three ST-ConvLSGRUs and one STAM-LSGRU, an encoder-decoder network is formed, which is currently the mainstream method for spatiotemporal sequence prediction. The ST-ConvLSGRU integrates the temporal information flow processing capabilities of the conventional GRU with a newly added spatial memory flow propagation mechanism, achieving simultaneous capture of temporal and spatial dependencies. The STAM-LSGRU predicts the next time step's radar echo image, not solely relying on the output of the previous time step, allowing for better handling of long sequence inputs and outputs. At a single time step, the vertical arrows represent the direction of memory and state updates along the spatial dimension, while the horizontal arrows represent the direction of updates along the time dimension. The spatiotemporal memory M is transferred from the lowest layer of the recurrent unit to the highest layer within a single time step, and is then transferred to the lowest layer of the following time step in a "Z"-shaped direction, first along the spatial dimension and then along the time dimension. The input radar data is downsampled by a factor of four and goes through three layers of ST-ConvLSGRU for information extraction and transformation before being input into the STAM-LSGRU. This enables the model to focus on past input states, avoiding error accumulation. The output is then upsampled to obtain the final prediction results. In order to improve the prediction of high echo areas, this paper proposes an enhanced loss function. In addition, the gating mechanisms has been optimized with an inception module for all the basic units of RNNs. Experimental results demonstrate that employing the STAM-LSGRU network leads to a significant improvement in prediction accuracy, thereby enabling more precise forecasting of future echo image sequences at different time points.

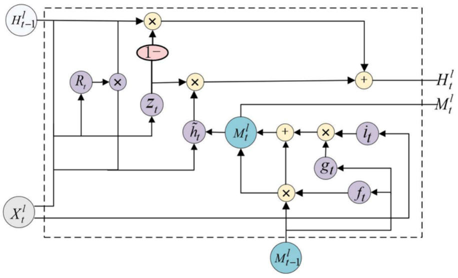


Fig. 2 The structure of Spatiotemporal Long Short Gated Recurrent Unit (ST-ConvLSGRU)

ST-ConvLSGRU

Inspired by ST-LSTM [35], this paper introduces the concept of LSTM memory units into the ConvGRU model, resulting in the ST-ConvLSGRU model depicted in Fig. 2 This model serves as the foundation for subsequent improvements. The original ST-LSTM model utilized a dual LSTM structure to process images with high spatiotemporal resolution effectively, storing and transmitting spatiotemporal information within and outside the memory cells. However, this structure tends to increase the model's complexity and the number of parameters, leading to overfitting issues. To address this challenge, this work integrates LSTM memory units into ConvGRU, aiming to reduce the model's parameter count while maintaining effective spatiotemporal information processing. This enhanced ST-ConvLSGRU model, compared to ST-LSTM, not only reduces the model's complexity but also enhances its capability to handle spatiotemporal information, avoiding overfitting and thus enabling more accurate predictions. As shown in Fig. 2, Z_t, R_t and X_t are update gates, reset gates and input states, respectively. $\tilde{h}_t$, i and f are new information, input gates, and forget gates, respectively. g is used as a temporary variable to update M. t denotes the t_{th} time step and l denotes that the loop cell is located at the l_{th} level of the stacked structure.

For a single ST-ConvLSGRU unit at time t, if the unit is located in the first layer (i.e., when l=1), the input state

X_t^l is a tensor converted from the radar echo map input at the current time. If the unit is not in the first layer (i.e., l >1), the hidden state H_t^{l-1} output at time t is used as the input state X_t^l for the unit. The ST-ConvLSGRU unit first passes the input state X_t^l and the hidden state H_{t-1}^l output by the unit in the same layer at time t-1 through a gating structure. Two different convolution filters are applied to obtain the reset gate R_t, the update gate Z_t, and the new information vector $\tilde{h}_t$. The calculation method for R_t and Z_t is consistent with that of ConvGRU, and is shown as follows, '*' denotes the convolution operation:

$$\begin{aligned} Z_t &= \sigma\left(W_z * \left[X_t, H_{t-1}^l\right] + b_z\right) \\ R_t &= \sigma\left(W_r * \left[X_t, H_{t-1}^l\right] + b_r\right) \end{aligned} \tag{1}$$

Similar to LSTM, the input state X_t^l and spatiotemporal memory M_{t-1}^l are fed into a gated structure. Three different convolution filters are applied to obtain the forget gate f_t, input gate i_t, and input modulation gate g_t. The forget gate f_t and element-wise Hadamard product are used to forget unimportant features of past time steps in the temporal memory M. The input gate and input modulation gate are used to update the features in memory through element multiplication, resulting in the updated spatiotemporal memory M_t^l. The '$\circ$' represents the Hadamard product. This process can be represented as follows:

$$\begin{aligned} g_t &= \tanh\left(W_{xg} * X_t + W_{hg} * H_{t-1}^l + b_g\right) \\ i_t &= \sigma\left(W_{xi} * X_t + W_{hi} * H_{t-1}^l + b_i\right) \\ f_t &= \sigma\left(W_{xf} * X_t + W_{hf} * H_{t-1}^l + b_f\right) \\ M_t^l &= f_t \circ M_t^{l-1} + i_t \circ g_t \end{aligned} \tag{2}$$

Next, the input state X_t^l, hidden state H_t^{l-1}, and updated spatiotemporal memory M_t^l are convolved using convolution filters to obtain new information $\tilde{h}_t$. The hidden state H_t^l is updated using the reset gate R_t and update gate Z_t, which extract abundant and rich spatiotemporal features through two gating mechanisms. As a result, the extrapolation network is able to accurately model the motion of radar echoes and make precise predictions on whether they will continue to expand or dissipate in the future. This process can be represented as follows:

$$\begin{aligned} \tilde{h}_t &= \sigma\left(W_{xh} * X_t^l + W_{hh} * H_{t-1}^l + W_{mh} * M_t^l + R_t \circ H_{t-1}^l + b_h\right) \\ H_t^l &= (1 - Z_t) * H_{t-1}^l + Z_t * \tilde{h}_t \end{aligned} \tag{3}$$

Spatiotemporal attention memory module

The extrapolation of radar echoes can also be classified as a regression problem. The extrapolation model theoretically has the ability to generate prediction sequences of arbitrary lengths. However, as the prediction length increases, there is a strong interdependence between adjacent frames, resulting in cumulative errors, which leads to blurred and distorted extrapolated images with missing details. To address this issue and enable the model to review the historical input sequence at each predicted time step, a STAM is constructed to utilize the input H_t^l of the l_{th} layer, to recall the historical input X_h^l. The model can adaptively learn the mapping from $X_{0:\mathrm{n}}$ to $X_{\mathrm{n}+1:T}$ based on a rich history of data.

$$\hat{X}_{\mathrm{n+1:T}} = \underset{X_{\mathrm{n+1:T}}}{\operatorname{argmax}}\, p(X_{\mathrm{n+1:T}} \mid X_{0:\mathrm{n}}) \tag{4}$$

The specific implementation is illustrated in Fig. 3, and its design is inspired by the dual-attention mechanism, which is named as STAM. STAM receives three inputs, including the predicted results of past time series $H_{t-\tau:t-1}^l$, the multi-layer results $H_t^{l-\tau:l-1}$ of the current time series, and the low-level results H_t^l of the current time series, where τ represents the step size. STAM consists of two modules, the attention module and the fusion module. The convolutional layer passes the current hidden state $H_t^l \in R^{C\times H\times W}$, and generates the query $Q \in R^{C\times H\times W}$ through a 1x1 convolution operation, where C, H, and W represent the number of channels, the length and width of the input data, respectively. Similarly, the key value $K_t \in R^{\tau\times C\times H\times W}$ and value $V_t \in R^{\tau\times C\times H\times W}$ can be obtained from the predicted results of the past time series $H_{t-\tau:t-1}^l \in R^{\tau\times C\times H\times W}$ through two independent convolutions. The weight matrix A_t is obtained by multiplying Q and K_t , and then applying sum and softmax operations:

$$A_t = \mathrm{softmax}(\mathrm{sum}(Q \circ K_t)) \tag{5}$$

Subsequently, the new temporal state T_t^l can be calculated according to the formula of temporal attention:

$$T_t^l = A_t \circ V_t \tag{6}$$

Finally, the reshaped T_t^l is resized to the same size as the original hidden state, and is used as the input of the fusion module.

The above approach enables adaptive learning of the historical input X_t^l in the temporal dimension. In order to address the problem of information loss during the propagation process from the low-level to high-level layers, the output of each convolutional neural network layer is kept in the multi-layer state. Then, the top-level hidden state H_t^l is used to recall $H_t^{l-\tau:l-1}$ and generate a new spatial hidden state S_t^l as the input of the fusion module.

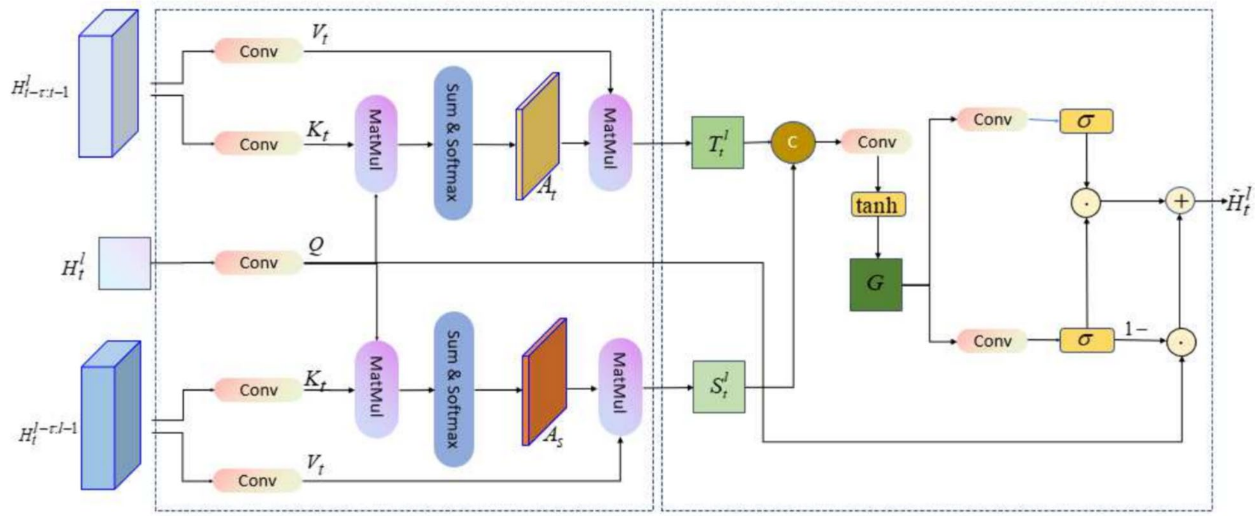


Fig. 3 The STAM consists of two components: the attention module and the fusion module

The computation process is akin to temporal attention, whereby the current hidden state $H_t^l \in R^{C\times H\times W}$ can be transformed into a query $Q \in R^{C\times H\times W}$ by reshaping the convolutional layer. Subsequently, the keys $K_s \in R^{\tau\times C\times H\times W}$ and values $V_s \in R^{\tau\times C\times H\times W}$ can be generated via two independent 1x1 convolutions using $H_t^{0:l-1} \in R^{\tau\times C\times H\times W}$. The weight matrix A_s is obtained by multiplying Q and K_s, followed by summation and softmax operations.

$$A_s = \text{soft max}\left(\text{sum}(Q \circ K_s)\right) \tag{7}$$

Subsequently, the new spatial state S_t^l can be calculated according to the formula for spatial attention:

$$S_t^l = A_s \circ V_s \tag{8}$$

Finally, the reshaped S_t^l is resized to the same size as the original hidden state and serves as the input to the fusion module.

The fusion module aggregates the temporal state T_t^l and the spatial state S_t^l, and uses gating mechanisms to control the output of the current time sequence. First, T_t^l and S_t^l are concatenated along the channel dimension, and the number of channels is adjusted through convolutional operations to obtain the fusion features:

$$G = \tanh\left(W_g * \left[T_t^l, S_t^l\right] + b_g\right) \tag{9}$$

Subsequently, to effectively control the fusion of historical attention information and the current hidden state H_t^l, two gating mechanisms are used:

$$\begin{aligned} e_i &= W_{ge} * G + b_e \\ e_f &= W_{gf} * G + b_f \end{aligned} \tag{10}$$

The fusion features G are used to generate the input gate e_i and the forget gate e_f through convolution, and the output of the STAM, represented as $\tilde{H}_t^l$, is given by:

$$\tilde{H}_t^l = e_i \circ e_f + H_t^l \circ (1 - e_i) \tag{11}$$

In the STAM-LSTM unit, the STAM will be embedded into the RNN unit shown in Fig. 4, forming the STAM-LSGRU.

Convolutional Inception optimization

The radar echo image exhibits varying strengths and sizes of echoes in different regions, and the commonly used 5x5 convolution is insufficient for capturing the multiscale radar echoes and high-order non-stationary

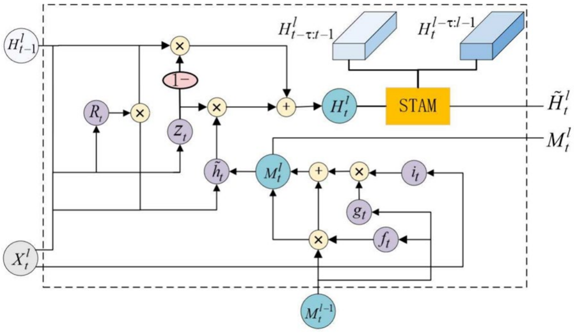


Fig. 4 The STAM-LSGRU unit incorporates the STAM module and optimized convolutional patterns on the basis of the ST-ConvLSGRU

information. In this paper, we propose an improved RNN unit that integrates an Inception network structure with different gating mechanisms and replaces the original 5x5 convolution. The Inception network architecture exhibits strong capabilities in extracting image features across various dimensions and orientations, enhancing the model's generalization ability and feature extraction performance [42]. Initially, it adopts a multi-scale feature extraction strategy, where multiple convolutions of varying sizes operating in parallel within the Inception modules can capture features at different scales simultaneously. This design allows the network to process both local and global features within a single layer, leading to a more comprehensive understanding of image content. Furthermore, the branches within the Inception module utilize convolutional kernels of different sizes and types, working in parallel to perform convolutions in various directions. This parallel operation facilitates the network's effective learning of multiple feature representations, encompassing both local details and global structures. Additionally, by employing multiple convolutions of different sizes in parallel, the Inception module significantly reduces the number of parameters in the network, thereby decreasing the risk of overfitting and enhancing the model's generalization capability. Lastly, the Inception module aggregates information by concatenating features of different scales along the channel dimension. This capacity for information aggregation enables the network to better integrate abstract features at different levels, further improving its understanding of image content.

As shown in Fig. 5, the enhanced convolutional network structure comprises three branches, each undergoing 1x1 convolution, 3x3 convolution, and two consecutive 3x3 convolution operations, respectively. By concatenating two consecutive 3x3 convolutions, our structure not only achieves the same receptive field as a single 5x5 convolution but also significantly reduces the parameter count compared to the latter, offering a more efficient computational approach. Furthermore, the improved Inception convolutional structure effectively captures features at different scales by synthesizing convolution kernels of different sizes, thereby demonstrating superior performance in capturing image features compared to a single-size 5x5 convolution. This design optimizes parameter usage, reduces computational burden, enhances the network's ability to capture multi-scale information in images, and improves the model's expressive power in handling complex image tasks.

Loss function optimization

Currently, most of the existing deep learning radar echo algorithms use mean squared error (MSE) loss as a loss function, which is a common loss function used to evaluate the difference between the predicted value and the real value of the model, and is suitable for regression problems. The smaller the value, the smaller the difference between the predicted and true results. The calculation of the MSE is as follows:

$$MSE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(\hat{y}_{ij} - y_{ij}\right)^2 \tag{12}$$

In this equation, H and W are the length and width of the radar image, and MSE loss is the sum of the squared errors for each pixel in every extrapolated radar image $\hat{y}$ and its corresponding true image y. However, since MSE is sensitive to outliers and penalizes large prediction errors, using MSE as the loss function on datasets with outliers can be influenced by those outliers. In practical images, noise or other interference may exist, which could greatly affect MSE and thereby impact the model's predictive ability. MSE only considers the difference between the predicted and actual values of each pixel, without considering the correlation between pixels. In image prediction, there is usually some correlation

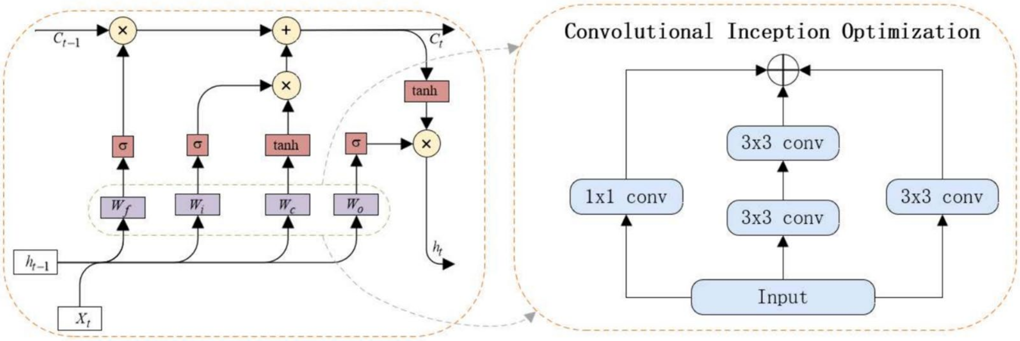


Fig. 5 The structure of the proposed convolutional inception optimization

between pixels, and ignoring this correlation may lead to a decrease in prediction performance. Therefore, to improve upon MSE, commonly used meteorological indicators such as the CSI and the HSS can be incorporated into the evaluation metrics. The improved loss function is as follows:

$$Loss = MSE + (1 - 0.5 \cdot \mathrm{sigmoid(CSI)} - 0.5 \cdot \mathrm{sigmoid(HSS)}) \tag{13}$$

Considering that the CSI and the HSS may not be differentiable, they can be made differentiable by applying the sigmoid function, and incorporated into the final differentiable loss function.

Experiments

Dataset

This paper utilizes the data from the China Central Meteorological Observatory's radar network in the eastern region of China from 2020 to 2022, with a spatial resolution of 0.01° and a temporal resolution of 6 minutes. The radar data is sliced at the central point to achieve a spatial resolution of 400x400. dBZ represents the radar echo value, with larger dBZ values indicating a higher likelihood and intensity of severe convective weather. Atmospheric motion exhibits periodicity, particularly in the same region, with many similar samples in both real-time observations and model forecasts, leading to overfitting. Additionally, severe convective weather occurs only a few days out of the year, thus filtering out some relatively mediocre data is necessary. Ultimately, 10,000 sequential samples are selected, with 6,000 sequences serving as the training set, 2,000 sequences as the validation set, and 2,000 sequences as the test set. The radar echo values range from 0 to 70, and normalization is employed to distribute the values between 0 and 1, facilitating better model convergence.

Implementation

For all experiments, a Nvidia GeForce RTX 3090 GPU was used for training. The default hyperparameters and experimental setup were as follows: the model was trained using a batch size of four image sequences, the Adam optimizer with an initial learning rate of 0.001, and momentum decay set to 0.90. The four-layer model was configured with 64 channels, and a total of 70,000 training steps were performed. At every 5000 training steps, evaluation metrics were recorded for both the training and validation sets. During training, the model was designed to predict the next 10 time sequences, with the attention mechanism of the STAM module focusing on a time step of 5. All experiments were conducted using the same set of hyperparameter values to ensure consistency and comparability. To enhance training performance and explore different strategies, various techniques were employed. These included the use of teacher forcing, which involved providing the correct sequence as input during training, and bidirectional training, where the model was trained in both forward and backward directions. To prevent overfitting, early stopping techniques were utilized. Specifically, training was terminated if the validation loss did not decrease for 10,000 consecutive steps, indicating that the model's performance had plateaued. This ensured that the model was not trained excessively and retained its generalization ability.

Evaluation indicators

This paper evaluates the effectiveness of various models based on MSE, CSI, HSS, and Structural Similarity (SSIM). MSE measures the average difference between model predictions and true values, reflecting the model's precision in predicting future radar echoes. CSI assesses the model's detection capability for precipitation events, taking into account false alarms and missed detections. HSS compares the correctness of the model's predictions to random forecasting, used to measure the predictive capability of the model. The SSIM is used to evaluate the degree of similarity between the model-predicted radar echo images and the actual radar echo images, considering aspects such as luminance, contrast, and structure. Considering the correlation between radar echo values and actual weather, three thresholds of 20dBZ, 35dBZ, and 45dBZ were chosen to evaluate the efficacy of the algorithm for radar extrapolation. Under these thresholds, radar echo images were binarized by designating a value of 1 if the echo value exceeded the threshold and a value of 0 otherwise. TP represents a predicted event that actually occurred, FP represents a predicted event that did not occur, FN represents an unpredicted event that did occur, and TN represents a predicted event that did not occur. Following are the formulas used to calculate the evaluation metrics in this paper:

$$\begin{aligned} CSI &= \frac{TP}{TP+FN+FP} \\ HSS &= \frac{2(TP \times TN - FN \times FP)}{(TP+FN)(FN+TN)+(TP+FP)(FP+TN)} \\ SSIM(x,y) &= \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{\left(\mu_x^2+\mu_y^2+c_1\right)\left(\sigma_x^2+\sigma_y^2+c_2\right)} \\ MSE &= \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2 \end{aligned} \tag{14}$$

where μ_x and μ_y are the means of x and y, respectively. σ_x^2 and σ_y^2 denote the variances. σ_{xy} is the covariance of x and y. $c1$ and $c2$ are constants.

Results and analysis

Currently, radar echo extrapolation models are primarily based on stacking multiple layers of basic Convolutional Recurrent Units. There is no fixed standard for the number of layers in the stacked radar echo extrapolation model, as it is contingent on variables such as data volume, data complexity, network structure, and hardware. Increasing the number of layers can improve the network's expressive power and the model's capacity to represent and abstract data, but it may also increase the network's computational and storage burden, resulting in overfitting and other issues. The number of layers is typically determined by the extent and complexity of the dataset as well as the training effect of the network. For smaller datasets and relatively straightforward problems, a shallower network structure may be optimal, whereas a deeper network structure may be preferable for larger datasets and more complex problems. Using experimental methods, this paper determines the optimal number of stacked layers to avoid overfitting and underfitting issues. Table 1 presents the MSE scores of various models at different numbers of layers. It is observed that as the number of layers increases, the MSE value of each model typically decreases to reach a minimum value before starting to rise again. For most models, the MSE value reaches its minimum when the number of layers is four, indicating that the models perform best at this depth. As the depth of the model increases, it is able to learn more complex features and deeper data representations. There comes a point where this learning capability is optimized, and further increases in the number of layers make the model more complex, increasing the number of parameters. This complexity can lead to gradients gradually vanishing or exploding during the backpropagation process, making it difficult to train the model. The STAM-LSGRU model achieves its best performance at four layers; when the number of layers is less than or greater than four, the model's performance metrics decrease. Therefore, this paper sets the number of layers of the STAM-LSGRU model to four.

Through ablation experiments and comparative experiments, we validated the best performance of radar echo extrapolation at different thresholds. The results are shown in Tables 2 and 3, where CSI and HSS scores were calculated at the thresholds of 20 dBZ, 35 dBZ, and 45 dBZ, while MSE and SSIM were averaged across all thresholds. The symbol ↑ indicates that higher values indicate better performance in radar echo extrapolation, while ↓ indicates the opposite.

The ablation experiment results are presented in Table 2, where evaluation metrics of the original ST-ConvLSGRU model and various improvement methods are compared. Across all metrics and thresholds, ST-ConvLSGRU-1, STAM-LSGRU-0, and STAM-LSGRU-1 outperform ST-ConvLSGRU. Specifically, STAM-LSGRU* demonstrates improvements of 6.87%, 6.45%, 5.8%, and 7.7% in CSI, HSS, SSIM, and MSE, respectively, compared to the ST-ConvLSGRU network. Figure 6 illustrates the radar echo extrapolation results in order to visually compare the experimental outcomes of various enhancement techniques. Indicating that these modules can enhance the performance of intensity prediction, the prediction results of STAM-LSGRU* are clearer, have more distinct edges, and pay closer attention to high echo regions.

Table 1 MSE score of 10 frame advance prediction for different depth models

Depth	ConvLSTM [21]	TrajGRU [34]	PredRNN [35]	MIM [37]	MotionRNN [39]	STAM-LSGRU
2	168.7	159.7	149.3	139.6	125.3	129.5
3	161.3	152.3	134.7	128.7	110.2	115.3
4	156.8	**145.7**	**127.5**	**121.6**	**103.4**	**106.7**
5	**153.2**	149.8	134.1	128.2	108.6	113.8
6	160.1	155.3	142.8	135.4	113.2	116.1

Table 2 The improved components were evaluated through a series of ablation experiments. The ST-ConvLSGRU served as the baseline model, while the ST-ConvLSGRU-1 incorporated an inception optimization. The STAM-LSGRU introduced an STAM module, and the loss function of STAM-LSGRU-1 was further optimized. Finally, the STAM-LSGRU* model integrated all the improvement strategies

Model	CSI↑			HSS↑			SSIM↑	MSE↓
	T=20	T=35	T=45	T=20	T=35	T=45		
ST-ConvLSGRU	0.557	0.457	0.298	0.586	0.496	0.322	0.583	115.3
ST-ConvLSGRU-1	0.565	0.461	0.309	0.599	0.504	0.334	0.598	111.2
STAM-LSGRU	0.571	0.465	0.305	0.604	0.513	0.329	0.590	**105.6**
STAM-LSGRU-1	0.581	0.470	0.311	0.605	0.519	0.335	0.601	119.7
STAM-LSGRU*	**0.589**	**0.485**	**0.324**	**0.615**	**0.529**	**0.347**	**0.617**	106.4

Table 3 The predictive performance of different models on radar echo data was compared. All models were trained using 5 time series and evaluated based on their ability to predict the next 10 time series. The evaluation metric was the average of the 10 predicted time series

Model	CSI↑			HSS↑			SSIM↑	MSE↓
	T=20	T=35	T=45	T=20	T=35	T=45		
ConvLSTM [21]	0.455	0.358	0.212	0.483	0.374	0.278	0.452	156.8
TrajGRU [34]	0.478	0.377	0.246	0.512	0.396	0.298	0.530	145.7
PredRNN [35]	0.502	0.433	0.265	0.534	0.468	0.315	0.535	127.5
MIM [37]	0.525	0.445	0.284	0.559	0.481	0.309	0.551	121.6
MotionRNN [39]	0.585	0.475	0.316	0.609	0.525	0.341	0.612	**103.4**
STAM-LSGRU*	**0.589**	**0.485**	**0.324**	**0.615**	**0.529**	**0.347**	**0.617**	106.4

As shown in Table 3, different evaluation metrics have been improved. Compared to the state-of-the-art MotionRNN model, the CSI score has increased by an average of 1.6%, the HSS score has increased by 1.1%, the SSIM score has increased by 2.7%, and the mean squared error has increased by 3.2%, under different thresholds. Figure 7 displays the hourly scores of each model in the prediction of the next hour. The models exhibit similar performances, but as time progresses, the scores of other models sharply decline, whereas STAM-LSGRU shows a more gradual decrease in scores.

For a more intuitive observation, a visual example is also shown in Fig. 8. From Fig. 8, it can be observed that the STAM-LSGRU model designed in this research outperformed the other five models. The ConvLSTM model produced smoother results than other methods and suffered from severe detail loss and prediction errors in the high reflectivity regions of radar images. TrajGRU and PredRNN had poor performance in predicting the central echo region and also suffered from distortion to some extent. Since radar image changes are a high-order nonstationary process, these methods were able to effectively predict the radar motion trend. MIM and STAM-LSGRU better captured the overall changing trend characteristics of the echo region, but STAM-LSGRU captured the high and low echo characteristics better than MIM. The prediction results of MotionRNN and STAM-LSGRU were similar, but the predicted image blocks of STAM-LSGRU were closer to the actual observation results. From the edge of the echo main body, its predicted image edges were more in line with the actual ones, and its blurring degree was lower, showing greater consistency with the real image.

Conclusion

This research introduces a neural network-based radar echo extrapolation algorithm named STAM-LSGRU. By deploying the STAM-LSGRU model in an edge computing environment, we not only achieve enhanced real-time data processing capabilities but also significantly reduce data transmission delays. Compared with traditional radar echo extrapolation algorithms and other deep learning-based algorithms, STAM-LSGRU exhibits markedly improved predictive performance in complex environments, particularly in heavy rain areas. This paper designs STAM to capture reliable inter-frame motion information by expanding the temporal and spatial receptive fields of the prediction units. The convolutional structure and loss function of the basic unit have been improved to

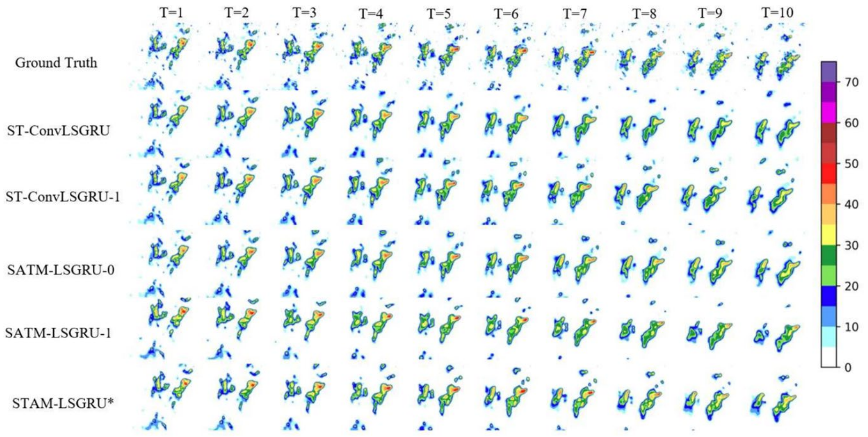
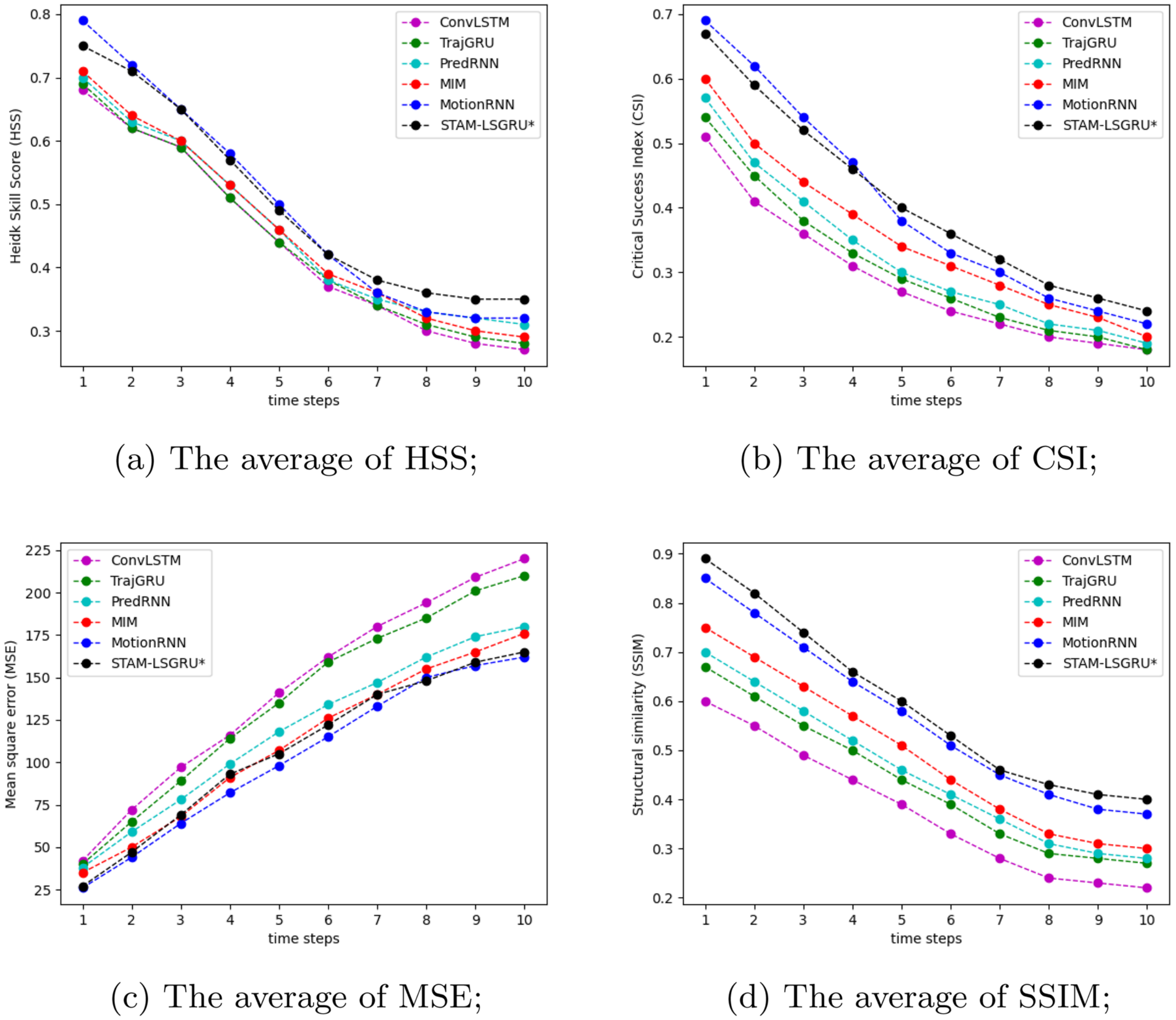


Fig. 6 Comparison of results of different improved versions based on ST-ConvLSGRU



(a) The average of HSS;

(b) The average of CSI;

(c) The average of MSE;

(d) The average of SSIM;

Fig. 7 Extrapolated objective indicators of 10 time series of different models

enhance the robustness of model predictions. Compared to the MotionRNN model, the CSI score has increased by an average of 1.6%, the HSS score by 1.1%, and the SSIM score by 2.7%. In the future, we plan to further advance meteorological forecasting by integrating more observational data and model outputs, aiming to improve the accuracy and timeliness of weather predictions. With the continuous advancements in MEC and AI technologies, along with the increasing abundance of meteorological observation data, we anticipate that the STAM-LSGRU model will demonstrate higher predictive capabilities in a wider range of meteorological scenarios, bringing new breakthroughs to the field of weather forecasting.

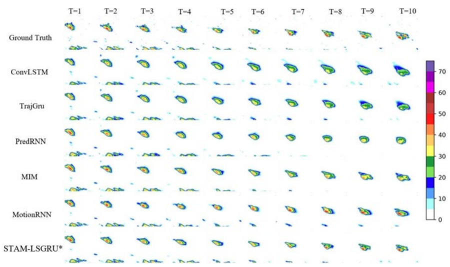


Fig. 8 The top row shows the labels, while the comparison results of different extrapolation methods are presented below. The pixel values correspond to the right side of the images. The darker the color, the higher the probability of the occurrence of severe convective weather

Authors' contributions

Hailang Cheng conducted thematic research, collected and organized data, and wrote the paper. Mengmeng Cui provided innovative points, and developed a detailed outline and structure. Yuzhe Shi undertook comprehensive editing and proofreading of the draft paper, including linguistic refinement, logical verification, and formatting. Mengmeng Cui is the corresponding author of this paper.

Funding

Not applicable.

Availability of data and materials

No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate

The research in this paper does not involve any illegal or unethical practices.

Consent for publication

The authors read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Received: 15 January 2024 Accepted: 28 April 2024
Published online: 14 May 2024

References

1. Alam F, Salam M, Khalil NA, khan O, Khan M (2021) Rainfall trend analysis and weather forecast accuracy in selected parts of khyber pakhtunkhwa, pakistan. SN Appl Sci 3:575
2. Guido Z, Lopus S, Waldman K, Hannah C, Zimmer A, Krell N, Knudson C, Estes L, Caylor K, Evans T (2021) Perceived links between climate change and weather forecast accuracy: new barriers to tools for agricultural decision-making. Clim Chang 168:1–20
3. Wang S, Wang T, Wang S, Fang Z, Huang J, Zhou Z (2023) MLAM: Multi-layer attention module for radar extrapolation based on spatiotemporal sequence neural network. Sensors 23(19):8065
4. Hu Z, Xu X, Zhang Y, Tang H, Cheng Y, Qian C, Khosravi MR (2022) Cloud–edge cooperation for meteorological radar big data: a review of data quality control. Complex Intell Syst 8:3789–3803. https://doi.org/10.1007/s40747-021-00581-w.
5. Xu X, Tang S, Qi L, Zhou X, Dai F, Dou W (2023) Cnn partitioning and offloading for vehicular edge networks in web3. IEEE Communications Magazine 61(8):36–42
6. Xu X, Yang C, Bilal M, Li W, Wang H (2023) Computation offloading for energy and delay trade-offs with traffic flow prediction in edge computing-enabled iov. IEEE Transactions on Intelligent Transportation Systems 24(12):15613–15623
7. Mehrabi M, You D, Latzko V, Salah H, Reisslein M, Fitzek FH (2019) Device-enhanced mec: Multi-access edge computing (mec) aided by end device computation and caching: A survey. IEEE Access 7(166):079–166108
8. Xu Y, Lu X, Tian Y, Huang Y (2022) Real-time seismic damage prediction and comparison of various ground motion intensity measures based on machine learning. J Earthq Eng 26(8):4259–4279
9. Kumar V, Azamathulla HM, Sharma KV, Mehta DJ, Maharaj KT (2023) The state of the art in deep learning applications, challenges, and future prospects: A comprehensive review of flood forecasting and management. Sustainability 15(13):10543
10. Kumar V, Kedam N, Sharma KV, Khedher KM, Alluqmani AE (2023) A comparison of machine learning models for predicting rainfall in urban metropolitan cities. Sustainability 15(18):13724
11. Luo C, Li X, Wen Y et al (2021) A novel lstm model with interaction dual attention for radar echo extrapolation. Remote Sens 13(2):164
12. Yang Z, Wu H, Liu Q et al (2023) A self-attention integrated spatiotemporal LSTM approach to edge-radar echo extrapolation in the Internet of Radars. ISA Trans 132:155–166
13. Zhang F, Lai C, Chen W (2022) Weather radar echo extrapolation method based on deep learning. Atmosphere 13(5):815
14. Sun N, Zhou Z, Li Q et al (2022) Three-dimensional gridded radar echo extrapolation for convective storm nowcasting based on 3d-convlstm model. Remote Sens 14(17):4256
15. Sun J, Xue M, Wilson JW et al (2014) Use of nwp for nowcasting convective precipitation: recent progress and challenges. Bull Am Meteorol Soc 95(3):409–426
16. Mehrkanoon S (2019) Deep shared representation learning for weather elements forecasting. Knowl-Based Syst 179:120–128
17. Monteiro MJ, Couto FT, Bernardino M et al (2022) A review on the current status of numerical weather prediction in portugal 2021: Surface-atmosphere interactions. Atmosphere 13(9):1356
18. Krogh A (2008) What are artificial neural networks? Nat Biotechnol 26(2):195–197
19. Noble WS (2006) What is a support vector machine? Nat Biotechnol 24(12):1565–1567
20. Myles AJ, Feudale RN, Liu Y et al (2004) An introduction to decision tree modeling. J Chemom J Chemom Soc 18(6):275–285
21. Shi X, Chen Z, Wang H, et al (2015) Convolutional lstm network: A machine learning approach for precipitation nowcasting. Adv Neural Inf Process Syst 802–810
22. Shi E, Li Q, Gu D, et al (2018) A method of weather radar echo extrapolation based on convolutional neural networks. In: Bai X, Mukherjee SS, Wu W, et al (eds) MultiMedia Modeling: 24th International Conference, MMM 2018, Bangkok, Thailand, February 5-7, 2018, Proceedings, Part I, pp 16–28
23. Deb SD, Jha RK (2023) Breast ultrasound image classification using fuzzy-rank-based ensemble network. Biomed Signal Process Control 85:104871
24. Palechor A, Bhoumik A, Günther M (2023) Large-scale open-set classification protocols for imagenet. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, IEEE, pp 42–51
25. Zhou Z, Chen X, Li E, Zeng L, Luo K, Zhang J (2019) Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proc IEEE 107:1738–62. https://api.semanticscholar.org/CorpusID:165163986
26. Al-Habob A, Dobre O (2020) Mobile edge computing and artificial intelligence: A mutually-beneficial relationship. Signal Process. arXiv:2005.03100
27. Huang S, Wang S, Wang R, Wen M, Huang K (2020) Reconfigurable intelligent surface assisted mobile edge computing with heterogeneous learning tasks. IEEE Trans Cogn Commun Netw 7:369–382
28. Deng S, Zhao H, Fang W, Yin J, Dustdar S, Zomaya AY (2020) Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet Things J 7(8):7457–7469
29. Yazid Y, Ez-zazi I, Guerrero-González A, Oualkadi AE, Arioua M (2021) Uav-enabled mobile edge-computing for iot based on ai: A comprehensive review. Drones 5(4):148. https://doi.org/10.3390/drones5040148
30. Dahmane S, Yagoubi M, Abdelaziz KC, Lorenz P, Lagraa N, Lakas A (2022) Toward a secure edge-enabled and artificially intelligent internet of flying things using blockchain. IEEE Internet Things Mag 5:90–95
31. Wang Y, Zhao J (2022) Mobile edge computing, metaverse, 6g wireless communications, artificial intelligence, and blockchain: Survey and their convergence. arXiv:2209.14147
32. Chakraborty S, Sukapuram R (2022) Multi-access edge computing for urban informatics. In: Proceedings of the 23rd International Conference on Distributed Computing and Networking 225–228. https://dl.acm.org/doi/abs/10.1145/3491003.3493332
33. Graves A, Jaitly N (2014) Towards end-to-end speech recognition with recurrent neural networks. In: International Conference on Machine Learning, PMLR, pp 1764–1772
34. Shi X, Gao Z, Lausen L, et al (2017) Deep learning for precipitation nowcasting: A benchmark and a new model. Adv Neural Inf Process Syst 5617–5627
35. Wang Y, Long M, Wang J, et al (2017) Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. Adv Neural Inf Process Syst 879–888
36. Wang Y, Gao Z, Long M, et al (2018) Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In: International Conference on Machine Learning, PMLR, pp 5123–5132

37. Wang Y, Zhang J, Zhu H, et al (2019) Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 9154–9162. https://openaccess.thecvf.com/content_CVPR_2019/html/Wang_Memory_in_Memory_A_Predictive_Neural_Network_for_Learning_Higher-Order_CVPR_2019_paper.html
38. Lin Z, Li M, Zheng Z et al (2020) Self-attention convlstm for spatiotemporal prediction. Proceedings of the AAAI Conference on Artificial Intelligence 34:11531–11538
39. Wu H, Yao Z, Wang J, Long M (2021) Motionrnn: A flexible model for video prediction with spacetime-varying motions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 15435–15444. https://openaccess.thecvf.com/content/CVPR2021/html/Wu_MotionRNN_A_Flexible_Model_for_Video_Prediction_With_Spacetime-Varying_Motions_CVPR_2021_paper.html
40. Chang Z, Zhang X, Wang S, et al (2022) Strpm: A spatiotemporal residual predictive model for high-resolution video prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 13946–13955. https://openaccess.thecvf.com/content/CVPR2022/html/Chang_STRPM_A_Spatiotemporal_Residual_Predictive_Model_for_High-Resolution_Video_Prediction_CVPR_2022_paper.html
41. Jin XB, Wang ZY, Kong JL et al (2023) Deep spatio-temporal graph network with self-optimization for air quality prediction. Entropy 25(2):247
42. Zhang X (2023) Improved three-dimensional inception networks for hyperspectral remote sensing image classification. IEEE Access 11:32648–32658

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.